SAE INTERNATIONAL

SURFACE VEHICLE INFORMATION REPORT

J2716™ APR2016

Issued 2007-04
Revised 2016-04

Superseding J2716 JAN2010

# (R) SENT - Single Edge Nibble Transmission for Automotive Applications

RATIONALE

1. Clarified clock variation refers to variation in clock tick time.

2. New appendix for High Speed 12 bit Sensors.

3. Changes for initialization and non-usage of serial message.

4. Changes to support option for independent 5V supply in sensor

5. Serial message cycle changed to ≤ 64 messages for greater flexibility and to allow more time for diagnostic information.

6. Clarification to EMC Susceptibility testing guidelines for Erroneous SENT frames which remain undetected by the SENT CRC

7. Added Appendix A.6 Temperature Sensor requirements

8. Added APPENDIX E

9. Added APPENDIX G for recommended connector

10. Created APPENDIX H (SENT Data Frame Formats) and moved generic sensor requirements from APPENDIX A to APPENDIX H

11. Changes to support higher current sensors

12. Added E.3 Error Messages and Signals

13. Added A.7 Position Sensors and Combined Position Temperature Sensors

14. Added *Overview of SENT standard structure* (3.2)



**TO PLACE A DOCUMENT ORDER:** **Tel: 877-606-7323 (inside USA and Canada)**
**Tel: +1 724-776-4970 (outside USA)**
**Fax: 724-776-0790**
**Email: CustomerService@sae.org**
**SAE WEB ADDRESS: http://www.sae.org**

**SAE values your input. To provide feedback on this Technical Report, please visit [http://www.sae.org/technical/standards/J2716_201604](http://www.sae.org/technical/standards/J2716_201604)**

TABLE OF CONTENTS

# 1. SCOPE

This document defines a level of standardization in the implementation of the digital pulse scheme for reporting sensor information via Single Edge Nibble Transmission (SENT) encoding. This standard will allow ECU and tool manufacturers to satisfy the needs of multiple end users with minimum modifications to the basic design. This standard will benefit vehicle Original Equipment Manufacturers (OEMs) by achieving lower ECU costs due to higher industry volumes of the basic design.

Requirements stated in this document provide a minimum standard level of performance to which all compatible ECUs and media shall be designed. This assures data communication among all connected devices regardless of supplier.

This document is a communication interface specification and no to be treated as product specification.

The intended audience includes, but is not limited to, ECU suppliers, sensor suppliers, component release engineers and vehicle system engineers.

## 1.1 Overview

The Single Edge Nibble Transmission encoding scheme (SENT) is intended for use in applications where high resolution sensor data needs to be communicated from a sensor to an Electronic Control Unit (ECU). It is intended as a replacement for the lower resolution methods of 10 bit A/D's and PWM and as a simpler low cost alternative to CAN or LIN. The implementation assumes that the sensor is a smart sensor containing a microprocessor or dedicated logic device (ASIC) to create the signal.

SENT is a unidirectional communications scheme from sensor / transmitting device to controller /receiving device which does not include a coordination signal from the controller/receiving device. The sensor signal is transmitted as a series of pulses with data encoded as falling to falling edge periods. Details of the signal encoding may vary for specific sensor applications described in various appendices of this specification.

# 2. REFERENCES

## 2.1 Applicable Documents

The following publications form a part of this specification to the extent specified herein. Unless otherwise indicated, the latest issue of SAE publications shall apply.

### 2.1.1 SAE Publications

Available from SAE International, 400 Commonwealth Drive, Warrendale, PA 15096-0001, Tel: 877-606-7323 (inside USA and Canada) or +1 724-776-4970 (outside USA), www.sae.org.

SAE J551 (All parts) Performance Levels and Methods of Measurement of Electromagnetic Compatibility for Vehicles and Devices

SAE J1113 (All parts) Electromagnetic Compatibility Measurement Procedures for Vehicle Components

SAE J1930 Electrical/Electronic Systems Diagnostic Terms, Definitions, Abbreviation and Acronyms

### 2.1.2 Other Publications

CISPR 25 Limits and Methods of Measurement of Radio Disturbance Characteristics for the Protection of Receivers Used On Board Vehicles (available at webstore.iec.ch).

ES-XW7T-1A278-AC Ford Component and Subsystem Electromagnetic Compatibility Worldwide Requirements and Test Procedures (available at www.fordemc.com. This document shall be referred to as the Ford EMC Spec.)

GMW3097 General Specification for Electrical / Electronic Components and Subsystems, Electromagnetic Compatibility (this document will be referred to as the GM EMC Spec.)

ANSI INCITS 4-1986 (R2007) American National Standards Institute, *Information Systems - Coded Character Sets - 7-Bit American National Standard Code for Information Interchange (7-Bit ASCII)*, 2007

USCAR United States Council for Automotive Research, USCAR EWCAP EWCAP Footprints Database, 1.2 mm Connectors (sealed), (available at http://www.uscar.org/guest/teams/10/Electrical-Wiring-Component-Applications-Partnership)

# 3. DEFINITION OF TERMS

## 3.1 GLOSSARY

### 3.1.1 Media

The physical entity that conveys the electrical (or equivalent means of communication) signal transmission between electronic devices.

### 3.1.2 Protocol

The formal set of conventions or rules for the exchange of information between electronic devices. This includes the specification of the signal frame administration, frame transfer and physical layer.

3.1.3 Message

One sequence of calibration pulse and specified number of nibble pulses for that implementation. The number of nibbles is constant for each implementation but the individual message times can vary depending on the specific values of the nibbles.

3.1.4 Radiated Emissions

The energy that radiates from the physical layer.

3.1.5 Radiated Immunity

The level of susceptibility of physical layer components to communication errors in the presence of high energy electromagnetic fields.

3.1.6 Receiver Module

The processor that receives the encoded signal. Generally an ECU with falling-edge detection and timing measurement capabilities.

3.1.7 Transmitter Module

The device that generates the message to the receiver module. Generally a smart sensor.

3.1.8 Nominal

Time period assuming no transmitter clock error.

3.1.9 Pulse Period

Time between consecutive falling edges of the transmitting signal.

3.1.10 Error

Indicates that a problem exists with current sample, data or message.

3.1.11 Fault

Indicates that enough errors have been detected (usually matured by counting X errors in-a-row or via up-down or X-out-of-Y counters or other filtering means) to develop into a fault which is latched until cleared.

3.1.12 Clock Tick Time

Fundamental time unit in transmitter used to construct SENT output signal.

3.1.13 Signal Ground Line

The reference point from which all SENT electrical interface voltage parameters are measured; transmitter requirements are defined with respect to the Signal Ground pin on the transmitting device; receiver requirements are defined with respect to the Return pin on the receiving module. An equivalent term for the Signal Ground Line is Signal Return Line. The suffix GND is linked to the Signal Ground Line.

3.1.14 Power Ground Line

The signal through which the primary power current for a module flows to a system level ground node. Under certain conditions it is possible, that Power Ground Line and Signal Ground Line are identical.

3.1.15 Medium Temperature

Measured temperature of a physical medium defined in the specification of the particular sensor of the transmitter.

3.1.16 Internal Reference Temperature

Measured internal temperature of some point within the transmitter (e.g., integrated circuit). This temperature can be used by the receiver as a reference to determine the status of the transmitter.

3.1.17 Reserved data ranges

Reserved data ranges shall be used, as specified in SAE J2716, or they are retained for future use by the SAE. These data ranges may not be assigned to any other use.

3.1.18 Fast Channel

The data transmitted on the data nibble pulses can carry the payload data of one or more Fast Channels. The data rate of these payload data channels is determined by the number of bits (e.g., 12 bits) in each signal data field and the period of the SENT transmission sequence. These channels are called Fast Channels, since their data rate is significantly larger than the data rate of the Serial Message Channel.

3.1.19 Slow Channel

The Serial Message Channel is also denoted as Slow Channel in earlier revisions of SAE J2716. The term Slow Channel is used as opposed to Fast Channel.

3.1.20 Serial Message Channel

Two bits per SENT data frame can be allocated to a serial message channel. Either Enhanced Serial Message Format or Short Serial Message Format can be used for data transmission over the serial message channel. The Enhanced Serial Message Format is recommended rather than the Short Serial Message Format.

3.1.21 Enhanced Serial Message Format

Recommended format of the Serial Message Channel frame.

3.1.22 Short Serial Message Format

Format of the Serial Message Channel frame.

3.1.23 Supplementary Data Channels

Enhanced Serial Message Format frames can carry supplementary data channels. These supplementary data channels can transmit measurement data from further sources at a lower rate than the data on the fast channels. A default assignment of supplementary data channels is defined for specific sensor classes.

## 3.2 OVERVIEW OF SENT STANDARD STRUCTURE

The main body of the standard defines the mandatory means of the physical as well as data link layer with limited degrees of freedom for implementation. The system designer has to specify the SENT compliant component by choosing the options and limit the variance given within physical and data link layer definitions of the main body.

The application specific definitions within the appendices are intended to help the system and component designer to limit the variance of implementations to support interchangeability of components, but following them is not mandatory.

The system designer should start by using the application specific protocols of APPENDIX A and additionally limit from the available ranges and choose from the set of available options for the transmitting component. Regardless the application specific protocols defined with APPENDIX A there are still some options which are open to be chosen for different implementations by the component designers. These options need to be defined with the component specifications and either be covered by the receiver design or have to be limited by the system designer.

A flexible receiver design adapting to different options is recommended and supports the interchangeability.

3.2.1 SENT Transmission Layers

Figure 3.2.1-1 illustrates the structure of the SENT standard and the relationship of as well the sections of the main body as the appendices of the document to the data transmission layers.

Since the functions which define one layer of the SENT protocol are required by the functions of the layer above it, this overview also explains the hierarchy of the sections of the SENT standard.

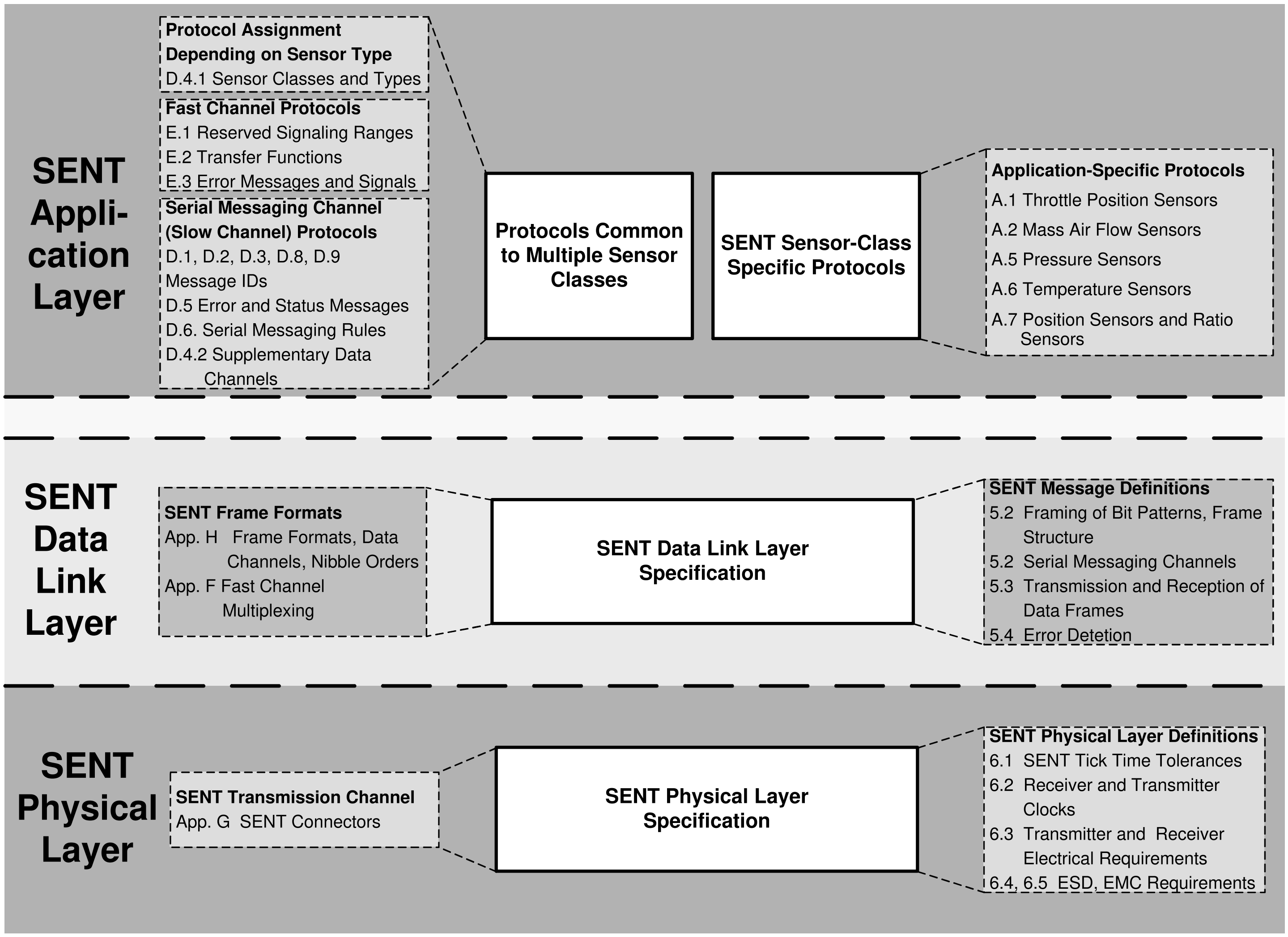



*Figure 3.2.1-1 – SENT standard structure and transmission layers*

3.2.2 SENT Appendices

Protocols and frame formats are common to multiple sensor types.

Application-specific sensor types, data transmission formats and protocols can be specified by references to respective sections that define generic functions and frame formats. SENT test conditions are specified in APPENDIX C.

## 4. ACRONYMS, ABBREVIATIONS, AND SYMBOLS

ASIC – Application Specific Integrated Circuit
CAN – Controller Area Network
ECU – Electronic Control Unit
EMC – Electromagnetic Compatibility
ESD – Electrostatic Discharge
FC – Frame Control (fast channel multiplexing)
ISO – International Organization for Standardization
kbits/sec – Thousands of data bits per second
LIN – Local Interconnect Network
LSN – least significant nibble
MAF – Mass Air Flow
MidLSN – middle least significant nibble
MidMSN – middle most significant nibble
MidN – middle nibble
MSN – most significant nibble
OEM – Original Equipment Manufacturer
RE – Radiated Emissions
RI – Radiated Immunity
SAE – Society of Automotive Engineers
SENT – Single Edge Nibble Transmission
TPS – Throttle Position Sensor

## 5. SENT SYSTEM REQUIREMENTS

### 5.1 General Requirements

Transmission occurs independently of any action of the receiver module, i.e., the transmission shall not require a synchronization signal from the receiver module.

Assumptions used to design the encoding scheme:

- Actual Transmission time may be dependent on the data values being sent and the transmitter clock variation.
- Message pulse order (i.e., message frame) is fixed for all transmitters.
- Transmitter is allowed a maximum clock tick time variation of ±20%.
- Transmission time for the longest data message and max transmitter clock variation is less than 1.0 millisecond at 3 microsecond clock tick time and 6 data nibbles.
- A transmitter specific nominal clock period (tick) between 3 microseconds and 90 microseconds.
- The encoding scheme defines a number of diagnostic tests to be implemented in the receiving module. However, for example, the CRC checksum is 4 bit and not as robust as other checking schemes (see section 5.4.1). Therefore, the encoding scheme is targeted at systems that can tolerate a low probability of intermittent faulted messages not being detected via the scheme’s diagnostic suite. In case additional robustness is needed, application level diagnostics should be used.

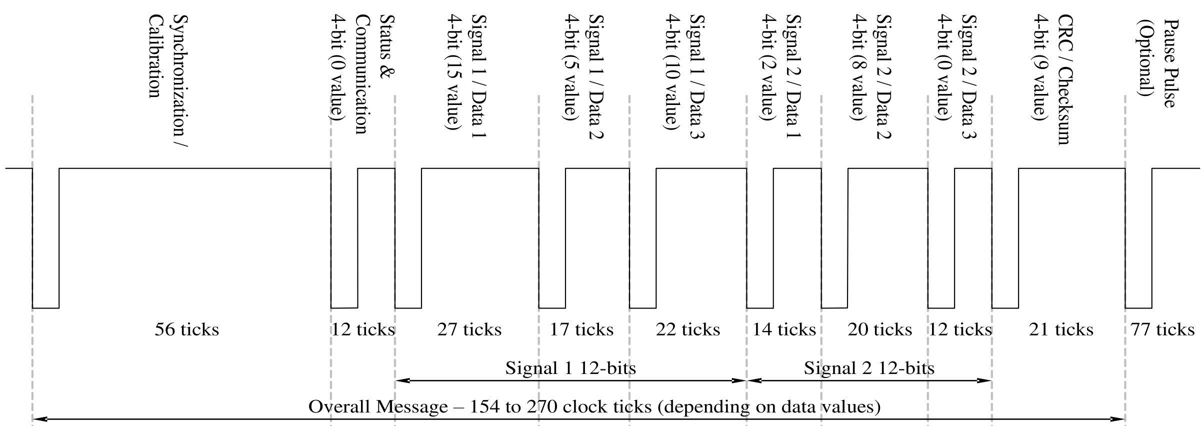
## 5.2 Message Definitions

### 5.2.1 Transmission Sequence

The encoding scheme consists of a sequence of pulses which is repeatedly sent by the transmitting module. The transmission consists of the following sequence (all times nominal):

- Calibration/Synchronization pulse period 56 clock ticks.
- One 4 bit Status and Serial Communication nibble pulse (defined in 5.2.4) of 12 to 27 clock ticks.
- A sequence of one up to six[1] 4 bit Data nibble pulses (12 to 27 clock ticks each) representing the values of the signal(s) to be communicated. The number of nibbles will be fixed for each application of the encoding scheme (i.e., throttle position sensors, mass air flow, etc.) but can vary between applications. For example, if two 12 bit values are transmitted, 6 nibbles will be communicated.
- One 4 bit Checksum nibble pulse (defined in 5.2.5) of 12 to 27 clock ticks.
- One optional pause pulse (defined in 5.2.6)

Figure 5.2.1-1 shows an example single message transmission for two 12 bit sensor values assuming a 3 microsecond clock tick. Note that the shortest length message, based on a valid checksum (determined using approach documented in section 5.4.2.2) consists of 154 clock ticks (Data nibbles [0 0 0 1 1 0] and Checksum 0, Data nibbles [0 0 1 0 0 0] and Checksum 1, and Data nibbles [1 0 0 0 0 1] and Checksum 0). Similarly, the longest message is 270 clock ticks (Data nibbles [$E $F $F $F $F $E] and Checksum $F, Data nibbles [$F $F $E $F $F $F] and Checksum $E, and Data nibbles [$F $F $F $E $E $F] and Checksum $F).




Minimum Nibble period = 36 μsec @ 3 μsec clock tick
Nibble encoded period = 36 μsec + x*(3 μsec) (where x=0,1,..,15)

***Figure 5.2.1-1 – Example encoding scheme for two 12 bit signals***

Signals conveyed using the Data nibbles are also referred to as “Fast Channel” signals with this specification.

---

[1] For SENT receiver implementations, it is highly recommended to allow for up to at least 8 data nibbles to enable a forward-compatibility with future revisions of the J2716 standard.

5.2.2 Transmission Properties of Calibration/Synchronization Pulse

- Nominal pulse period is 56 clock ticks.
- More than 4 clock ticks driven low time (all remaining clock ticks driven high).
- Actual period measured by receiving module to correct for transmitter clock variation.

5.2.3 Transmission Properties of Nibble Pulse

- Minimum pulse period is 12 clock ticks.
- More than 4 clock ticks driven low time (all remaining clock ticks driven high).
- Each nibble count is 1 clock tick (0 – 15 counts ⇒ 0 – 45 µs at a 3 µs clock tick).
- Minimum nibble pulse period (transmission value of 0) = 12 clock ticks (36 µs at a 3 µs clock tick).
- Maximum nibble pulse period (transmission value of 15) = 12 + 15 = 27 clock ticks (36 + 45 = 81 µs at a 3 µs clock tick).

See Figure 5.2.3-1 and Figure 5.2.3-2 for pictorial representation of some nibble properties.

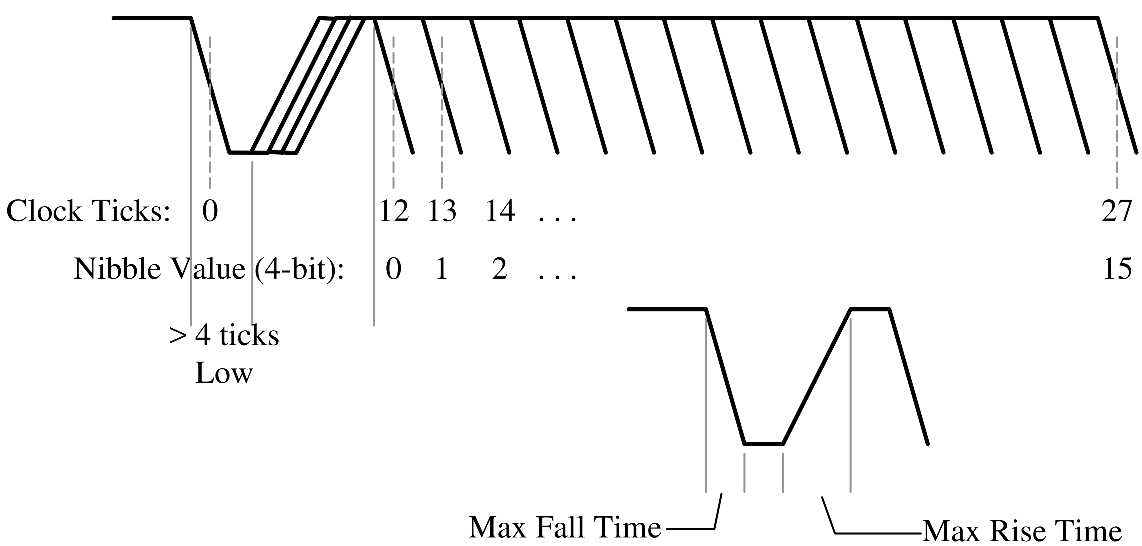



*Figure 5.2.3-1 – Example nominal nibble times*

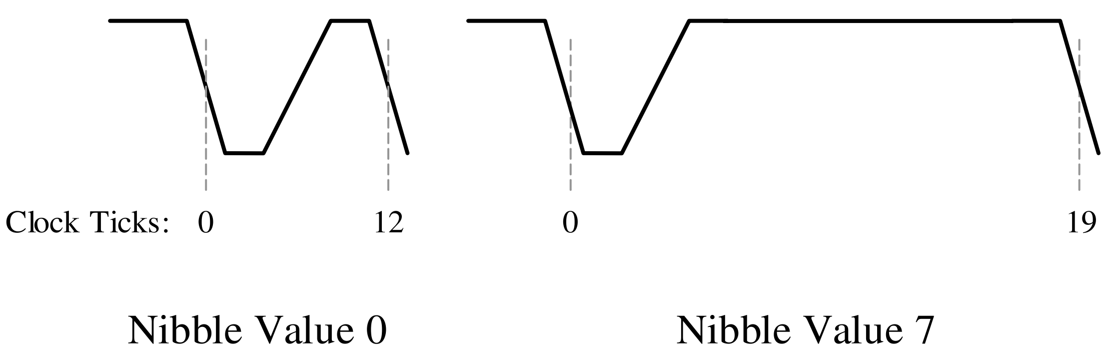



*Figure 5.2.3-2 – Continued example nibble values*

### 5.2.4 Status and Communication Nibble

This nibble is reserved to enable the sensor to transmit miscellaneous information such as part numbers or error code information. The nibble is defined in the following table:

***Table 5.2.4-1 – Status and communication nibble description***

<table>
<tr><th>Bit Number</th><th>Bit Function</th></tr>
<tr><td>0 (least significant)</td><td>Reserved for specific application</td></tr>
<tr><td>1</td><td>Reserved for specific application</td></tr>
<tr><td>2</td><td rowspan="2">See section 5.2.4.1 and 5.2.4.2 for definition of serial data message bits.</td></tr>
<tr><td>3 (most significant)</td></tr>
</table>

Note that the Status and Communication Nibble is not included in the frame CRC calculation and therefore can have a higher occurrence of non-detected errors.

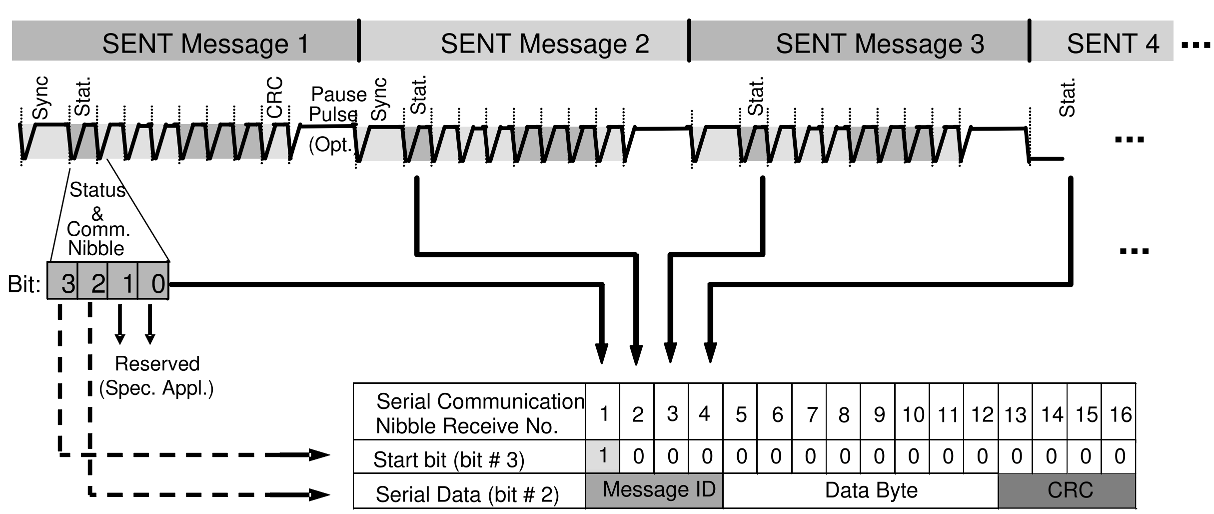
Bits 2 and 3 define a transmitter optional serial message channel which can be implemented either as a short serial message format or an enhanced serial message format as specified below. SENT receivers shall support both the short serial message format as well as the enhanced serial message format. The Enhanced Serial Message Format is recommended rather than the Short Serial Message Format.

#### 5.2.4.1 Short Serial Message Format

Serial data is transmitted (bit by bit) in bit 2 (of the Status and Communication nibble) of consecutive messages from the transmitter. Serial data will be communicated in a 16-bit sequence as shown in Figure 5.2.4.1-1. The starting bit of a serial message is indicated by a “1” in bit 3 of the Status and Communication nibble. The next 15 received frames must contain a value of “0” in this same bit 3 position, as shown in Figure 5.2.4.1-1. All 16 frames must be successfully received (no errors, calibration pulse variation, data nibble CRC error, etc.) for the serial value to be received (Figure 5.2.4.1-1).

The 16-bit message consists of a 4 bit Message ID, a Data Byte, and a 4 bit CRC checksum. The CRC checksum is derived for the Message ID and Data Byte and is the same checksum algorithm as used to calculate the SENT CRC nibble (see 5.4). The Message ID is used to identify the type of data being communicated in the Data Byte. Actual Serial Data message IDs and data values are application specific.

All data transmitted in the Serial Data Bit (bit #2) is sent in the order Most Significant Bit to Least Significant Bit.




One serial message is composed of 16 SENT consecutive error-fee messages.

***Figure 5.2.4.1-1 – Construction of short serial data message from 16 SENT messages***

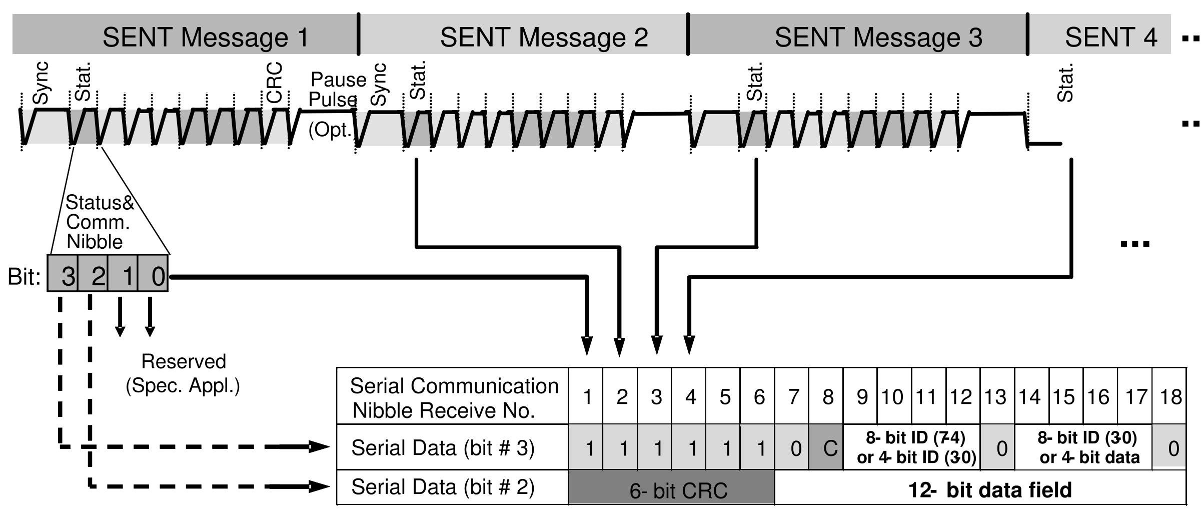
### 5.2.4.2 Enhanced Serial Message Format

An optional enhanced serial message format can be used by sensors, which require a serial communication channel with a larger data field and a larger set of message IDs. If the enhanced serial message format is used, serial data is transmitted in bit #2 and bit #3 of the status and communication nibble. A serial message frame stretches over 18 consecutive SENT data messages from the transmitter as shown in Figure 5.2.4.2-1. All 18 frames must be successfully received (no errors, calibration pulse variation, data nibble CRC error, etc.) for the serial value to be received.

The table in Figure 5.2.4.2-1 defines the frame format. The frame start of a serial message is indicated by the unique pattern “01111110” in bit 3 of the status and communication nibble (SENT message #18, #1 to #7). The first “1” in a series of six ones (after a “0”) indicates the start of a serial message frame. The “0” in SENT message #13 is defined to ensure the uniqueness of the start pattern. At initialization, it is recommended that the transmitter send the Serial data bit #3 sequence “0…01111110” (start with one or more SENT messages equivalent to #18). One starting “0” will guarantee the minimum latency start sequence and no interrupted message (due to sensor reset) will appear to be completed and incorrectly pass the CRC.

The serial message frame contains 20 bits of payload data. Two different configurations can be chosen determined by the configuration bit (serial data bit #3, serial communication nibble No. 8):

- 12-bit data and 8-bit message ID (configuration bit = 0)
- 16-bit data and 4-bit message ID (configuration bit = 1)




One serial message is composed of 18 SENT consecutive error-fee messages.

***Figure 5.2.4.2-1 – Construction of enhanced serial data message from 18 SENT messages***

All data (data field, message ID and CRC) that is transmitted in the serial message channel is sent in the order MSB (most significant bit) to LSB (least significant bit). The mapping and the order of the data bits, the message ID, the configuration bit and the CRC is detailed in Figure 5.2.4.2-2 for the configuration 12-bit data + 8-bit ID and in Figure 5.2.4.2-3 for the configuration 16-bit data and 4-bit ID.

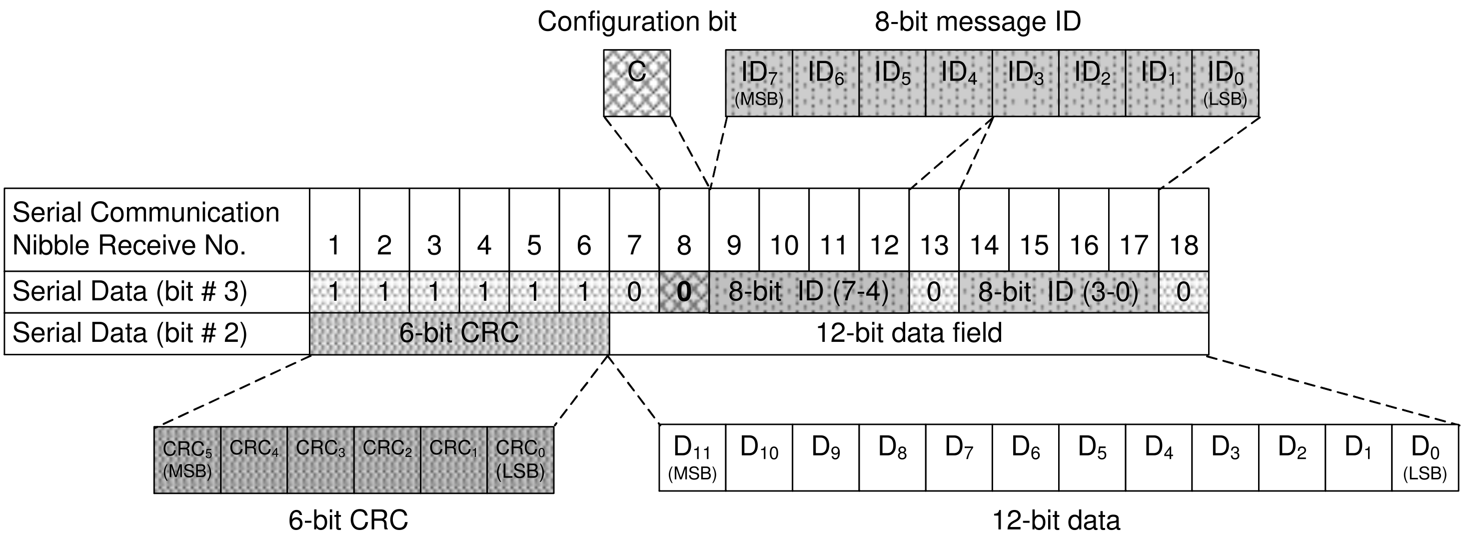



***Figure 5.2.4.2-2 – Enhanced serial message format with 12-bit data field and 8-bit message id***

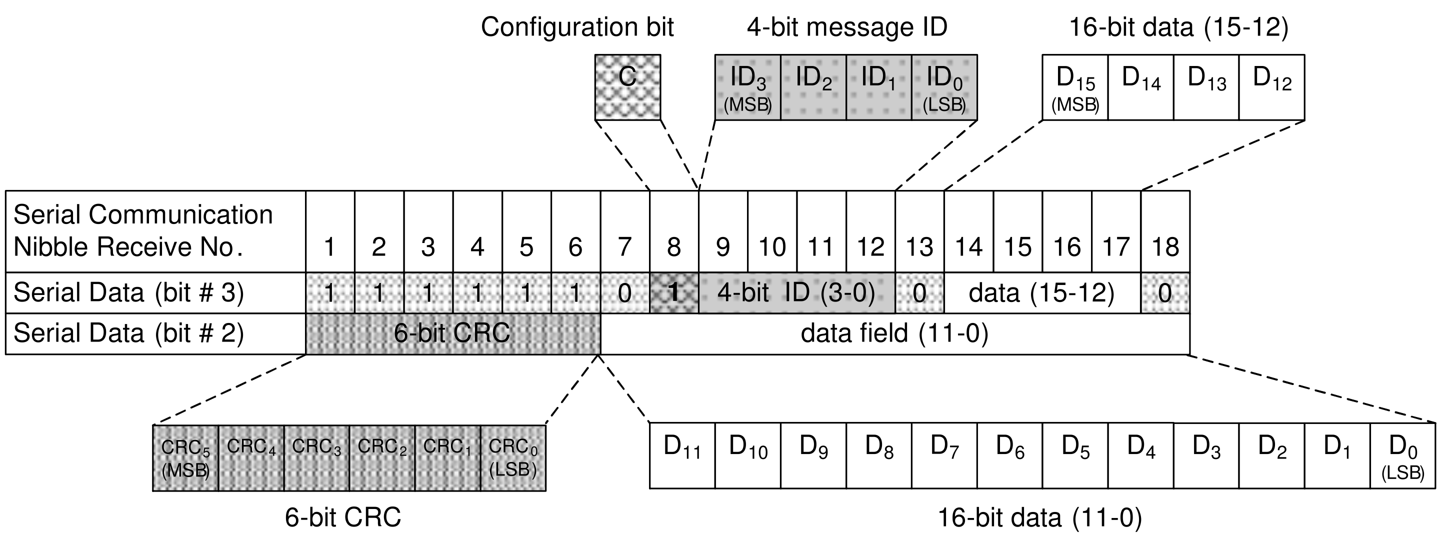



***Figure 5.2.4.2-3 – Enhanced serial message format with 16-bit data field and 4-bit message id***

#### 5.2.4.3 Enhanced Serial Message Format CRC

This CRC-value is computed as a function of the contents of Serial data message bits #2 and #3 for frames 7-18 (the 12-bit data field, the 8-bit message ID, the configuration bit and "0" bits 7, 13 and 18). For purposes of the CRC calculation, the bits shall be ordered: $\boldsymbol{m}$ = [Frame 7 bit#2, Frame 7 bit #3, Frame 8 bit #2, Frame 8 bit #3 .. Frame 18 bit #2, Frame 18, bit #3], see Figure 5.2.4.3-1. These bits contribute to the message data $\boldsymbol{m} = [m_0 \;\; m_1 \;\; m_2 \;\ldots\ldots\; m_{21} \;\; m_{22} \;\; m_{23}]$. Bit $m_0$ (which corresponds to frame bit $D_{11}$ is the first received bit that is read-in for CRC generation, as illustrated in the lower half of Figure 5.2.4.3-1.

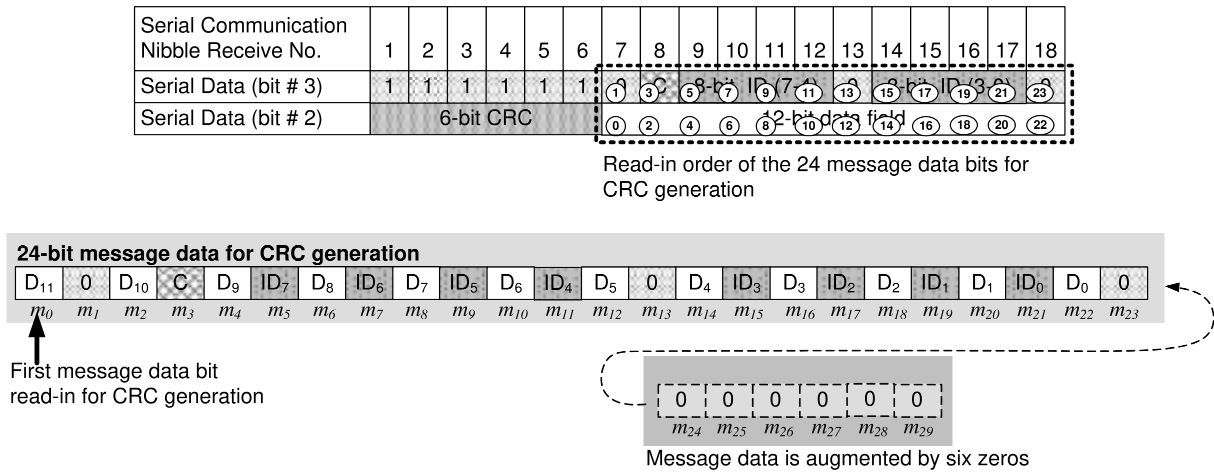



***Figure 5.2.4.3-1 – Order of the message bits, 24-bit message used for CRC generation***

The encoding is defined by the generating polynomial.

$G(x) = x^6+x^4+x^3+1$

with seed value 010101 (0x15 hex, 21 dec). For CRC generation, the message will be augmented by six zeros $\boldsymbol{m}_{crc} = [m_0\ m_1\ m_2 \ldots\ldots m_{21}\ m_{22}\ m_{23}\ 0\ 0\ 0\ 0\ 0\ 0]$ and the 6-bit data for the CRC calculation processed as follows.

data(1) = { m0 (=MSB), m1, m2, m3, m4, m5 (=LSB) }
data(2) = { m6 (=MSB), m7, ,m8, m9, m10, m11 (=LSB) }
etc.

The 6-bit CRC polynomial $x^6+x^4+x^3+1$ allows to detect all 1- bit, 2-bit, 3-bit, 5-bit errors. The number of undetected 4-bit errors depends on the number of message data bits as shown in the Table 5.2.4.3-1.

***Table 5.2.4.3-1 – Undetected 3 and 4 bit errors***

| **Message Data** | **Number of undetected 3-bit errors** | **Number of undetected 4-bit errors** |
|---|---|---|
| 24 bit | 0 | 365 |
| 24 bit + error detection of frame bits 7, 13 and 18 | 0 | 208 |

The number of possible 4-bit error patterns is $\binom{24}{4}$. The percentage of undetected 4-bit errors is approximately 3.4%.

The three serial data bits #3 of frames No. 7, 13 and 18 are known to be zero. If any of the three serial data bits #3 of nibbles No. 7, 13 and 18 differs from zero, the frame shall be considered as erroneous. This reduces the number of undetected 4-bit errors from 365 to 208.

5.2.4.3.1 Implementation Example for Enhanced Serial Message Format CRC

The CRC checksum can be implemented via a bit-wise exclusive OR with a 64 array lookup. The checksum is determined by reading-in 6-bit groups (in decimal representation) of the 24-bit message data in sequence, and then checksumming the result with an extra zero value (augmentation by six 0-bits). ). An example MATLAB implementation is:

```
  % Table-based CRC generation with Matlab
  % 24-bit message data
  % data = 4 x 6-bit  (Attention: Difference to 4-bit nibbles)
  % data(1) =   { m0 (=MSB), m1, m2, m3, m4, m5 (=LSB) }
  % data(2) =   { m6 (=MSB), m7, ,m8, m9, m10, m11 (=LSB) }
  % etc.

  % CRC table for poly = 0x59  (x^6+x^4+x^3+1)
  % Seed 0x15  (21)
    crc6_table = [ ...
   0  25  50  43  61  36  15  22  35  58  17   8  30   7  44  53 ...
  31   6  45  52  34  59  16   9  60  37  14  23   1  24  51  42 ...
  62  39  12  21   3  26  49  40  29   4  47  54  32  57  18  11 ...
  33  56  19  10  28   5  46  55   2  27  48  41  63  38  13  20 ];

CheckSum64 = 21; % initialize checksum
for I=1:4
  CheckSum64 = bitxor(uint8(data(I)), uint8(crc6_table(CheckSum64+1)));
end
% checksum with an extra "0" value (message is augmented by six zeros)
CheckSum64 = bitxor(uint8(0), uint8(crc6_table(CheckSum64+1)));
```

Examples of the CRC computation are given in Appendix B.2.

5.2.4.4 Non-usage of Serial Message

If no serial message is to be transmitted, both serial data bits shall be set to zero for both formats. If the transmitter needs to interrupt an already commenced serial data sequence, it is the responsibility of the transmitter to ensure that the serial message to be received fails the checksum calculation by corrupting either the data or CRC transmitted. This requirement is not intended to cover transmitter resets.

5.2.5 Checksum Nibble

See 5.4

5.2.6 Pause Pulse (Optional)

The SENT transmission may optionally contain an extra fill pulse which is transmitted after the checksum nibble. For example, this pulse can be used to create a SENT transmission with a constant number of clock ticks. If implemented, the pause pulse shall have the following properties:

- Minimum Length 12 ticks (equivalent to a nibble with 0 value)
- Maximum Length 768 ticks (3 * 256)

For a message with 6 data nibbles, the minimum constant length SENT message would be 282 ticks (assuming correct checksum calculation) for a pause pulses range of 12 ticks to 128 ticks. If a maximum message length of less than 1ms is desired then the transmitter clock tick time tolerance must be decreased to (-20%, +18%) (see Table 6.2.1-1).

## 5.3 Transmission and Reception of Message

### 5.3.1 Transmission of Message

Each specific implementation of the encoding scheme (e.g., throttle sensors, pedal sensors, mass air sensors, etc.) shall have a defined sequence, a calibration pulse followed by a constant number of nibbles (and possibly a pause pulse). The transmitting module shall send the defined sequence repeatedly.

### 5.3.2 Adjustment of Received Message Nibble Lengths

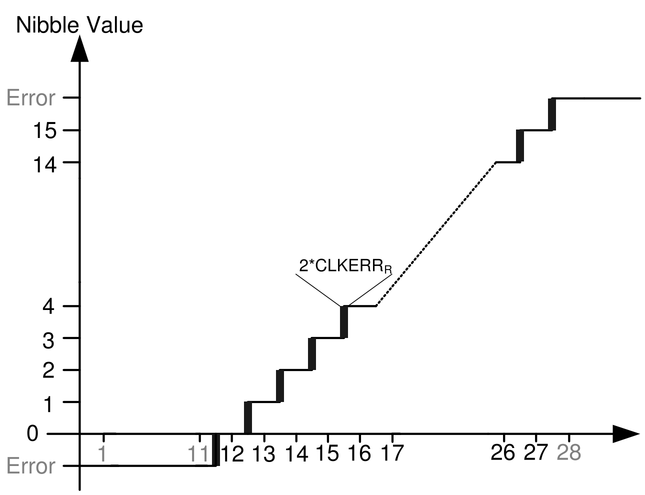
The ECU shall receive the encoded signal by monitoring the time between falling edges (the pulse period). The measurement of a pulse period equal to the calibration pulse ±20% shall signal the start of a new message. The actual period of the calibration pulse received shall be used to adjust the period of the succeeding nibble pulses within the message. This calibration pulse is required to correct for clock tick time variation in the sensor/transmitting device.

The ratio Rcal between the measured calibration pulse period and the nominal (56 clock tick) period shall be calculated. Each of the message's remaining measured (nibble) pulse periods is then divided by this ratio to correct them back to the equivalent nominal period. These corrected periods then have the 12 clock tick minimum period subtracted and the resulting data length period divided by the nominal clock tick period and rounded to an integer value to obtain the data nibble value. The process is summarized by the following algorithm:

Rcal = Measured Calibration pulse period/(56*Ticknom)
Data Value N = Round[(Measure pulse period N/Rcal – 12*Ticknom)/Ticknom]

Figure 5.3.2-1 shows the interpretation in a graphical way.

This section is intended to describe the theory of how data nibbles are to be interpreted by a receiver. Other methods which achieve this same end result are also acceptable.




$$\frac{Received\ Nibble\ Period}{Received\ Calib\ Pulse\ Period} * 56$$

***Figure 5.3.2-1 – Quantisation of sampled and calibrated nibble***

### 5.3.3 Received Message Diagnostics

The receiver module shall monitor the message for the following conditions. If any of the below conditions are met, the receiver module shall declare that a message error has occurred and ignore the entire message. Note, not all diagnostics need to be run for each individual message if the message has already been deemed to be in error by a previous diagnostic.

- Calibration pulse length < 56 clock ticks – 25% or > 56 clock ticks + 25%.

- Not the expected number of falling edges between calibration pulses. (Message length is pre-defined by each sensor device)

- Checksum error.

- Any nibble data values measured as < 0 or > 15.

- One of the following two conditions
  - Successive calibration pulses differ by > +1.5625% (1/64) or < -1.5625%. See section 5.3.3.1 for more detail
  - (for pause pulse with fixed message length only) Ratio of calibration pulse to message length varies by >1.5625% or <-1.5625%. To minimize message latency, this diagnostic can be used as the receiver does not need to wait for the next calibration pulse to diagnose the current received frame. It is only recommended for precisely controlled constant message length.

Any of the errors above shall also cause the receiver to begin searching for a valid calibration pulse to re-synchronize reception to the transmitting device start of message. The diagnostics above will allow the reception of message with ±25% clock tick timing range. However, only systems with a maximum ±20% timing range will be certified as meeting the SENT requirements. Note, messages exceeding the 20% clock tolerance should be received correctly but may exceed the 1 ms transmission time target (3 µs clock tick).

For systems with a pause pulse, during synchronization or re-synchronization of reception, if calibration pulses are detected one immediately following the other, the first calibration pulse shall be ignored as it may be a pause pulse with duration matching the calibration pulse range.

#### 5.3.3.1 Successive Calibration Pulses

The diagnostic for detecting successive calibration pulses differing by > 1.5625% shall be implemented by one of two methods. Options are described with reference to Figure 5.3.3.1-1 where the calibration pulse of Frame n differs from Frames n-1, n+1 and n+2 by > 1.5625% threshold.

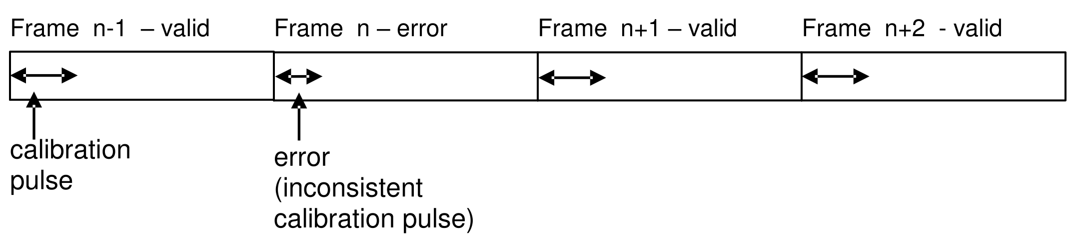



***Figure 5.3.3.1-1 – Error pattern for successive calibration pulse detection***

Preferred Option:

The calibration pulse of current frame is compared to the calibration pulse of the preceding frame. If they differ by > the threshold 1.5625%, the preceding frame is declared to be bad and ignored. No special initialization or resynchronization is required.

i. Calibration pulse n-1 is compared to calibration pulse n and the entire message of Frame n-1 is declared to be bad even though the calibration pulse is correct.
ii. Calibration pulse n is compared to calibration pulse n+1 and the entire message of Frame n is declared to be bad.
iii. Calibration pulse n+1 is compared to calibration pulse n+2. Message n+1 passes the test.

Advantages of the Preferred Option:

- “Too many nibbles between calibration pulses” can be detected
- Clock variation affecting calibration pulse and immediately preceding or succeeding nibbles can be detected.

Disadvantages of the Preferred Option:

- Potentially, a valid frame (frame n-1) could be declared to be bad
- There is an extra latency of the length of the calibration pulse in processing the message in the receiver (see Figure 5.3.3.1-2).

Option 2 (should only be used if extra latency to process second calibration pulse cannot be tolerated)

The calibration pulse of current frame is compared to the calibration pulse of the last valid preceding frame (i.e., frame with no errors). If they differ by > the threshold 1.5625%, the frame is declared to be bad and ignored.

Initialization: The first calibration pulse shall be considered valid unless the message frame contains other errors.

Resynchronization: On the third successive calibration pulse error, the current calibration pulse value shall be considered as valid and the message accepted unless the message frame contains other errors.

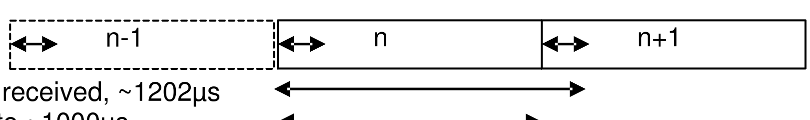
i. Calibration pulse n is compared to calibration pulse n-1 and the entire message of Frame n is declared to be bad.
ii. Calibration pulse n+1 is compared to calibration pulse n-1 and the entire message of Frame n+1 is declared to be valid.
iii. Calibration pulse n+2 is compared to calibration pulse n+1. Message n+2 passes the test.

Advantages of Option 2:

- Message processing can be completed within 1 ms worst case timing

Disadvantages of the Option 2:

- “Too many nibbles between calibration pulses” cannot be detected as the next calibration pulse has not been received.
- Clock variation affecting calibration pulse and immediately succeeding nibbles cannot be detected by calibration pulse comparison.




1) CRC & next calibration pulse received, ~1202µs
2) CRC received, frame complete ~1000µs

***Figure 5.3.3.1-2 – Successive calibration pulse detection***

#### 5.3.4 Reset

Should the transmitting module reset for any reason, it shall start transmitting a new message immediately following its initialization procedure. The output signal shall be held in the high state until initialization is completed.

If the SENT interface of the transmitting module is initialized after a reset, while the measurement unit of the transmitting module is still initializing, the sensor shall not transmit a value within the measurement data range. It is recommended to tranmit the initialization message (value zero), as defined in E.1.

### 5.4 Checksum Nibble Details

The Checksum nibble is a 4 bit CRC of the Data nibbles only. The Status and Communication Nibble is not included in the CRC calculation.

The CRC is calculated using polynomial x^4 + x^3 + x^2 + 1 with seed value of 0101. For CRC generation, the data nibbles will be augmented by a nibble with data = 0. Refer to Appendix B.1 for details.

#### 5.4.1 Errors Detected and Not Detected by Checksum

The following bit flip errors are detectable using the polynomial:

1. All single bit errors.
2. All odd number of nonconsecutive bit errors.
3. All single burst errors of length ≤ 4.
4. 87.5% of single burst errors of length = 5.
5. 93.75% of single burst errors of length > 5.

The fault mode of most concern for the signal encoding scheme is a shifted falling edge which would cause one nibble to appear to be a longer (shorter) period followed by a nibble with a shorter (longer) period than intended by the transmitting device. The CRC detects all cases of 1, 2, 4, 8, 9, 10, 12, 13, 14 and 15 count shifts over two nibbles.

The CRC polynomial will not detect all cases of 3, 5, 6, 7, 11 count shifts. These undetectable patterns are listed in Table 5.4.1-1. For example, row 3 of Table 3 has error pattern 101 followed by 11. If nibble 1 was 4 (100) and nibble 2 0 (000), after the bit flips nibble 1 becomes 1 (001) and nibble 2 becomes 3 (011). Nibble 1 has shifted by –3 and nibble 2 by +3. It is assumed that the risk associated with not detecting these types of faults is acceptable given the high update rate available to the encoding scheme.

***Table 5.4.1-1 – Bit Flip patterns over two nibbles not detectable using the CRC polynomial***

| Nibble 1 Error Pattern | | Nibble 2 Error Pattern | |
|---|---|---|---|
| 8 | 1000 | 1 | 1 |
| 13 | 1101 | 2 | 10 |
| 5 | 101 | 3 | 11 |
| 7 | 111 | 4 | 100 |
| 15 | 1111 | 5 | 101 |
| 10 | 1010 | 6 | 110 |
| 2 | 10 | 7 | 111 |
| 14 | 1110 | 8 | 1000 |
| 6 | 110 | 9 | 1001 |
| 3 | 11 | 10 | 1010 |

| Nibble 1 Error Pattern | | Nibble 2 Error Pattern | |
|---|---|---|---|
| 11 | 1011 | 11 | 1011 |
| 9 | 1001 | 12 | 1100 |
| 1 | 1 | 13 | 1101 |
| 4 | 100 | 14 | 1110 |
| 12 | 1100 | 15 | 1111 |

Table 5.4.1-2 lists the missed detection rate for the shifted edge failure resulting in a longer followed by a correspondingly equally shortened pulse. Table 5.4.1-3 lists the missed detection rate for the same situation except that the shifted edge results in an offset of one in the change of the shortened versus the longer pulse.

***Table 5.4.1-2 – Checksum missed detection rate, edge shift between two nibbles (valid frames only, recommended implementation)***

| Shift Nibble X | Shift Nibble X+1 | Not detected [%] |
|---|---|---|
| -15 | 15 | 0.00 |
| -14 | 14 | 0.00 |
| -13 | 13 | 0.00 |
| -12 | 12 | 0.00 |
| -11 | 11 | 16.00 |
| -10 | 10 | 0.00 |
| -9 | 9 | 8.16 |
| -8 | 8 | 0.00 |
| -7 | 7 | 4.94 |
| -6 | 6 | 16.00 |
| -5 | 5 | 6.61 |
| -4 | 4 | 0.00 |
| **-3** | **3** | **11.83** |
| **-2** | **2** | **0.00** |
| **-1** | **1** | **0.00** |
| **1** | **-1** | **0.00** |
| **2** | **-2** | **0.00** |
| **3** | **-3** | **11.83** |
| 4 | -4 | 0.00 |
| 5 | -5 | 6.61 |
| 6 | -6 | 16.00 |
| 7 | -7 | 4.94 |
| 8 | -8 | 0.00 |
| 9 | -9 | 8.16 |
| 10 | -10 | 0.00 |
| 11 | -11 | 16.00 |
| 12 | -12 | 0.00 |
| 13 | -13 | 0.00 |
| 14 | -14 | 0.00 |
| 15 | -15 | 0.00 |

***Table 5.4.1-3 – Checksum missed detection rate, asymmetrical edge shift offset by 1 between two nibbles (valid frames only, recommended implementation)***

| Shift Nibble X | Shift Nibble X+1 | Not detected [%] |
|---|---|---|
| -14 | 15 | 0.00 |
| -13 | 14 | 0.00 |
| -12 | 13 | 33.33 |
| -11 | 12 | 0.00 |
| -10 | 11 | 0.00 |
| -9 | 10 | 0.00 |
| -8 | 9 | 0.00 |
| -7 | 8 | 0.00 |
| -6 | 7 | 17.78 |
| -5 | 6 | 0.00 |
| -4 | 5 | 3.03 |
| -3 | 4 | 10.26 |
| **-2** | **3** | **8.79** |
| **-1** | **2** | **0.00** |
| **0** | **1** | **0.00** |
| **1** | **0** | **0.00** |
| **2** | **-1** | **7.62** |
| **3** | **-2** | **8.79** |
| 4 | -3 | 2.56 |
| 5 | -4 | 12.12 |
| 6 | -5 | 0.00 |
| 7 | -6 | 0.00 |
| 8 | -7 | 0.00 |
| 9 | -8 | 0.00 |
| 10 | -9 | 0.00 |
| 11 | -10 | 0.00 |
| 12 | -11 | 20.00 |
| 13 | -12 | 0.00 |
| 14 | -13 | 0.00 |
| 15 | -14 | 0.00 |
| | | |
| -15 | 14 | 0.00 |
| -14 | 13 | 0.00 |
| -13 | 12 | 0.00 |
| -12 | 11 | 20.00 |
| -11 | 10 | 0.00 |
| -10 | 9 | 0.00 |
| -9 | 8 | 0.00 |
| -8 | 7 | 0.00 |
| -7 | 6 | 0.00 |
| -6 | 5 | 0.00 |
| -5 | 4 | 12.12 |
| -4 | 3 | 2.56 |
| **-3** | **2** | **8.79** |
| **-2** | **1** | **7.62** |
| **-1** | **0** | **0.00** |
| **0** | **-1** | **0.00** |
| **1** | **-2** | **0.00** |
| **2** | **-3** | **8.79** |

| Shift Nibble X | Shift Nibble X+1 | Not detected [%] |
|---|---|---|
| 3 | -4 | 10.26 |
| 4 | -5 | 3.03 |
| 5 | -6 | 0.00 |
| 6 | -7 | 17.78 |
| 7 | -8 | 0.00 |
| 8 | -9 | 0.00 |
| 9 | -10 | 0.00 |
| 10 | -11 | 0.00 |
| 11 | -12 | 0.00 |
| 12 | -13 | 33.33 |
| 13 | -14 | 0.00 |
| 14 | -15 | 0.00 |

#### 5.4.2 CRC Implementation

SENT receivers shall be capable of implementing both checksum implementations below but sensors should use the recommended implementation. The same 4-bit CRC implementation shall be used for both a per frame basis (i.e. checksum of all data nibbles) and for Short Serial Message Format checking.

##### 5.4.2.1 Legacy Implementation (as in revision FEB2008 and older)

The CRC checksum can be implemented as a series of shift left by 4 (multiply by 16) followed by a 256 array lookup. An example MATLAB implementation is:

numNibbles is the number of nibbles to be check summed. The data array contains the nibble to be check summed and the 256 table is given below in MATLAB form:

```
% CRC 4 nibble lookup table
CRC4Table= [ ...
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 ...
13 12 15 14 9 8 11 10 5 4 7 6 1 0 3 2 ...
7 6 5 4 3 2 1 0 15 14 13 12 11 10 9 8 ...
10 11 8 9 14 15 12 13 2 3 0 1 6 7 4 5 ...
14 15 12 13 10 11 8 9 6 7 4 5 2 3 0 1 ...
3 2 1 0 7 6 5 4 11 10 9 8 15 14 13 12 ...
9 8 11 10 13 12 15 14 1 0 3 2 5 4 7 6 ...
4 5 6 7 0 1 2 3 12 13 14 15 8 9 10 11 ...
1 0 3 2 5 4 7 6 9 8 11 10 13 12 15 14 ...
12 13 14 15 8 9 10 11 4 5 6 7 0 1 2 3 ...
6 7 4 5 2 3 0 1 14 15 12 13 10 11 8 9 ...
11 10 9 8 15 14 13 12 3 2 1 0 7 6 5 4 ...
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 ...
2 3 0 1 6 7 4 5 10 11 8 9 14 15 12 13 ...
8 9 10 11 12 13 14 15 0 1 2 3 4 5 6 7 ...
5 4 7 6 1 0 3 2 13 12 15 14 9 8 11 10 ...
];

CheckSum = 5; % initialize checksum
for I=1:numNibbles
    tempCS = data(I)+CheckSum*16;
    CheckSum = CRC4Table(tempCS+1);
end
```

This implementation is included for backwards compatibility. Based on valid patterns, it is subject to a 42 % missed detection rate of edge shifts between the final data nibble and the checksum nibble and is not recommended.

#### 5.4.2.2 Recommended Implementation

The recommended CRC calculation utilizes the same seed value and polynomial as the legacy calculation but also augments the message data with an extra 4 zero bits in the CRC calculation.

Example implementations are given below. Other implementations are possible. For example, the CRC can also be implemented as a series of bit-wise XOR and shift operations or a reduced sized table can be used. Example checksum values are given in APPENDIX B.

##### 5.4.2.2.1 Implementation using 256 Element Array

The CRC checksum can be implemented as a series of shift left by 4 (multiply by 16) followed by a 256 element array lookup. The checksum is determined by using all data nibbles in sequence and then check summing the result with an extra zero value. An example MATLAB implementation is (same CRC4Table as in section 5.4.2.1):

```
CheckSum = 5; % initialize checksum
for I=1:numNibbles
    tempCS = data(I)+CheckSum*16;
    CheckSum = CRC4Table(tempCS+1);
end
% checksum with an extra “0” value
tempCS = CheckSum*16;
CheckSum = CRC4Table(tempCS+1);
```

##### 5.4.2.2.2 Implementation using 16 Element Array

The CRC checksum can be implemented as a bit-wise exclusive OR with a 16 element array lookup. The checksum is determined by using all data nibbles in sequence and then check summing the result with an extra zero value. An example MATLAB implementation is:

```
numNibbles=6; data=[7 4 8 7 4 8]; % initialization & data

%CRC table for poly = 0x1d
crc4_table = [0 13 7 10 14 3 9 4 1 12 6 11 15 2 8 5 ];

CheckSum16 = 5; % initialize checksum
for I=1:numNibbles
  CheckSum16 = bitxor(uint8(data(I)), uint8(crc4_table(CheckSum16+1)));
end
% checksum with an extra “0” value
CheckSum16 = bitxor(uint8(0), uint8(crc4_table(CheckSum16+1)));

% expected result for 748748: Checksum16=3
```

## 6. SENT PHYSICAL LAYER OPERATION

The physical layer is responsible for providing a method of transferring digital data encoded as time between two falling edges of a signal through the communication medium. This results in repeatable and time accurate switching of the input circuit of the physical layer interface which consists of power, ground and signal wires.

### 6.1 Normal Communication Mode and Transmission Rate

Transmission bit rate is variable depending on the clock tick period, data values sent and the transmitter nominal clock tick time variance and +/-20% clock tolerance. The longest transmission time for 6 Data nibbles is 270 clock ticks or 972 μs at a 3 μs clock tick period, this results in a worst case transmission rate of ~24.7 kBits/s. Similarly, the shortest valid 6 Data nibble message is 154 clock ticks which, at –20% clock tick time tolerance, results in a transmission rate of ~64.9 kBits/sec. These calculations ignore the contribution made by Serial data transmission (see 5.2.4).

## 6.2 Clock Tolerance

### 6.2.1 Transmitter–Receiver Communications

Table 6.2.1-1 lists the timing and resolution requirements for the transmitting and receiving modules. To avoid requiring a high cost oscillator in the transmitter, the transmitter clock tick time variation is allowed to be large. Requirements for Clock Drift Error and Jitter at intermediate clock tick periods shall be linearly interpolated between the minimum and maximum clock tick periods in Table 6.2.1-1.

***Table 6.2.1-1 – Communication clock tolerance***

| Device Type | Symbol | Parameter | Clock Tolerance |
|---|---|---|---|
| Transmitter | $\Delta T_{osc}/T_{osc}$ | Clock Accuracy | ≤ ±20% on clock tick time without pause pulse<br>> -20%, < + 18%, on clock tick time, with pause pulse (optional) |
| Transmitter | $ClkErr_T$ | Clock Jitter and Drift Error (including systematic phase errors) | ≤ 0.3 µs(1) variation of maximum nibble time(2) compared to the expected time derived from the calibration pulse time at a 3 µs clock tick(3) |
| Transmitter | $Tick_{nom}$ | Clock Tick Period Ranges | 3 to 90 µs nominal(3) |
| Receiver | $ClkErr_R$ | Clock Jitter and Drift Error (including sampling quantization) | ≤ 0.3 µs(1) variation of maximum nibble time(2) compared to the expected time derived from the calibration pulse time at a 3 µs clock tick(3) |

[1] In case of Gaussian distributed errors the time variation limit shall be compared to 6σ, where σ is the standard deviation of the absolute difference of measured maximum nibble times to the expected time derived from the corresponding calibration pulses. In case of other distributions the same error margin as for 6σ Gaussian distribution shall be guaranteed. The value for the transmitter is defined for an ideal receiver and the value for the receiver for an ideal transmitter. The value for the transmitter is defined at any constant trigger level as specified in Table 6.3.2-1 c and e.

[2] The maximum nibble time corresponds to nibbles with 27 clock tick nominal time.

[3] For clock tick times greater than 3 µs, transmitter and receiver "Clock Jitter and Drift Error" can be increased linearly proportionally to the 3 µs clock tick values (see Figure 6.2.1-1).

NOTE: Increased values of the clock jitter and drift error compared to Revision JAN2010 are due to a more precise and inclusive definition of the parameters. This change does not affect the design both of the transmitter and receiver.

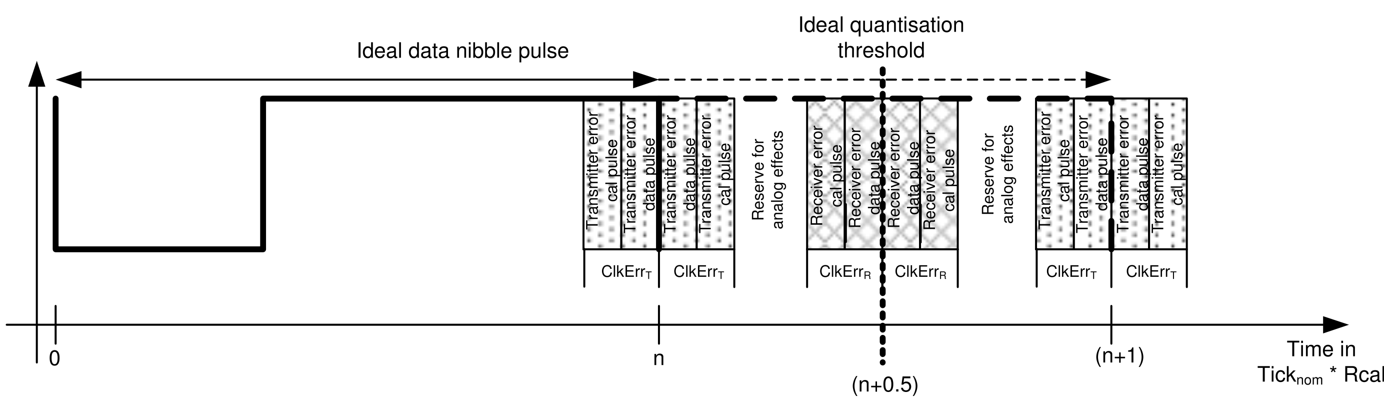



***Figure 6.2.1-1 – Illustration of clock error accumulation as given in Table 6.2.1-1.***

## 6.3 Electrical Interface Requirements

The SENT signal can be seen as a nominal 5 V square wave signal, but for low radiated emissions, signal shaping is required. To minimize ground and supply offset effects, the receiving device shall provide a regulated 5 V supply and ground reference to the transmitting device as specified below. This signal is intended to operate in automotive under-hood control system applications. The driver and receiver shall be protected from shorts to ground or vehicle voltages and various transient pulses as required below and by the specific vehicle EMC requirements.

Figure 6.3-1 illustrates a possible equivalent circuit for a SENT transmitter and receiver. The Transmitter portion of this circuit is an example only and should not be taken as a direct implementation requirement. All interface requirements in this document shall be met whether the implementation follows the example circuit topology or not.

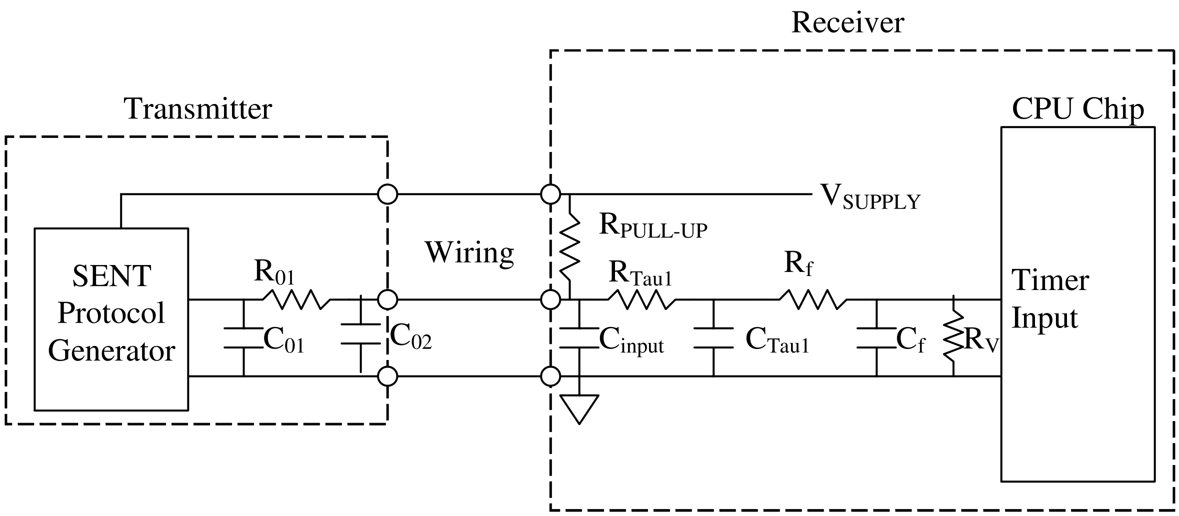



***Figure 6.3-1 – Example SENT system interface circuit topology***

### 6.3.1 Transmitter Driver Requirements

The transmitting device shall provide a driver with a configuration and technology suitable for the automotive environment. The driver shall ensure that the signal pulse parameter as defined in Figure 6.3.1-2, Table 6.3.1-1, Table 6.3.1-2 and Table 6.3.1-3 are met when driving into the receiver passive load defined in section 6.3.2. Protection requirements from damage due to voltage or current conditions on the output signal outside those defined in Table 6.3.1-1 shall be defined by the OEM specification. The requirements in Table 6.3.1-1, Table 6.3.1-2 and Table 6.3.1-3 shall be met over the entire transmitter environmental requirement range. Any loss or corruption of the transmitted signal due to open circuit or short of any of the transmitter pins shall be detected by the receiver via the message diagnostic tests defined in Section 5.3.3.

In addition an EMC filter consisting of a capacitor followed by a resistor in series with the output pin is recommended to attenuate RF energy coupled on the external signal line. See Figure 6.3-1. In order to meet radiated emission EMC requirements, it is expected that a “shaped” waveform (controlled slope and corner rounding) will need to be driven by the transmitter. Figure 6.3.1-1 illustrates an example shaped waveform (falling and rising edge) that could meet emissions requirements. SENT transmit pulse parameters are explained in Figure 6.3.1-2.

The transmitter shall meet the input threshold level at the receiver with maximum time constants of the input filters.

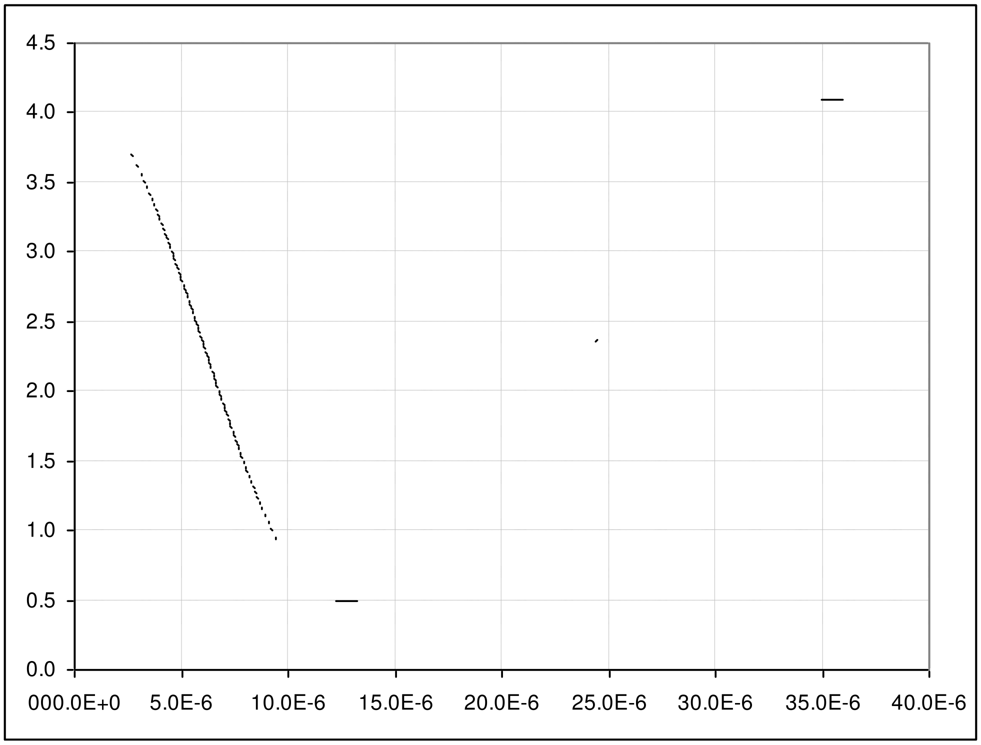



***Figure 6.3.1-1 – Example SENT shaped waveform transmitter output***

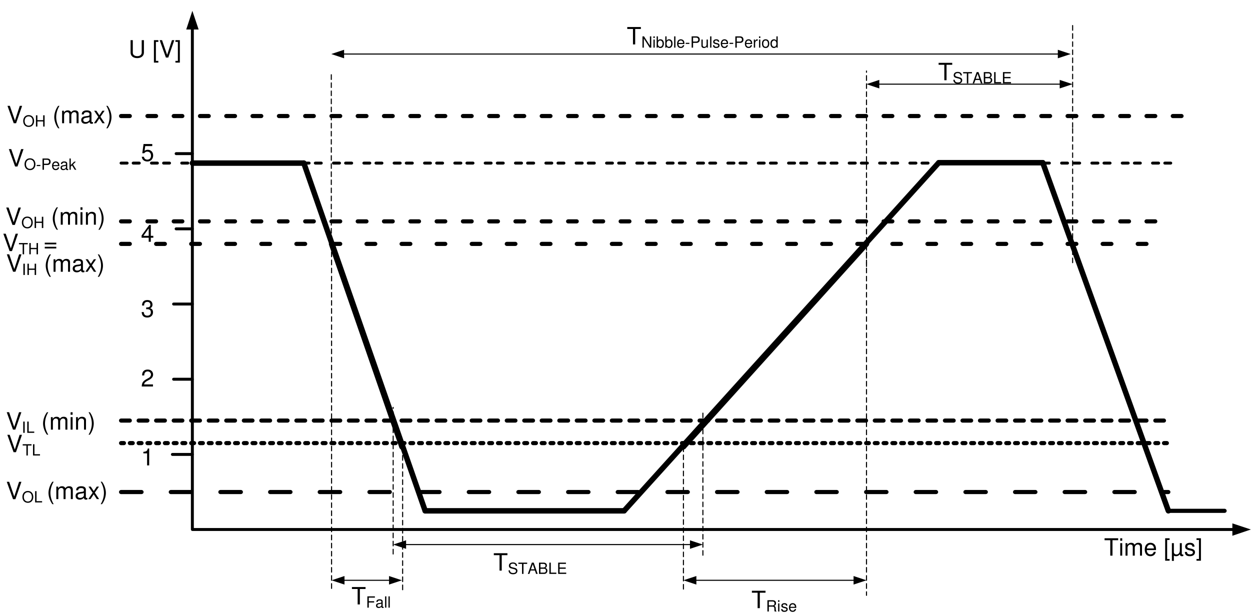



***Figure 6.3.1-2 – SENT transmit pulse parameters***

***Table 6.3.1-1 – Transmitter driver requirements***

| Parameter | | | Limits (including component tolerances) Min | Max | Units | Test Conditions / Definitions (on wire) |
|---|---|---|---|---|---|---|
| a. | $V_{OL}$ | = | | 0.5 | V | Low state Voltage @ 0.52 mA DC load current |
| b. | $V_{OH}$ | = | 4.1 | | V | High state Voltage @ 0.10 mA DC load current |
| c1. | $I_{SUP}$ | = | | 50 | mA | Average current consumption from Receiver 5V supply over one message |
| c2. | $I_{GND}$ | | | 50 | mA | Average current through (Signal) Ground Line over one message |
| d. | $I_{SUP\text{-}RIPPLE}$ | = | | 9.0[3] | mA | Peak-to-peak variation in supply current consumption over one message at frequencies up to $f_c$= 30 kHz measured as described in C.2 |
| e1. | $T_{FALL}$ | = | | 6.5[1] | µs | From $V_{TH}$ = 3.8 V to $V_{TL}$ = 1.1 V, $I_{SUP} \leq$ 20 mA |
| e2. | $T_{FALL}$ | = | | 5.0[1] | µs | From $V_{TH}$ = 3.8 V to $V_{TL}$ = 1.1 V, 20 mA < $I_{SUP} \leq$ 50 mA |
| f. | $T_{RISE}$ | = | | 18[1] | µs | From $V_{TL}$ = 1.1V to $V_{TH}$ = 3.8V |
| g. | $\Delta T_{FALL}$ | = | | 0.1[1] | µs | Edge to edge jitter with static environment for any pulse period |
| h. | $T_{STABLE}$ | = | 6[2] | | µs | Signal stabilization time below $V_{IL}$ (min) = 1.39V (low signal) or above $V_{IH}$ (max) = 3.8V (high signal) |

[1] For 3µs nominal clock tick including clock accuracy. For higher clock tick times these values can be increased proportionally.

[2] If "$T_{FALL}$", "$T_{RISE}$" or "$\Delta T_{FALL}$" are increased, $T_{STABLE}$ has to be increased by the same ratio as well.

[3] Devices or loads on the transmitter side other than one SENT output circuit may lead to higher current ripple, uncorrelated to that SENT transmission. In this case the current ripple has to be specified and considered in the design of the supply on the receiver side.

In case the transmitter is supplied from the receiver with a regulated 5V power supply as defined in Table 6.3.2-1 the transmitter shall satisfy those requirements defined in 6.3.1, that specify the SENT pulse shape together with the specific requirements in Table 6.3.1-1. The circuit topology for this case is depicted in Figure 6.3.1-3.

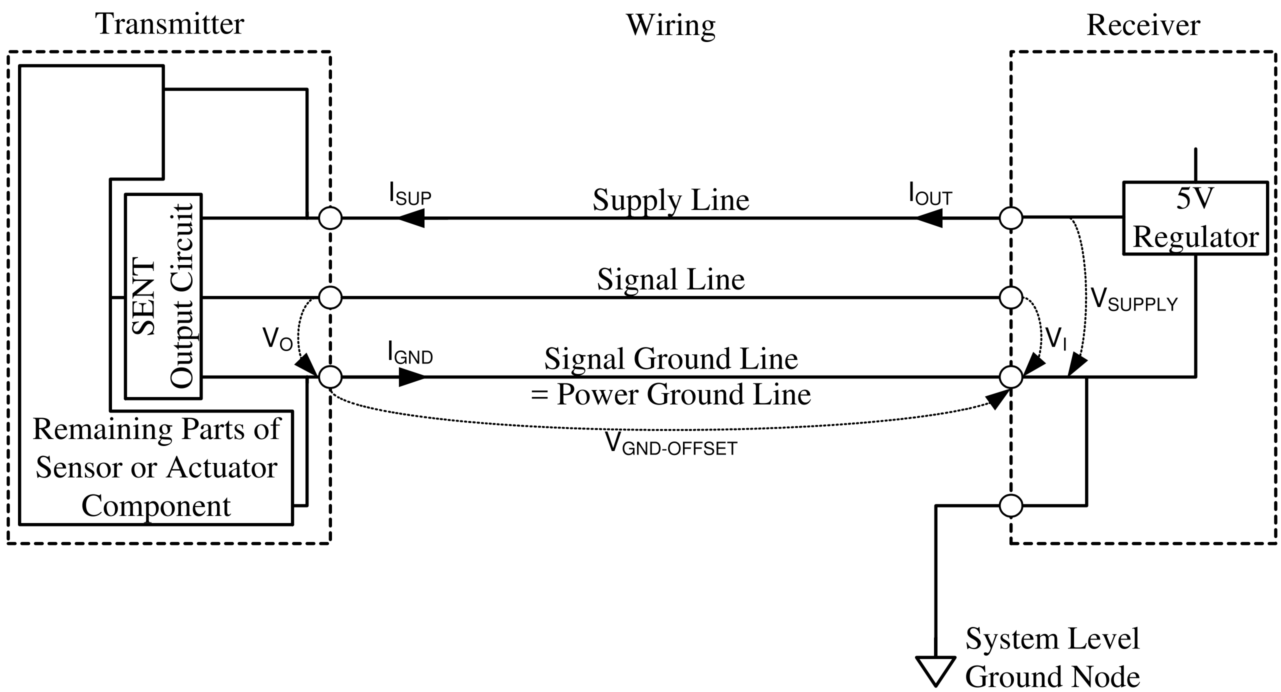



*Figure 6.3.1-3 – SENT system interface circuit topology for a regulated 5v power supply of the transmitter by the receiver*

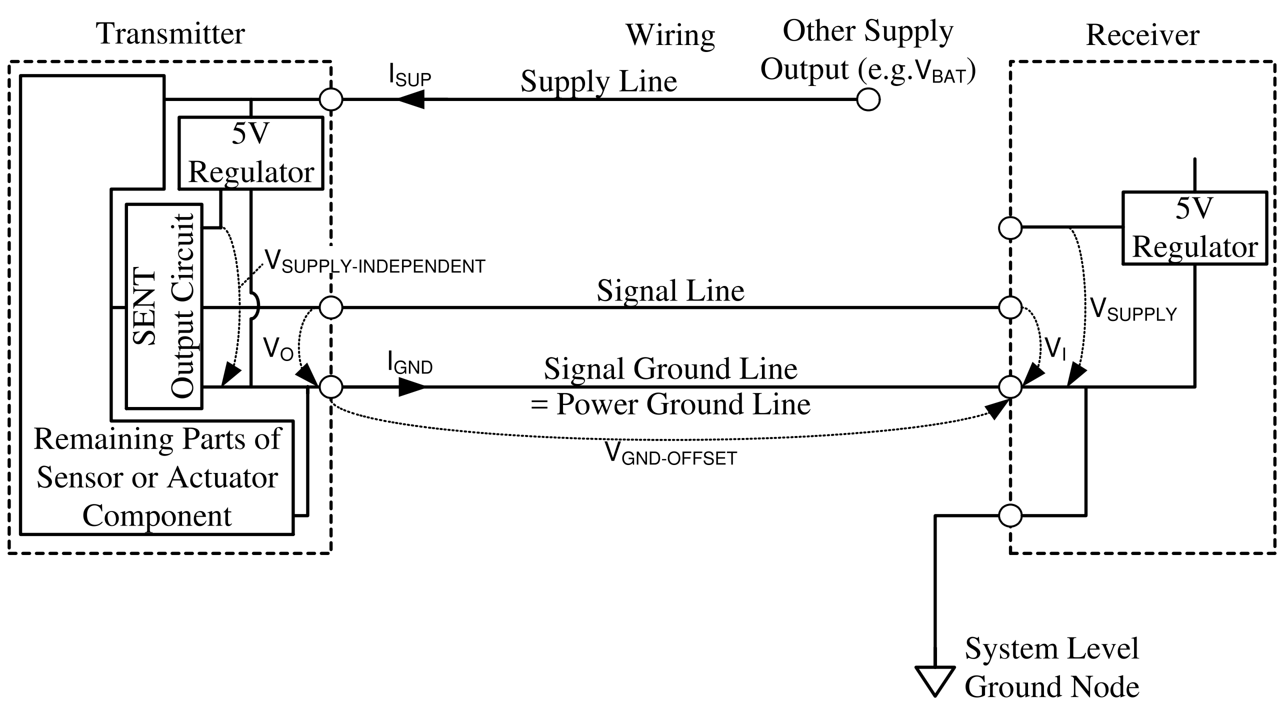



*Figure 6.3.1-4 – SENT System Interface Circuit Topology for an Independent 5V Power Supply in the Receiver with Identical Signal Ground Line and Power Ground Line*

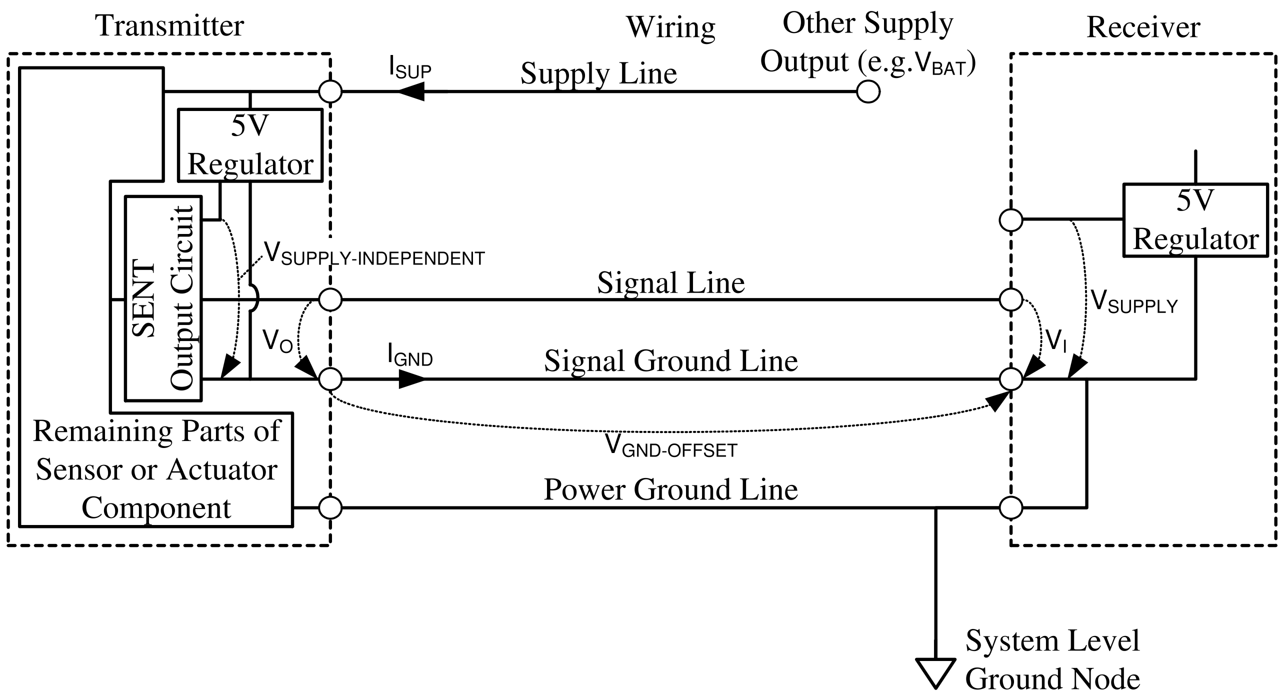



***Figure 6.3.1-5 – SENT system interface circuit topology for an independent 5v power supply in the receiver with separate signal ground line and power ground line***

The transmitting device may optionally implement a 5V regulator to power the SENT signal driver independently of the power supply from the receiver. In this option, the transmitter shall satisfy those driver requirements defined in 6.3.1, that specify the SENT pulse shape together with the specific requirements for independent 5V powered transmitters, as specified in Table 6.3.1-2. These requirements ensure that the optional transmitter interface will remain compatible with the standard SENT receiver requirements (Table 6.3.2-1). Additional specific requirements for independent 5V powered transmitters that require higher power consumption are specified in Table 6.3.1-3.

Note that while the 5V power may be independent of the receiver, the SENT transmitter shall connect its signal ground line to the ground signal provided by the SENT receiver to ensure that excessive ground offset will not disturb the SENT communication. This may be done with the signal ground line being identical to the power ground line as shown in Figure 6.3.1-4, if the transmitter design ensures sufficiently small ground offset $V_{GND\text{-}OFFSET}$ as defined in Table 6.3.1-3 m. Otherwise the power ground line must be isolated from the signal ground line on the transmitter side as shown in Figure 6.3.1-5, and the transmitter design must accommodate any voltage offset between these two signals.

***Table 6.3.1-2 – Independent 5v powered transmitter requirements***

| Parameter | | | Limits (including component tolerances) | | Units | Test Conditions / Definitions (on wire) |
|---|---|---|---|---|---|---|
| | | | Min | Max | | |
| a. | $V_{OL}$ | = | | 0.5 | V | Low state Voltage @ 0.52 mA DC load current |
| b. | $V_{OH}$ | = | 4.1 | 5.5 | V | High state Voltage @ 0.10 mA DC load current |
| c. | $I_{GND}$ | = | | 50 | mA | Average current through Signal Ground Line over one message. |
| e1. | $T_{FALL}$ | = | | 6.5(1) | µs | From $V_{TH}$ = 3.8 V to $V_{TL}$ = 1.1 V, $I_{SUP} \le 20$ mA |
| e1. | $T_{FALL}$ | = | | 5.0(1) | µs | From $V_{TH}$ = 3.8 V to $V_{TL}$ = 1.1 V, 20 mA < $I_{SUP} \le 50$ mA |
| f. | $T_{RISE}$ | = | | 18(1) | µs | From $V_{TL}$ = 1.1V to $V_{TH}$ = 3.8V |

| Parameter | | | Limits (including component tolerances) Min | Max | Units | Test Conditions / Definitions (on wire) |
|---|---|---|---|---|---|---|
| g. | $\Delta T_{FALL}$ | = | | 0.1 (1) | μs | Edge to edge jitter with static environment for any pulse period |
| h. | $T_{STABLE}$ | = | 6 (2) | | μs | Signal stabilization time below $V_{IL}$ (min) = 1.39V (low signal) or above $V_{IH}$ (max) = 3.8V (high signal) |
| i. | $V_{O\text{-}Peak\text{-}Range}$ | | | 0.3 | V | Stability of pulse peaks $V_{O\text{-}Peak\text{-}Range} = V_{O\text{-}Peak}(max) - V_{O\text{-}Peak}(min)$ over one SENT message at $T_{TICK}$ = 3 μs |
| j. | $I_{GND\text{-}RIPPLE}$ | = | | 9.0 (3) | mA | Peak-to-peak variation in ground current over one message at frequencies up to $f_c$= 30 kHz |

[1] For 3μs nominal clock tick including clock accuracy. For higher clock tick times these values can be increased proportionally.

[2] If "$T_{FALL}$", "$T_{RISE}$" or "$\Delta T_{FALL}$" are increased, $T_{STABLE}$ has to be increased by the same ratio as well.

[3] Devices or loads on the transmitter side other than one SENT output circuit may lead to higher current ripple, uncorrelated to that SENT transmission. In this case the current ripple has to be specified and considered in the design of the supply on the receiver side.

***Table 6.3.1-3 – Independent 5v supply high power transmitter additional requirements***

| Parameter | | | Limits (including component tolerances) Min | Max | Units | Test Conditions / Definitions (on wire) |
|---|---|---|---|---|---|---|
| k1. | $I_{GND\text{-}Increased}$ | = | | 100 | mA | Average current through Signal Ground Line over one message (increased value, resulting in l1) |
| k2. | $I_{GND\text{-}High}$ | = | | 200 | mA | Average current through Signal Ground Line over one message (highly increased value, resulting in l2) |
| l1. | $T_{TICK}$ | = | 2.6 | | μs | Sensors with increased ground current $I_{GND\text{-}Increased}$: Minimum SENT clock tick period required to maintain compatible timing margin with receiver<br>NOTE: 15% clock tolerance needed to achieve 1 ms worst case message transmission time and compatible timing margin |
| l2. | $T_{TICK}$ | = | 3.0 | | μs | Sensors with with highly increased ground current $I_{GND\text{-}High}$: Minimum SENT clock tick period required to maintain compatible timing margin with receiver<br>NOTE: 10% clock tolerance needed to achieve 1 ms worst case message transmission time and compatible timing margin |
| m. | $V_{GND\text{-}OFFSET}$ | = | | 200 | mV | Sensors with current consumption $I_{SUP\text{-}High}$: Ground offset from transmitter to receiver connectors due to transmitter ground current |

### 6.3.2 Receiver Input Requirements

Figure 6.3.2-1 and Figure 6.3.2-2 show two alternative receiver circuit topologies. One of the two or an alternative circuit (Figure 6.3.2-3) shall be used; however, Figure 6.3.2-1 is included for backwards compatibility with specification version FEB2008, is not recommended for new implementations and cannot be used with Open Drain Type Transmitters. Figure 6.3.2-2 represents the recommended approach for new implementations.

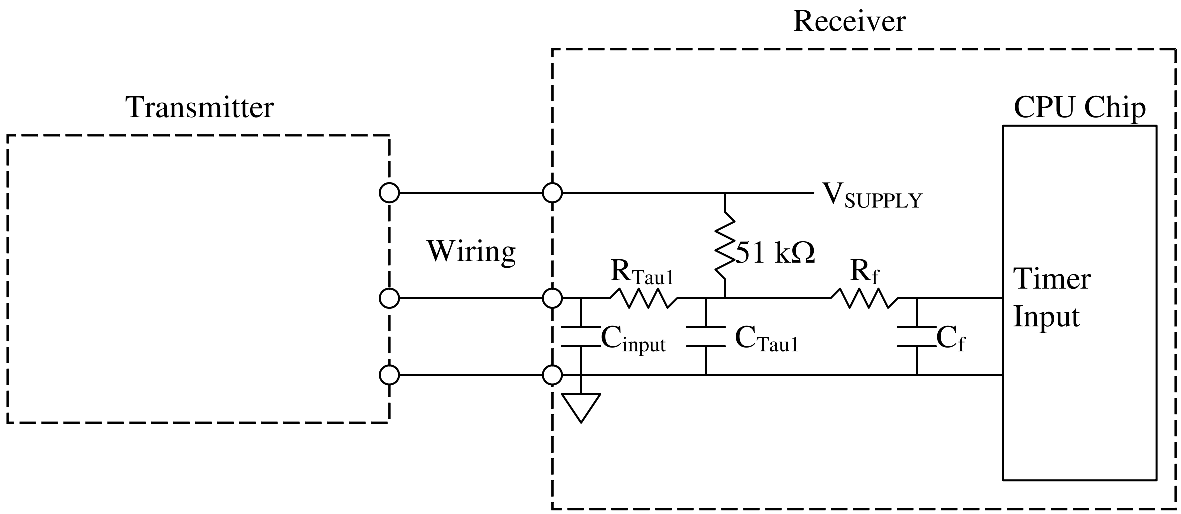



*Figure 6.3.2-1 – Legacy SENT receiver interface circuit topology*

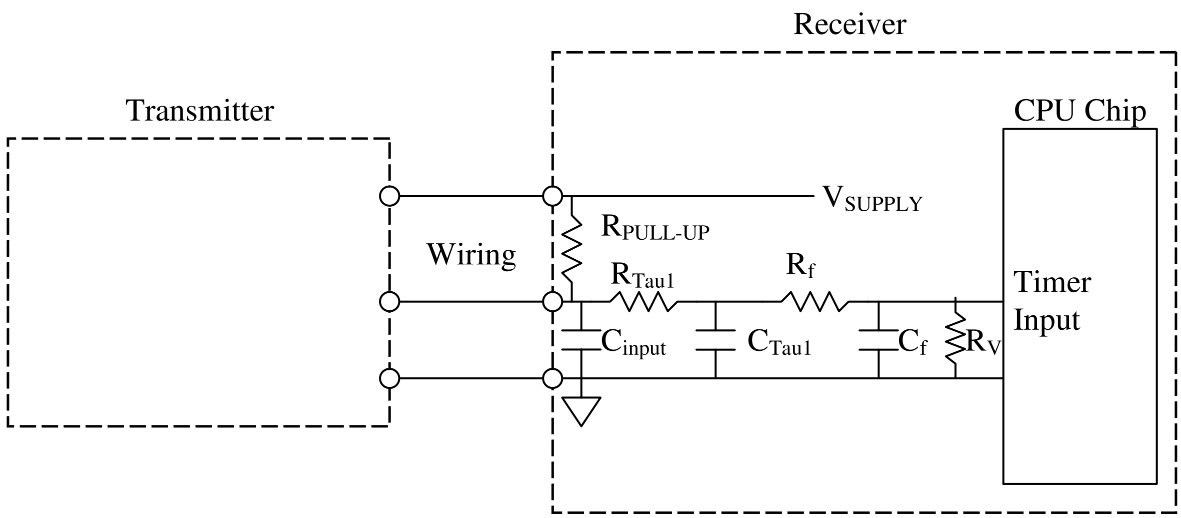



*Figure 6.3.2-2 – Recommended SENT system interface circuit topology*

The $R_V$ resistor is an optional element used to create a Voltage divider to translate the SENT signal level into a level required by the CPU chip. The recommended discrete receiver input circuit topology (Figure 6.3.2-2) shall be used for transmitter testing and evaluation.

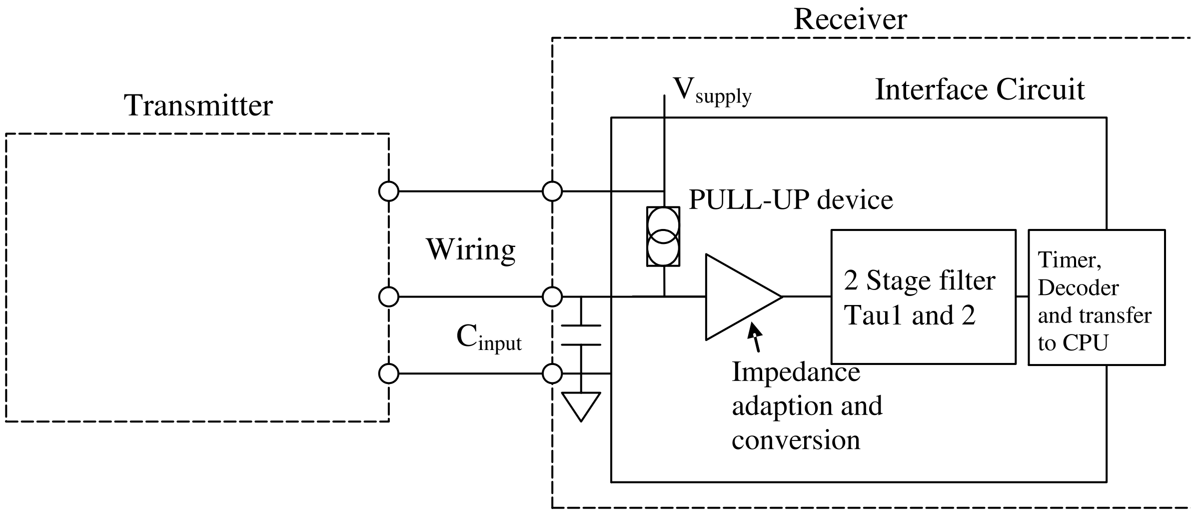



***Figure 6.3.2-3 – Example alternative SENT system interface circuit topologiy***

An alternative receiver circuit may be used (e.g.,, current source instead of the pull-up resistor). The alternative receiver circuit shall be designed such that the function of the SENT sub-system (transmitter and receiver) is equivalent to a sub-system with the reference receiver circuit of Figure 6.3.2-2. See APPENDIX I for more details.

EMC and EMI compliance of transmitter and receiver circuits are not in the scope of this standard, as these requirements are specified and may vary by the **OEM**. Note: both nonlinear behavior and parasitic parameters of the actual receiver circuits will have some influence on the transmitter's EMC performance.

The Receiving device shall provide a pull-up device to the supply voltage, a controlled threshold digital receiver, and a timing channel (to measure pulse periods). The receiver shall meet the requirements of Table 6.3.2-1 over the entire Receiving device environmental requirement range.

***Table 6.3.2-1 – General receiver input requirements***

| Parameter | | | Limits (including component tolerances) Min | Max | Units | Test Conditions / Definitions |
|---|---|---|---|---|---|---|
| c. | $V_{IL}$ | = | 1.39 | | V | Input low state threshold range |
| d. | $V_{IL\text{-}DELTA}$ | = | -50 | 50 | mV | Maximum input low state threshold variation over 1 millisecond interval with supply Voltage constant |
| e. | $V_{IH}$ | = | | 3.8 | V | Input high state threshold range |
| f. | $V_{HYST}$ | = | 0.3 | | V | Input electrical hysteresis ($V_{HYST} = V_{IH} - V_{IL}$) |
| g. | $Tau_1$ | = | 0.74 | 1.73 | µs | Input filter time constant – first stage |
| h. | $Tau_2$ | = | 0.6 | 1.4 | µs | Input filter time constant – second stage determined by $R_V$, $R_f$ and $C_f$ |

If the legacy or recommended discrete receiver input circuits given in Figure 6.3.2-1 and Figure 6.3.2-2 are used, the elements of the circuit shall meet the requirements of Table 6.3.2-2 over the entire Receiving device environmental range. Otherwise, Table 6.3.2-2 shall be used for the definition of equivalent receiver input circuits. See APPENDIX I for analysis requirements.

In case the Receiving device also provides a regulated 5V power supply voltage to the Transmitter, it shall meet the requirements in Table 6.3.2-3 over the entire Receiving device environmental range.

*Table 6.3.2-2 – Discrete receiver component values*

| Parameter | | | Limits (including component tolerances) | | Units | Test Conditions / Definitions |
|---|---|---|---|---|---|---|
| | | | Min | Max | | |
| a. | $R_{PULL-UP}$ | = | 10000 | 55000 | Ohms | Input pull-up resistance |
| b. | $C_{INPUT}$ | = | | 0.1 | nF | Parasitic input capacitance and ESD protection |
| i. | $R_{Tau1}$ | | 448 | 672 | Ohms | 560 +/- 20% overall tolerance must meet Tau1 requirement |
| j. | $C_{Tau1}$ | | 1.54 | 2.86 | nF | 2.2 +/- 30% overall tolerance must meet Tau1 requirement |
| k. | Rf | | 4 | | kΩ | Rf should be chosen in such a way that the 0.1 mA sourcing capability of the transmitter is not exceeded. |

*Table 6.3.2-3 – Receiver power supply requirements*

| Parameter | | | Limits (including component tolerances) | | Units | Test Conditions / Definitions |
|---|---|---|---|---|---|---|
| | | | Min | Max | | |
| a. | $V_{SUPPLY}$ | = | 4.85 | 5.15 | V | Supply Voltage |
| b. | $I_{OUT}$ | = | (1) | | mA | Available current for Transmitting device |
| c. | $C_{LOAD}$ | = | | (2) | µF | Supply to return load capacitance in transmitter |
| d. | $V_{GND-OFFSET}$ | = | | 0.02 | V | Signal return ground offset in Receiving device |
| e. | $V_{3.3SUPPLY}$ | = | 3.234 | 3.366 | V | Internal 3.3 V supply (if needed by Receiver) |

[1] The Receiver Power Supply shall be dimensioned application specific according to the Power Supply needs and current consumption of the connected sensors. The Power Supply dimensioning needs also include increased start up and ripple currents.
[2] OEM or application specific requirements may limit capacitive loading at sensor supply outputs.

#### 6.3.2.1 ESD Transient Suppressor

If necessary, a circuit element such as a transorb (back-to-back zener) or a varistor device may be added in one or more places to provide ESD protection. However, when these devices are used they may add capacitance or introduce voltage and/or temperature variability to the signal time constant. When such devices are used the device load capacitor shall be reduced by an amount equivalent to the capacitance of the ESD transient suppressor.

### 6.3.3 Bus Wiring Harness and ECU Connectors

Combined resistance for all connectors and wiring shall have less than 1 Ohms per line over total vehicle life. The bus wiring shall utilize cables with less than 0.1nF per meter of wire length. The maximum cable length shall be 5 meters.

## 6.4 ESD Immunity

The transmitting and receiving module I/O pins shall withstand electrostatic discharges without any damage to the transmitting or receiving module when subjected to testing per the vehicle manufacturer’s component Electrostatic Discharge test. The particular vehicle manufacturer’s component technical specification shall state the Criticality Level of the device.

## 6.5 EMC Testing Requirements

The physical layer, when incorporated into an ECU or sensor design, shall function as specified in the device's intended electromagnetic environment. Additionally, the electromagnetic emissions produced during signal related operations shall not interfere with the normal operation of other ECU's or subsystems.

The particular vehicle manufacturer's component technical specification shall state the EMC testing requirements of the device.

## 6.6 Fault Protection Modes

Data communications may be interrupted under short of any impedance to supply or ground.

1. Wiring Short to Ground – There shall be no damage to any device when the signal or supply line is shorted to ground. An impedance of less than 50 ohms between the line and ground shall be considered a short to ground. Upon removal of the fault, a transmitter reset may occur and normal operation shall resume within the transmitter reset time. Operator intervention (for example a key cycle) shall not be required.
2. Wiring Short to Supply – There shall be no damage to any device when the signal or ground line is shorted to sensor supply $V_{OUT}$. An impedance of less than 50 ohms between the line and $V_{OUT}$ shall be considered a short. Upon removal of the fault, a transmitter reset may occur and normal operation shall resume within the transmitter reset time. Operator intervention (for example a key cycle) shall not be required.

# 7. CONFIGURATION SHORTHAND

## 7.1 Shorthand Definition

A shorthand description for the various optional configurations is constructed using the following format and Table. Elements of the table are separated by dashes to form the final shorthand notation.

***Table 7.1-1 – SENT configuration shorthand definition***

| Shorthand Description | Format | Requirements |
|---|---|---|
| SENT SAE J2716 revision | SENTxxxx | One of the following values depending on SENT revision supported:<br>• 2007<br>• 2008<br>• 2010<br>• 2016 |
| Clock Tick Length [µs] | xx.xus | 03.0 ≤ xx.x ≤ 90.0 |
| Number of Data Nibbles | xdn | 1 ≤ x ≤ 6 |
| Pause Pulse Option | npp<br>pp<br>ppc(xxx.x) | no pause pulse<br>pause pulse<br>pause pulse with constant frame length, xx is number of clock ticks, 282.0 ≤ xxx.x ≤ 922.0 (for 6 data nibbles) |
| Use of Serial Protocol | ssp<br>esp<br>nsp | short serial protocol<br>enhanced serial protocol<br>no serial protocol |
| Sensor Type | Ns<br><br><br>A.1<br>A.3<br>A.4<br>D.1xxx<br><br>… | Not specified here (see manufacturer specification)<br>Additional appendices may be referenced.<br>Some examples:<br>Dual Throttle Position sensors described in Appendix A.1<br>Single Secure sensors described in Appendix A.3<br>Single sensors described in Appendix A.4<br>Sensors class given in Table D.4.1-1 , xxx are nibble values for bits $D_{11}$ through $D_0$ (APPENDIX D represents the sensors defined in Appendices A.2 and A.5) |

Example Shorthand Description: SENT2010-03.0us-6dn-ppc(282.0)-nsp-A.1

# 8. NOTES

## 8.1 Revision Indicator

A change bar (I) located in the left margin is for the convenience of the user in locating areas where technical revisions, not editorial changes, have been made to the previous issue of this document. An (R) symbol to the left of the document title indicates a complete revision of the document, including technical revisions. Change bars and (R) are not used in original publications, nor in documents that contain editorial changes only.

PREPARED BY THE SAE SENT TASK FORCE
OF THE SAE VEHICLE ARCHITECTURE FOR DATA COMMUNICATIONS COMMITTEE

# APPENDIX A - RECOMMENDED APPLICATION SPECIFIC PROTOCOLS

This Appendix gives SENT protocol details for certain sensor applications.

## A.1 DUAL THROTTLE POSITION SENSORS

Specific additional requirements for the application of the protocol to dual sensor Electronic Throttle Control systems shall follow the requirements of Section H.1 with the exception of the following:

- When an individual sensor is determined to be in error, the nibbles for that sensor will be all set to 0xF (TPS1) or 0x0 (TPS2).
- Details serial message: Not required for throttle application.

## A.2 MASS AIR FLOW SENSORS

This appendix describes the application of the SENT interface for Mass Air Flow (MAF) sensors. In addition to the MAF signal (on fast channel 1), MAF sensors optionally can deliver air pressure or air temperature data over the SENT fast channel 2.

Air temperature at MAF (medium temperature), relative humidity, temperature at humidity sensor element and barometric pressure can be optionally transmitted over the supplementary data channels on the serial message channel. The default assignment of those measurement signals on particular supplementary data channels is specified in Section D.4.2.

Three different distributions of the 24 data bits (six data nibbles) can be chosen to allocate 2 fast channels

- 16-bit fast channel 1 and 8-bit fast channel 2 (H.7)
- 14-bit fast channel 1 and 10-bit fast channel 2 (H.6)
- 12-bit fast channel 1 and 12-bit fast channel 2 (H.1)

### A.2.1 Data Channels for MAF Sensors

Table A.2.1-1 lists the various sensor configurations for MAF sensor applications as an extract of Table D.4.1-1. MAF sensors with high-resolution MAF signal and limited requirements on the pressure signal partition the six data nibbles (24 bit) into 16 bit for the MAF signal and 8 bit for the air pressure at the MAF. The air pressure at the MAF can be for example the Inlet Depression (Air pressure downstream of the air filter).

***Table A.2.1-1 – Data channels for MAF sensors***

| MAF Sensor Type | Data Channels | | | 12-Bit Code (see D.4.1) |
|---|---|---|---|---|
| | Fast Channel 1 | Fast Channel 2 | Supplementary data channels | |
| not specified MAF Sensors | Not specified | Not specified | Not specified | $010 |
| MAF (hi-res, lin) | MAF (16-bit), linear | Zeros (8-bit) | If additional sensor data is transmitted over supplementary data channels, the default assignment is according to Section D.4.2.<br><br>NOTE:<br>The transmission of additional sensor data is optional for | $011 |
| MAF (hi-res, non-lin) | MAF (16-bit), non-linear | Zeros (8-bit) | | $012 |
| MAF (hi-res, lin,) / Pressure | MAF (16-bit), linear | Pressure (8-bit) | | $013 |
| MAF (hi-res, non-lin) / Pressure | MAF (16-bit), non-linear | Pressure (8-bit) | | $014 |
| MAF (lin) / Pressure | MAF (14-bit), linear | Pressure (10-bit) | | $015 |
| MAF (non-lin) / Pressure | MAF (14-bit), non-linear | Pressure (10-bit) | | $016 |
| MAF (lin) | MAF (14-bit), linear | Zeros (10-bit) | | $017 |

| MAF Sensor Type | Data Channels | | | 12-Bit Code (see D.4.1) |
|---|---|---|---|---|
| | Fast Channel 1 | Fast Channel 2 | Supplementary data channels | |
| MAF (non-lin) | MAF (14-bit), non-linear | Zeros (10-bit) | each of the specified MAF sensor types. | $018 |
| MAF (lin) / Temperature | MAF (14-bit), linear | Temperature (10-bit) | | $019 |
| MAF (non-lin) / Temperature | MAF (14-bit), non-linear | Temperature (10-bit) | | $01A |
| MAF (hi-res, lin) / Temperature | MAF (16-bit) linear | Temperature (8-bit) | | $01B |
| MAF (hi-res, non-lin) / Temperature | MAF (16-bit), non-linear | Temperature (8-bit) | | $01C |
| MAF (low-res, non-lin) | MAF (12-bit), non-linear | Zeros (12-bit) | | $01D |
| MAF (low-res, non-lin) / Pressure | MAF (12-bit), non-linear | Pressure (12-bit) | | $01E |
| MAF (low-res, non-lin) / Temperature | MAF (12-bit), non-linear | Temperature (12-bit) | | $01F |

- MAF sensors with a 16-bit fast channel 1 (MAF data) and an optional 8-bit fast channel 2 (pressure or temperature of the air flow at the MAF) shall use the frame format specified in Section H.1.

- MAF sensors with a 14-bit fast channel 1 (MAF data) and an optional 10-bit fast channel 2 (pressure or temperature of the air flow at the MAF) shall use the frame format specified in Section H.1.

- MAF sensors with a 12-bit fast channel 1 (MAF data) and an optional 12-bit fast channel 2 (pressure or temperature of the air flow at the MAF) shall use the frame format specified in Section H.1.

The scaling of MAF data transmitted over fast channel 1 is specified in Section A.2.6.1. The scaling of data transmitted over fast channel 2 is specified in Section A.2.7 for pressure and A.2.8 for temperature data.

Sensors using this format shall use the Enhanced Serial Message Format if serial messages are transmitted.

For new applications the definitions for Reserved Signaling Code Ranges of Section A.2.3.2 shall be used to transmit error indicators and specific messages.

For new applications the definitions for Reserved Signaling Code Ranges of Section A.2.3.2 shall be used to transmit error indicators and specific messages.

Older applications, compatible with specification Version FEB2008, can use the following error indications:

- When Reserved Signaling Code Ranges of Section A.2.3.2 are not used and the mass air flow is determined to be in error, the nibbles for that sensor will be all set to 0 (transmitting 0).

- When Signaling Code Ranges of Section A.2.3.2 are not used and the pressure or temperature at the MAF sensor is determined to be in error, the nibbles for that sensor will be all set to 0 (transmitting 0 kPa).

### A.2.2 Status and Communication Nibble Error Flags

Status and error messages, error flags and error flag handling recommendation shall be used as specified in Section E.3.

### A.2.3 Overview of Message IDs

Sections A.2.3.1 and A.2.3.2 gives two alternative sets of message IDs. One of the two shall be used; however, Section A.2.3.1 is included for backwards compatibility with specification version FEB2008 and should not be used for new implementations. Section A.2.3.2 represents the recommended approach for new implementations.

A.2.3.1 Short Serial Message Format (legacy MAF sensors)

Table A.2.3.1-1 lists specific Message IDs when Short Serial Message Format is used.

Individual serial message channel sensor components (for example, air temperature, humidity or barometric pressure) are optional components and messages for these components do not need to be transmitted if the component is not present.

All scalings are application defined.

***Table A.2.3.1-1 – Short serial message format message IDs for mass air flow sensor serial data***

| Message ID | Sensed Value for Data Field |
|---|---|
| 0 | Air Temperature (high byte) |
| 1 | Air Temperature (low byte) |
| 2 | Relative Humidity (high byte) |
| 3 | Relative Humidity (low byte) |
| 4 | Barometric Pressure(high byte) |
| 5 | Barometric Pressure(low byte) |
| 6 | Reserved |
| 7 | Reserved |
| 8 | Reserved |
| 9 | Reserved |
| A | Reserved |
| B | Reserved |
| C | Reserved |
| D | Reserved |
| E | Reserved |
| F | Error Codes |

Short Serial Message Format Serial message error codes bit encoding are listed in Table A.2.3.1-2.

If no errors are present, message F does not need to be transmitted.

***Table A.2.3.1-2 – Error codes for message ID F, short serial message format mass air flow sensor serial data***

| Error Code Bit Encoding | Bit (least significant to most significant) |
|---|---|
| Air Temperature Out-of-range High | 0 |
| Air Temperature Out-of-range Low | 1 |
| Relative Humidity Out-of-range High | 2 |
| Relative Humidity Out-of-range Low | 3 |
| Barometric Pressure Out-of-range High | 4 |
| Barometric Pressure Out-of-range Low | 5 |
| Reserved | 6 |
| Reserved | 7 |

A.2.3.2 Enhanced Serial Message Format

Data formats and protocols of the enhanced serial message format shall be used as specified in APPENDIX D.

### A.2.4 Reserved Signaling Code Ranges

The Reserved Signaling Code Ranges are required when the Enhanced Serial Message is implemented. Reserved signaling code ranges shall be used as specified in Section E.1.2. They can be optionally used for sensors that do not implement the Enhanced Serial Message.

If legacy MAF modes are used, the full data ranges from 0 to 65535 and 0 to 255 shall be used. The receiver module must know a priori the scaling used.

### A.2.5 Reserved Signaling Ranges

Data nibble reserved values shall be used as specified in section E.1.2. and E.1.3.

### A.2.6 Encoding Mass Air Flow Transfer Functions

#### A.2.6.1 Common Definition for Linear and Sensor Specific Nonlinear Transfer Functions

The partitioning of the 12-bit, 14-bit or 16-bit MAF Data shall be as specified in Section E.2.1.2 independent if a linear Mass Air Flow Transfer Function or a Sensor specific nonlinear Transfer Function is applied.

##### A.2.6.1.1 Nonlinear Mapping of Mass Air Flow Measurement Values onto $n$-bit SENT Data

In case of a nonlinear Transfer Function this function shall be defined sensor specific. The application-specific transfer characteristics have to be known by the receiver.

##### A.2.6.1.2 Linear Mapping of Mass Air Flow Measurement Values onto $n$-bit SENT Data

In case of linear mapping of mass air flow measurement values onto $n$-bit SENT data values the mapping shall be carried out as specified in Section E.2.6.

##### A.2.6.1.3 Transmission of Mass Air Flow Sensor Transfer Characteristic Functions

Transfer characteristic functions of mass air flow sensors shall be transmitted from the sensor to the ECU over the Serial Message Channel, using the Enhanced Serial Message Format as specified in E.2.1.5, if this Serial Message Channel is implemented. Messages with IDs $07 through $0E (see APPENDIX D) are used to transmit the node characteristic values.

The transfer characteristic function for Fast Channel 1 can be determined by serial message channel message IDs $07 through $0A (see section D.1). The transfer characteristic function for Fast Channel 2 can be determined by serial message channel message IDs $0B through $0E (see section D.1).

If the optional serial message channel is not implemented or nonlinear transfer functions are used, application-specific transfer characteristics have to be known by the receiver.

### A.2.7 Encoding Pressure Transfer Functions

The mapping of pressure data on $n$ = 8-bit, 10-bit or 12-bit SENT data values transmitted over fast channel 2 as well as supplementary data channel shall be made by using the linear pressure transfer functions as specified in Appendix E.2.4.

### A.2.8 Encoding Temperature Transfer Functions

The mapping of temperature data on $n$ = 8-bit, 10-bit or 12-bit SENT data values transmitted over fast channel 2 as well as supplementary data channel shall be made by using the default linear temperature transfer characteristic functions as specified in Section E.2.2.1.

### A.2.9 Encoding Humidity Transfer Functions

The mapping of relative humidity data in percent on 12-bit SENT data values transmitted over supplementary data channel shall be made by using the ratio sensing transfer functions as specified in Section E.2.3.

## A.3 SINGLE SECURE SENSORS

An example of a single secure sensor is a pedal position sensor which is used to determine driver intent. Secure sensors require sufficient diagnostics to ensure the integrity of the data being transmitted. For example, the rolling count is used to detect a stuck message or stuck message processing.

Single Secure Sensors shall follow the requirements of Section H.4.

## A.4 SINGLE SENSORS

Single Sensors shall follow the requirements of one of the Sections H.2, H.3, H.4, or H.5. It has to be specified, which of these options is used.

## A.5 PRESSURE SENSORS AND COMBINED PRESSURE AND TEMPERATURE SENSORS

Specific additional requirements for the application of the protocol to Pressure (P) sensors and Combined Pressure (P) and Temperature (T) measurement systems are listed in this section. These systems are characterized as follows:

- 1 or 2 fast data channel(s): P or T, P + P, P+T
- Serial message channel (optional)
- 12-bit digital representation of pressure and temperature data can be interpreted by means of transfer characteristic functions
- The serial message channel of the status and communication nibble may be optionally used (enhanced serial message format).
- Some 12-bit data values are reserved for initialization information and error codes
- The serial message message channel can optionally be used to transmit sensor transfer characteristic function(s) parameters
- The optional serial message message channel can be used to convey measurement data from additional sources. In this way supplementary data channels can be implemented using the serial message channel.

Simple sensors which do not implement the serial channel use default sensor parameters. These sensor parameters have to be known by the receiver-side to be able to interpret the 12-bit measurement data values.

### A.5.1 Data Channels for Pressure Sensors and Combined Pressure and Temperature Sensors

#### A.5.1.1 Pressure Sensor Configurations and Allocations of Data Channels

##### A.5.1.1.1 Default Encoding Scheme

Pressure sensors and combined pressure and temperature sensors shall make use of the following data channels provided by the SENT interface (H.1)

- SENT signal 1 with data nibbles 1-3 carries a 12-bit data value. In the context of this appendix this denotes the “fast channel 1”
- SENT signal 2 with data nibbles 4-6 carries a 12-bit data value. In the context of this appendix this denotes the “fast channel 2”
- Bits 0 and 1 of the status and communication nibble serve as error indicators for fast channels 1 and 2.
- Sensors shall use the Enhanced Serial Message Format if serial messages are transmitted.

A.5.1.1.2 High-speed Encoding Scheme

If the special high-speed encoding scheme is used, pressure sensors shall make use of the following data channels provided by the SENT interface

- SENT signal 1 with data nibbles 1-4 carries a 12-bit data value as specified in H.3.
- Bits 0 of the status and communication nibble serves as an error indicator for fast channel 1.
- Sensors shall use the Enhanced Serial Message Format if serial messages are transmitted.

A.5.1.1.3 Overview of Pressure Sensor Frame Formats and Protocols

The principal sensor data is transmitted over Fast Channel 1 and Fast Channel 2 (if defined by sensor type). The allocation of the Fast Channels to the particular SENT frame formats is specified in APPENDIX H.

Ancillary sensor data can be transmitted over the optional serial message data channel. If a serial message channel is implemented, the enhanced serial message format with 12-bit data field and 8-bit message ID shall be used (see Section 5.2.4.2). The status and communication nibble shall be used as specified in Table H-2.

The various pressure sensor types shall use frame formats and protocols specified as follows:

- The first pressure data signal shall be transmitted over Fast Channel 1 (encoded, as specified in Section A.5.3.1)
- Fast Channel 2 (if present) either carries a second pressure data signal (encoded, as specified in Section A.5.3.1), temperature data, secure sensor information or it's stuffed with zeros.
- If additional measurement data (e.g., medium temperature and/or internal reference temperature) is transmitted over the serial message channel, supplementary data channels as specified in D.1 shall be used. For the transmission of additional measurement data the default assignment of supplementary data channels for pressure sensors, as specified in section D.4.2 and Table D.4.2-1, shall be used.

Table A.5.1-1 specifies the data formats and protocols of the different pressure sensor types.

NOTE: The transmission of medium temperature and/or internal reference temperatures over the serial message channels is optional for each of the specified pressure sensor types.

The default temperature characteristic according to E.2.2.1 is used unless the particular sensor type explicitly specifies a sensor specific temperature characteristic.

***Table A.5.1-1 – Overview of pressure sensor frame formats and protocols***

| Pressure sensor frame format | Number of nibbles / data nibbles | Data Frame Format (see APPENDIX H) | Fast Channel 2 (data nibbles 4-6) | Supplementary data channel # 1,1 for medium temperature (ID $ 1 0) | Pressure Sensor Type [12-Bit Code] (see D.4) |
|---|---|---|---|---|---|
| Pressure sensors with a 12-bit fast channel 1 (pressure data) and a 12-bit fast channel 2 (pressure, temperature or other data) | 8 / 6 | H.5 | Zeros (12 bit) | Optional with linear default characteristic E.2.2.1 | P/- [$002] |
| | | | | Sensor-specific temperature characteristic function (see A.5.3.2) | P/-/t [$00C] |

| Pressure sensor frame format | Number of nibbles / data nibbles | Data Frame Format (see APPENDIX H) | Fast Channel 2 (data nibbles 4-6) | Supplementary data channel # 1,1 for medium temperature (ID $ 1 0) | Pressure Sensor Type [12-Bit Code] (see D.4) |
|---|---|---|---|---|---|
| Pressure sensors with a 12-bit fast channel 1 (pressure data) and a 12-bit fast channel 2 (pressure, temperature or other data) | 8 / 6 | H.1 | Pressure signal # 2 (12 bit) | Optional with linear default characteristic E.2.2.1 | P/P [$006] |
| | | | | Default characteristic E.2.2.1 | P/P/t [$009] (1) |
| | | | | Sensor-specific temperature characteristic function (see A.5.3.2) | P/P/t [$00A] |
| | | | Temperature 12 bit, linear default characteristic E.2.2.1 | Optional with linear default characteristic E.2.2.1 | P/T [$007] |
| | | | Temperature 12 bit, (sensor-specific characteristic function see A.5.3.2) | Sensor-specific temperature characteristic function (see A.5.3.2) | P/T [$008] |
| Pressure sensors with only one 12-bit fast channel | 5 / 3 | H.2 | Not implemented | Optional with linear default characteristic E.2.2.1 | P [$001] |
| | | | | Sensor-specific temperature characteristic function (see A.5.3.2) | P/t [$00B] |
| Pressure sensors with high-speed 12 bit message – format | 6 / 4 | H.3 | Not implemented | Optional with linear default characteristic E.2.2.1 | P (high speed) [$00D] |
| | | | | Sensor-specific temperature characteristic function (see A.5.3.2) | P/t (high speed) [$00E] |
| Pressure sensors with 12-bit fast channel 1 and 12 bit fast channel 2 for secure sensor information | 8 / 6 | H.4 | 8-bit rolling counter (nibbles 4-5) and inverted MSN of fast channel 1 (nibble 6) | Optional with linear default characteristic E.2.2.1 | P/S [$003] |
| | | | | Default characteristic E.2.2.1 | P/S/t [$004] (1) |
| | | | | Sensor-specific temperature characteristic function (see A.5.3.2) | P/S/t [$005] |
| Pressure sensors with frame format and protocol not specified by App. A.5 & D.4 | Not speci-fied | Not speci-fied | Not specified | Not specified | [$00F] |

[1] Note: Legacy pressure sensor types are contained in the table for backward compatibility reasons and are not recommended. Since the default assignment of supplementary data channels according to D.4.2 is to be used, these legacy types have become redundant. Example: P/S/t [$004] corresponds to P/S [$003] and P/P/t [$009] corresponds to P/P [$006], if medium temperature is transmitted over supplementary data channel #1,1 can be both P/S [$003] and P/S/t [$004] (legacy).

### A.5.2 Reserved Signaling Ranges

Data nibble reserved values shall be used as specified in section E.1.2. and E.1.3.

### A.5.3 Encoding Sensor Characteristic Transfer Functions

#### A.5.3.1 Pressure Sensor Transfer Functions

##### A.5.3.1.1 Mapping of Pressure Measurement Values onto n = 12 bit SENT Data

The mapping of pressure measurement values onto $n$ = 12 bit SENT data values shall be carried out with the linear pressure transfer functions, as specified in Section E.2.4.

##### A.5.3.1.2 Transmission of Pressure Transfer Characteristic Functions

Linear transfer characteristic functions of pressure sensors shall be transmitted from the sensor to the ECU over the Serial Message Channel, using the Enhanced Serial Message Format as specified in E.2.1.5, if this Serial Message Channel is implemented. Messages with IDs $07 through $0E (see APPENDIX D) are used to transmit the node characteristic values.

The transfer characteristic function for Fast Channel 1 can be determined by serial channel message IDs $07 through $0A (see section D.1). The transfer characteristic function for Fast Channel 2 can be determined by serial channel message IDs $0B through $0E (see section D.1).

If the optional serial message channel is not implemented, application-specific transfer characteristics have to be known by the receiver.

#### A.5.3.2 Temperature Sensor Transfer Characteristic Functions

The range of possible temperature values is small compared with the variety of pressure ranges. Therefore the standard defines in section E.2.2.1 the default characteristic function that can be applicable to most sensors.

In addition there exists also an option to specify sensor specific temperature characteristics specified by sensor type in combination with a configuration code.

Examples of such sensor specific temperature characteristic are:

- Sensor types with nonlinear temperature scale(non-Kelvin, e.g., un-linearized NTC output). The particular non-linear temperature characteristic can be indicated by Message 04 (Configuration Code).

- Sensor types with non-default linear characteristics, for example, wider temperature range and coarser resolution. A linear Temperature Characteristic that differs from the default characteristic can be transmitted using Fast Channel 2 Characteristic Nodes $X_1$, $Y_1$ and $X_2$ and $Y_2$ using Messages 0B, 0C, 0D, 0E.

If temperature is transmitted as fast-channel data, it shall always be transmitted in fast channel 2 when combined with pressure data. If temperature is transmitted over the serial message channel, supplementary data message #1,1 (with ID# 1 0) shall carry the 12-bit (medium) temperature values(see section D.4.2). Note: Supplementary data channel #4,1 (ID 2 3) can optionally carry an internal reference temperature (see section D.4.2).

### A.5.4 Serial Message Channel Definition

The serial message channel for sensors defined in Section A.5 shall be as defined in APPENDIX D.

### A.5.5 SENT Status and Error Messages

Implementation of status and error messages, error flags and handling recommendation has been moved to section E.3

### A.5.6 12-bit Data Nibble Reserved Values

Data nibble reserved values shall be used as specified in section E.1.2. and E.1.3.

## A.6 TEMPERATURE SENSORS

### A.6.1 Overview: Temperature Sensors

This section specifies the SENT interface of sensors whose primary function is the transmission of temperature measurement values (Fast Channel 1 carries temperature data values). The temperature sensors specified in this section use linear transfer characteristic functions, which determine the mapping of the temperature values (in Kelvin [K]) onto $n$-bit SENT data values (E.2.2).

Temperature sensors can be categorized by their transfer characteristic functions and temperature ranges:

- Temperature sensors for measurements within the default temperature range (-73.025 °C … 437.85°C) covered by the default temperature transfer characteristic function (E.2.2.1). This characteristic function covers the temperature ranges that are relevant for a large share of automotive applications.

- Temperature sensors for measurements within the high temperature range (-72.82 °C … 1289.51°C) covered by the high-temperature transfer characteristic function (E.2.2.2)

- Temperature sensors for measurements within a special sensor-specific temperature range, which is defined by an application-specific special transfer characteristic function (E.2.2.3)

Temperature sensor types, their transfer characteristic functions and frame formats are specified in Table D.4.1-1. Transfer characteristic functions used for data on supplementary data channels of the serial message channel are specified in Table D.4.2-1. An overview of data frame formats for temperature sensors is given in Section A.6.3.

### A.6.2 Temperature Transfer Characteristic Functions

The receiver can interpret the n-bit SENT data values as temperature measurement values (in Kelvin) based on the sensor-type specific transfer characteristic. The applicable transfer characteristic functions are determined, as specified in Table A.6.2.1.

***Table A.6.2.1 – Interpretation of n-bit temperature data***

| **Specification of SENT $n$-bit Temperature Data in Fast Channel or Supplementary Data Channel (according to Appendix D.4)** | **Temperature Transfer Characteristic Functions (Fast Channels and/or Serial Message Channels)** | **Appendix** |
|---|---|---|
| Temperature characteristic function is not explicitly specified for temperature data channel of particular sensor type | Default linear temperature transfer characteristic | E.2.2.1 |
| Temperature characteristic function is not specified and no special transfer characteristic function of a temperature channel is transmitted over the serial message channel | Default linear temperature transfer characteristic | E.2.2.1 |
| Default temperature transfer characteristic specified for data channel | Default linear temperature transfer characteristic | E.2.2.1 |
| High temperature transfer characteristic specified for data channel | High temperature linear transfer characteristic | E.2.2.2 |
| Special temperature transfer characteristic specified for data channel | Special linear temperature transfer characteristic Transfer characteristic node values are transmitted over serial message channel as specified in E.2.1.5 and D.1. If the optional serial data channel is not implemented, application-specific transfer characteristics have to be known by the receiver. | E.2.2.3 |

Note that sensor output signals do not need to cover the entire temperature range of their particular transfer characteristic function.

### A.6.3 Temperature Sensor Frame Formats

The temperature measurement data (12-bit) is transmitted over Fast Channel 1 and Fast Channel 2 (if defined by sensor type) or over the Multiplexing Fast Channel, as specified in D.4. The allocation of the Fast Channels to the particular SENT frame formats is specified in APPENDIX H and APPENDIX F. Table A.6.3-1 specifies the data frame formats and protocols of the different temperature sensor types using standard H.x or not specified data frame formats. Table A.6.3-2 specifies the data frame formats and protocols of the different temperature sensor types using fast channel multiplexing data frame formats.

***Table A.6.3-1 – Overview of temperature sensor frame formats and protocols***

| Temperature sensor frame format | Total number of nibbles / data nibbles | Data Frame Format | Fast Channel 1 (data nibbles 1-3) | Fast Channel 2 (data nibbles 4-6) | Multiplexing Channel Fast Channel (data nibbles 3-5) | Temperature Sensor Type [12-Bit Code] (see D.4) |
|---|---|---|---|---|---|---|
| Temperature sensors with a 12-bit fast channel 1 and a 12-bit fast channel 2 | 8 / 6 | H.5 | Temperature (E.2.2.1) | Zeros (12 bit) | - | $ 041 |
| | | H.1 | | Temperature (E.2.2.1) | - | $ 043 |
| | | H.5 | High Temperature (E.2.2.2) | Zeros (12 bit) | - | $ 045 |
| | | H.1 | | High Temperature (E.2.2.2) | - | $ 047 |
| | | H.5 | Special Temperature (E.2.2.3) | Zeros (12 bit) | - | $ 049 |
| | | H.1 | | Special Temperature (E.2.2.3) | - | $ 04B |
| Temperature sensors with 12-bit fast channel 1 and 12 bit fast channel 2 for secure sensor information | 8 / 6 | H.4 | Temp (E.2.2.1) | 8-bit rolling counter (nibbles 4-5) and inverted MSN of fast channel 1 (nibble 6) | - | $ 042 |
| | | | High Temperature (E.2.2.2) | | - | $ 046 |
| | | | Special Temperature (E.2.2.3) | | - | $ 04A |
| Temperature sensors with frame format and protocol not specified by Appendix A.6 and D.4 | Not specified | Not speci-fied | Not specified | Not specified | - | $ 040 |

***Table A.6.3-2 – Overview of temperature sensor frame formats and protocols using fast channel multiplexing***

| Temperature sensor frame format | Total number of nibbles / data nibbles | Data Frame Format | Total number of Frame Controls (FC) | Sequence of Frame control values | Multiplexing Channel Fast Channel 1 | Temperature Sensor Type [12-Bit Code] (see D.4) |
|---|---|---|---|---|---|---|
| Temperature sensor clusters with Fast Channel Multiplexing | 8 / 6 | All frames: F.6.1 Data Frames with one multiplexed fast channel and 12-bit sensor data (F1.5) | 1…16 | The sequence of FC values is depending on amount of used FCs and with ascending order | All fast channel 1: Temp (E.2.2.1) | $ 044 |
| | | | | | All fast channel 1: High Temperature (E.2.2.2) | $ 048 |
| | | | | | All fast channel 1: Special Temperature (E.2.2.3) | $ 04C |

A.6.4 Status and Communication Nibble

Bit 0 and Bit 1 of the status and communication nibble shall be used as error indicator bits for fast channel 1 and fast channel 2 as specified in Sections H.1 and H.4.

Sensors with fast channel multiplexing use bit 0 and bit 1 of the status and communication nibble as specified in APPENDIX F.

Bit 2 and Bit 3 of the status and communication nibble can optionally be used to carry a serial message channel. Ancillary sensor data can be transmitted over the optional serial message data channel. If a serial message channel is implemented, the enhanced serial message format with 12-bit data field and 8-bit message ID shall be used (see Section 5.2.4.2). If additional measurement data is transmitted over the serial message channel, supplementary data channels, as specified in D.1, shall be used. For the transmission of additional measurement data the default assignment of supplementary data channels for temperature sensors, as specified in section D.4.2 and Table D.4.2-1 shall be used.

A.6.5 Reserved Signaling Ranges

Data nibble reserved values shall be used as specified in section E.1.2 and E.1.3.

A.7 POSITION SENSORS AND RATIO SENSORS

A position sensor in the context of this section is defined as a sensor which measures either

- a linear position in the physical unit meter,
- an angle in degree,
- or a relative position or a relative angle scaled in percent.

A ratio sensor in the context of this section is defined as a non-position sensor which measures a physical quantity scaled in percent (e.g., relative humidity sensor).

Specific additional requirements for the application of the protocol to position sensors and ratio sensors are listed in this section. These systems are characterized as follows:

- 1 or 2 fast data channel(s):
    - X, X+X, X+T, X+S for position sensors
    - R, R+R, R+T, R+P, R+S for ratio sensors
    - Legend:
      X: Position channel
      R: Ratio channel
      T: Temperature channel
      P: Pressure channel
      S: Secure information

- 12-bit digital representation of position, ratio, temperature, and pressure data can be interpreted by means of transfer characteristic functions. Specific data values are reserved for initialization information and error indicators

- The serial message channel of the status and communication nibble may be optionally used.

    - Optional transmission of sensor transfer characteristic function(s) parameters.

    - Optional transmission of measurement data from additional sources by supplementary data channels.

- Optional pause pulse.

Position and ratio sensors with one or two dimensions have been considered. For position or ratio sensors with more dimensions or higher resolution than 12-bit the sensor type “not specified position sensors” respectively “not specified ratio sensors” or “not specified multi dimension position sensors“ shall be used (see Table D.4.1-1 ).

Simple sensors which do not implement the serial channel use default sensor parameters. The receiver-side shall know the sensor parameters and characteristic to be able to interpret the measurement data values when the serial channel is not implemented at sensor-side

A.7.1 Default Encoding Scheme and Frame Formats

Position sensors and ratio sensors shall make use of the following data channels provided by the SENT interface

- SENT signal 1 with data nibbles 1-3 carries a 12-bit data value. In the context of this appendix this denotes the “fast channel 1”.

- Optional SENT signal 2 with data nibbles 4-6 carries a 12-bit data value. In the context of this appendix this denotes the “fast channel 2”. This channel shall carry either a second position respectively ratio data signal, temperature data, pressure data, secure sensor information (see H.4) or it is stuffed with zeros.

- The status and communication nibble shall be used as specified in Table H-2 – Allocation of the bits of the status and communication nibble.

- For transmitting of optional ancillary sensor data, the sensor shall use the Enhanced Serial Message Format with 8 bit ID and 12 bit data (see Section 5.2.4.2). Supplementary data channels as specified in D.1 and the default assignment of supplementary data channels for position sensors, as specified in section D.4.2 and, Table D.4.2-1 shall be used.

The data formats and protocols of the different position and ratio sensor types are defined in Table D.4.1-1 SENT Sensor Types. The allocation of the Fast Channels to the particular SENT frame formats is specified in APPENDIX H. See section A.7.3 for sensor transfer characteristic function.

A.7.2 Reserved Signaling Ranges

Data nibble reserved values shall be used as specified in section E.1.2 and E.1.3.

A.7.3 Encoding Sensor Transfer Characteristic Functions

Details for the transfer characteristic functions for encoding position, ratio, temperature and pressure are described in the sections below.

Linear transfer function characteristics of position and pressure channels can optionally be transmitted from the sensor to the ECU using serial message channel Message IDs. These messages define the coordinates of the nodes $X_1, Y_1$ and $X_2$, $Y_2$ (see section E.2.1). Section D.1 specifies the particular serial messages which carry the 12-bit values $X_1$, $Y_1$ and $X_2$, $Y_2$. If the optional serial data channel is not implemented, application-specific transfer characteristics have to be known by the receiver.

A.7.3.1 Position Sensor Transfer Functions

See section E.2.5 for encoding of linear position, angle, and relative position or angle.

A.7.3.2 Temperature Sensor Transfer Function Characteristics

The default temperature characteristic according to E.2.2.1 is used unless the particular sensor type explicitly specifies a sensor specific temperature characteristic.

A.7.3.3 Ratio Sensor Transfer Function Characteristics

The default ratio characteristic according to E.2.3 is used.

A.7.3.4 Pressure Sensor Transfer Function Characteristics

The pressure transfer characteristic according to E.2.4 is used.

A.7.4 Serial Message Channel Definition

The serial message channel for sensors defined in Section A.7 shall be as defined in APPENDIX D.

A.7.5 SENT Status and Error Messages

See section E.3.

# APPENDIX B - CHECKSUM EXAMPLES

In order to facilitate CRC implementations and to avoid ambiguities, example values of the various CRC implementations are given in this appendix.

## B.1 4-BIT CHECKSUM EXAMPLES

This section includes example 4-bit checksum values calculated for specific data nibble values. Table B.1-1 contains examples for the legacy implementation and Table B.1-2 contains examples for the recommended implementation

***Table B.1-1 – Example 4-bit checksum calculations, legacy implementation***

| Signal 1 Data 1 Nibble 2 | Signal 1 Data 2 Nibble 3 | Signal 1 Data 3 Nibble 4 | Signal 2 Data 1 Nibble 5 | Signal 2 Data 2 Nibble 6 | Signal 2 Data 3 Nibble 7 | Checksum |
|---|---|---|---|---|---|---|
| 5 | 3 | 14 | 5 | 3 | 14 | 12 |
| 7 | 4 | 8 | 7 | 4 | 8 | 5 |
| 4 | 10 | 12 | 4 | 10 | 12 | 3 |
| 7 | 8 | 15 | 7 | 8 | 15 | 15 |
| 9 | 1 | 13 | 9 | 1 | 13 | 10 |
| 0 | 0 | 0 | 0 | 0 | 0 | 15 |

***Table B.1-2 – Example 4-bit checksum calculations, recommended implementation***

| Signal 1 Data 1 Nibble 2 | Signal 1 Data 2 Nibble 3 | Signal 1 Data 3 Nibble 4 | Signal 2 Data 1 Nibble 5 | Signal 2 Data 2 Nibble 6 | Signal 2 Data 3 Nibble 7 | Checksum |
|---|---|---|---|---|---|---|
| 5 | 3 | 14 | 5 | 3 | 14 | 15 |
| 7 | 4 | 8 | 7 | 4 | 8 | 3 |
| 4 | 10 | 12 | 4 | 10 | 12 | 10 |
| 7 | 8 | 15 | 7 | 8 | 15 | 5 |
| 9 | 1 | 13 | 9 | 1 | 13 | 6 |
| 0 | 0 | 0 | 0 | 0 | 0 | 5 |

The following example explains polynomial division for CRC calculation for line 3 of the examples.

***Table B.1-3 – Example 4-bit CRC calculation by polynomial division***

| | | seed | | | | nibble1 | | | | nibble2 | | | | nibble3 | | | | nibble4 | | | | nibble5 | | | | nibble6 | | | | augmentation | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Seed/Aug | 5/0 | | | | 5 | | | | | | | | | | | | | | | | | | | | | | | | | | | | 0 |
| data (hex) | ***4AC4AC*** | | | | | | | | 4 | | | | A | | | | C | | | | 4 | | | | A | | | | C | | | | |
| data (bin) | | **0** | **1** | **0** | **1** | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | **0** | **0** | **0** | **0** |
| polynomial | 0x1D | | ***1*** | ***1*** | ***1*** | ***0*** | ***1*** | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| polynomial | 0x1D | | | ***1*** | ***1*** | ***1*** | ***0*** | ***1*** | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| polynomial | 0x1D | | | | ***1*** | ***1*** | ***1*** | ***0*** | ***1*** | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| polynomial | 0x1D | | | | | | ***1*** | ***1*** | ***1*** | ***0*** | ***1*** | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| polynomial | 0x1D | | | | | | | | | ***1*** | ***1*** | ***1*** | ***0*** | ***1*** | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| polynomial | 0x1D | | | | | | | | | | | | | | ***1*** | ***1*** | ***1*** | ***0*** | ***1*** | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| polynomial | 0x1D | | | | | | | | | | | | | | | 1 | 1 | 1 | 0 | 1 | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| polynomial | 0x1D | | | | | | | | | | | | | | | | | ***1*** | ***1*** | ***1*** | ***0*** | ***1*** | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| polynomial | 0x1D | | | | | | | | | | | | | | | | | | ***1*** | ***1*** | ***1*** | ***0*** | ***1*** | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| polynomial | 0x1D | | | | | | | | | | | | | | | | | | | ***1*** | ***1*** | ***1*** | ***0*** | ***1*** | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| polynomial | 0x1D | | | | | | | | | | | | | | | | | | | | | ***1*** | ***1*** | ***1*** | ***0*** | ***1*** | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| polynomial | 0x1D | | | | | | | | | | | | | | | | | | | | | | | ***1*** | ***1*** | ***1*** | ***0*** | ***1*** | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| polynomial | 0x1D | | | | | | | | | | | | | | | | | | | | | | | | ***1*** | ***1*** | ***1*** | ***0*** | ***1*** | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| polynomial | 0x1D | | | | | | | | | | | | | | | | | | | | | | | | | | | ***1*** | ***1*** | ***1*** | ***0*** | ***1*** | |
| CRC (bin) | | | | | | | | | | | | | | | | | | | | | | | | | | | | 0 | 0 | **1** | **0** | **1** | **0** |
| ***CRC (hex)*** | ***0xA*** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | **A** |

Result: The CRC for the data 0x4AC4AC with the polynomial 0x1D and the seed 5 is 0xA.
Received data and CRC are used for verification calculation of the CRC.

***Table B.1-4 – Example 4-bit CRC verification by polynomial division***

| | | seed | | | | nibble1 | | | | nibble2 | | | | nibble3 | | | | nibble4 | | | | nibble5 | | | | nibble6 | | | | CRC | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Seed | 5 | 5 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| received data | ***4AC4AC*** | | | | | 4 | | | | A | | | | C | | | | 4 | | | | A | | | | C | | | | | | | |
| received CRC | 0xA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | A | | | |
| | | **0** | **1** | **0** | **1** | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | **1** | **0** | **1** | **0** |
| polynomial | 0x1D | | 1 | 1 | 1 | 0 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| polynomial | 0x1D | | | 1 | 1 | 1 | 0 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| polynomial | 0x1D | | | | 1 | 1 | 1 | 0 | 1 | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| polynomial | 0x1D | | | | | | 1 | 1 | 1 | 0 | 1 | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| polynomial | 0x1D | | | | | | | | | 1 | 1 | 1 | 0 | 1 | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| polynomial | 0x1D | | | | | | | | | | | | | | 1 | 1 | 1 | 0 | 1 | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| polynomial | 0x1D | | | | | | | | | | | | | | | 1 | 1 | 1 | 0 | 1 | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| polynomial | 0x1D | | | | | | | | | | | | | | | | | 1 | 1 | 1 | 0 | 1 | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| polynomial | 0x1D | | | | | | | | | | | | | | | | | | 1 | 1 | 1 | 0 | 1 | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| polynomial | 0x1D | | | | | | | | | | | | | | | | | | | 1 | 1 | 1 | 0 | 1 | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| polynomial | 0x1D | | | | | | | | | | | | | | | | | | | | | 1 | 1 | 1 | 0 | 1 | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| polynomial | 0x1D | | | | | | | | | | | | | | | | | | | | | | | 1 | 1 | 1 | 0 | 1 | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 |
| polynomial | 0x1D | | | | | | | | | | | | | | | | | | | | | | | | 1 | 1 | 1 | 0 | 1 | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 |
| polynomial | 0x1D | | | | | | | | | | | | | | | | | | | | | | | | | | | 1 | 1 | 1 | 0 | 1 | |
| ***Result:*** | 0x0 | | | | | | | | | | | | | | | | | | | | | | | | | | | **0** | **0** | **0** | **0** | **0** | **0** |

The verification is a pass, if the calculation result is 0.

## B.2 6- BIT CHECKSUM EXAMPLES

In order to facilitate CRC implementations and to avoid ambiguities, example values of the 6-bit CRC of the enhanced serial message communications format are given in Table B.2-1.

***Table B.2-1 – Example checksum calculations, 6-bit CRC***

| **24-bit message data *m*** | | | | | **Checksum 6-bit** |
|---|---|---|---|---|---|
| **Data value** | ***m*(0-5)** | ***m*(6-11)** | ***m*(12-17)** | ***m*(18-23)** | |
| $000000_{hex}$ | $00_{hex,6}$ ($0_{dec}$) | $00_{hex,6}$ ($0_{dec}$) | $00_{hex,6}$ ($0_{dec}$) | $00_{hex,6}$ ($0_{dec}$) | $26_{hex,6}$ ($38_{dec}$) |
| $ffffff_{hex}$ | $3f_{hex,6}$ ($63_{dec}$) | $3f_{hex,6}$ ($63_{dec}$) | $3f_{hex,6}$ ($63_{dec}$) | $3f_{hex,6}$ ($63_{dec}$) | $02_{hex,6}$ ($2_{dec}$) |
| $555555_{hex}$ | $15_{hex,6}$ ($21_{dec}$) | $15_{hex,6}$ ($21_{dec}$) | $15_{hex,6}$ ($21_{dec}$) | $15_{hex,6}$ ($21_{dec}$) | $0d_{hex,6}$ ($13_{dec}$) |
| $aaaaaa_{hex}$ | $2a_{hex,6}$ ($42_{dec}$) | $2a_{hex,6}$ ($42_{dec}$) | $2a_{hex,6}$ ($42_{dec}$) | $2a_{hex,6}$ ($42_{dec}$) | $29_{hex,6}$ ($41_{dec}$) |
| $53e53e_{hex}$ | $14_{hex,6}$ ($20_{dec}$) | $3e_{hex,6}$ ($62_{dec}$) | $14_{hex,6}$ ($20_{dec}$) | $3e_{hex,6}$ ($62_{dec}$) | $02_{hex,6}$ ($02_{dec}$) |
| $748748_{hex}$ | $1d_{hex,6}$ ($29_{dec}$) | $08_{hex,6}$ ($8_{dec}$) | $1d_{hex,6}$ ($29_{dec}$) | $08_{hex,6}$ ($8_{dec}$) | $16_{hex,6}$ ($22_{dec}$) |
| $4ac4ac_{hex}$ | $12_{hex,6}$ ($18_{dec}$) | $2c_{hex,6}$ ($44_{dec}$) | $12_{hex,6}$ ($18_{dec}$) | $2c_{hex,6}$ ($44_{dec}$) | $25_{hex,6}$ ($37_{dec}$) |
| $78f78f_{hex}$ | $1e_{hex,6}$ ($30_{dec}$) | $0f_{hex,6}$ ($15_{dec}$) | $1e_{hex,6}$ ($30_{dec}$) | $0f_{hex,6}$ ($15_{dec}$) | $3b_{hex,6}$ ($59_{dec}$) |

NOTE: Message data vector **m** corresponds to a message polynomial $P_m(x) = m_0 \cdot x^{23} + m_1 \cdot x^{22} + \ldots + m_{22} \cdot x + m_{23}$. The augmentation with six zeros corresponds to $P_m(x) \cdot x^6$.

# APPENDIX C - TESTING GUIDELINES

## C.1 EMC SUSCEPTIBILITY TESTING GUIDELINES

### C.1.1 Susceptibility Testing Failure Criteria

1) A maximum of 3 out of any consecutive 100 SENT frames is allowed to be erroneous.
2) Maximum 2 successive frames are allowed to be erroneous. (Note: for the two options in Section 5.3.3.1, one error can lead to two successive wrong frames)

### C.1.2 Error Determination of the Received SENT Frames

The receiver shall be configured with the option from section 5.3.3.1 intended for the application. During the testing, it is not necessary to count disturbed frames (which is very difficult), as only valid frames are used for evaluation of the error. Frames are evaluated based on time durations of the error which will be explained in more detail below.

Definitions:

“valid frame”: For a valid frame the receiver diagnostics mentioned in 5.3.3 detect no error.

**“**Error time**”**: The error time is measured between the last falling edge of the message, either the CRC nibble or pause pulse, of a valid frame to the first falling edge of the calibration pulse of the next valid frame. In case of two valid succeeding frames, the last falling CRC or pause pulse edge of frame 1 and the first falling calibration edge of frame 2 is exactly the same falling edge and thus the error time is 0, Figure C.1.2.1-1, Case A. This is different in case of an error in one or more frames, Figure C.1.2.1-1 Case B.

#### C.1.2.1 Determination of Received Message Error Times

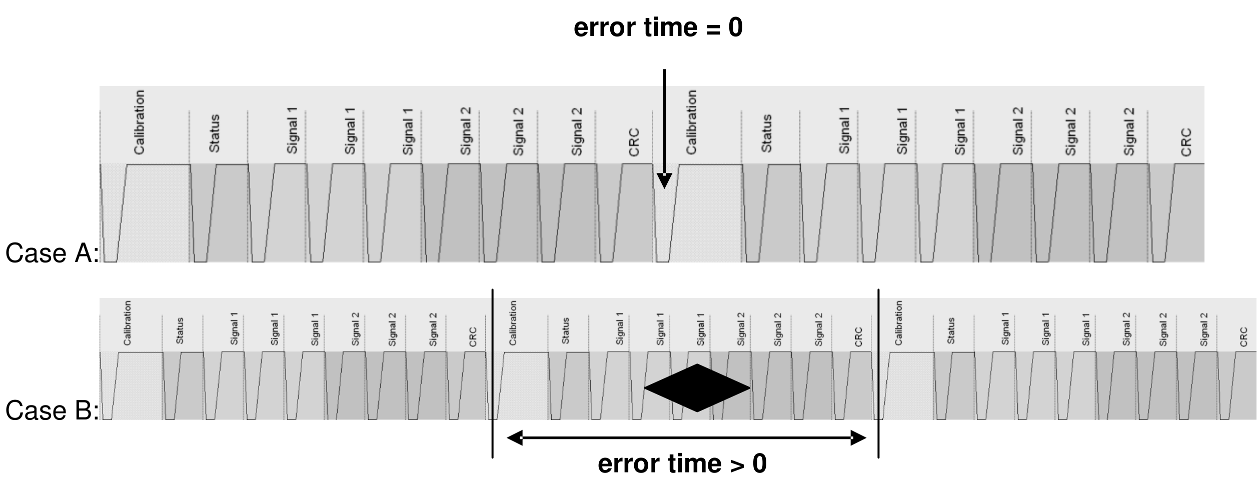



***Figure C.1.2.1-1 – Error time examples***

Discussion of Figure C.1.2.1-1

Case A: Shows the case of two valid succeeding frames for an example without pause pulse. The last falling CRC edge of frame 1 and the first falling calibration edge of frame 2 is exactly the same falling edge (black arrow) and thus the error time is 0.

Case B: Shows the case of an error in one frame (black diamond). The error time between the last falling edge of a valid frame and the first falling edge of the next valid frame is >0 and indicated (error time).

Error times are translated into a number of wrong frames as follows:

*Table C.1.2.1-1 – Number of wrong frames for error time*

| Error time | | | | | Number of wrong frames |
|---|---|---|---|---|---|
| 0 | < | error time | < | MaxFrameLength | 1 Error |
| MaxFrameLength | < | error time | < | 2xMaxFrameLength | 2 Errors |
| | | error time | > | 2xMaxFrameLength | >2 Errors, (failure) |

Table C.1.2.1-1 shows how to determine the number of errors for different error times. MaxFrameLength is the maximum nominal frame length increased by the allowable tick tolerance (see section 5.3.3). This table should also be used if no frames or pulses are received at all for some time.

C.1.2.2 Example Calculation of Worst-Case Data Transmission Delay

For a 3µs clock tick system, the proposed robustness leads to the following worst case age of the signal at the ECU (max. once during the transmission time of 100 frames):

signal age = 1ms transmission time + sensor measurement delay
(+ 200µs calibration pulse length for chapter 5.3.3 option 1) ≈ 1.5ms
worst case update rate = 2ms error time + 1ms transmission time = 3ms

worst case age of value at ECU = signal age + worst case update rate ≈ 4.5ms

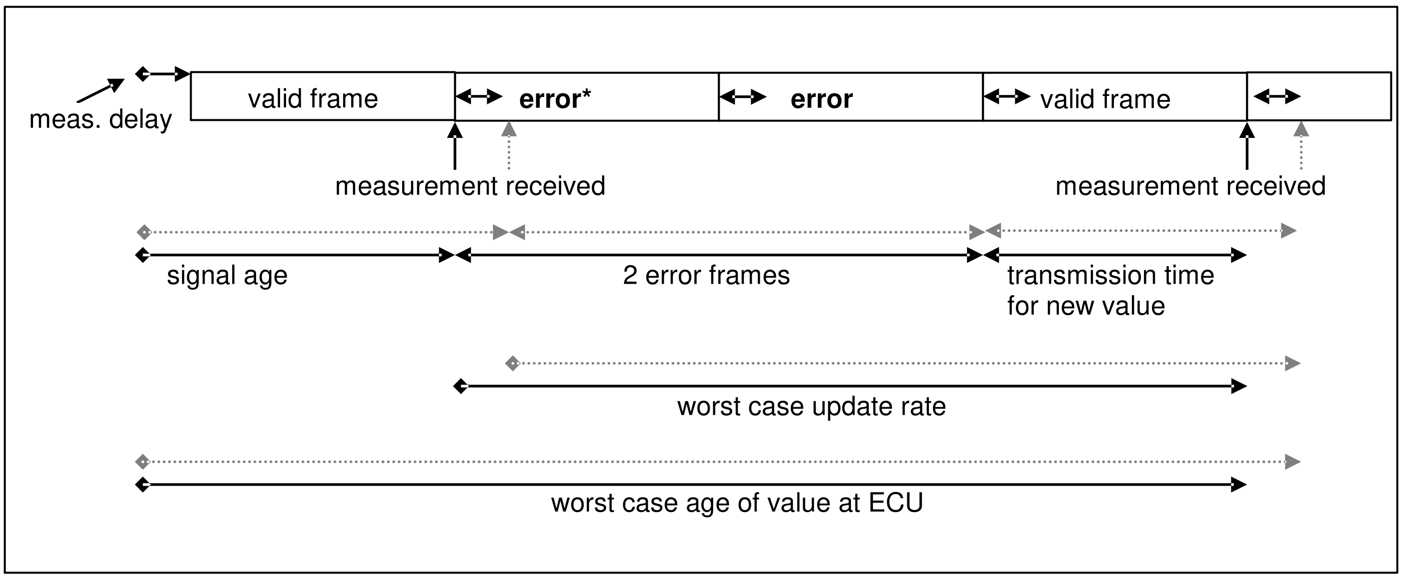



***Figure C.1.2.2-1 – Worst case age for signal as received by ECU***

Discussion of Figure C.1.2.2-1

Origin of the worst case usage of the ECU signal: For section 5.3.3.1, preferred option (grey dotted) and option 2 (black), the error is assumed to occur only after the calibration pulse (otherwise the first valid frame in Figure C.1.2.2-1 would not be valid for chapter 5.3.3.1 option 1). In the second error frame an error can be anywhere.

C.1.3 Evaluation of Received SENT Frames

C.1.3.1 Erroneous SENT Frames Undetected by the SENT CRC

Section 5.4.1 specifies the error patterns that cannot be detected by the 4-bit CRC. Therefore, due to SENT transmission errors, received data values with valid checksum may appear as deviations from the expected measurement values. It is the responsibility of the end-item user to determine acceptance/rejection criteria for data frames received in conformance with this standard.

C.1.3.2 Sensor-specific Tolerance Ranges for Applicable Fast Channel and Serial Message Data

Due to the inherent functional principles of particular sensor types, the transmitted sensor measurement values may deviate from the expected value within a sensor-specific tolerance range under disturbance conditions. These sensor-specific tolerance ranges can be derived from the tolerance ranges of their counterparts with analog sensor interfaces. It is the responsibility of the end-item user to determine acceptance/rejection criteria for data frames received in conformance with this standard.

C.2 TEST SETUP FOR SUPPLY CURRENT RIPPLE

Table 6.3.1-1 defines the peak-to-peak variation in supply current consumption over one message at a given frequency range. Since the measurement setup has some influence on the measured current ripple, the test setup as described in Figure C.2-1 shall be used. Note, that this is a test setup for the transmitter, not a specification for the receiver. Either a linear power supply or a battery may be used as a supply, but the DC output resistance $R_i$ of that source must comply to Table C.2-1. Either a) a current monitor probe or b) a differential measurement of the voltage drop across a shunt Rshunt may be used for the current measurement. The current ripple shall be evaluated after a simple 1st order low pass filter with the -3dB corner frequency $f_c$ as defined in Table 6.3.1-1

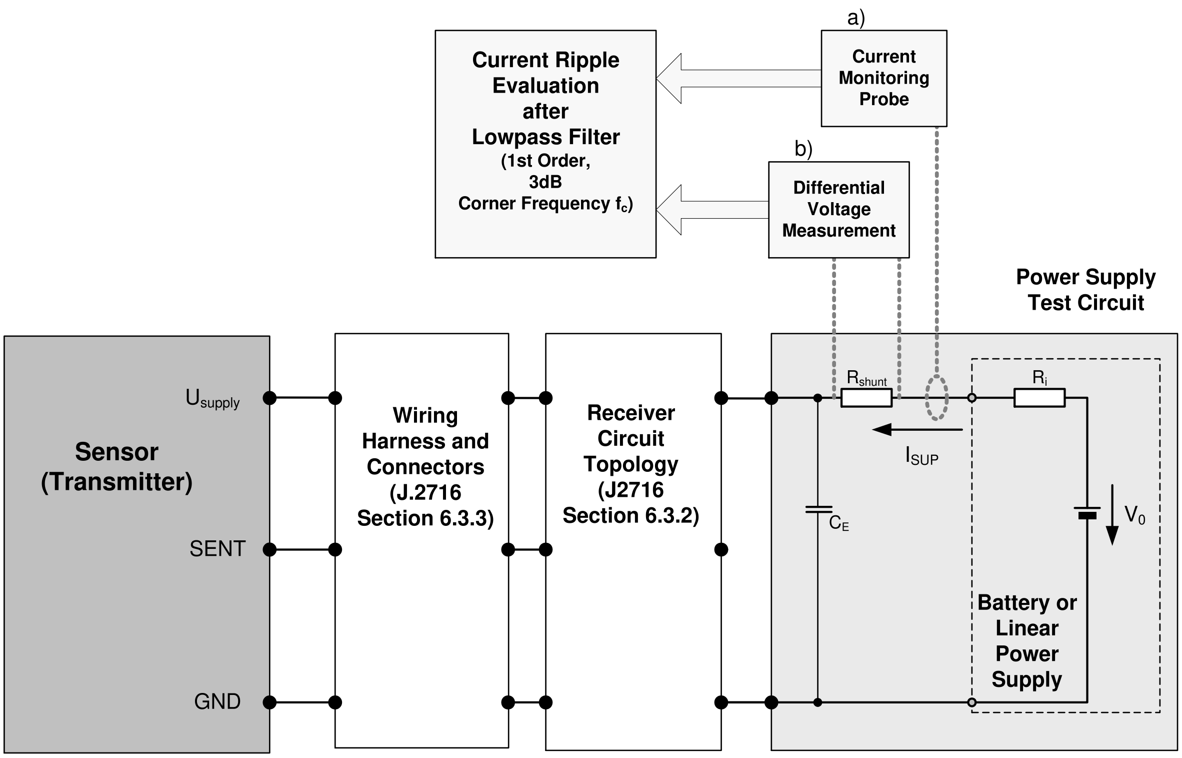



*Figure C.2-1 – Supply current ripple test setup*

*Table C.2-1 – Supply current ripple test setup circuit parameters*

| Parameter | Symbol | min | max |
|---|---|---|---|
| Open circuit output voltage of supply | $V_0$ | 4.85 V | 5.15 V |
| Output capacitance of supply | $C_E$ | - | 100 nF |
| Total DC output resistance of supply | $R_i + R_{shunt}$ | - | 3 Ω |
| Corner frequency (-3dB) of evaluation filter | $f_c$ | (1) | - |

[1] see Table 6.3.1-1.

# APPENDIX D - SERIAL CHANNEL MESSAGE IDS

## D.1 LIST OF 8-BIT MESSAGE IDS ENHANCED SERIAL MESSAGE FORMAT

Unless otherwise defined for a specific application in APPENDIX A, the list of 8-bit message IDs for Enhanced Serial Message Format shall be as given in Table D.1-1. The 8-bit IDs are transmitted when the configuration bit = 0 (serial data bit #3, serial communication nibble No. 8, see section 5.2.4.2).

The specific set of messages implemented by the sensor is optional. A sensor can implement any number of messages, including none. For the purpose of sensor and protocol identification, a recommended minimum content is defined in Table D.1-1.

***Table D.1-1 – Serial message channel 8-bit message IDs***

| Message ID (8 bit) | Recommended minimum content | Definition | Comment |
|---|---|---|---|
| $00 | | Not assigned | |
| $01 | | Error and Status Codes | See section D.5 for details |
| $02 | | reserved | |
| $03 | x | Channel 1 / 2 Sensor type | See section D.4 for details |
| $04 | | Configuration Code | Detailed specification of sensor type defined in Message $03 (Channel 1 / 2 Sensor type) |
| $05 | x | Manufacturer code | Specific codes are assigned by the SAE SENT Task Force |
| $06 | x | Protocol standard revision | See section D.3 for details |
| $07 | | Fast channel 1 Characteristic $X_1$ | Physical unit and encoding $X_{D,1}$ defined in application-specific appendices |
| $08 | | Fast channel 1 Characteristic $X_2$ | Physical unit and encoding $X_{D,2}$ defined in application-specific appendices |
| $09 | | Fast channel 1 Characteristic $Y_1$ | Encoding $Y_{D,1}$ defined in E.2.1.5. If message $09 is not transmitted, default $Y_1$ is used. Default values are specified in E.2.1.4. |
| $0A | | Fast channel 1 Characteristic $Y_2$ | Encoding $Y_{D,2}$ defined in E.2.1.5. If message $0A is not transmitted, default $Y_2$ is used. Default values are specified in E.2.1.4. |
| $0B | | Fast channel 2 Characteristic $X_1$ | See comments for Fast Channel 1 (ID $07…$0A) |
| $0C | | Fast channel 2 Characteristic $X_2$ | |
| $0D | | Fast channel 2 Characteristic $Y_1$ | |
| $0E | | Fast channel 2 Characteristic $Y_2$ | |
| $0F | | Not assigned | |
| $10 | | Supplementary data channel #1,1 | Measurement data signal |
| $11 | | Supplementary data channel #1,2 | Measurement data signal, extension |
| $12 | | Supplementary data channel #1 Characteristic $X_1$ | See comments for Fast Channel 1 (ID $07…$0A) |
| $13 | | Supplementary data channel #1 Characteristic $X_2$ | |
| $14 | | Supplementary data channel #1 Characteristic $Y_1$ | |
| $15 | | Supplementary data channel #1 Characteristic $Y_2$ | |

| Message ID (8 bit) | Recommended minimum content | Definition | Comment |
|---|---|---|---|
| $16 | | Supplementary data channel #2,1 | Measurement data signal |
| $17 | | Supplementary data channel #2,2 | Measurement data signal, extension |
| $18 | | Supplementary data channel #2 Characteristic $X_1$ | See comments for Fast Channel 1 (ID $07…$0A) |
| $19 | | Supplementary data channel #2 Characteristic $X_2$ | |
| $1A | | Supplementary data channel #2 Characteristic $Y_1$ | |
| $1B | | Supplementary data channel #2 Characteristic $Y_2$ | |
| $1C | | Supplementary data channel #3,1 | Measurement data signal |
| $1D | | Supplementary data channel #3,2 | Measurement data signal, extension |
| $1E | | Supplementary data channel #3 Characteristic $X_1$ | See comments for Fast Channel 1 (ID $07…$0A) |
| $1F | | Reserved | |
| $20 | | Supplementary data channel #3 Characteristic $X_2$ | See comments for Fast Channel 1 (ID $07…$0A) |
| $21 | | Supplementary data channel #3 Characteristic $Y_1$ | |
| $22 | | Supplementary data channel #3 Characteristic $Y_2$ | |
| $23 | | Supplementary data channel #4,1 | Measurement data signal |
| $24 | | Supplementary data channel #4,2 | Measurement data signal, extension |
| $25 | | Supplementary data channel #4 Characteristic $X_1$ | See comments for Fast Channel 1 (ID $07…$0A) |
| $26 | | Supplementary data channel #4 Characteristic $X_2$ | |
| $27 | | Supplementary data channel #4 Characteristic $Y_1$ | |
| $28 | | Supplementary data channel #4 Characteristic $Y_2$ | |
| $29 | x | Sensor ID #1 | |
| $2A | | Sensor ID #2 | |
| $2B | | Sensor ID #3 | |
| $2C | | Sensor ID #4 | |
| $2D – $7F | | Reserved | |
| $80 - $8F | | To be defined by OEM/Supplier | Not recommended to use $8F in order to increase robustness of frame start sequence |
| $90 - $97 | | ASCII character OEM codes | 6-bit representation of ASCII characters specified in 0 |
| $98 - $FF | | To be defined by OEM/Supplier | Not recommended to use $xF or $Fx in order to increase robustness of frame start sequence |

Message $07 through $0A can optionally be used for non-linear transfer characteristics.

## D.2 LIST OF 4-BIT MESSAGE IDS ENHANCED SERIAL MESSAGE FORMAT

Unless otherwise define for a specific application in APPENDIX A, the list of 4-bit message IDs for Enhanced Serial Message Format shall be as given in Table D.2-1. The 4-bit IDs are transmitted when the configuration bit = 1 (serial data bit #3, serial communication nibble No. 8, see section 5.2.4.2).

***Table D.2-1 – Serial message channel 4-bit message IDs***

| Message ID (4 bit) | Sensed Value for Data Field |
|---|---|
| $0 | Air Temperature (word) |
| $1 | Undefined |
| $2 | Humidity (word) |
| $3 | Undefined |
| $4 | Barometric Pressure(word) |
| $5 | Undefined |
| $6 | Undefined |
| $7 | Undefined |
| $8 | Undefined |
| $9 | Undefined |
| $A | Undefined |
| $B | Undefined |
| $C | Undefined |
| $D | Undefined |
| $E | Undefined |
| $F | Undefined |

## D.3 SENT STANDARD REVISION

The SENT Revision code is communicated in the Enhanced Serial Message 12-bit data field when the Enhanced Serial Message ID is $06 (Protocol standard revision). These codes are given in Table D.3-1.

***Table D.3-1 – SENT revision codes***

| 12-bit Code | Definition | Comment |
|---|---|---|
| $000 | Not specified | |
| $001 | J2716 FEB2007<br>SENT Revision 1 | Standard revision does not support enhanced serial message format |
| $002 | J2716 FEB2008<br>SENT Revision 2 | Standard revision does not support enhanced serial message format |
| $003 | J2716 JAN2010<br>SENT Revision 3 | |
| $004 | J2716 APR2016<br>SENT Revision 4 | Revision permits the use of serial message cycles which are composed of ≤ 64 messages. Whereas SENT Revision J 2716 JAN2010 specifies a serial message cycle of ≤ 32 messages. |
| $005 – $FFF | reserved | |

## D.4 SENT SENSOR CLASSES

### D.4.1 SENT Sensor Types

The definition of the sensor types transmitted in the Enhanced Serial Message 12-bit data field when the Enhanced Serial Message ID is $03 (Channel 1/2 Sensor Type) is given in Table D.4.1-1. The assignment of the sensor type definitions is done with approval of the J2716 Task Force and is tracked in a common database

The assignment at publication of this revision of this document is given in Table D.4.1-1. To request a new type or obtain the most recent list of types, contact the SENT Task Force chairperson.

***Table D.4.1-1 – SENT sensor types***

| 12-bit Code | Definition | | | Data Frame Nibble Definition | Transfer Characteristic |
|---|---|---|---|---|---|
| $000 | not specified Sensor Type | | | | |
| $001 | Pressure Sensors | P | Pressure (short) | H.2 One 12-bit fast channel<br>ch.1: Pressure | ch.1: E.2.4 Linear Pressure Transfer Characteristic Function |
| $002 | | P/- | Pressure | H.5 Two 12-bit fast channels<br>ch.1: Pressure<br>ch.2: all zeroes | ch.1: E.2.4 Linear Pressure Transfer Characteristic Function |
| $003 | | P/S | Pressure / Secure sensor | H.4 Secure Sensors with 12-bit fast channel 1 and 12-bit secure sensor information<br>ch.1: Pressure | ch.1: E.2.4 Linear Pressure Transfer Characteristic Function |
| $004 | | P/S/t | Pressure / Secure sensor / Temperature in supplementary channel | H.4 Secure Sensors with 12-bit fast channel 1 and 12-bit secure sensor information<br>ch.1: Pressure | ch.1: E.2.4 Linear Pressure Transfer Characteristic Function |
| $005 | | P/S/t | Pressure / Secure sensor / Temperature in supplementary channel | H.4 Secure Sensors with 12-bit fast channel 1 and 12-bit secure sensor information<br>ch.1: Pressure | ch.1: E.2.4 Linear Pressure Transfer Characteristic Function<br>supplementary ch.: E.2.2.3 Special Linear Temperature Transfer Characteristic Functions |
| $006 | | P/P | Pressure 1 / Pressure 2 | H.1 Two 12-bit fast channels<br>ch.1: Pressure<br>ch.2: Pressure | ch.1 / ch.2: E.2.4 Linear Pressure Transfer Characteristic Function |
| $007 | | P/T | Pressure / Temperature | H.1 Two 12-bit fast channels<br>ch.1: Pressure<br>ch.2: Temperature | ch.1: E.2.4 Linear Pressure Transfer Characteristic Function<br>ch.2: E.2.2.1 Default Linear Temperature Transfer Characteristic Function |
| $008 | | P/T | Pressure / Temperature | H.1 Two 12-bit fast channels<br>ch.1: Pressure<br>ch.2: Temperature | ch.1: E.2.4 Linear Pressure Transfer Characteristic Function<br>ch.2: E.2.2.3 Special Linear Temperature Transfer Characteristic Functions |
| $009 | | P/P/t | Pressure 1 / Pressure 2 / Temperature in supplementary channel | H.1 Two 12-bit fast channels<br>ch.1: Pressure<br>ch.2: Pressure | ch.1 / ch.2: E.2.4 Linear Pressure Transfer Characteristic Function |
| $00A | | P/P/t | Pressure 1 / Pressure 2 / Temperature in supplementary channel | H.1 Two 12-bit fast channels<br>ch.1: Pressure<br>ch.2: Pressure | ch.1 / ch.2: E.2.4 Linear Pressure Transfer Characteristic Function<br>supplementary ch.: E.2.2.3 Special Linear Temperature Transfer Characteristic Functions |

| 12-bit Code | Definition | | | Data Frame Nibble Definition | Transfer Characteristic |
|---|---|---|---|---|---|
| $00B | | P/t | Pressure (short) / Temperature in supplementary channel | H.2 One 12-bit fast channel<br>ch.1: Pressure | ch.1: E.2.4 Linear Pressure Transfer Characteristic Function<br>supplementary ch.: E.2.2.3 Special Linear Temperature Transfer Characteristic Functions |
| $00C | | P/-/t | Pressure / Temperature in supplementary channel | H.1 Two 12-bit fast channels<br>ch.1: Pressure<br>ch.2: all zeroes | ch.1: E.2.4 Linear Pressure Transfer Characteristic Function<br>supplementary ch.: E.2.2.3 Special Linear Temperature Transfer Characteristic Functions |
| $00D | | P (high Speed) | Pressure (high speed) | H.3 High-speed with one 12-bit fast channel<br>ch.1: Pressure | ch.1: E.2.4 Linear Pressure Transfer Characteristic Function |
| $00E | | P/t (high Speed) | Pressure (high speed) / Temperature in supplementary channel | H.3 High-speed with one 12-bit fast channel<br>ch.1: Pressure | ch.1: E.2.4 Linear Pressure Transfer Characteristic Function<br>supplementary ch.: E.2.2.3 Special Linear Temperature Transfer Characteristic Functions |
| $00F | | not specified Pressure Sensor | | | |
| $010 | MAF Sensors | not specified MAF Sensors | | | |
| $011 | | MAF (hi-res, lin) | | H.7: Two fast channels with 16-bit fast channel 1 and 8-bit fast channel 2<br>ch.1: MAF (16-bit)<br>ch.2: all zeroes (8-bit) | ch.1: E.2.6 Encoding of Linear MAF Transfer Function |
| $012 | | MAF (hi-res, non-lin) | | H.7: Two fast channels with 16-bit fast channel 1 and 8-bit fast channel 2<br>ch.1: MAF (16-bit)<br>ch.2: all zeroes (8-bit) | ch.1: Sensor specific MAF Transfer Function |
| $013 | | MAF (hi-res, lin) / Pressure | | H.7: Two fast channels with 16-bit fast channel 1 and 8-bit fast channel 2<br>ch.1: MAF (16-bit)<br>ch.2: Pressure (8-bit) | ch.1: E.2.6 Encoding of Linear MAF Transfer Function<br>ch 2: E.2.4 Linear Pressure Transfer Characteristic Function |
| $014 | | MAF (hi-res, non-lin) / Pressure | | H.7: Two fast channels with 16-bit fast channel 1 and 8-bit fast channel 2<br>ch.1: MAF (16-bit)<br>ch.2: Pressure (8-bit) | ch.1: Sensor specific MAF Transfer Function<br>ch 2: E.2.4 Linear Pressure Transfer Characteristic Function |
| $015 | | MAF (lin) / Pressure (hi-res) | | H.6: Two fast channels with 14-bit fast channel 1 and 10-bit fast channel 2<br>ch.1: MAF (14-bit)<br>ch.2: Pressure (10-bit) | ch.1: E.2.6 Encoding of Linear MAF Transfer Function<br>ch 2: E.2.4 Linear Pressure Transfer Characteristic Function |

| 12-bit Code | Definition | | Data Frame Nibble Definition | Transfer Characteristic |
|---|---|---|---|---|
| $016 | | MAF (non-lin) / Pressure (hi-res) | H.6: Two fast channels with 14-bit fast channel 1 and 10-bit fast channel 2<br>ch.1: MAF (14-bit)<br>ch.2: Pressure (10-bit) | ch.1: Sensor specific MAF Transfer Function<br>ch 2: E.2.4 Linear Pressure Transfer Characteristic Function |
| $017 | | MAF (lin) | H.6: Two fast channels with 14-bit fast channel 1 and 10-bit fast channel 2<br>ch.1: MAF (14bit)<br>ch.2: all zeroes (10-bit) | ch.1: E.2.6 Encoding of Linear MAF Transfer Function |
| $018 | | MAF (non-lin) | H.6: Two fast channels with 14-bit fast channel 1 and 10-bit fast channel 2<br>ch.1: MAF (14bit)<br>ch.2: all zeroes (10-bit) | ch.1: Sensor specific MAF Transfer Function |
| $019 | | MAF (lin) / Temperature | H.6: Two fast channels with 14-bit fast channel 1 and 10-bit fast channel 2<br>ch.1: MAF (14-bit)<br>ch.2: Temperature (10-bit) | ch.1: E.2.6 Encoding of Linear MAF Transfer Function<br>ch 2: E.2.2.1 Default Linear Temperature Transfer Characteristic Function |
| $01A | | MAF (non-lin) / Temperature | H.6: Two fast channels with 14-bit fast channel 1 and 10-bit fast channel 2<br>ch.1: MAF (14-bit)<br>ch.2: Temperature (10-bit) | ch.1: Sensor specific MAF Transfer Function<br>ch 2: E.2.2.1 Default Linear Temperature Transfer Characteristic Function |
| $01B | | MAF (hi-res, lin) / Temperature | H.7: Two fast channels with 16-bit fast channel 1 and 8-bit fast channel 2<br>ch.1: MAF (16-bit)<br>ch.2: Temperature (8-bit) | ch.1: E.2.6 Encoding of Linear MAF Transfer Function<br>ch 2: E.2.2.1 Default Linear Temperature Transfer Characteristic Function |
| $01C | | MAF (hi-res, non-lin) / Temperature | H.7: Two fast channels with 16-bit fast channel 1 and 8-bit fast channel 2<br>ch.1: MAF (16-bit)<br>ch.2: Temperature (8-bit) | ch.1: Sensor specific MAF Transfer Function<br>ch 2: E.2.2.1 Default Linear Temperature Transfer Characteristic Function |
| $01D | | MAF (low-res, non-lin) | H.1 Two 12-bit fast channels<br>ch.1: MAF (12bit)<br>ch.2: all zeroes (12-bit) | ch.1: Sensor specific MAF Transfer Function |
| $01E | | MAF (low-res, non-lin) / Pressure | H.1 Two 12-bit fast channels<br>ch.1: MAF (12-bit)<br>ch.2: Pressure (12-bit) | ch.1: Sensor specific MAF Transfer Function<br>ch 2: E.2.4 Linear Pressure Transfer Characteristic Function |

| 12-bit Code | Definition | | Data Frame Nibble Definition | Transfer Characteristic |
|---|---|---|---|---|
| $01F | | MAF (low-res, non-lin) / Temperature | H.1 Two 12-bit fast channels<br>ch.1: MAF (12-bit)<br>ch.2: Temperature (12-bit) | ch.1: Sensor specific MAF Transfer Function<br>ch 2: E.2.2.1 Default Linear Temperature Transfer Characteristic Function |
| $020 - $02F | | reserved for MAF Sensors | | |
| $030 - $03F | Pressure Sensors (cont'd) | reserved for Pressure Sensors (cont'd) | | |
| $040 | Temperature Sensors | not specified Temperature Sensors | | |
| $041 | | Temperature | H.1 Two 12-bit fast channels<br>ch.1: Temperature<br>ch.2: all zeroes | ch.1: E.2.2.1 Default Linear Temperature Transfer Characteristic Function |
| $042 | | Temperature / Secure Sensor | H.4 Secure Sensors with 12-bit fast channel 1 and 12-bit secure sensor information<br>ch.1: Temperature | E.2.2.1 Default Linear Temperature Transfer Characteristic Function |
| $043 | | Temperature / Temperature | H.1 Two 12-bit fast channels<br>ch.1: Temperature<br>ch.2: Temperature | ch.1 / ch.2: E.2.2.1 Default Linear Temperature Transfer Characteristic Function |
| $044 | | Temperature Sensor Cluster | F.6.1 Data Frames with one multiplexed fast channel and 12-bit sensor data (F1.5); 1-16 Frame Controls (FC)<br>all ch.: Temperature | all channel: E.2.2.1 Default Linear Temperature Transfer Characteristic Function |
| $045 | | High Temperature | H.1 Two 12-bit fast channels<br>ch.1: Temperature<br>ch.2: all zeroes | ch.1: E.2.2.2 Linear High Temperature Transfer Characteristic Functions |
| $046 | | High Temperature / Secure Sensor | H.4 Secure Sensors with 12-bit fast channel 1 and 12-bit secure sensor information<br>ch.1: Temperature | E.2.2.2 Linear High Temperature Transfer Characteristic Functions |
| $047 | | High Temperature / High Temperature | H.1 Two 12-bit fast channels<br>ch.1: Temperature<br>ch.2: Temperature | ch.1 / ch.2: E.2.2.2 Linear High Temperature Transfer Characteristic Functions |
| $048 | | High Temperature Sensor Cluster | F.6.1 Data Frames with one multiplexed fast channel and 12-bit sensor data (F1.5); 1-16 Frame Controls (FC)<br>all ch.: Temperature | all channel: E.2.2.2 Linear High Temperature Transfer Characteristic Functions |
| $049 | | Special Temperature | H.1 Two 12-bit fast channels<br>ch.1: Temperature<br>ch.2: all zeroes | ch.1: E.2.2.3 Special Linear Temperature Transfer Characteristic Functions |

| 12-bit Code | Definition | | Data Frame Nibble Definition | Transfer Characteristic |
|---|---|---|---|---|
| $04A | | Special Temperature / Secure Sensor | H.4 Secure Sensors with 12-bit fast channel 1 and 12-bit secure sensor information<br>ch.1: Temperature | ch.1: E.2.2.3 Special Linear Temperature Transfer Characteristic Functions |
| $04B | | Special Temperature / Special Temperature | H.1 Two 12-bit fast channels<br>ch.1: Temperature<br>ch.2: Temperature | ch.1 / ch.2: E.2.2.3 Special Linear Temperature Transfer Characteristic Functions |
| $04C | | Special Temperature Sensor Cluster | F.6.1 Data Frames with one multiplexed fast channel and 12-bit sensor data (F1.5); 1-16 Frame Controls (FC)<br>all ch.: Temperature | all channel: E.2.2.3 Special Linear Temperature Transfer Characteristic Functions |
| $04E - $04F | | reserved for Temperature Sensors | | |
| $050 | Position Sensors | not specified Position Sensors | | |
| $051 | | Linear Position | H.2 One 12-bit fast channel<br>ch.1: Linear Position | ch.1: E.2.5 Encoding of Position Sensor Transfer Functions |
| $052 | | Linear Position | H.1 Two 12-bit fast channels<br>ch.1: Linear Position<br>ch.2: all zeroes | ch.1: E.2.5 Encoding of Position Sensor Transfer Functions |
| $053 | | Linear Position (high speed) | H.3 High-speed with one 12-bit fast channel<br>ch.1: Linear Position | ch.1: E.2.5 Encoding of Position Sensor Transfer Functions |
| $054 | | Linear Position 1 / Linear Position 2 | H.1 Two 12-bit fast channels<br>ch.1: Linear Position<br>ch.2: Linear Position | ch.1 / ch.2: E.2.5 Encoding of Position Sensor Transfer Functions |
| $055 | | Linear Position / Secure Sensor | H.4 Secure Sensors with 12-bit fast channel 1 and 12-bit secure sensor information<br>ch.1: Linear Position | ch.1: E.2.5 Encoding of Position Sensor Transfer Functions |
| $056 | | Linear Position / Temperature | H.1 Two 12-bit fast channels<br>ch.1: Linear Position<br>ch.2: Temperature | ch.1: E.2.5 Encoding of Position Sensor Transfer Functions<br>ch.2: E.2.2.1 Default Linear Temperature Transfer Characteristic Function |
| $057 | | Linear Position / Ratio | H.1 Two 12-bit fast channels<br>ch.1: Linear Position<br>ch.2: Ratio | ch.1: E.2.5 Encoding of Position Sensor Transfer Functions<br>ch.2: E.2.3 Encoding Ratio Sensing Transfer Functions |
| $058 - $05F | | reserved for Linear Position Sensors | | |
| $060 | | Angle Position | H.2 One 12-bit fast channel<br>ch.1: Angle Position | ch.1: E.2.5 Encoding of Position Sensor Transfer Functions |

| 12-bit Code | Definition | | Data Frame Nibble Definition | Transfer Characteristic |
|---|---|---|---|---|
| $061 | | Angle Position | H.1 Two 12-bit fast channels<br>ch.1: Angle Position<br>ch.2: all zeroes | ch.1: E.2.5 Encoding of Position Sensor Transfer Functions |
| $062 | | Angle Position (high speed) | H.3 High-speed with one 12-bit fast channel<br>ch.1: Angle Position | ch.1: E.2.5 Encoding of Position Sensor Transfer Functions |
| $063 | | Angle Position 1 / Angle Position 2 | H.1 Two 12-bit fast channels<br>ch.1: Angle Position<br>ch.2: Angle Position | ch.1 / ch.2: E.2.5 Encoding of Position Sensor Transfer Functions |
| $064 | | Angle Position / Secure Sensor | H.4 Secure Sensors with 12-bit fast channel 1 and 12-bit secure sensor information<br>ch.1: Angle Position | ch.1: E.2.5 Encoding of Position Sensor Transfer Functions |
| $065 | | Angle Position Sensor / Temperature | H.1 Two 12-bit fast channels<br>ch.1: Angle Position<br>ch.2: Temperature | ch.1: E.2.5 Encoding of Position Sensor Transfer Functions<br>ch.2: E.2.2.1 Default Linear Temperature Transfer Characteristic Function |
| $066 - $06F | | reserved for Angle Position Sensors | | |
| $070 | | Relative Position | H.2 One 12-bit fast channel<br>ch.1: Relative Position | ch.1: E.2.5 Encoding of Position Sensor Transfer Functions |
| $071 | | Relative Position | H.1 Two 12-bit fast channels<br>ch.1: Relative Position<br>ch.2: all zeroes | ch.1: E.2.5 Encoding of Position Sensor Transfer Functions |
| $072 | | Relative Position (high speed) | H.3 High-speed with one 12-bit fast channel<br>ch.1: Relative Position | ch.1: E.2.5 Encoding of Position Sensor Transfer Functions |
| $073 | | Relative Position 1 / Relative Position 2 | H.1 Two 12-bit fast channels<br>ch.1: Relative Position<br>ch.2: Relative Position | ch.1 / ch.2: E.2.5 Encoding of Position Sensor Transfer Functions |
| $074 | | Relative Position / Secure Sensor | H.4 Secure Sensors with 12-bit fast channel 1 and 12-bit secure sensor information<br>ch.1: Relative Position | ch.1: E.2.5 Encoding of Position Sensor Transfer Functions |
| $075 | | Relative Position / Temperature | H.1 Two 12-bit fast channels<br>ch.1: Relative Position<br>ch.2: Temperature | ch.1: E.2.5 Encoding of Position Sensor Transfer Functions<br>ch.2: E.2.2.1 Default Linear Temperature Transfer Characteristic Function |
| $076 | | Relative Position / Ratio | H.1 Two 12-bit fast channels<br>ch.1: Relative Position<br>ch.2: Ratio | ch.1: E.2.5 Encoding of Position Sensor Transfer Functions<br>ch.2: E.2.3 Encoding Ratio Sensing Transfer Functions |
| $077 - $07F | | reserved for Relative Position Sensors | | |

| 12-bit Code | Definition | | Data Frame Nibble Definition | Transfer Characteristic |
|---|---|---|---|---|
| $080 | | Coded Position | H.2 One 12-bit fast channel<br>ch.1: Coded Position | ch.1: application specific |
| $081 | | Coded Position | H.1 Two 12-bit fast channels<br>ch.1: Coded Position<br>ch.2: all zeroes | ch.1: application specific |
| $082 | | Coded Position (high speed) | H.3 High-speed with one 12-bit fast channel<br>ch.1: Coded Position | ch.1: application specific |
| $083 | | Coded Position 1 / Coded Position 2 | H.1 Two 12-bit fast channels<br>ch.1: Coded Position<br>ch.2: Coded Position | ch.1 / ch.2: application specific |
| $084 | | Coded Position / Secure Sensor | H.4 Secure Sensors with 12-bit fast channel 1 and 12-bit secure sensor information<br>ch.1: Coded Position | ch.1: application specific |
| $085 | | Coded Position / Temperature | H.1 Two 12-bit fast channels<br>ch.1: Coded Position<br>ch.2: Temperature | ch.1: application specific<br>ch.2: E.2.2.1 Default Linear Temperature Transfer Characteristic Function |
| $086 - $08F | | reserved for Coded Position Sensors | | |
| $090 | | not specified multi Dimension Position Sensors | Appendix F.6 Basic Fast Channel Multiplexing SENT Data Frame Formats | |
| $091 - $09F | | reserved for multi Dimension Position Sensors | | |
| $0A0 - $0AF | | reserved for Position Sensors | | |
| $0B0 | Ratio Sensors | not specified Ratio Sensors | | |
| $0B1 | | Ratio | H.2 One 12-bit fast channel<br>ch.1: Ratio | ch.1: E.2.3 Encoding Ratio Sensing Transfer Functions |
| $0B2 | | Ratio | H.1 Two 12-bit fast channels<br>ch.1: Ratio<br>ch.2: all zeroes | ch.1: E.2.3 Encoding Ratio Sensing Transfer Functions |
| $0B3 | | Ratio (high speed) | H.3 High-speed with one 12-bit fast channel<br>ch.1: Ratio | ch.1: E.2.3 Encoding Ratio Sensing Transfer Functions |
| $0B4 | | Ratio 1 / Ratio 2 | H.1 Two 12-bit fast channels<br>ch.1: Ratio<br>ch.2: Ratio | ch.1 / ch.2: E.2.3 Encoding Ratio Sensing Transfer Functions |
| $0B5 | | Ratio / Secure Sensor | H.4 Secure Sensors with 12-bit fast channel 1 and 12-bit secure sensor information<br>ch.1: Ratio | ch.1: E.2.3 Encoding Ratio Sensing Transfer Functions |

| 12-bit Code | Definition | | Data Frame Nibble Definition | Transfer Characteristic |
|---|---|---|---|---|
| $0B6 | | Ratio / Temperature | H.1 Two 12-bit fast channels<br>ch.1: Ratio<br>ch.2: Temperature | ch.1: E.2.3 Encoding Ratio Sensing Transfer Functions<br>ch.2: E.2.2.1 Default Linear Temperature Transfer Characteristic Function |
| $0B7 | | Ratio / Pressure | H.1 Two 12-bit fast channels<br>ch.1: Ratio<br>ch.2: Pressure | ch.1: E.2.3 Encoding Ratio Sensing Transfer Functions<br>ch.2: E.2.4 Linear Pressure Transfer Characteristic Function |
| $0B8 - $0BF | | reserved for Ratio Sensors | | |
| $0C0 | Acceleration Sensors | not specified Acceleration Sensors | | |
| $0C1 - $0CF | | reserved for Acceleration Sensors | | |
| $0D0 - $FFE | reserved for further Sensor Classes Definitions | | | |
| $FFF | not specified Sensor Type | | | |

NOTE:

- If not otherwise defined the supplementary channel definition is according to generic definition within Appendix D.4.2.
- Sensor class only to be assigned if the definition is complete and unambiguous in terms of Data Frame Definition and Transfer Characteristics

D.4.2 Default Assignment of Supplementary Data Channels for Specific Sensor Classes

A default assignment of supplementary data channels is defined for specific sensor classes. This avoids definition of additional sensor types for each possible combination of supplementary data channels.

A sensor type according to Section D.4.1 is specified by the data transmitted on the particular fast channels. If data is transmitted on the supplementary data channels and the sensor type definition does not define a specific use of the supplementary data channel, the default definition according to Table D.4.2-1 is used. The presence or absence of the supplementary data channels shows whether the sensor supports the specified measurement data of the respective supplementary channel.

Supplementary data channels which differ from the assignments listed in Table D.4.2-1 are possible. This is indicated by specific sensor types in Table D.4.1-1.

Example:

Sensor Type $001 (P) uses the default assignment of the supplementary data channels, if the respective supplementary data channels are used.

Sensor type $00B (P/t) transmits a medium temperature on supplementary channel #1 with a sensor specific temperature characteristic (not the default characteristic).

NOTE: Sensor types $004 (P/S/t) and $009 (P/P/t) use the supplementary data channels compliant with Table D.4.2-1. These sensor types are kept in Table D.4.1-1 for compatibility reasons with SENT J2716 JAN2010.

For not specified sensor types (e.g., $000, $00F, $010, etc.) the assignment of the supplementary data channels is not specified.

*Table D.4.2-1 – SENT sensor classes*

| Sensor Class | Supplementary Data Channels on Serial Message Channel | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | #1,1 ID# 1 0 | #1,2 ID# 1 1 | #2,1 ID# 1 6 | #2,2 ID# 1 7 | #3,1 ID# 1 C | #3,2 ID# 1 D | #4,1 ID# 2 3 | #4,2 ID# 2 4 |
| | Device 1 | Device 2 | Device 1 | Device 2 | Device 1 | Device 2 | Device 1 | Device 2 |
| Pres-sure Sensors | Medium temper-ature (1) | Medium temper-ature (1) | Reserved | Reserved | To be defined by OEM/ supplier | To be defined by OEM/ supplier | Internal reference temper-ature (1) | Internal reference temper-ature (1) |
| Mass Air Flow Sensors | Medium temper-ature (1) | Tempe-rature at humidity sensor element (1) | Humidity | To be defined by OEM/ supplier | Baro-metric pressure | Reserved | Internal reference temper-ature (1) | Internal reference temper-ature (1) |
| Position Sensors | Medium temper-ature (1) | Reserved | Reserved | Reserved | To be defined by OEM/ supplier | To be defined by OEM/ supplier | Internal reference temper-ature (1) | Internal reference temper-ature (1) |
| Temper-ature Sensors | Medium temper-ature (2) | Medium temper-ature (2) | Reserved | Reserved | To be defined by OEM/ supplier | To be defined by OEM/ supplier | Internal reference temper-ature (1) | Internal reference temper-ature (1) |
| Ratio Sensors | Medium temper-ature (1) | Reserved | Reserved | Reserved | To be defined by OEM/ supplier | To be defined by OEM/ supplier | Internal reference temper-ature (1) | Internal reference temper-ature (1) |

[1] Linear default temperature characteristic according to E.2.2.1

[2] Same temperature characteristic function as specified by particular sensor type (if the sensor type specifies a transfer characteristic function)

## D.5 LIST OF ERROR AND STATUS MESSAGES

The list of error and status messages transmitted in the 12-bit Enhanced Serial Message data field when Enhance Serial Message ID is $01 (Error and Status Codes) is given in Table D.5-1.

In case more than one error is detected, the behavior has to be specified by the sensor specification.

*Table D.5-1 – Definition of error and status messages, transmitted over the serial channel*

| 12-bit Code | Definition | | Comment |
|---|---|---|---|
| $000 | No error | | |
| $001 | Fast Channel Error and Status | Channel 1 out of range high | |
| $002 | | Channel 1 out of range low | |
| $003 | | Initialization error (Channel 1) | |
| $004 | | Channel 2 out of range high | |
| $005 | | Channel 2 out of range low | |
| $006 | | Initialization error (Channel 2) | |
| $007 | | Channel 1 and 2 rationality error | |
| $008 - $01F | | Reserved for Fast Channel Status | |
| $020 | Sensor Error and Status | Undervoltage Status | threshold to be determined depending on the requirements of the specific sensor implementation |
| $021 | | Overvoltage Status | |
| $022 | | Overtemperature Status | |
| $023 - $02F | | Reserved for Sensor Error and Status | |
| $030 - $100 | Reserved | | |
| $1[FC]1 | Fast Channel Error and Status with Fast Channel Multiplexing | First Channel of multiplexed Frame defined by FC out of range high | Fast Channel Allocation specified by frame control nibble (FC) |
| $1[FC]2 | | First Channel of multiplexed Frame defined by FC out of range low | |
| $1[FC]3 | | Initialization error (First Channel of multiplexed Frame defined by FC) | |
| $1[FC]4 | | Second Channel of multiplexed Frame defined by FC out of range high | |
| $1[FC]5 | | Second Channel of multiplexed Frame defined by FC out of range low | |
| $1[FC]6 | | Initialization error (Second Channel of multiplexed Frame defined by FC) | |
| $1[FC]7 | | First and Second Channel of multiplexed Frame defined by FC rationality error | |
| $1[FC]8 - $1[FC]F | | Reserved for multiplexed Fast Channel Status | |
| $200 - $400 | Reserved | | |
| $401 | Supplementary Data Channel Error and Status | Supplementary Data Channel #1,1 signal out of range high | |
| $402 | | Supplementary Data Channel #1,1 signal out of range low | |
| $403 | | Supplementary Data Channel #1,1 signal initialization error | |
| $404 | | Supplementary Data Channel #2,1 signal out of range high | |
| $405 | | Supplementary Data Channel #2,1 signal out of range low | |
| $406 | | Supplementary Data Channel #2,1 signal initialization error | |
| $407 | | Supplementary Data Channel #3,1 signal out of range high | |
| $408 | | Supplementary Data Channel #3,1 signal out of range low | |
| $409 | | Supplementary Data Channel #3,1 signal initialization error | |
| $40A | | Supplementary Data Channel #4,1 signal out of range high | |
| $40B | | Supplementary Data Channel #4,1 signal out of range low | |
| $40C | | Supplementary Data Channel #4,1 signal initialization error | |
| $40D | | Supplementary Data Channel #1,2 signal out of range high | |
| $40E | | Supplementary Data Channel #1,2 signal out of range low | |

| 12-bit Code | Definition | | Comment |
|---|---|---|---|
| $40F | | Supplementary Data Channel #1,2 signal initialization error | |
| $410 | | Supplementary Data Channel #2,2 signal out of range high | |
| $411 | | Supplementary Data Channel #2,2 signal out of range low | |
| $412 | | Supplementary Data Channel #2,2 signal initialization error | |
| $413 | | Supplementary Data Channel #3,2 signal out of range high | |
| $414 | | Supplementary Data Channel #3,2 signal out of range low | |
| $415 | | Supplementary Data Channel #3,2 signal initialization error | |
| $416 | | Supplementary Data Channel #4,2 signal out of range high | |
| $417 | | Supplementary Data Channel #4,2 signal out of range low | |
| $418 | | Supplementary Data Channel #4,2 signal initialization error | |
| $419 - $41F | | Reserved for Supplementary Data Channel Error and Status | |
| $420 - $7FF | Reserved | | |
| $800 - $FFF | To be defined by OEM / supplier | | |

## D.6 SERIAL MESSAGE CHANNEL SCHEDULE AND RULES

The catalog of serial messages is defined in Section D.1. It is important to note that all serial messages are optional. The sensor determines the subset of serial messages that are required by the particular application.

One serial message cycle shall be composed of ≤ 64 messages

- All messages that are used by the sensor are transmitted within this cycle
- ECU knows after ≤ 64 messages which set of message IDs is being used
- Message IDs can be used multiple times within one cycle
- If message IDs are not used within the message cycle, default values as specified by the standard are assumed by the ECU
- Example: Rate of error and status messages or supplementary data channels can be changed dynamically when necessary

The maximum number of serial messages within one serial message cycle is limited for two reasons: The receiver has to know which serial messages (message IDs) from the catalog are being used, and the receiver has to know which messages from the catalog are not being used. The second point is very important, because the receiver knows then, which default values are to be applied (e.g., default values $Y_1$ and $Y_2$, if transfer characteristic values $X_1$ and $X_2$ are transmitted, but no messages with $Y_1$ and $Y_2$ are transmitted).

The maximum length of a serial message cycle is limited to 64 messages since the receiver has to store and compare the IDs of the received messages in order to determine the set of messages used by the transmitter. Moreover, the length of a message cycle limits the time it takes for the receiver to obtain all sensor-specific parameters.

The application of the serial message channel schedule and rules is illustrated by the example in Table D.6-1 and Figure D.6-1.

***Table D.6-1 – Example: Sub-set of message IDs that are used by a sensor within one serial message cycle***

| Message number in serial message cycle | 8-bit Message ID | Comment |
|---|---|---|
| 1 | $01 Error and status codes | |
| 2 | $03 Channel 1 / 2 sensor type | |
| 3 | $05 Manufacturer code | |
| 4 | $06 Protocol standard revision | |
| 5 | $07 Fast channel 1 characteristic $X_1$ | Receiver uses default $Y_1$ |
| 6 | $23 Supplementary data channel #4,1 | e.g., internal reference temperature |
| 7 | $10 Supplementary data channel #1,1 | e.g., medium temperature |
| 8 | $08 Fast channel 1 characteristic $X_2$ | Receiver uses default $Y_2$ |
| 9 | $01 Error and status codes | |
| 10 | $29 Sensor ID #1 | |
| 11 | $2A Sensor ID #2 | |
| 12 | $2B Sensor ID #3 | |
| 13 | $2C Sensor ID #4 | |
| 14 | $10 Supplementary data channel #1,1 | e.g., medium temperature |

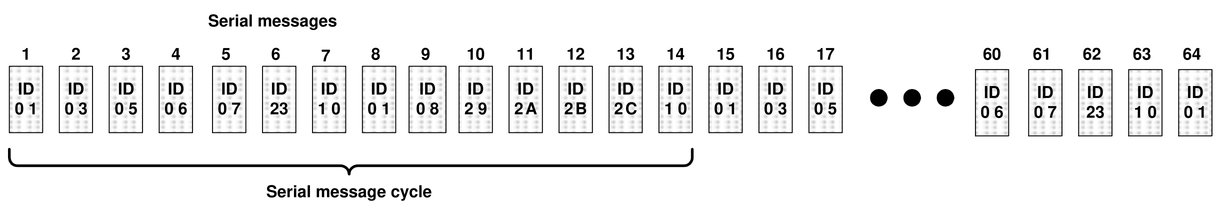



Figure D.6-1 – Example: Serial message cycle with 14 messages (Table D.6-1) is transmitted continuously. Serial message cycle can be captured within 64 messages.

## D.7 INITIALIZATION

For sensors using the enhanced serial message to communicate sensor configuration information to the receiver (ECU), the following items apply:

- Sensor-specific parameters transmitted over SENT serial channel are not available during ECU initialization (see Figure D.7-1)
- Sensor-specific parameters available when ECU is in execution phase
- Duration of one serial frame can exceed ECU initialization phase
- Prior to receiving all the configuration information the receiver either uses stored sensor characteristic function from previous execution, default values

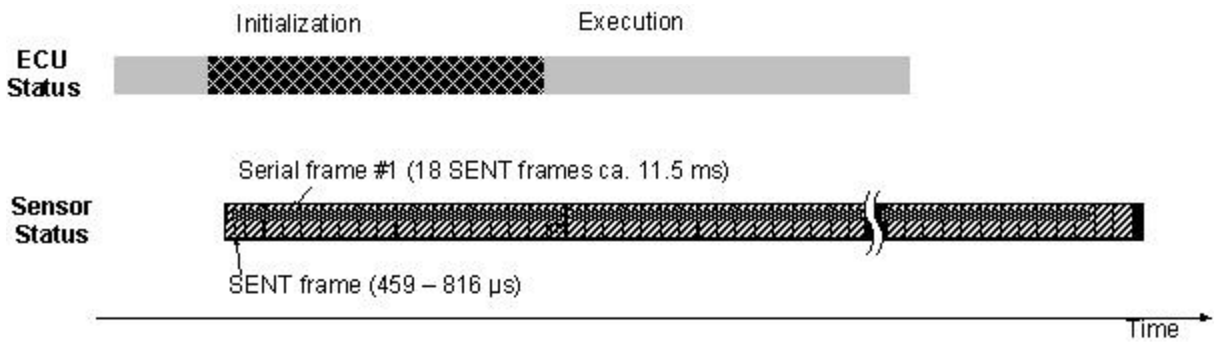



***Figure D.7-1 – Start-up phase of the sensor and the ECU***

D.7.1 Initialization Error

The following requirements apply to an initialization error detected in the transmitting module:

- After an initialization error is detected in the transmitter and communicated to the receiver, the transmitter shall reset and try to initialize again (if possible) as opposed to staying in some indeterminate state. This allows the transmitter to recover if the initialization problem does not reoccur and for the receiver to debounce the initialization error.

- While transmitting the initialization error to the receiver module using the serial message channel, the transmitter may attempt to send data in the fast channel but if the integrity of the data is in question, the error bit shall be set and it is highly recommended to set the data to a specific error data range value.

## D.8 MANUFACTURER CODE

The predefined Manufacturer Code is communicated in the Enhanced Serial Message 12-bit data field when the Enhanced Serial Message ID is $05 (Manufacturer code). The assignment of the Manufacturer Code is done with approval of the J2716 Task Force and is tracked in a common database. Exceptions are the 12-bit code values of $000 and $FFF which are not to be assigned.

The assignment at publication of this revision of this document is given in Table D.8-1. Codes unspecified in the Table are reserved for future use. To request a code or obtain the most recent list of codes, contact the SENT Task Force chairperson.

**Table D.8-1 – Manufacturers codes**

| Manufacturer code | | Definition | Comment | |
|---|---|---|---|---|
| 12-bit | 8-bit | | *ASCII* | |
| $000 | $00 | not specified | *NUL* | |
| $001 | $01 | Bosch | *SOH* | Not for future use |
| $002 | $02 | Hitachi | *STX* | |
| $003 | $03 | Continental | *ETX* | Continental also assigned $080 / $80 and $043 / $43 |
| $004 | $04 | Infineon | *EOT* | Infineon also assigned $049 / $49 |
| $005 | $05 | Sensata | *ENQ* | |
| $006 | $06 | Melexis | *ACK* | |
| $007 | $07 | Micronas | *BEL* | |
| $008 | $08 | Austria Micro Systems | *BS* | |
| $009 | $09 | Denso | *HT* | |
| $010 | $10 | Bosch | *DLE* | Not for future use |
| $012 | $12 | Stoneridge Inc | *DC2* | |
| $020 | $20 | SiemensVDO | *SP* | Not for future use |
| $032 | $32 | i2s Intelligente Sensorsysteme | *2* | |
| $040 | $40 | Autoliv | *@* | Not for future use |
| $041 | $41 | Autoliv | *A* | |
| $042 | $42 | Bosch | *B* | Bosch also assigned $001 / $01 and $010 / $10 |
| $043 | $43 | Continental | *C* | Continental also assigned $003 / $03 and $080 / $80 |
| $045 | $45 | Elmos | *E* | |
| $046 | $46 | Freescale | *F* | |
| $048 | $48 | Hella | *H* | |
| $049 | $49 | Infineon | *I* | |
| $04E | $4E | NXP Semiconductors | *N* | |
| $04F | $4F | OnSemi | *O* | |
| $053 | $53 | ST Microelectronics | *S* | |
| $054 | $54 | TRW | *T* | |
| $056 | $56 | Valeo | *V* | |
| $05A | $5A | ZMDI | *Z* | |
| $069 | $69 | IHR | *i* | |
| $073 | $73 | Seskion | *s* | |
| $080 | $80 | Continental | | Not for future use |
| $0FF - $FFF | $FF | not specified | | |

## D.9 ASCII CHARACTER OEM CODES

OEM codes composed of 16 characters with ASCII representation (ANSI INCITS 4-1986, R2007) can be transmitted with messages $90 to $97 over the serial message channel. ASCII character OEM codes as well as the use of serial messages $90 - $97 is optional. These 16 characters can, for example, specify OEM part numbers, version numbers or OEM-specific manufacturer codes.

A 6-bit representation of the ASCII characters makes it possible to map two ASCII characters onto one 12-bit data field of an enhanced serial message. The 6-bit representation of ASCII characters permits to encode a subset of printable ASCII characters which comprises uppercase letters, digits and punctuation marks. The subset of 64 ASCII characters from “space” to “_”, corresponds to ASCII values $20. $5F. Sixteen ASCII Characters (N=1.16) are mapped on the data fields of eight serial messages (M=1..8) with IDs $90 .. $97.

The 6-bit representation of ASCII characters is determined as follows:

6bit_code = ASCII_value - $20

12bit_data_field (M) = 6bit_code(Character[2·M]) · $40 + 6bit_code(Character[2·M-1])

The encoding and transmission of 16 ASCII characters over the serial messages $90 to $97 is illustrated by the example in Table D.9-1.

***Table D.9-1 – Transmission of ASCII characters over $90 to $97: Encoding example***

| SENT Message No. (M) | 1 | | 2 | | 3 | | 4 | | 5 | | 6 | | 7 | | 8 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **SENT Serial Message ID** | **$90** | | **$91** | | **$92** | | **$93** | | **$94** | | **$95** | | **$96** | | **$97** | |
| ASCII Character No. (N) | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| **ASCII-Character** | **H** | **E** | **L** | **L** | **O** | **_** | **S** | **E** | **N** | **T** | **_** | **W** | **O** | **R** | **L** | **D** |
| ASCII-Value [dec] | 72 | 69 | 76 | 76 | 79 | 95 | 83 | 69 | 78 | 84 | 95 | 87 | 79 | 82 | 76 | 68 |
| ASCII-Value -32 ( $20) [dec] | 40 | 37 | 44 | 44 | 47 | 63 | 51 | 37 | 46 | 52 | 63 | 55 | 47 | 50 | 44 | 36 |
| Data [bin] [5:3] | 101 | 100 | 101 | 101 | 101 | 111 | 110 | 100 | 101 | 110 | 111 | 110 | 101 | 110 | 101 | 100 |
| Data [bin] [2:0] | 000 | 101 | 100 | 100 | 111 | 111 | 011 | 101 | 110 | 100 | 111 | 111 | 111 | 010 | 100 | 100 |
| Data [dec] | 2408 | | 2860 | | 4079 | | 2419 | | 3374 | | 3583 | | 3247 | | 2348 | |
| **Data [hex]** | **968** | | **B2C** | | **FEF** | | **973** | | **D2E** | | **DFF** | | **CAF** | | **92C** | |
| Data [bin] [11:6] | 100101 | | 101100 | | 111111 | | 100101 | | 110100 | | 110111 | | 110010 | | 100100 | |
| Data [bin] [5:0] | 101000 | | 101100 | | 101111 | | 110011 | | 101110 | | 111111 | | 101111 | | 101100 | |

# APPENDIX E - COMMON SENSOR DEFINITIONS

Definitions which apply to multiple application specific protocols from APPENDIX A are specified in APPENDIX E. The respective application-specific appendices A.x point to definitions in appendices E.x. APPENDIX E specifies SENT default implementations. However, an application-specific appendix A.x can still define other implementations and protocols which differ from the default definitions from APPENDIX E.

## E.1 DATA NIBBLE RESERVED VALUES

### E.1.1 Application Specific Sensor Implementations with Reserved Fast and Supplementary Channel Data Values

Certain application specific sensor implementations of SENT listed in APPENDIX A have reserved fast and supplementary channel data values. Most application specific sensor implementations listed in APPENDIX A use the default reserved data values for $n$-bit data channels ($n$ = channel bit width, e.g., $n$ = 16, 14, 12, 10 or 8). However some application-specific sensor implementations in APPENDIX A specify special application specific reserved data values.

### E.1.2 Reserved Signaling Ranges

Eight of the possible $2^n$ values transmitted in each fast or supplementary channel are used for signaling purposes related to the measurement data of the particular channel.

- Address $0$ Initialization
- Address $(2^n\text{-}7)\ldots\ldots(2^n\text{-}1)$ Error indicators / specific messages

The particular width of the fast or supplementary channel is denoted by $n$. Table E.1.2-1 explains the partitioning of the data range of fast and supplementary channels with examples for specific channel bit widths. The particular data values of the signaling codes are specified in Section E.1.3.

***Table E.1.2-1 – Examples: Measurement data and signaling data regions of fast and supplementary channel data***

| Data Channel Signal | Data Class | 16-bit Values | 14-bit Values | 12-bit Values | 10-bit Values | 8-bit Values |
|---|---|---|---|---|---|---|
| Error indicators / specific messages | Signaling | 65529<br>…<br>65535 | 16377<br>…<br>16383 | 4089<br>…<br>4095 | 1017<br>…<br>1023 | 249<br>…<br>255 |
| Fast or supplementary channel data | Measure-ment Data | 1<br>…<br>65528 | 1<br>…<br>16376 | 1<br>…<br>4088 | 1<br>…<br>1016 | 1<br>…<br>248 |
| Initialization | Signaling | 0 | 0 | 0 | 0 | 0 |

### E.1.3 Data Nibble Reserved Values

Table E.1.3-1 specifies the assignment of the eight reserved data values to error indicators, specific messages and the initialization message.

***Table E.1.3-1 – Error indicators / specific messages / initialization message***

| $n$ -bit Data Value | 12-bit Data Value (Example) | Indicator Message | Definition | Comment |
|---|---|---|---|---|
| $2^n$ -1 | 4095 | IM1 | Production state | Used to indicate production state of sensor (e.g., for manufacturing purposes) |
| $2^n$-2 | 4094 | IM2 | Free to define by OEM/Supplier | |
| $2^n$-3 | 4093 | IM3 | Reserved | |
| $2^n$-4 | 4092 | IM4 | Reserved | |
| $2^n$-5 | 4091 | IM5 | Sensor error indication | Generic error (1) |
| $2^n$-6 | 4090 | IM6 | Sensor functionality and processing error indication | Sensor status: Signal processing and sensor functionality errors (1) |
| $2^n$-7 | 4089 | IM7 | Not a valid measurement value | Invalid data values, not-a-number (or measurements with reduced reliability)(1, 2) |
| 0 | 0 | IM0 | Initialization | The initialization message is transmitted during the sensor initialization phase until valid measurement values are available. |

1 Optionally: Serial message channel status and error indicators (Message ID 0 1) can carry details on the error state indications and specific code messages
2 E.g., applicable for safety-relevant sensors

The use of Data Nibble Reserved Values to report a default value (for error indicators/specific messages/initialization message) instead of the measurement value according to the Table E.1.3-1 should be reviewed with the system integrator in order to avoid violation of specific requirements at system level, such as emission relevant diagnostic(s) or functional safety goal(s).

## E.2 ENCODING SENSOR CHARACTERISTIC TRANSFER FUNCTIONS

### E.2.1 Generic Linear Transfer Characteristic Functions

#### E.2.1.1 Calculation of Generic Linear Transfer Characteristic Functions

Measurement data values can be mapped onto $n$-bit SENT data by means of a linear transfer characteristic function ($n$ = channel bit width). Slope and offset of such a transfer characteristic function are determined by the coordinates of the node values $X_1$, $Y_1$ and $X_2$, $Y_2$.

Sensor measurement values $SMV_{Phy}$ (physical unit or number) shall be converted into $n$-bit SENT measurement data $SMD_{val}$ with linear transfer characteristic functions, which are specified by the formula

$$SMD_{Val} = \underbrace{Y_1}_{Offset\ Y} + \underbrace{\frac{Y_2 - Y_1}{X_2 - X_1}}_{Slope} \cdot \left( SMV_{Phy}[physical\ unit\ or\ number] - \underbrace{X_1}_{Offset\ X} \right)$$

for $SMD_{val} \geq 1$ and $SMD_{val} \leq 2^n$-8.

Correspondingly, the conversion of $n$-bit SENT sensor measurement data $SMD_{val}$ into physical sensor measurement values $SMV_{Phy}$ (physical unit or number) shall be made with the following formula:

$$SMV_{Phy}\,[physical\ unit\ or\ number] = \underbrace{X_1}_{Offset\ X} + \underbrace{\frac{X_2 - X_1}{Y_2 - Y_1}}_{Slope} \cdot \left( SMD_{Val} - \underbrace{Y_1}_{Offset\ Y} \right)$$

for $SMD_{val} > 1$ and $SMD_{val} < 2^n\text{-}8$.

The $n$-bit SENT measurement data $SMD_{val}$ are $n$-bit unsigned fixed-point integer numbers in the range $1\ ..\ 2^n\text{-}8$. The node values $X_1$ and $X_2$ either have a physical unit or are a dimensionless number corresponding to the measurement data values $SMV_{Phy}$. *Values* $Y_1$ and $Y_2$ are dimensionless numbers.

The example in Figure E.2.1.1-1 illustrates a linear transfer characteristic function which is determined by the coordinates of the nodes $X_1$, $Y_1$ and $X_2$, $Y_2$ for channel bit width $n = 12$.

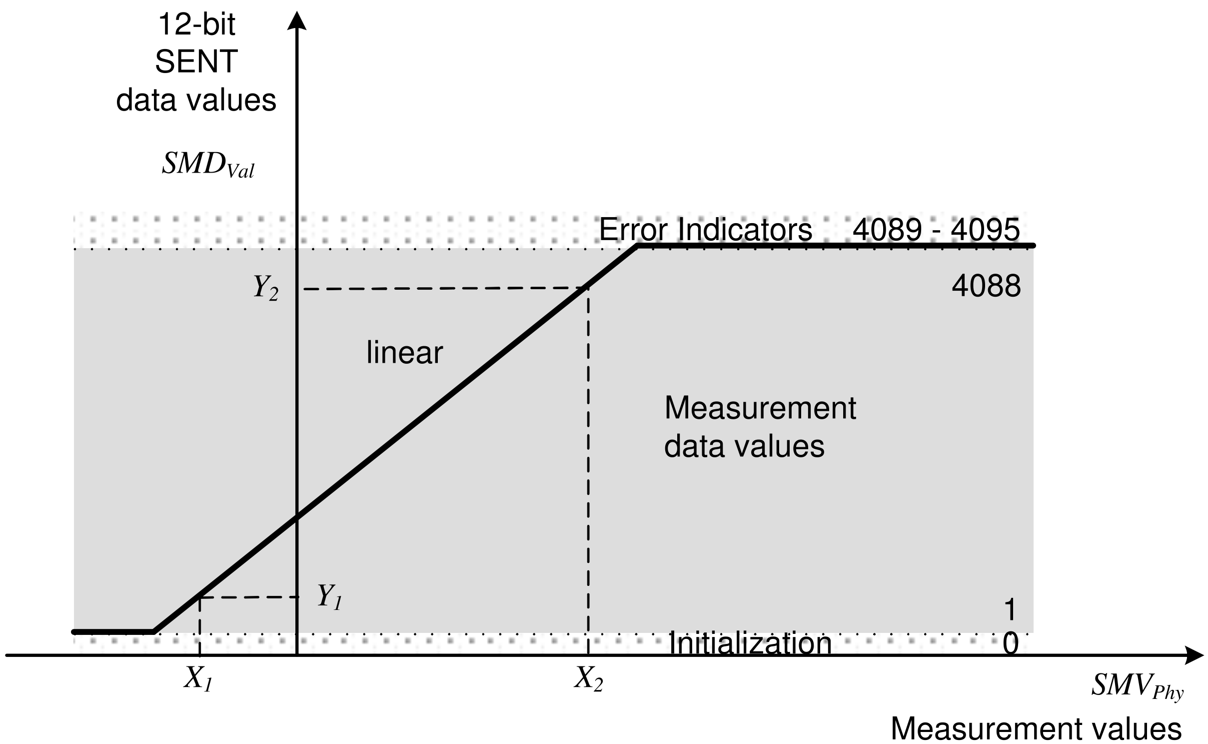



***Figure E.2.1.1-1 – Nominal characteristic function of a sensor (with 12-bit data values)***

E.2.1.2 Measurement Value Data Range

Digital measurement data values are transmitted over SENT data channels with bit width $n$. Eight of the $2^n$ data values are assigned for signaling purposes, as specified in Section E.1.

The $n$-bit measurement data range is determined by

- The high clamp $(2^n\text{-}8)$ and low clamp $(1)$ values, which are the highest and the lowest digital SENT values $SMD_{val}$ , which can be transmitted.
- Physical measurement data beyond these clamp values are mapped onto the respective clamp values.
- The values $Y_1$ and $Y_2$ have a dual function. First, they define the limits of the core measurement data range. Second, they specify, in combination with the physical values $X_1$ and $X_2$, the nodes which determine the linear transfer characteristic functions (see Section E.2.1).
- Data values from $Y_2 + 1 \ldots.. 2^n\text{-}9$ are referred to as the upper data range.
- Data values from $2 \ldots (Y_1 - 1)$ are referred to as the lower data range.
- The use of the core, the upper and the lower data range is specified in the sections that define the particular transfer function type.

Table E.2.1.2-1 explains the partitioning of the measurement data range into a core measurement data range (limited by $Y_1$ and $Y_2$) as well as an upper measurement data range and a lower measurement data range.

***Table E.2.1.2-1 – Partitioning of the signal data space with default $Y_1$ and $Y_2$***

| Data Channel Signal | Data Class | $n$-bit Values | |
|---|---|---|---|
| | | Data Value | Number of Values |
| Error Indicators | Signaling | $2^n\text{-}7 \ldots 2^n$ | 7 |
| High Clamp | Measurement data ($\log_2(2^n\text{-}8)$ bit digital resolution) | $2^n\text{-}8$ | 1 |
| Upper range | | $Y_2 + 1 \ldots.. 2^n\text{-}9$ | $2^n\text{-}9 - Y_2$ |
| Core range | | $Y_1 \ldots Y_2$ | $Y_2 - Y_1 + 1$ |
| Lower Range | | $2 \ldots (Y_1 - 1)$ | $Y_1 - 2$ |
| Low Clamp | | 1 | 1 |
| Initialization | Signaling | 0 | 1 |

Figure E.2.1.2-1 illustrates the partitioning of the measurement data range. See section E.1 for definitions of the reserved signaling ranges. Note, the terms *upper* and *lower* data range refer to the $n$-bit digital values. The physical measurement values $SMV_{Phy}$ that are mapped to the upper or lower range of $SMD_{val}$ do not need to have the larger or smaller magnitude (e.g.,, in case of transfer functions with negative slope).

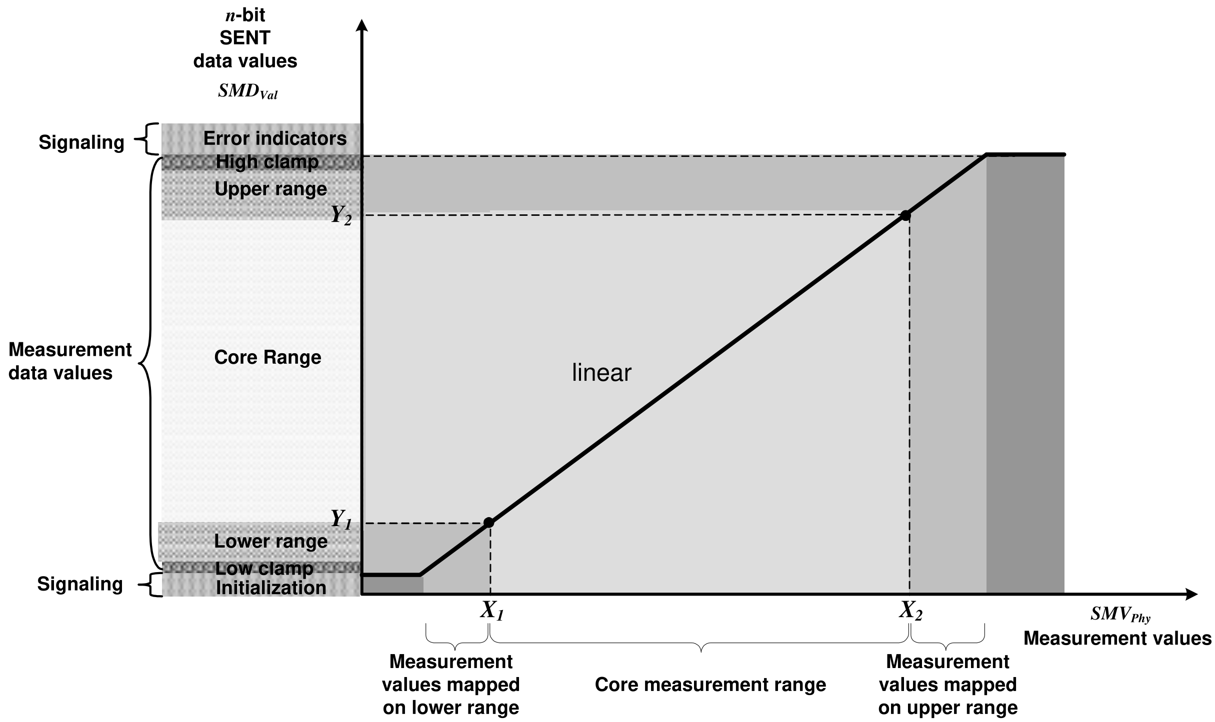



***Figure E.2.1.2-1 – Partitioning of measurement data ranges and mapping onto SENT data values (for illustration)***

### E.2.1.3 Node Index Values $X_1$, $X_2$ for Physical Measurement Data

The node index values $X_1$, $X_2$ for physical measurement data shall be encoded with 12-bit data values $X_{D,i}$ composed of an $N_m$-bit mantissa $X_{m,i}$ and an $N_e$-bit exponent $X_{e,i}$ ($N_m + N_e = 12$, $X_{D,i} = [X_{m,i}(N_{m-1}:0)\ X_{e,i}(N_{e-1}:0)]$). The representation of the physical measurement data node indices $X_1$, $X_2$ is determined by the formula

$$X_i = X_{m,i} \cdot 10^{(Xe,i+Xe\,offset)}$$

.

The node index values $X_1$ and $X_2$ have the physical unit of the respective measurement data. The bit width $N_m$ of the mantissa and $N_e$ of the exponent and their binary representation (unsigned or signed two's complement) as well as the offset value $X_{e\ offset}$ is specified for each transfer function type in the following subsections of E.2.

### E.2.1.4 Default Values for $Y_1$ and $Y_2$

The SENT $n$-bit digital node indices $Y_1$ and $Y_2$ can either be specified as

- Default values, or
- Sensor-specific values transmitted over the serial message channel.

Default values for $Y_1$ and $Y_2$ are determined by the following rule:

$$Y_1 = ceil(0.047 \cdot (2^n - 8))$$

$$Y_2 = 2^n - 7 - Y_1$$

Table E.2.1.4-1 shows the default $Y_1$ and $Y_2$ values for channel bit widths $n$ = 6…16 calculated from the formula above for illustration.

***Table E.2.1.4-1 – Default $Y_1$ and $Y_2$ values***

| Bit width $n$ | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Default $Y_1$ | 3 | 6 | 12 | 24 | 48 | 96 | 193 | 385 | 770 | 1540 | 3080 |
| Default $Y_2$ | 54 | 115 | 237 | 481 | 969 | 1945 | 3896 | 7800 | 15607 | 31221 | 62449 |

### E.2.1.5 Transmission of Generic Linear Transfer Characteristic Functions with Enhanced Serial Data Messages

Node coordinate values $X_1$, $Y_1$ and $X_2$, $Y_2$ which define linear transfer function characteristics of sensors shall be communicated from the transmitter (sensor) to the receiver (ECU) using the 12-bit data fields of Enhanced Serial Data Messages. For mapping onto the 12-bit data field D(11:0), node coordinate values are represented in a 12-bit format as $X_{D,i}$ and $Y_{D,i}$. Section D.1 specifies the particular serial messages which carry the 12-bit values $X_{D,1}$, $X_{D,2}$ and (optionally) $Y_{D,1}$, $Y_{D,2}$ of the transfer characteristic nodes. The SENT $n$-bit node indices $Y_1$ and $Y_2$ shall be within the data range of $n$-bit SENT measurement data ($SMD_{val}$ ) which is defined by low clamp $(=1) \leq SMD_{val} \leq$ high clamp $(= 2^n\text{-}8)$.

Some sensor types specify default $Y_1$, $Y_2$ values. If $X_1$, $X_2$ are transmitted, but no $Y_1$, $Y_2$ values are transmitted over the serial message channel, the a priori known default $Y_1$, $Y_2$ values for the particular sensor type are used by the receiver to determine the transfer characteristic function. Default $Y_1$, $Y_2$ values are specified in E.2.1.4.

If the values $Y_1$ and $Y_2$ of the transfer characteristic nodes are transmitted in the 12-bit data fields of the Enhanced Serial Data Messages, sensor-specific partitioning of the data space (other than the default $Y_1$, $Y_2$) can be specified (it is still possible to transmit the default $Y_1$, $Y_2$ in the respective Enhanced Serial messages).

If the bit width $n$ of the data channel is $12$, the 12-bit values $Y_i$ correspond to $Y_{D,i}$ the data format for transmission over the Enhanced Serial Messages. If the bit width $n$ of the data channel differs from the 12-bit data field of the Enhanced Serial Data Message, the following rules for mapping $Y_i$ onto $Y_{D,i}$ apply: If the bit width $n$ of the data channel is smaller than 12 bit, $Y_1[n\text{-}1:0]$ and $Y_2[n\text{-}1:0]$ are mapped onto the LSBs of the 12-bit data fields $Y_{D,i}[11:0]$ of the Enhanced Serial Messages (e.g., $n=10$, $Y_{D,i}[11]=0$, $Y_{D,i}[10]=0$, $Y_{D,i}[9:0] = Y_1[9:0]$). If the bit width $n$ of the data channel is larger than 12 bit, the LSBs of $Y_1[n\text{-}1:0]$ and $Y_2[n\text{-}1:0]$ are assumed to be zero and their MSBs are mapped onto $Y_{D,i}[11:0]$ – MSB aligned (e.g., $n=16$, $Y_{D,i}[11:0] = Y_i[15:4]$, $Y_i[3:0] = 0$ or $n=14$, $Y_{D,i}[11:0] = Y_i[13:2]$, $Y_i[1:0] = 0$).

The mapping of the encoded pressure index values $X_{D,i}$ with the $N_m$ bits of the mantissa and the $N_e$ bits of the exponent onto the 12 bit data field of an enhanced serial message is explained in Table E.2.1.5-1.

***Table E.2.1.5-1 – Transmission of generic linear transfer characteristic node values $X_i$ with 12-bit $X_{D,i}$***

| **Data bit** | $D_{11}$ | $D_{10}$ | …. | $D_{Ne+1}$ | $D_{Ne}$ | $D_{Ne-1}$ | … | $D_0$ |
|---|---|---|---|---|---|---|---|---|
| **Encoded value $X_{D,i}$** | $X_{m,i,Nm-1}$ | $X_{m,i,Nm-2}$ | …. | $X_{m,i,1}$ | $X_{m,i,0}$ | $X_{e,i,Ne-1}$ | … | $X_{e,i,0}$ |
| $X_i = X_{m,i} \cdot 10^{(Xe,i+Xe\,offset)}$ | Mantissa | | | | | Exponent | | |

### E.2.2 Linear Temperature Transfer Characteristic Functions

Temperature signals, which are transmitted as $n$-bit data values over SENT fast channels or supplementary data channels, are encoded using either

- the default linear temperature transfer characteristic function, or
- the linear high-temperature transfer characteristic function, or
- a special linear (non-default) temperature transfer characteristic function.

The application of these transfer characteristic functions is explained in Table E.2.2-1. If not specified otherwise for a sensor type, (and no transfer characteristic node values are transmitted), the default transfer characteristic function is assumed and used for decoding the $n$-bit SENT measurement data $SMD_{val}$ at the receiver side.

*Table E.2.2-1 – Overview of temperature transfer characteristic functions*

| Temperature Transfer Characteristic Functions | Appendix | Formula for $n$-bit SENT measurement data $SMD_{val}$ |
|---|---|---|
| Default Linear Temperature Transfer Characteristic | E.2.2.1 | $SMD_{Val} = round\lfloor S_{TF} \cdot (T[K] - T_{Offset}) \rfloor$ |
| Linear High Temperature Transfer Characteristic | E.2.2.2 | $SMD_{Val}$: $n$-bit unsigned fixed-point integer numbers in the range $1$ .. $2^n$-8<br>$S_{TF}$: Transfer function slope [LSB/K]<br>$T_{Offset}$: Transfer function offset [K]<br>$S_{TF}$ and $T_{Offset}$ are specified in E.2.2.1 and E.2.2.2 |
| Special Linear Temperature Transfer Characteristic determined by node values $X_1[K]$, $Y_1[n\text{-}bit]$ and $X_2[K]$, $Y_2\ [n\text{-}bit]$ | E.2.2.3 | Specified in E.2.1 |

E.2.2.1 Default Linear Temperature Transfer Characteristic Function

The default linear temperature characteristic function can be applied to most sensors.

***Table E.2.2.1-1 – Transfer function parameters for default linear temperature data channels***

| | Default linear temperature transfer characteristic | | |
|---|---|---|---|
| **Channel bit width $n$ [bit]** | **12** | **10** | **8** |
| **Transfer function slope $S_{TF}$ [LSB/K]** | 8 | 4 | 1 |
| **Transfer function offset $T_{Offset}$ [K]** | 200 | 220 | 220 |
| **T(min)** | 200.125 K (-73.025°C) | 220.25 K (-52.90°C) | 221 K (-52.15°C) |
| **T(max)** | 711K (437.85°C) | 474 K (200.85°C) | 468 K (194.85°C) |

The digital resolution is $1/S_{TF}$ [K/LSB].

Figure E.2.2.1-1 illustrates the default linear temperature transfer characteristic function for $n$=12.

**Default:**

$SMD_{val} = round[8 \cdot (T[K] - 200\ K)/[K]]$

$SMD_{val}$: 12-bit SENT data values
$T$: Temperature range 200.125 K .. 711 K or: -73.025°C .. 437.85°C

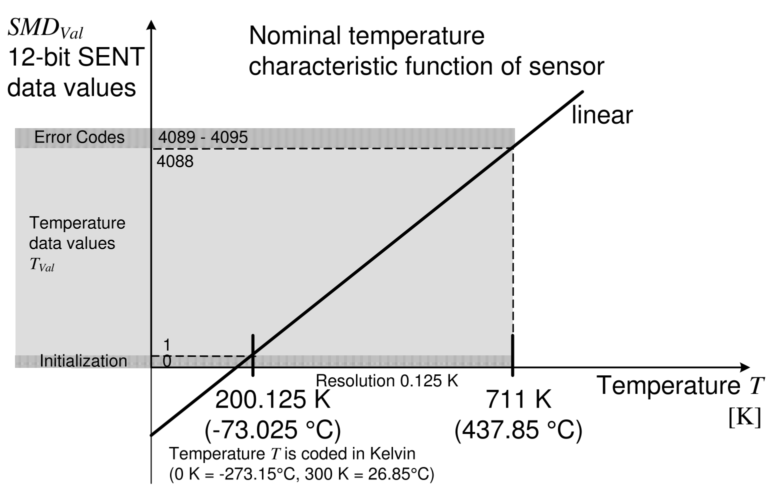



***Figure E.2.2.1-1 Default temperature characteristic function (for Illustration)***

NOTE: Sensors which use the default temperature characteristic function do not need to cover the entire temperature range of the default characteristic function.

E.2.2.2 Linear High Temperature Transfer Characteristic Functions

The linear high temperature characteristic function can be applied to special sensors for temperature ranges up to more than 1200°C.

TABLE E.2.2.2-1 – TRANSFER FUNCTION PARAMETERS FOR DEFAULT LINEAR HIGH TEMPERATURE DATA CHANNELS

| | **Linear high temperature transfer characteristic** | |
|---|---|---|
| **Channel bit width $n$ [bit]** | **12** | |
| **Transfer function slope $S_{TF}$ [LSB/K]** | 3 | |
| **Transfer function offset $T_{Offset}$ [K]** | 200 | |
| **T(min)** | 200.33 K | (-72.82°C) |
| **T(max)** | 1562.67 K | (1289.52°C) |

Figure E.2.2.2-1 illustrates the linear high temperature transfer characteristic function for $n$=12.

**High temperature:**

$SMD_{val} = round[3 \cdot (T[K] - 200\ K)/[K]]$

$SMD_{val}$: 12-bit SENT data values
$T$: Temperature range 200.33 K .. 1562.67 K or: -72.82°C .. 1289.52°C

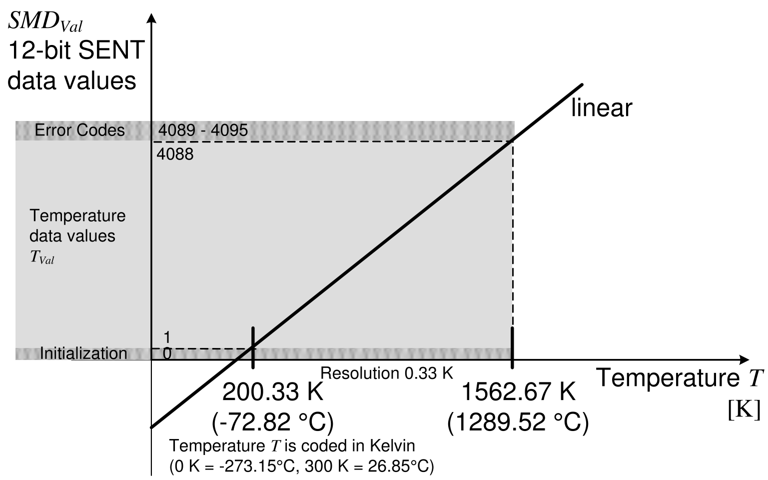



***Figure E.2.2.2-1 High temperature characteristic function (for Illustration)***

E.2.2.3 Special Linear Temperature Transfer Characteristic Functions

Some sensor types have non-default linear characteristics, for example to represent a smaller temperature range with finer resolution. Special linear temperature measurement data values $SMV_{Phy}$ can be mapped onto $n$-bit SENT data values $SMD_{val}$ by means of the generic linear transfer characteristic function as specified in E.2.1 ($n$ = channel bit width). Slope and offset of such a transfer characteristic function is determined by the coordinates of the node values $X_1$ [K], $Y_1$ [$n$ -bit] and $X_2$ [K], $Y_2$ [$n$-bit]. The physical sensor measurement values ($SMV_{Phy}$ in Kelvin) shall be converted into $n$-bit SENT measurement data ($SMD_{val}$) with the formulas, as specified in E.2.1. Parameters that determine the transfer function of Section E.2.1.1 and the node index values of Section E.2.1.3 are defined in Table E.2.2.3-1.

***Table E.2.2.3-1 – Transfer function parameters for special linear temperature data channels***

| | Parameter | Definition | Comment |
|---|---|---|---|
| **Physical unit** | $SMV_{Phy}$ | Kelvin [K] | For special temperature data channels |
| **Measurement value data range** | Upper range | Measurement data with reduced accuracy. | |
| | Core Range | Range of temperature data with guaranteed tolerances | |
| | Lower Range | Measurement data with reduced accuracy. | |
| **Node value mantissa** | $N_m$ | 11 | Bit width of $X_{m,i}$ |
| | Binary representation of the mantissa $X_{m,i}$ | unsigned | |
| | Set of allowed mantissa values $X_{m,i}$ | 0 … 2047 | |
| **Node value exponent** | $N_e$ | 1 | Bit width of $X_{e,i}$ |
| | Binary representation of the exponent $X_{e,i}$ | Signed, two's complement | |
| | Set of allowed exponent values $X_{e,i}$ | -1 … 0 | |
| | $X_{e\ offset}$ | 0 | |

Example 1: The 12-bit linear high-temperature characteristic function could be represented with $X_1$ = 233 K, $Y_1$ = 99, and $X_2$ =1560 K, $Y_2$ = 4080.

Example 2: The 12-bit default temperature characteristic function with a temperature range with guaranteed tolerances between 233 K (-40.15 °C) and 414 K (140.85 °C) could be represented with $X_1$ = 233 K, $Y_1$ = 264, and $X2$ =414 K, $Y_2$ = 1712.

E.2.2.4 Internal Reference Temperature

Some sensor elements or ICs have devices that derive a reference temperature. This temperature information can be transmitted over a Fast Channel or a supplementary data channel. The default temperature characteristic function (with $T_{Val} = 8\cdot(T[K]-200\ K)/\ [K]$ ) according to E.2.2.1 shall be used if no channel characteristics ($X_1,Y_1$ and $X_2,Y_2$) are specified.

E.2.3 Encoding Ratio Sensing Transfer Functions

This section gives the requirements for encoding ratio measurements which are scaled 0 to 100 %. The transmitted data can be, for example the ratio of components of a medium or absolute humidity.

Ratio is transmitted in a range from 0% to 100% using a 12 bit data format. The digital resolution (LSB value) of this format is 0.03125 % (1/32 %). Ratio values $R$ from -13.84% to 113.875 % shall be mapped on the SENT data values 1 to 4088. This allows for a 13% over/undershoot of the signal (10% of data range) and for the representation of integer percentage values and fractions (e.g., 1% 0.5%, 50%). 12-bit reserved data values according to appendix E.1.3 are used.

The ratio values $R_{val}$ that are transmitted as 12-bit SENT data shall be calculated with the formula:

$$R_{Val} = round[32.0 \cdot R[\%]/1\% + 444]$$

$R_{val}$ are 12-bit unsigned numbers (data range 1 to 4088; -13.84% to 113.875 %), see Figure E.2.3-1.

$R_{val}$ = round[32.0 · R [%] / 1% +444 ]

$R_{val}$: 12-bit unsigned number
(data range 1 .. 4088)

$R[\%]$: Ratio range
(-13.84 % … 113.875 %)

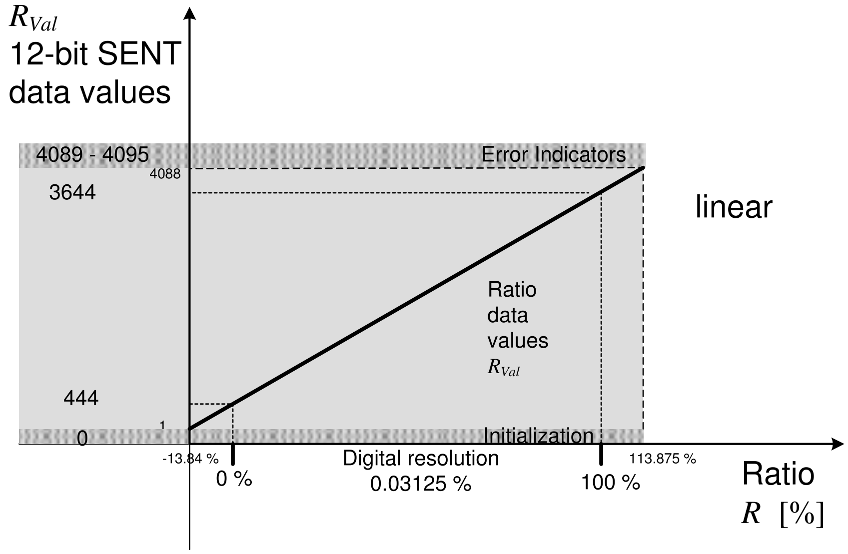



***Figure E.2.3-1 – Ratio transfer function***

NOTE: Ratio values can have a corresponding temperature of the Ratio measurement element. The temperature of the ratio sensor element is transmitted over a dedicated temperature channel, as specified for the particular sensor type.

E.2.4 Encoding of Pressure Sensor Transfer Functions

Physical pressure measurement data values $SMV_{Phy}$ can be mapped onto $n$-bit SENT data values $SMD_{val}$ by means of a linear transfer characteristic function ($n$ = channel bit width). Slope and offset of such a transfer characteristic function is determined by the coordinates of the node values $X_1$ [Pa], $Y_1$ [$n$-bit] and $X_2$ [Pa], $Y_2$ [$n$-bit]. The physical sensor measurement values ($SMV_{Phy}$ in Pascal) shall be converted into $n$-bit SENT measurement data ($SMD_{val}$) with the formulas, as specified in E.2.1.1. Parameters that determine the transfer function and the node index values of Section E.2.1.3 are defined in Table E.2.4-1.

***Table E.2.4-1 – Transfer function parameters for pressure data channels***

| | Parameter | Definition | Comment |
|---|---|---|---|
| **Physical unit** | $SMV_{Phy}$ | Pascal [Pa] | For pressure data channels |
| **Measurement value data range** | Upper range | Measurement data with reduced accuracy. | |
| | Core Range | Range of pressure data with guaranteed tolerances | Normal operation range |
| | Lower Range | Measurement data with reduced accuracy. | |
| **Node value mantissa** | $N_m$ | 9 | Bit width of $X_{m,i}$ |
| | Binary representation of the mantissa $X_{m,i}$ | Signed, two's complement | |
| | Set of allowed mantissa values $X_{m,i}$ | -256 … 255 | |
| **Node value exponent** | $N_e$ | 3 | Bit width of $X_{e,i}$ |
| | Binary representation of the exponent $X_{e,i}$ | Unsigned | |
| | Set of allowed exponent values $X_{e,i}$ | 0 … 7 | |
| | $X_{e\ offset}$ | 0 | |

The $n$-bit pressure data range is partitioned as specified in E.2.1.2. The use of the core, the upper and the lower data range is specified in Table E.2.4-1. Node values $Y_1$, $Y_2$ can either be default values, as specified in E.2.1.4, or sensor-specific values $Y_1$, $Y_2$.

Ranges for $X_1$, $X_2$ for all possible exponents $X_{e,i}$ are given in Table E.2.4-2.

***Table E.2.4-2 – Illustration of possible values for $X_1$ and $X_2$ for pressure data channels***

| **Pressure Range** for $X_1$ **and** $X_2$ | **Step-width of Limiting Pressure Values** $X_1$ **and** $X_2$ | **Exponent** $X_{e,i}$ |
|---|---|---|
| -2.560e+002 …. 2.550e+002 [Pa]<br>-2.560e-003 …. 2.550e-003 [bar] | 1.0e+000 [Pa]<br>1.0e-005 [bar] | 0 |
| -2.560e+003 …. 2.550e+003 [Pa]<br>-2.560e-002 …. 2.550e-002 [bar] | 1.0e+001 [Pa]<br>1.0e-004 [bar] | 1 |
| -2.560e+004 …. 2.550e+004 [Pa]<br>-2.560e-001 …. 2.550e-001 [bar] | 1.0e+002 [Pa]<br>1.0e-003 [bar] | 2 |
| -2.560e+005 …. 2.550e+005 [Pa]<br>-2.560e+000 …. 2.550e+000 [bar] | 1.0e+003 [Pa]<br>1.0e-002 [bar] | 3 |
| -2.560e+006 …. 2.550e+006 [Pa]<br>-2.560e+001 …. 2.550e+001 [bar] | 1.0e+004 [Pa]<br>1.0e-001 [bar] | 4 |
| -2.560e+007 …. 2.550e+007 [Pa]<br>-2.560e+002 …. 2.550e+002 [bar] | 1.0e+005 [Pa]<br>1.0e+000 [bar] | 5 |
| -2.560e+008 …. 2.550e+008 [Pa]<br>-2.560e+003 …. 2.550e+003 [bar] | 1.0e+006 [Pa]<br>1.0e+001 [bar] | 6 |
| -2.560e+009 …. 2.550e+009 [Pa]<br>-2.560e+004 …. 2.550e+004 [bar] | 1.0e+007 [Pa]<br>1.0e+002 [bar] | 7 |

Table E.2.4-3 explains the mapping of the pressure node value mantissa and exponent onto the 12-bit encoded pressure value $X_{D,i}$ for transmission over the Serial Message channel, using the Enhanced Serial Message Format (for information).

***Table E.2.4-3 – Transmission of pressure transfer characteristic node values $X_i$ with 12-bit $X_{D,i}$***

| **Data bit** | $D_{11}$ | $D_{10}$ | $D_9$ | $D_8$ | $D_7$ | $D_6$ | $D_5$ | $D_4$ | $D_3$ | $D_2$ | $D_1$ | $D_0$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Encoded pressure value** $X_{D,i}$ | $X_{m,i,8}$ | $X_{m,i,7}$ | $X_{m,i,6}$ | $X_{m,i,5}$ | $X_{m,i,4}$ | $X_{m,i3}$ | $X_{m,i,2}$ | $X_{m,i,1}$ | $X_{m,i,0}$ | $X_{e,i,2}$ | $X_{e,i,1}$ | $X_{e,i,0}$ |
| $X_i = X_{m,i} \cdot 10^{(Xe,i + Xe\,offset)}$ | Mantissa | | | | | | | | | Exponent | | |

E.2.4.1 Pressure Application Examples

Examples of encoding for pressure sensors and temperature sensors are illustrated in the next sections. In both cases the Sensor Type (8 bit ID see D.1 and 12-bit Sensor type see D.4) defined in APPENDIX D is transmitted in the serial message channel.

E.2.4.1.1 High Pressure Common Rail Sensor

$X_1$, $X_2$ transmitted in serial message channel:
$X_{m,1}$=0, $X_{e,1}$=0
$X_{m,2}$=28, $X_{e,2}$=7
Defaults for $Y_1$, $Y_2$ (these are not transmitted): $Y_1$=193, $Y_2$=3896
Figure E.2.4.1.1-1 shows the scaling and diagnostic regions for these sensors.

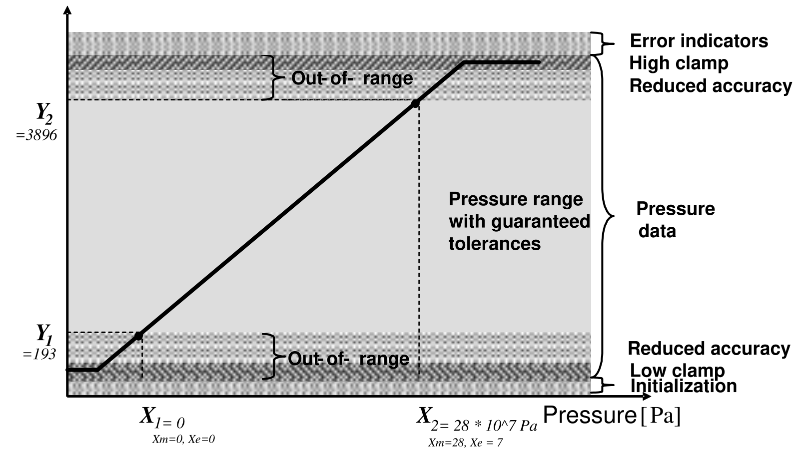



***Figure E.2.4.1.1-1 – Application with default $Y_1$, $Y_2$***

E.2.4.1.2 Differential Pressure Sensor

Tank pressure sensor -2.5 kPa … 2.5 kPa
Transfer characteristic specified by $X_1$, $X_2$, $Y_1$ and $Y_2$

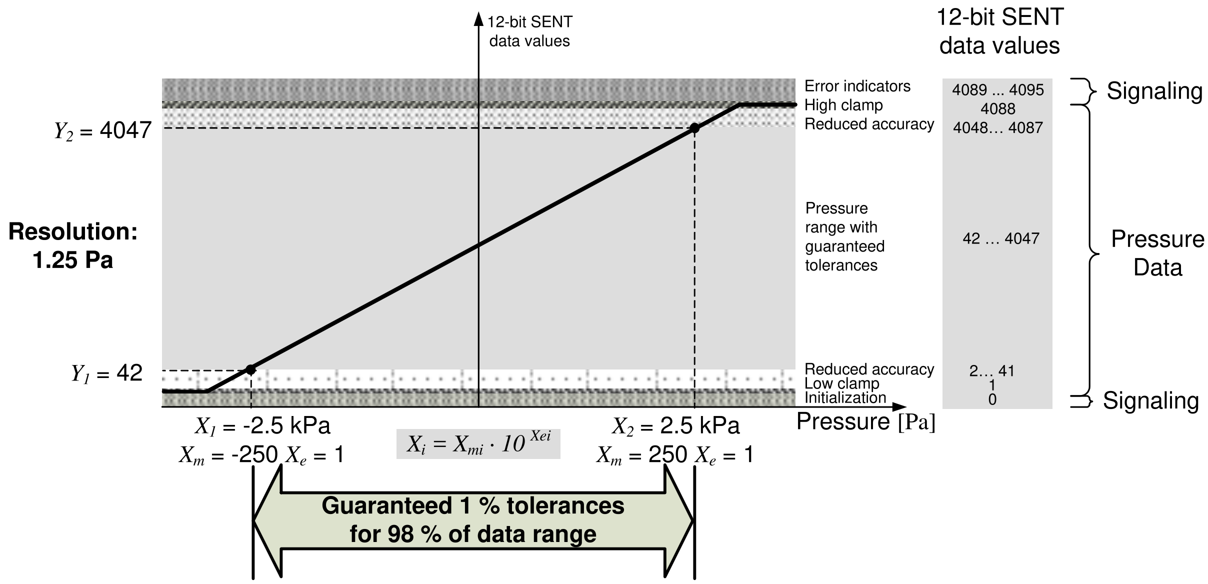



***Figure E.2.4.1.2-1 – Application with sensor-specific $Y_1$, $Y_2$***

Node-values $X_1$, $Y_1$ and $X_1$, $Y_2$ can be transmitted in the serial message channel. Figure E.2.4.1.2-1 shows the scaling and diagnostic regions for these sensors.

E.2.5 Encoding of Position Sensor Transfer Functions

This section specifies the mapping of position data to $n$-bit data values (numbers) for linear position, angle, and relative position or angle data channels.

The linear transfer function and the node values $X_1$, $X_2$ and $Y_1$, $Y_2$ are specified in sections E.2.1.1 toE.2.1.5. Application specific parameters are defined in Table E.2.5-1. Possible values for $X_1$ and $X_2$ and reserved exponents are illustrated in Table E.2.5-2, Table E.2.5-3, and Table E.2.5-4.

***Table E.2.5-1 – Transfer function parameters for position data channels***

| | Parameter | Definition | Comment |
|---|---|---|---|
| **Physical unit** | $SMV_{Phy}$ | Meter [m] | For linear position data channels. |
| | | Degree [°] | For angle position data channels. |
| | | Percent [%] | For relative position and relative angle data channels. |
| **Measurement value data range** | Upper range | Overshoot margin for inaccuracies | |
| | Core Range | Normal operation range for typical system parameters | |
| | Lower Range | Undershoot margin for inaccuracies | |
| **Node value mantissa** | $N_m$ | 9 | Bit width of $X_{m,i}$. |
| | Binary representation of the mantissa $X_{m,i}$ | Signed two's complement | |
| | Set of allowed mantissa values $X_{m,i}$ | -256 to 255 | For linear position and angle position data channels. |
| | | 0, -100 | For $X_{m,1}$ of relative position and relative angle data channels. |
| | | 100 | For $X_{m,2}$ of relative position and relative angle data channels. |
| **Node value exponent** | $N_e$ | 3 | Bit width of $X_{e,i}$. |
| | Binary representation of the exponent $X_{e,i}$ | Unsigned | |
| | Set of allowed exponent values $X_{e,i}$ | 0 to 7 | For linear position data channels. |
| | | 1 to 5 | For angle position data channels. The numbers 0, 6, 7 are reserved. |
| | | 3 | For relative position and relative angle data channels. All other numbers are reserved. |
| | $X_{e\ offset}$ | -6 | Exponent offset for linear position data channels. |
| | | -3 | Exponent offset for angle position data channels. |
| | | | Exponent offset for relative position and relative angle data channels. |

***Table E.2.5-2 – Illustration of possible values for $X_1$ and $X_2$ for linear position data channels***

| **Linear Position Range for $X_1$ and $X_2$** | **Step-width of Limiting Position values $X_1$ and $X_2$** | **Exponent $X_{e,i}$** |
|---|---|---|
| -2.560e-004 …. 2.550e-004 [m] | 1.0e-006 [m] | 0 |
| -2.560e-003 …. 2.550e-003 [m] | 1.0e-005 [m] | 1 |
| -2.560e-002 …. 2.550e-002 [m] | 1.0e-004 [m] | 2 |
| -2.560e-001 …. 2.550e-001 [m] | 1.0e-003 [m] | 3 |
| -2.560e+000 …. 2.550e+000 [m] | 1.0e-002 [m] | 4 |
| -2.560e+001 …. 2.550e+001 [m] | 1.0e-001 [m] | 5 |
| -2.560e+002 …. 2.550e+002 [m] | 1.0e+000 [m] | 6 |
| -2.560e+003 …. 2.550e+003 [m] | 1.0e+001 [m] | 7 |

***Table E.2.5-3 – Illustration of possible values for $X_1$ and $X_2$ for angle data channels***

| Absolute Angle Range for $X_1$ and $X_2$ | Step-width of Limiting Position values $X_1$ and $X_2$ | Exponent $X_{e,i}$ |
|---|---|---|
| Reserved | | 0 |
| -2.560e+000 …. 2.550e+000 [degree] | 1.0e-002 [°] | 1 |
| -2.560e+001 …. 2.550e+001 [degree] | 1.0e-001 [°] | 2 |
| -2.560e+002 …. 2.550e+002 [degree] | 1.0e+000 [°] | 3 |
| -2.560e+003 …. 2.550e+003 [degree] | 1.0e+001 [°] | 4 |
| -2.560e+004 …. 2.550e+004 [degree] | 1.0e+002 [°] | 5 |
| Reserved | | 6 |
| Reserved | | 7 |

***Table E.2.5-4 – Illustration of possible values for $X_1$ and $X_2$ for relative position and relative angle data channels***

| Range for $X_1$ and $X_2$ | Step-width of Limiting Position values $X_1$ and $X_2$ | Exponent $X_{e,i}$ |
|---|---|---|
| Reserved | | 0 |
| Reserved | | 1 |
| Reserved | | 2 |
| -100 [%] and 0 for $X_1$<br>+100 [%] for $X_2$ | 1.0e+000 [%] | 3 |
| Reserved | | 4 |
| Reserved | | 5 |
| Reserved | | 6 |
| Reserved | | 7 |

E.2.6 Encoding of Linear MAF Transfer Functions

Linear MAF measurement data values $SMV_{Phy}$ can be mapped onto $n$-bit SENT data values $SMD_{val}$ by means of the generic linear transfer characteristic function as defined in Section E.2.1 ($n$ = channel bit width). Slope and offset of such a transfer characteristic function is determined by the coordinates of the node values $X_1$ [kg/h], $Y_1$ [$n$-bit] and $X_2$ [kg/h], $Y_2$ [$n$-bit]. The physical sensor measurement values ($SMV_{Phy}$ in kg/h) shall be converted into $n$-bit SENT measurement data ($SMD_{val}$) with the formulas, as specified in E.2.1. Parameters, that determine the transfer function of Section E.2.1.1 and the node index values of Section E.2.1.3 are defined in Table E.2.6-1.

***Table E.2.6-1 – Transfer function parameters for linear MAF data channel***

| | Parameter | Definition | Comment |
|---|---|---|---|
| **Physical unit** | $SMV_{Phy}$ | [kg/h] | |
| **Measurement value data range** | Upper range | Measurement data with reduced accuracy. | |
| | Core Range | Range of MAF data with guaranteed tolerances | |
| | Lower Range | Measurement data with reduced accuracy. | |
| **Node value mantissa** | $N_m$ | 10 | Bit width of $X_{m,i}$ |
| | Binary representation of the mantissa $X_{m,i}$ | signed | |
| | Set of allowed mantissa values $X_{m,i}$ | -512…511 | |
| **Node value exponent** | $N_e$ | 2 | Bit width of $X_{e,i}$ |
| | Binary representation of the exponent $X_{e,i}$ | unsigned | |
| | Set of allowed exponent values $X_{e,i}$ | 0…3 | |
| | $X_{e\ offset}$ | 0 | |

NOTE: The mass air flow transfer characteristic depends as well on the MAF sensor itself as on the particular mounting position (engine-specific air duct design). Therefore, mass air flow values $X_1$, $X_2$ of a sensor have to be calibrated for the engine-specific air duct design.

## E.3 ERROR MESSAGES AND SIGNALS

### E.3.1 SENT Status and Error Messages

Sensors can transmit error indicators, specific messages and the initialization message information to the ECU-side. The occurrence of an error shall be indicated by the error flags for fast channel 1 and fast channel 2 (bit 0 and bit 1 of the status and communication nibble), as specified in APPENDIX H.

In addition to the Status and Communication nibble error flags, fast-channel indicator messages can also be transmitted instead of measurement data using the data nibble reserved values, as specified in E.1.3. Transmission of error and status indicator messages over the optional serial message channel (as specified in D.5) can be made in parallel to the transmission of measurement data over the fast data channels.

### E.3.2 Setting of Error Flags and Transmission of Reserved Data Values

Table E.3.2-1 specifies the setting of the error and status messages and signals for the different frame formats (H.1…H.7) in case of distinct error types. Bit 0 and bit 1 of the status and communication nibble are used as error flags, as specified in APPENDIX H. Indicator messages (IM0. IM7) for error indicators, specific messages and the initialization message are used as specified in E.1.3.

***Table E.3.2-1 – Setting of error messages and signals***

| | | | measurement data – no error | Production state | measurement data erroneous | sensor functionality and/or processing error | invalid measurement data | measurement with reduced reliability | initialization |
|---|---|---|---|---|---|---|---|---|---|
| H.1 H.6, H.7 | Fast Channel 1 | Bit 0 | 0 | x | 1 | 1 | 1 | 1 | 0 |
| H.1 H.6, H.7 | Fast Channel 1 | Bit 1 | x | x | x | x | x | x | x |
| H.1 H.6, H.7 | Fast Channel 1 | Data Nibble 1 – β | MD | IM1 | IM5 or IM6 | IM6 | IM7 | MD or IM7[2] | IM0 |
| H.1 H.6, H.7 | Fast Channel 1 | Data Nibble χ - 6 | x | x | x | x | x | x | x |
| H.1 H.6, H.7 | Fast Channel 2 | Bit 0 | x | x | x | x | x | x | x |
| H.1 H.6, H.7 | Fast Channel 2 | Bit 1 | 0 | x | 1 | 1 | 1 | 1 | 0 |
| H.1 H.6, H.7 | Fast Channel 2 | Data Nibble 1 - β | x | x | x | x | x | x | x |
| H.1 H.6, H.7 | Fast Channel 2 | Data Nibble χ - 6 | MD | IM1 | IM5 or IM6 | IM6 | IM7 | MD or IM7[2] | IM0 |
| H.2 | | Bit 0 | 0 | x | 1 | 1 | 1 | 1 | 0 |
| H.2 | | Bit 1 | x | x | x | x | x | x | x |
| H.2 | | Data Nibble 1 - 3 | MD | IM1 | IM5 or IM6 | IM6 | IM7 | MD or IM7[2] | IM0 |
| H.3 | | Bit 0 | 0 | x | 1[1] | 1[1] | 1[1] | 1[1] | 0 |
| H.3 | | Bit 1 | x | x | x | x | x | x | x |
| H.3 | | Data Nibble 1 - 4 | MD | IM1 | IM5 or IM6 | IM6 | IM7 | MD or IM7[2] | IM0 |

| | | measurement data – no error | Production state | measurement data erroneous | sensor functionality and/or processing error | invalid measurement data | measurement with reduced reliability | initialization |
|---|---|---|---|---|---|---|---|---|
| H.4 | Bit 0 | 0 | x | 1 | 1 | 1 | 1 | 0 |
| | Bit 1 | x | x | x | x | x | x | x |
| | Data Nibble 1 - 3 | MD | IM1 | IM5 or IM6 | IM6 | IM7 | MD or IM7[2)] | IM0 |
| | Data Nibble 4 - 6 | SI | x | SI | x | SI | SI | x |
| H.5 | Bit 0 | 0 | x | 1 | 1 | 1 | 1 | 0 |
| | Bit 1 | x | x | x | x | x | x | x |
| | Data Nibble 1 - 3 | MD | IM1 | IM5 or IM6 | IM6 | IM7 | MD or IM7[2)] | IM0 |
| | Data Nibble 4 - 6 | 0 | x | 0 | 0 | 0 | 0 | 0 |

[1] Optional use of error flags (reduction of H.3 frame length)
[2] e.g., applicable for safety relevant sensors

NOTE:

| | |
|---|---|
| $\beta$, $\chi$ | $\beta = 3$, $\chi = 4$ for H.1; $\beta = 4$, $\chi = 4$ for H.6; and $\beta = 4$, $\chi = 5$ for H.7, |
| MD | measurement data |
| SI | secure sensor information (8-bit rolling counter and inverted copy of nibble 1) |
| x | don't care |
| IM0 - IM7 | error indicators / specific messages / initialization messages |

E.3.3 Receiver Error Flag Handling Recommendation

The Status and Communication nibble is not protected by the CRC of the SENT data frames. It is recommended that the receiver should contain a mechanism that takes into account the reduced reliability of single error flags (e.g., some filtering or debouncing of the error flag).

E.3.4 Example Usage of Error Messages

Error scenarios:

**Measurement data is available with restricted accuracy**

- Error indication (general):
  - Error flag raised
- Error status (specific):
  - Error and status indicator messages over serial message channel (if implemented)

Example: Unreliable pressure data due to overheating of sensor

**No measurement data or unreliable measurement data**

- Error indication (general):
    - Error flag raised
- Error status (specific):
    - Error and status indicator messages over serial message channel (if implemented)
    - Error indicator ($n$-bit, fast channel)
    - Information in serial channel messages takes priority (if available)

Example: No valid measurement data due to mechanical failure of sensor

# APPENDIX F - FAST CHANNEL MULTIPLEXING

## F.1 FAST CHANNEL MULTIPLEXING DEFINITION

Fast channel multiplexing specifies an optional time-division multiplexing scheme for the SENT interface. SENT messages with fast channel data from multiple data sources (e.g., sensor measurement data, sensor status information) are multiplexed over one SENT interface. A multiplexing sequence can be composed of up to 16 SENT messages. Each message is identified by its 4-bit Frame Control (FC) value. The length of a sequence may be larger than the number of unique FC values used. Optionally, SENT messages from different sources which constitute a group are marked by a common 4-bit data consistency counter (DCC) value.

Application examples are given hereafter:

- 3D position Sensor (x-, y-,z-position data each 12/14-bit)
- High Temperature Sensor Cluster (more than two 12-bit temperature sensor data)

## F.2 GENERIC DEFINITION OF DATA FRAME FORMAT

If fast channel multiplexing is used the first data nibble pulse is defined as Frame Control nibble (FC) with the remaining data nibbles containing the multiplexed sensor data, as shown in Table F.2-1. For each fast channel multiplexed SENT interface a Frame Control Matrix is defined that links the multiplexed sensor data format and content to the corresponding Frame Control value. Within this Frame Control Matrix each 4-bit multiplexed Nibble can be defined independently from the others. One or more sequences of Frame Control values also must be defined for the recurrent transmission of the multiplexed sensor data. If more than one sequence of Frame Control values is defined, the conditions under which this transmission schedule would change must also be defined.

Optionally the second data nibble of each message defined by Frame Control values can be assigned as 4-bit Data Consistency Counter (DCC) to tag multiplexed sensor data belonging together (e.g., same sampling time from sensor cluster). The DCC value is defined as a rolling counter and incremented with each recurrence of the Frame Control sequence. If the DCC value is used, it is not required to be included in all messages defined by Frame Control values within a message sequence.

***Table F.2-1 – Fast channel multiplexing data frame formats examples with six data nibbles***

| | **Data Nibbles** | | | | | |
|---|---|---|---|---|---|---|
| | **1** | **2** | **3** | **4** | **5** | **6** |
| Data Frame Format w/o Data Consistency Counter (DCC) | Frame Control (FC) | 4-bit mux. Nibble 1 | 4-bit mux. Nibble 2 | 4-bit mux. Nibble 3 | 4-bit mux. Nibble 4 | 4-bit mux. Nibble 5 |
| Data Frame Format with Data Consistency Counter (DCC) | Frame Control (FC) | Data Consistency Counter (DCC) | 4-bit mux. Nibble 1 | 4-bit mux. Nibble 2 | 4-bit mux. Nibble 3 | 4-bit mux. Nibble 4 |

A fast channel multiplexing data frame consists of a minimum of two data nibbles (one Frame Control nibble and one multiplexed nibble) and shall have the same number of data nibbles for the whole multiplexing sequence.

This version of the standard and APPENDIX F currently defines up to 6 data nibbles only, but more data nibbles are considered for future revisions (see chapter 5.2.1).

If sensor data is transmitted with multiplexed fast channels, the following definitions of mapping the data to the multiplexed nibbles shall be applied similar to non-multiplexed data frame definitions in APPENDIX H.

- If only one sensor data channel is mapped into a Frame Control Matrix entry the bit order is always MSN with the first multiplexed Nibble to LSN with the last multiplexed Nibble.

- If two sensor data channels are mapped into a Frame Control Matrix entry the bit order for the first sensor data shall be MSN in the first multiplexed Nibble to LSN in the last multiplexed Nibble used for the first sensor data. The second sensor data is subsequently mapped with LSN in the first multiplexed Nibble to MSN in the last multiplexed Nibble used for the second sensor data.

- If the partitioning of two sensor data channels is such that one multiplexed Nibble is used for transmitting bits of both sensor data channels, the MSBs of the shared nibble are used as LSBs for sensor data channel one and the LSBs of the shared Nibble are used as LSBs for sensor data channel two. This mapping of the bits corresponds to the mapping onto the shared nibble defined in Table H.6-1 – Nibble and bit orders for sensors with 14-bit fast channel 1 and 10-bit fast channel 2.

***Table F.2-2 – Examples for mapping of sensor data into fast channel multiplexing data frame format***

<table>
<tr><th rowspan="2"></th><th colspan="24">Data Nibbles</th></tr>
<tr><th colspan="4">1</th><th colspan="4">2</th><th colspan="4">3</th><th colspan="4">4</th><th colspan="4">5</th><th colspan="4">6</th></tr>
<tr><td rowspan="3">One Sensor data (12-bit) w/o DCC</td><td colspan="4" rowspan="2">FC</td><td colspan="12">Sensor 1 (12-bit) Sensor Data</td><td colspan="4" rowspan="3"></td><td colspan="4" rowspan="3"></td></tr>
<tr><td colspan="4">MSN</td><td colspan="4">MidN</td><td colspan="4">LSN</td></tr>
<tr><td>$2^3$</td><td>$2^2$</td><td>$2^1$</td><td>$2^0$</td><td>$2^{11}$</td><td>$2^{10}$</td><td>$2^9$</td><td>$2^8$</td><td>$2^7$</td><td>$2^6$</td><td>$2^5$</td><td>$2^4$</td><td>$2^3$</td><td>$2^2$</td><td>$2^1$</td><td>$2^0$</td></tr>
<tr><td rowspan="3">One Sensor data (12-bit) w/ DCC</td><td colspan="4" rowspan="2">FC</td><td colspan="4" rowspan="2">DCC</td><td colspan="12">Sensor 1 (12-bit) Sensor</td><td colspan="4" rowspan="3"></td></tr>
<tr><td colspan="4">MSN</td><td colspan="4">MidN</td><td colspan="4">LSN</td></tr>
<tr><td>$2^3$</td><td>$2^2$</td><td>$2^1$</td><td>$2^0$</td><td>$2^3$</td><td>$2^2$</td><td>$2^1$</td><td>$2^0$</td><td>$2^{11}$</td><td>$2^{10}$</td><td>$2^9$</td><td>$2^8$</td><td>$2^7$</td><td>$2^6$</td><td>$2^5$</td><td>$2^4$</td><td>$2^3$</td><td>$2^2$</td><td>$2^1$</td><td>$2^0$</td></tr>
<tr><td rowspan="3">Two Sensor data (12-bit and 8-bit) w/o DCC</td><td colspan="4" rowspan="2">FC</td><td colspan="12">Sensor 1 (12-bit) Sensor Data</td><td colspan="8">Sensor 2 (8-bit) Sensor Data</td></tr>
<tr><td colspan="4">MSN</td><td colspan="4">MidN</td><td colspan="4">LSN</td><td colspan="4">LSN</td><td colspan="4">MSN</td></tr>
<tr><td>$2^3$</td><td>$2^2$</td><td>$2^1$</td><td>$2^0$</td><td>$2^{11}$</td><td>$2^{10}$</td><td>$2^9$</td><td>$2^8$</td><td>$2^7$</td><td>$2^6$</td><td>$2^5$</td><td>$2^4$</td><td>$2^3$</td><td>$2^2$</td><td>$2^1$</td><td>$2^0$</td><td>$2^3$</td><td>$2^2$</td><td>$2^1$</td><td>$2^0$</td><td>$2^7$</td><td>$2^6$</td><td>$2^5$</td><td>$2^4$</td></tr>
<tr><td rowspan="3">Two Sensor data (10-bit) w/o DCC</td><td colspan="4" rowspan="2">FC</td><td colspan="10">Sensor 1 (10-bit) Sensor Data</td><td colspan="10">Sensor 2 (10-bit) Sensor Data</td></tr>
<tr><td colspan="4">MSN</td><td colspan="4">MidN</td><td colspan="4">LSN</td><td colspan="4">MidN</td><td colspan="4">MSN</td></tr>
<tr><td>$2^3$</td><td>$2^2$</td><td>$2^1$</td><td>$2^0$</td><td>$2^9$</td><td>$2^8$</td><td>$2^7$</td><td>$2^6$</td><td>$2^5$</td><td>$2^4$</td><td>$2^3$</td><td>$2^2$</td><td>$2^1$</td><td>$2^0$</td><td>$2^1$</td><td>$2^0$</td><td>$2^5$</td><td>$2^4$</td><td>$2^3$</td><td>$2^2$</td><td>$2^9$</td><td>$2^8$</td><td>$2^7$</td><td>$2^6$</td></tr>
</table>

## F.3 INITIALIZATION

During Initialization phase of the transmitter any multiplexed channel that has no sensor data available shall transmit the initialization message (data value = 0). The transmitter may optionally keep the frame control value at 0 until some or all of the sensor channel data are available; then a normal operation transmission sequence is started.

## F.4 STATUS AND COMMUNICATION NIBBLE

Bit 0 and Bit 1 of each Frame Control Matrix entry message shall be used as error indication of the current fast channel as defined hereafter.
In case only one sensor data channel is mapped into the Frame Control Matrix entry message:

- Bit 0 shall be set to 1 in case the sensor is determined to be in error, otherwise Bit 0 shall be set to 0.

- Bit 1: don’t care.

In case two sensor data channels are mapped into the Frame Control Matrix entry message:

- Bit 0 shall be set to 1 in case the first sensor is determined to be in error, otherwise Bit 0 shall be set to 0.
- Bit 1 shall be set to 1 in case the second sensor is determined to be in error, otherwise Bit 0 shall be set to 0.

Bit 2 and Bit 3 of the status and communication nibble can optionally be used to carry serial message data. Fast Channel Multiplexed transmitters shall use the Enhanced Serial Message format, if serial messages are transmitted.

In contrast to Bit 0 and Bit 1 which are linked with each multiplexed fast channel, Bit 2 and Bit 3 used as serial message channel shall not being multiplexed. The serial message channel is independent from the fast channel multiplexing and transmitted with each consecutive message as defined in section 5.2.4.2.

***Table F.4-1 – Allocation of the bits of the status and communication nibble***

| Bit Number | Bit Function | Comment |
|---|---|---|
| 0<br>(least significant) | Single or First Channel Indicator<br>(1 = error, 0 otherwise) | Error Flags |
| 1 | Optional Second Channel Indicator<br>(1 = error, 0 otherwise) | |
| 2 | Serial Data message bits | Serial Message Channel<br>(serial message channel) |
| 3<br>(most significant) | Serial Data message bits | |

## F.5 APPLICATION EXAMPLE

In Figure F.5-1 an example is given which shows the fast channel multiplexing with transmission of 3 x 16-bit Sensor data (Data 1…3) from one Sensor using 3 Frame Control values and the Data Consistency Counter (DCC).

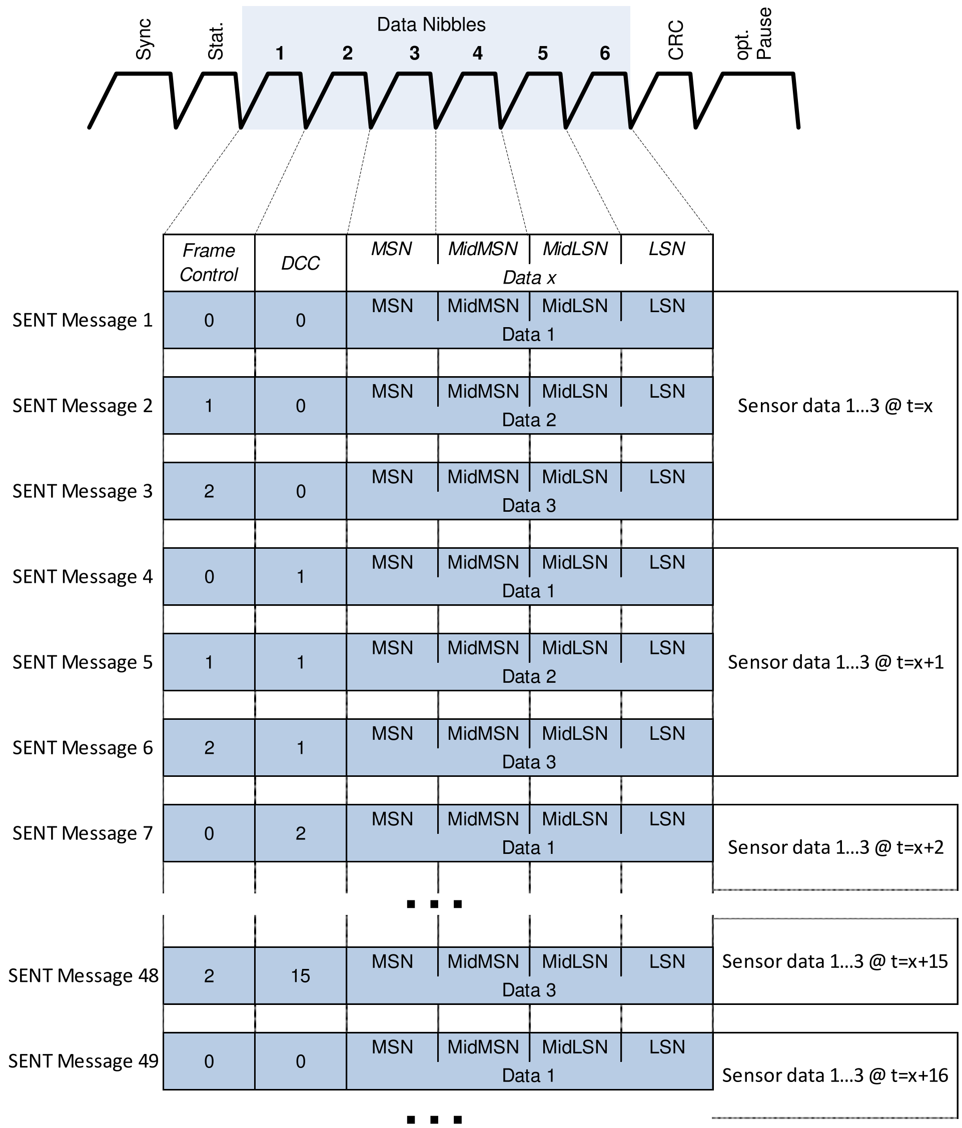



***Figure F.5-1 – Example of fast channel multiplexing with DCC***

Examples of how to set Bit 0 and Bit 1 of each multiplexed fast channel as error indication of the current fast channel are shown in Table F.5-1 and Table F.5-2 for different failure cases.

Table F.5-1 provides an example for transmission of only one 12-bit Sensor Data in each of the four Frame Control Matrix entry messages together with the Data consistency Counter (DCC).

***Table F.5-1 – Example of failure cases with single sensor data channel***

| Failure Case Description | Status & Comm. Nibble | | | | Data Nibbles | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | 1 | 2 | 3 | 4 | 5 | 6 |
| | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Frame Control | Data Consistency Counter | Sensor data (12-bit) | | | e.g., reserved |
| **no Error** | | | | | | | | | | |
| | x | x | x | 0 | 0 | DCC | Sensor 1 (12-bit)<br>*Sensor Data* | | | reserved |
| | x | x | x | 0 | 1 | DCC | Sensor 2 (12-bit)<br>*Sensor Data* | | | reserved |
| | x | x | x | 0 | 2 | DCC | Sensor 3 (12-bit)<br>*Sensor Data* | | | reserved |
| | x | x | x | 0 | 3 | DCC | Sensor 4 (12-bit)<br>*Sensor Data* | | | reserved |
| **Error with Sensor 1 and 3** | | | | | | | | | | |
| Error with Sensor 1 but transmission of Sensor Data possible (e.g., Overvoltage) | x | x | x | 1 | 0 | DCC | Sensor 1 (12-bit)<br>*Sensor Data* | | | reserved |
| | x | x | x | 0 | 1 | DCC | Sensor 2 (12-bit)<br>*Sensor Data* | | | reserved |
| Error with Sensor 3 w/o transmission of Sensor Data | x | x | x | 1 | 2 | DCC | Sensor 3 (12-bit)<br>*error indicator (e.g., IM5)* | | | reserved |
| | x | x | x | 0 | 3 | DCC | Sensor 4 (12-bit)<br>*Sensor Data* | | | reserved |
| **Sensor Module Error but transmission of Sensor Data is possible (e.g., Over-temperature, Over-/Under-voltage)** | | | | | | | | | | |
| | x | x | x | 1 | 0 | DCC | Sensor 1 (12-bit)<br>*Sensor Data* | | | reserved |
| | x | x | x | 1 | 1 | DCC | Sensor 2 (12-bit)<br>*Sensor Data* | | | reserved |
| | x | x | x | 1 | 2 | DCC | Sensor 3 (12-bit)<br>*Sensor Data* | | | reserved |
| | x | x | x | 1 | 3 | DCC | Sensor 4 (12-bit)<br>*Sensor Data* | | | reserved |
| **Sensor Module Error w/o transmission of Sensor Data** | | | | | | | | | | |
| | x | x | x | 1 | 0 | DCC | Sensor 1 (12-bit)<br>*error indicator (e.g., IM5)* | | | reserved |
| | x | x | x | 1 | 1 | DCC | Sensor 2 (12-bit)<br>*error indicator (e.g., IM5)* | | | reserved |
| | x | x | x | 1 | 2 | DCC | Sensor 3 (12-bit)<br>*error indicator (e.g., IM5)* | | | reserved |
| | x | x | x | 1 | 3 | DCC | Sensor 4 (12-bit)<br>*error indicator (e.g., IM5)* | | | reserved |

Table F.5-2 provides an example for transmission of two 10bit Sensor Data in each of the four Frame Control Matrix entry messages.

***Table F.5-2 – Example of failure cases with two sensor data channels***

<table>
<tr><th rowspan="3"></th><th colspan="4" rowspan="2">Status & Comm. Nibble</th><th colspan="8">Data Nibbles</th></tr>
<tr><th>1</th><th>2</th><th>3</th><th colspan="2">4</th><th>5</th><th>6</th></tr>
<tr><th>Bit 3</th><th>Bit 2</th><th>Bit 1</th><th>Bit 0</th><th>Frame Control</th><th colspan="3">Sensor data (10-bit)</th><th colspan="3">Sensor data (10-bit)</th></tr>
<tr><td colspan="12">no Error</td></tr>
<tr><td rowspan="2"></td><td rowspan="2">x</td><td rowspan="2">x</td><td rowspan="2">0</td><td rowspan="2">0</td><td rowspan="2">0</td><td colspan="3">Sensor 1 (10-bit)</td><td colspan="3">Sensor 2 (10-bit)</td></tr>
<tr><td colspan="3">Sensor Data</td><td colspan="3">Sensor Data</td></tr>
<tr><td rowspan="2"></td><td rowspan="2">x</td><td rowspan="2">x</td><td rowspan="2">0</td><td rowspan="2">0</td><td rowspan="2">1</td><td colspan="3">Sensor 3 (10-bit)</td><td colspan="3">Sensor 4 (10-bit)</td></tr>
<tr><td colspan="3">Sensor Data</td><td colspan="3">Sensor Data</td></tr>
<tr><td rowspan="2"></td><td rowspan="2">x</td><td rowspan="2">x</td><td rowspan="2">0</td><td rowspan="2">0</td><td rowspan="2">2</td><td colspan="3">Sensor 5 (10-bit)</td><td colspan="3">Sensor 6 (10-bit)</td></tr>
<tr><td colspan="3">Sensor Data</td><td colspan="3">Sensor Data</td></tr>
<tr><td rowspan="2"></td><td rowspan="2">x</td><td rowspan="2">x</td><td rowspan="2">0</td><td rowspan="2">0</td><td rowspan="2">3</td><td colspan="3">Sensor 7 (10-bit)</td><td colspan="3">Sensor 8 (10-bit)</td></tr>
<tr><td colspan="3">Sensor Data</td><td colspan="3">Sensor Data</td></tr>
<tr><td colspan="12">Error with Sensor 1, 4, 7, 8</td></tr>
<tr><td rowspan="2">Error with Sensor 1 but transmission of Sensor Data possible (e.g., Overvoltage)</td><td rowspan="2">x</td><td rowspan="2">x</td><td rowspan="2">0</td><td rowspan="2">1</td><td rowspan="2">0</td><td colspan="3">Sensor 1 (10-bit)</td><td colspan="3">Sensor 2 (10-bit)</td></tr>
<tr><td colspan="3">Sensor Data</td><td colspan="3">Sensor Data</td></tr>
<tr><td rowspan="2">Error with Sensor 4 w/o transmission of Sensor Data</td><td rowspan="2">x</td><td rowspan="2">x</td><td rowspan="2">1</td><td rowspan="2">0</td><td rowspan="2">1</td><td colspan="3">Sensor 3 (10-bit)</td><td colspan="3">Sensor 4 (10-bit)</td></tr>
<tr><td colspan="3">Sensor Data</td><td colspan="3">error indicator (e.g., IM5)</td></tr>
<tr><td rowspan="2"></td><td rowspan="2">x</td><td rowspan="2">x</td><td rowspan="2">0</td><td rowspan="2">0</td><td rowspan="2">2</td><td colspan="3">Sensor 5 (10-bit)</td><td colspan="3">Sensor 6 (10-bit)</td></tr>
<tr><td colspan="3">Sensor Data</td><td colspan="3">Sensor Data</td></tr>
<tr><td rowspan="2">Error with Sensor 7 and 8 w/o transmission of Sensor Data</td><td rowspan="2">x</td><td rowspan="2">x</td><td rowspan="2">1</td><td rowspan="2">1</td><td rowspan="2">3</td><td colspan="3">Sensor 7 (10-bit)</td><td colspan="3">Sensor 8 (10-bit)</td></tr>
<tr><td colspan="3">error indicator (e.g., IM5)</td><td colspan="3">error indicator (e.g., IM5)</td></tr>
<tr><td colspan="12">Sensor Module Error but transmission of Sensor Data is possible (e.g., Over-temperature, Over-/Under-voltage)</td></tr>
<tr><td rowspan="2"></td><td rowspan="2">x</td><td rowspan="2">x</td><td rowspan="2">1</td><td rowspan="2">1</td><td rowspan="2">0</td><td colspan="3">Sensor 1 (10-bit)</td><td colspan="3">Sensor 2 (10-bit)</td></tr>
<tr><td colspan="3">Sensor Data</td><td colspan="3">Sensor Data</td></tr>
<tr><td rowspan="2"></td><td rowspan="2">x</td><td rowspan="2">x</td><td rowspan="2">1</td><td rowspan="2">1</td><td rowspan="2">1</td><td colspan="3">Sensor 3 (10-bit)</td><td colspan="3">Sensor 4 (10-bit)</td></tr>
<tr><td colspan="3">Sensor Data</td><td colspan="3">Sensor Data</td></tr>
<tr><td rowspan="2"></td><td rowspan="2">x</td><td rowspan="2">x</td><td rowspan="2">1</td><td rowspan="2">1</td><td rowspan="2">2</td><td colspan="3">Sensor 5 (10-bit)</td><td colspan="3">Sensor 6 (10-bit)</td></tr>
<tr><td colspan="3">Sensor Data</td><td colspan="3">Sensor Data</td></tr>
<tr><td rowspan="2"></td><td rowspan="2">x</td><td rowspan="2">x</td><td rowspan="2">1</td><td rowspan="2">1</td><td rowspan="2">3</td><td colspan="3">Sensor 7 (10-bit)</td><td colspan="3">Sensor 8 (10-bit)</td></tr>
<tr><td colspan="3">Sensor Data</td><td colspan="3">Sensor Data</td></tr>
<tr><td colspan="12">Sensor Module Error w/o transmission of Sensor Data</td></tr>
<tr><td rowspan="2"></td><td rowspan="2">x</td><td rowspan="2">x</td><td rowspan="2">1</td><td rowspan="2">1</td><td rowspan="2">0</td><td colspan="3">Sensor 1 (10-bit)</td><td colspan="3">Sensor 2 (10-bit)</td></tr>
<tr><td colspan="3">error indicator (e.g., IM5)</td><td colspan="3">error indicator (e.g., IM5)</td></tr>
<tr><td rowspan="2"></td><td rowspan="2">x</td><td rowspan="2">x</td><td rowspan="2">1</td><td rowspan="2">1</td><td rowspan="2">1</td><td colspan="3">Sensor 3 (10-bit)</td><td colspan="3">Sensor 4 (10-bit)</td></tr>
<tr><td colspan="3">error indicator (e.g., IM5)</td><td colspan="3">error indicator (e.g., IM5)</td></tr>
<tr><td rowspan="2"></td><td rowspan="2">x</td><td rowspan="2">x</td><td rowspan="2">1</td><td rowspan="2">1</td><td rowspan="2">2</td><td colspan="3">Sensor 5 (10-bit)</td><td colspan="3">Sensor 6 (10-bit)</td></tr>
<tr><td colspan="3">error indicator (e.g., IM5)</td><td colspan="3">error indicator (e.g., IM5)</td></tr>
<tr><td rowspan="2"></td><td rowspan="2">x</td><td rowspan="2">x</td><td rowspan="2">1</td><td rowspan="2">1</td><td rowspan="2">3</td><td colspan="3">Sensor 7 (10-bit)</td><td colspan="3">Sensor 8 (10-bit)</td></tr>
<tr><td colspan="3">error indicator (e.g., IM5)</td><td colspan="3">error indicator (e.g., IM5)</td></tr>
</table>

## F.6 BASIC FAST CHANNEL MULTIPLEXING SENT DATA FRAME FORMATS

Many application-specific SENT protocols with fast channel multiplexing may use the same frame formats, data channels and nibble orders. Those basic fast channel multiplexing SENT data frame formats, which are used by several application-specific protocols (as specified in APPENDIX A), are defined hereafter. The application specific definitions can combine the different basic data frame formats according to the generic definitions of section F.2, e.g., same nibble count and implementation of optional DCC for all multiplexed fast channels. The implementation of pause pulse is optional for all fast channel multiplexing SENT Data frame formats.

Basic SENT data frame formats for multiplexing fast channels are defined as follows:

- F1.x: One sensor data with 12-bit representation
- F2.x: One sensor data with 16-bit representation
- F3.x: Two sensor data with different bit representations
- F4.x: Free to use data definition

F.6.1 Data Frames with One Multiplexed Fast Channel and 12-bit Sensor Data

Table F.6.1-1 defines the standardized Frame Control Matrix entries with one multiplexed fast channel which contains one sensor data with 12-bit representation in combination with or without DCC. The possible combinations also list the usage of the remaining nibbles either as application specific, filled with zeros or not implemented.

***Table F.6.1-1 – Data frames with 12-bit sensor data***

<table>
<tr><th rowspan="2">ID</th><th colspan="6">Data Nibbles</th></tr>
<tr><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th></tr>
<tr><td rowspan="2">F1.1</td><td rowspan="2">FC</td><td colspan="3">12-bit Sensor Data</td><td colspan="2" rowspan="2">application specific</td></tr>
<tr><td>MSN</td><td>MidN</td><td>LSN</td></tr>
<tr><td rowspan="2">F1.2</td><td rowspan="2">FC</td><td colspan="3">12-bit Sensor Data</td><td colspan="2" rowspan="2">Zeros</td></tr>
<tr><td>MSN</td><td>MidN</td><td>LSN</td></tr>
<tr><td rowspan="2">F1.3</td><td rowspan="2">FC</td><td colspan="3">12-bit Sensor Data</td><td colspan="2" rowspan="2"></td></tr>
<tr><td>MSN</td><td>MidN</td><td>LSN</td></tr>
<tr><td rowspan="2">F1.4</td><td rowspan="2">FC</td><td rowspan="2">DCC</td><td colspan="3">12-bit Sensor Data</td><td rowspan="2">application specific</td></tr>
<tr><td>MSN</td><td>MidN</td><td>LSN</td></tr>
<tr><td rowspan="2">F1.5</td><td rowspan="2">FC</td><td rowspan="2">DCC</td><td colspan="3">12-bit Sensor Data</td><td rowspan="2">Zeros</td></tr>
<tr><td>MSN</td><td>MidN</td><td>LSN</td></tr>
<tr><td rowspan="2">F1.6</td><td rowspan="2">FC</td><td rowspan="2">DCC</td><td colspan="3">12-bit Sensor Data</td><td rowspan="2"></td></tr>
<tr><td>MSN</td><td>MidN</td><td>LSN</td></tr>
</table>

F.6.2 Data Frames with One Multiplexed Fast Channel and 16-bit Sensor Data

Table F.6.2-1 defines the standardized Frame Control Matrix entries with one multiplexed fast channel which contains one sensor data with 16-bit representation in combination with or without DCC. The possible combinations also list the usage of the remaining nibbles either as application specific, filled with zeros or not implemented.

***Table F.6.2-1 – Data frames with 16-bit sensor data***

<table>
<tr><th rowspan="2">ID</th><th colspan="6">Data Nibbles</th></tr>
<tr><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th></tr>
<tr><td rowspan="2">F2.1</td><td rowspan="2">FC</td><td colspan="4">16-bit Sensor Data</td><td rowspan="2">application specific</td></tr>
<tr><td>MSN</td><td>MidMSN</td><td>MidLSN</td><td>LSN</td></tr>
<tr><td rowspan="2">F2.2</td><td rowspan="2">FC</td><td colspan="4">16-bit Sensor Data</td><td rowspan="2">Zeros</td></tr>
<tr><td>MSN</td><td>MidMSN</td><td>MidLSN</td><td>LSN</td></tr>
<tr><td rowspan="2">F2.3</td><td rowspan="2">FC</td><td colspan="4">16-bit Sensor Data</td><td rowspan="2"></td></tr>
<tr><td>MSN</td><td>MidMSN</td><td>MidLSN</td><td>LSN</td></tr>
<tr><td rowspan="2">F2.4</td><td rowspan="2">FC</td><td rowspan="2">DCC</td><td colspan="4">16-bit Sensor Data</td></tr>
<tr><td>MSN</td><td>MidMSN</td><td>MidLSN</td><td>LSN</td></tr>
</table>

### F.6.3 Data Frames with Two Multiplexed Fast Channels with Sensor Data

Table F.6.3-1 defines possible data frames with two multiplexed fast channels.

With 6 data nibbles, frames are defined which contain either 12-bit sensor data and 8-bit sensor data or 10-bit sensor data and 10-bit sensor data. These combinations require all 6 data nibbles and are therefore not possible with DCC.

With 8 data nibbles (the standard currently defines up to 6 data nibbles only, but more data nibbles are considered for future revisions (see chapter 5.2.1)), higher bit widths are possible for the two sensor values in a given Frame Control Matrix entry. Frames without DCC are defined for 16-bit sensor data and 12-bit sensor data or 14-bit sensor data and 14-bit sensor data. With DCC a data frame is defined with 12-bit sensor data and 12-bit sensor data.

***Table F.6.3-1 – Data frames with two sensor data channels***

<table>
<tr><th rowspan="2">ID</th><th colspan="8">Data Nibbles</th></tr>
<tr><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7[1]</th><th>8[1]</th></tr>
<tr><td rowspan="2">F3.1</td><td rowspan="2">FC</td><td colspan="3">12-bit Sensor Data</td><td colspan="2">8-bit Sensor Data</td><td rowspan="2"></td><td rowspan="2"></td></tr>
<tr><td>MSN</td><td>MidN</td><td>LSN</td><td>LSN</td><td>MSN</td></tr>
<tr><td rowspan="2">F3.2</td><td rowspan="2">FC</td><td colspan="3">10-bit Sensor Data</td><td colspan="2">10-bit Sensor Data</td><td rowspan="2"></td><td rowspan="2"></td></tr>
<tr><td>MSN</td><td>MidN</td><td>LSN</td><td>MidN</td><td>MSN</td></tr>
<tr><td rowspan="2">F3.3[1]</td><td rowspan="2">FC</td><td colspan="4">16-bit Sensor Data</td><td colspan="3">12-bit Sensor Data</td></tr>
<tr><td>MSN</td><td>MidMSN</td><td>MidLSN</td><td>LSN</td><td>LSN</td><td>MidN</td><td>MSN</td></tr>
<tr><td rowspan="2">F3.4[1]</td><td rowspan="2">FC</td><td colspan="4">14-bit Sensor Data</td><td colspan="3">14-bit Sensor Data</td></tr>
<tr><td>MSN</td><td>MidMSN</td><td>MidLSN</td><td>LSN</td><td>MidLSN</td><td>MidMSN</td><td>MSN</td></tr>
<tr><td rowspan="2">F3.5[1]</td><td rowspan="2">FC</td><td rowspan="2">DCC</td><td colspan="3">12-bit Sensor Data</td><td colspan="3">12-bit Sensor Data</td></tr>
<tr><td>MSN</td><td>MidN</td><td>LSN</td><td>LSN</td><td>MidN</td><td>MSN</td></tr>
</table>

### F.6.4 Data Frames with Free to Use Data

Table F.6.4-1 defines possible data frames containing free to use data in variants with or without DCC and different nibble count which can be combined with data frames containing sensor data as defined in Section F.6.1 to F.6.3.

***Table F.6.4-1 – Data frames with free to use data***

<table>
<tr><th rowspan="2">ID</th><th colspan="8">Data Nibbles</th></tr>
<tr><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7[1]</th><th>8[1]</th></tr>
<tr><td>F4.1[1]</td><td>FC</td><td colspan="7">free to use</td></tr>
<tr><td>F4.2[1]</td><td>FC</td><td colspan="6">free to use</td><td></td></tr>
<tr><td>F4.3</td><td>FC</td><td colspan="5">free to use</td><td></td><td></td></tr>
<tr><td>F4.4</td><td>FC</td><td colspan="4">free to use</td><td></td><td></td><td></td></tr>
<tr><td>F4.5</td><td>FC</td><td colspan="3">free to use</td><td></td><td></td><td></td><td></td></tr>
<tr><td>F4.6[1]</td><td>FC</td><td>DCC</td><td colspan="6">free to use</td></tr>
<tr><td>F4.7[1]</td><td>FC</td><td>DCC</td><td colspan="5">free to use</td><td></td></tr>
<tr><td>F4.8</td><td>FC</td><td>DCC</td><td colspan="4">free to use</td><td></td><td></td></tr>
<tr><td>F4.9</td><td>FC</td><td>DCC</td><td colspan="3">free to use</td><td></td><td></td><td></td></tr>
</table>

[1] This version of the standard currently defines up to 6 data nibbles only, but more data nibbles are considered for future revisions (see chapter 5.2.1)

# APPENDIX G - SENT CONNECTORS

This appendix provides the mechanical and electrical specifications for the SENT transmitter connectors and pin configurations that can be optionally used to connect SENT devices to the receiver (e.g., control unit).

The connectors described in this section are optional. The SENT standard permits the use of other pin configurations and connectors than those described in this section.

NOTE: This revision of J2716 only specifies SENT 3-way connectors.

## G.1 SENT CONNECTOR TERMINATION DATA

### G.1.1 SENT 3-Way Connectors

#### G.1.1.1 Pin Configuration

Table G.1.1.1-1 specifies the recommended termination assignments of the contacts by number and electrical signal name for the following connector types:

- SENT connectors as specified in G.2.1
- Other connector types

***Table G.1.1.1-1 – SENT connector termination assignment***

| Contact Number | Signal Name |
|---|---|
| 1 | SENT |
| 2 | GND |
| 3 | $V_{SUPPLY}$ |

The example in Figure G.1.1.1-1 illustrates the assignment of the electrical signals to the connector contacts for the connector as specified in G.2.1.

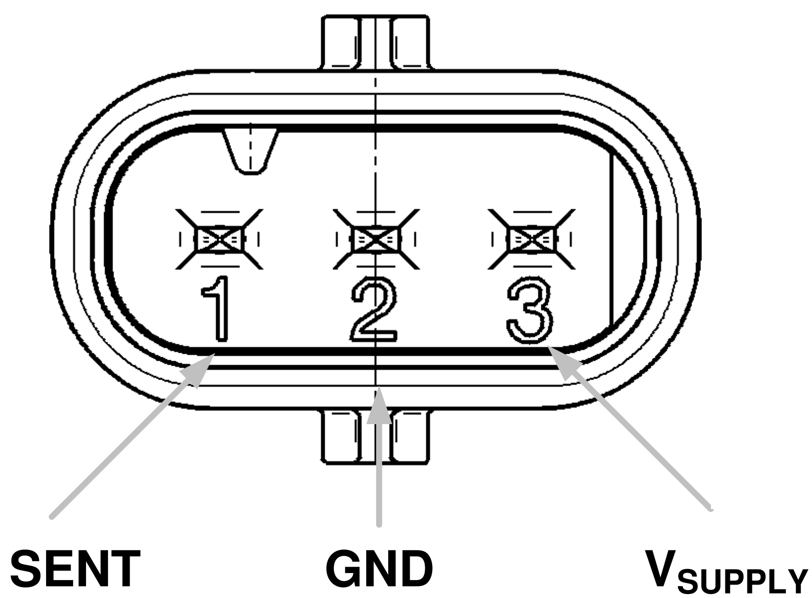



***Figure G.1.1.1-1 – Assignment example of the electrical signals to the contact numbers (SENT AK connector)***

## G.2 SPECIFICATION OF PLUG AND SOCKET CONNECTORS

Electrical and mechanical interface configurations for SENT plugs at the transmitter (e.g., sensor) as specified here are optional.

### G.2.1 SENT 3-Way Plug

#### G.2.1.1 Connector Geometry

The geometry of the 3-way plug and socket is specified in “USCAR EWCAP footprint 1.2 mm connector 120-S-003-1-Z01 Rev. D” (*Direct connect interface, 3 way (1 x 3), 1.2 mm, sealed male*).

Please see reference in 2.1.2 (USCAR)

#### G.2.1.2 Keying

Keying options A, B, C, D and Z, as specified in USCAR 120-S-003-1-Z01 Rev. D can be used. However, keying A is the preferred keying for the SENT 3-way plug.

## APPENDIX H - SENT DATA FRAME FORMATS

Many application-specific SENT protocols use the same frame formats, data channels and nibble orders. Those basic SENT data frame formats, which are used by several application-specific protocols (as specified in APPENDIX A), are put together in APPENDIX H.

SENT sensor types are specified in section D.4.1. Either a basic frame format from APPENDIX H or an application-specific frame format is assigned to each sensor type. If one of the basic frame formats from APPENDIX H is used, a reference to the respective frame format is made in the application-specific APPENDIX A and the sensor type definition D.4.1.

APPENDIX H defines the following basic SENT frame formats

H.1: Two 12-bit fast channels (6 data nibbles)

H.2: One 12-bit fast channel (3 data nibbles)

H.3: High-speed with one 12-bit fast channel (4 data nibbles, where only values 0-7 are used)

H.4: Secure sensor with 12-bit fast channel 1 and secure sensor information on fast channel 2 (6 data nibbles)

H.5: Single sensor with 12-bit fast channel 1 and zero value on fast channel 2 (6 data nibbles)

H.6: Two fast channels with 14-bit fast channel 1 and 10-bit fast channel 2 (6 data nibbles)

H.7: Two fast channels with 16-bit fast channel 1 and 8-bit fast channel 2 (6 data nibbles)

The nibble orders of the basic SENT frame formats are summarized in Table H-1. It currently defines up to 6 data nibbles only, but more data nibbles are considered for future revisions (see chapter 5.2.1). The definitions in Table H-1 follow these general rules:

- Fast channel 1 nibble and bit order:
  - The data vector (ordered MSB to LSB) is divided into 4-bit groups corresponding to the nibbles and mapped in the same order to the assigned nibbles of fast channel 1.
- Fast channel 2 nibble and bit order
  - Data vector (ordered MSB to LSB) is first divided into 4-bit groups corresponding to the nibbles and then mapped in the reversed order to the assigned nibbles of fast channel 2. An exception is the secure sensor information.
- Fast channel 1 and 2 sharing one nibble (e.g., H.6)
  - If the partitioning of two sensor data channels is in that way that one nibble is used for transmitting bits of both fast channels, the MSBs of the shared nibble are used as LSBs for fast channel 1 and the LSBs of the shared nibble are used as LSBs for fast channel 2.
    The sharing of one nibble and mapping of the bits is applicable with H.6.

***Table H-1 – Basic SENT frame formats, data channels and nibble orders***

| Frame Format | Data Nibbles | Data 1 | Data 2 | Data 3 | Data 4 | Data 5 | Data 6 |
|---|---|---|---|---|---|---|---|
| H.1 Two 12-bit fast channels | 6 | Ch1 MSN | Ch1 MidN | Ch1 LSN | Ch2 LSN | Ch2 MidN | Ch2 MSN |
| H.2 One 12-bit fast channel | 3 | Ch1 MSN | Ch1 MidN | Ch1 LSN | Not imple-mented | Not imple-mented | Not imple-mented |
| H.3 High-speed with one 12-bit fast channel | 4 | Most signifi-cant bits 11 - 9 | Bits 8 – 6 | Bits 5 – 3 | Least significa nt bits 2 - 0 | Not imple-mented | Not imple-mented |
| H.4 Secure sensor with 12-bit fast channel 1 and secure sensor information on fast channel 2 | 6 | Ch1 MSN | Ch1 MidN | Ch1 LSN | Counter MSN | Counter LSN | Inverted Copy Ch1 MSN |
| H.5 Single sensor with 12-bit fast channel 1 and zero value on fast channel 2 | 6 | Ch1 MSN | Ch1 MidN | Ch1 LSN | Zero | Zero | Zero |
| H.6: Two fast channels with 14-bit fast channel 1 and 10-bit fast channel 2 | 6 | Ch1 MSN | Ch1 MidMSN | Ch1 MidLSN | Ch1/Ch 2 LSN | Ch2 MidN | Ch2 MSN |
| H.7: Two fast channels with 16-bit fast channel 1 and 8-bit fast channel 2 | 6 | Ch1 MSN | Ch1 MidMSN | Ch1 MidLSN | Ch1 LSN | Ch2 LSN | Ch2 MSN |

NOTE: If data nibbles are not implemented, this means that they do not exist. In contrast, if data nibbles are not used, they shall be set to zero.

The allocation of the bits of the status and communication nibble is described in Table H-2.

***Table H-2 – Allocation of the bits of the status and communication nibble***

| Bit Number | Bit Function | Comment |
|---|---|---|
| 0 (least significant) | Channel 1 Indicator (1 = error, 0 otherwise) | Error Flags (use of error flags is optional in High-speed frame format H.3) |
| 1 | Channel 2 Indicator (1 = error, 0 otherwise) | |
| 2 | Serial Data message bits | Serial Message Data Channel - Serial message channel (optional depending on application-specific protocol) |
| 3 (most significant) | Serial Data message bits | |

The serial message channel can be optionally used. This data channel can be used to transmit, at a slower rate, additional information such as status information, sensor-specific information and it can also carry the supplementary data channels.

## H.1 SENSORS WITH 6-NIBBLE 12-BIT FAST CHANNEL 1 AND 12-BIT FAST CHANNEL 2 DATA

The data channels of sensors with two 12-bit fast channels are defined as follows (see Table H.1-1):

- Fast Channel 1
  - Data order:
    - data nibble 1: Channel 1 measurement data MSN
    - data nibble 2: Channel 1 measurement data MidN
    - data nibble 3: Channel 1 measurement data LSN
  - Error Flags
    - If the fast channel 1 measurement data is determined to be in error, bit 0 of the Status Nibble shall be set to 1 otherwise bit 0 shall be set to 0 (see Table H-2)
- Fast Channel 2
  - Data order:
    - data nibble 4: Channel 2 measurement data LSN
    - data nibble 5: Channel 2 measurement data MidN
    - data nibble 6: Channel 2 measurement data MSN
  - Data vector (ordered MSB to LSB) is first divided into 4-bit groups corresponding to the nibbles. The nibble order of Channel 2 is then reversed.
  - Error Flags
    - If the fast channel 2 measurement data is determined to be in error, bit 1 of the Status Nibble shall be set to 1 otherwise bit 1 shall be set to 0 (see Table H-2)
- Pause pulse (optional)

*Table H.1-1 – Nibble and bit orders for sensors with two 12-bit fast channels*

| Sensor Data | Bit Weight | SENT Nibble Bits | SENT Nibble |
|---|---|---|---|
| S&C [3] Serial Data Channel | - | S&C [3] | Status and Communication |
| S&C [2] Serial Data Channel | - | S&C [2] | |
| S&C [1] Channel 2 Indicator | - | S&C [1] | |
| S&C [0] Channel 1 Indicator | - | S&C [0] | |
| Channel 1 Data [11] | $2^{11}$ | Channel 1 MSN [3] | Data 1 Channel 1 MSN |
| Channel 1 Data [10] | $2^{10}$ | Channel 1 MSN [2] | |
| Channel 1 Data [9] | $2^{9}$ | Channel 1 MSN [1] | |
| Channel 1 Data [8] | $2^{8}$ | Channel 1 MSN [0] | |
| Channel 1 Data [7] | $2^{7}$ | Channel 1 MidN [3] | Data 2 Channel 1 MidN |
| Channel 1 Data [6] | $2^{6}$ | Channel 1 MidN [2] | |
| Channel 1 Data [5] | $2^{5}$ | Channel 1 MidN [1] | |
| Channel 1 Data [4] | $2^{4}$ | Channel 1 MidN [0] | |
| Channel 1 Data [3] | $2^{3}$ | Channel 1 LSN [3] | Data 3 Channel 1 LSN |
| Channel 1 Data [2] | $2^{2}$ | Channel 1 LSN [2] | |
| Channel 1 Data [1] | $2^{1}$ | Channel 1 LSN [1] | |
| Channel 1 Data [0] | $2^{0}$ | Channel 1 LSN [0] | |
| Channel 2 Data [3] | $2^{3}$ | Channel 2 LSN [3] | Data 4 Channel 2 LSN |
| Channel 2 Data [2] | $2^{2}$ | Channel 2 LSN [2] | |
| Channel 2 Data [1] | $2^{1}$ | Channel 2 LSN [1] | |
| Channel 2 Data [0] | $2^{0}$ | Channel 2 LSN [0] | |
| Channel 2 Data [7] | $2^{7}$ | Channel 2 MidN [3] | Data 5 Channel 2 MidN |
| Channel 2 Data [6] | $2^{6}$ | Channel 2 MidN [2] | |
| Channel 2 Data [5] | $2^{5}$ | Channel 2 MidN [1] | |
| Channel 2 Data [4] | $2^{4}$ | Channel 2 MidN [0] | |
| Channel 2 Data [11] | $2^{11}$ | Channel 2 MSN [3] | Data 6 Channel 2 MSN |
| Channel 2 Data [10] | $2^{10}$ | Channel 2 MSN [2] | |
| Channel 2 Data [9] | $2^{9}$ | Channel 2 MSN [1] | |
| Channel 2 Data [8] | $2^{8}$ | Channel 2 MSN [0] | |

Figure H.1-1 explains the allocation of the SENT data nibbles to the fast channel 1, fast channel 2 and the status and communication nibble that conveys the serial message channel and the error flags.

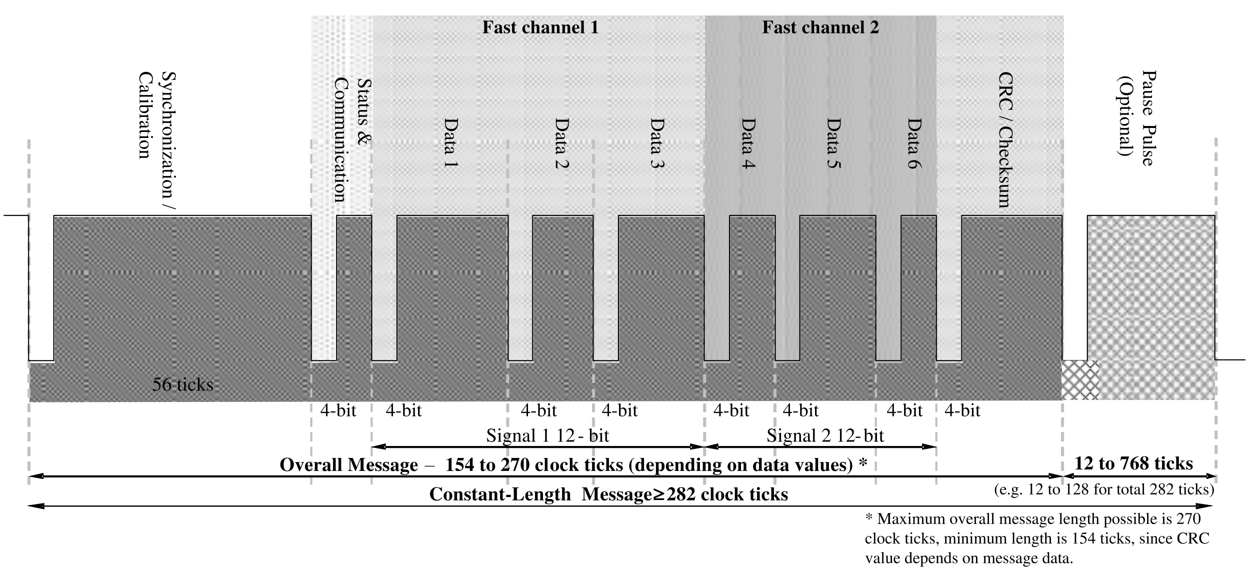



*Figure H.1-1 – Format and data channels of sensors with two fast channels*

## H.2 SENSORS WITH 12-BIT FAST CHANNEL 1 (5-NIBBLE FRAME)

For this format, channel 2 is not implemented and the SENT message shall contain only 3 data nibbles (5 nibble frame).

The data channels of sensors with one fast channel are defined as follows:

- Fast Channel 1
    - Data order:
        - data nibble 1: Channel 1 measurement data MSN
        - data nibble 2: Channel 1 measurement data MidN
        - data nibble 3: Channel 1 measurement data LSN
    - Error Flags
        - If the fast channel 1 measurement data is determined to be in error, bit 0 of the Status Nibble shall be set to 1 otherwise bit 0 shall be set to 0 (see Table H-2).
- Pause pulse (optional)

Figure H.2-1 explains the allocation of the SENT data nibbles to the fast channel 1and the status and communication nibble that conveys the serial message channel.

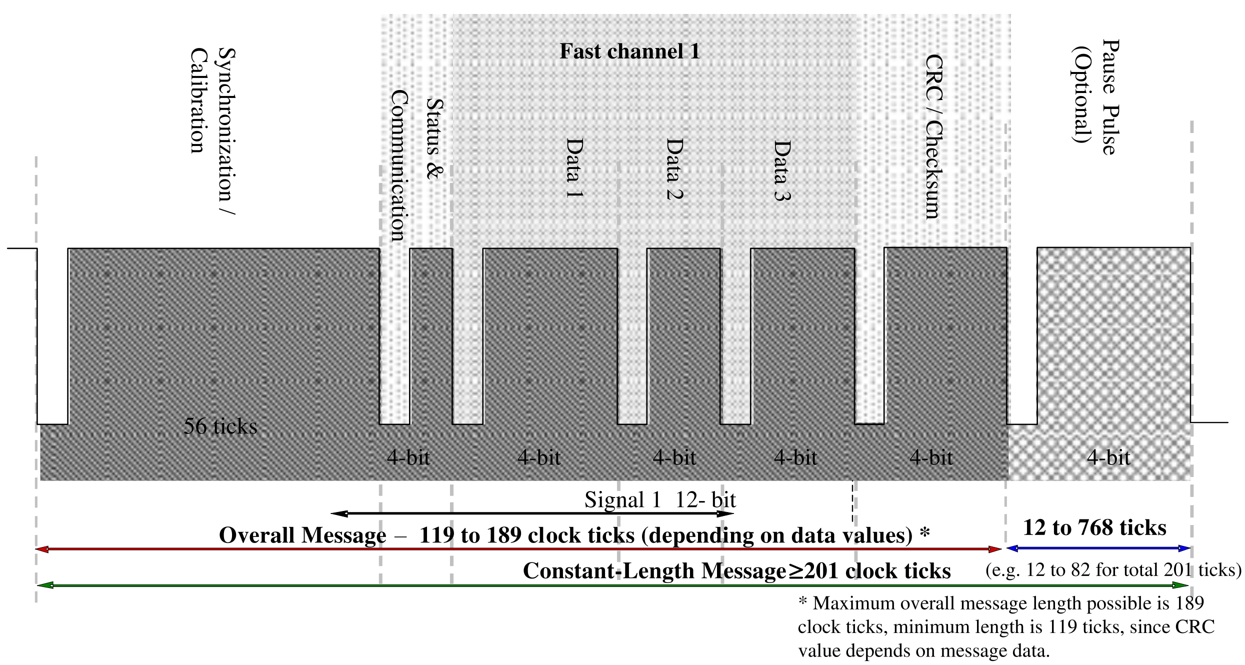



*Figure H.2-1 – Format and data channels of sensors with one fast channel*

## H.3 HIGH SPEED 12-BIT SENSORS

This section contains specific additional requirements for sensors whose application protocol is optimized with respect to the transmission of 12-bit data with very short SENT data frames (and a very high frame rate). For example the high-speed 12 bit message provides the possibility for a worst case message length < 500 µs for frames with a zero Status and Communications Nibble and < 550 µs for frames with a zero Status but serial message channel Communications enabled.

The high-speed 12-bit data format is fully compliant with the SENT data transmission, as described in the main body of the SENT standard. Thus, the calculation of the CRC is carried out as described in 5.4.

### H.3.1 High-Speed 12-bit Message – Format

The high-speed data format is characterized as follows:

- Data Nibbles
  - Sensor value is transmitted as a 12 bit value in 4 data nibbles where only values 0 to 7 are used
    - Data order:
      - data nibble 1: Most significant bits 11 – 9 (MSN)
      - data nibble 2: bits 8 – 6 (MidMSN)
      - data nibble 3: bits 5 – 3 (MidLSN)
      - data nibble 4: bits 2 – 0 (LSN)
- Status and Communication Nibble
  - The use of bit 0 as an error flag is optional. If the sensor is determined to be in error, bit 0 of the Status Nibble is set to 1, otherwise bit 0 is set to 0~~)~~.
  - Serial message channel: optional.
- Pause pulse (optional)

In order to provide a possibility to reduce the message length the use of the error flag (bit 0) and the serial message channel (bits 2 and 3) is optional. Figure H.3.1-1 illustrates the structure of a high-speed SENT frame.

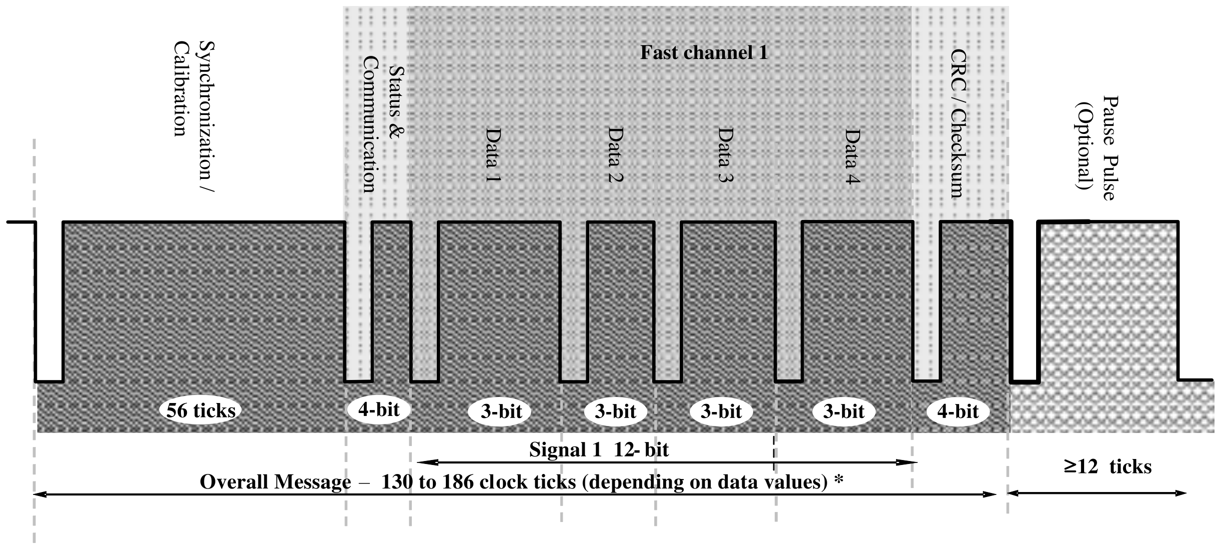



*Figure H.3.1-1 – Structure of the high-speed SENT frame*

Note, that the most significant bit of data nibbles 1, 2, 3 and 4 is always ‘0’. Thus, the high-speed SENT data format does not affect the calculation of the 4-bit CRC as described in 5.4.

*Table H.3.1-1 – Nibble and bit orders for high-speed 12-bit sensors*

| Sensor Data | Bit Weight | SENT Nibble Bits | SENT Nibble |
|---|---|---|---|
| S&C [3] Serial Data Channel | - | S&C [3] | Status and Communication |
| S&C [2] Serial Data Channel | - | S&C [2] | |
| S&C [1] Channel 2 Indicator | - | S&C [1] | |
| S&C [0] Channel 1 Indicator | - | S&C [0] | |
| Always ‘0’ | - | Channel 1 MSN [3] | Data 1 Channel 1 Most significant bits 11-9 |
| Channel 1 Data [11] | $2^{11}$ | Channel 1 MSN [2] | |
| Channel 1 Data [10] | $2^{10}$ | Channel 1 MSN [1] | |
| Channel 1 Data [9] | $2^{9}$ | Channel 1 MSN [0] | |
| Always ‘0’ | - | Channel 1 MidMSN [3] | Data 2 Channel 1 Bits 8-6 |
| Channel 1 Data [8] | $2^{8}$ | Channel 1 MidMSN [2] | |
| Channel 1 Data [7] | $2^{7}$ | Channel 1 MidMSN [1] | |
| Channel 1 Data [6] | $2^{6}$ | Channel 1 MidMSN [0] | |
| Always ‘0’ | - | Channel 1 MidLSN [3] | Data 3 Channel 1 Bits 5-3 |
| Channel 1 Data [5] | $2^{5}$ | Channel 1 MidLSN [2] | |
| Channel 1 Data [4] | $2^{4}$ | Channel 1 MidLSN [1] | |
| Channel 1 Data [3] | $2^{3}$ | Channel 1 MidLSN [0] | |
| Always ‘0’ | - | Channel 1 LSN [3] | Data 4 Channel 1 Least significant bits 2-0 |
| Channel 1 Data [2] | $2^{2}$ | Channel 1 LSN [2] | |
| Channel 1 Data [1] | $2^{1}$ | Channel 1 LSN [1] | |
| Channel 1 Data [0] | $2^{0}$ | Channel 1 LSN [0] | |

H.3.2 High-Speed 12-bit Message – Clock Accuracy Examples

- Transmission Clock Accuracy is ±10%
- Nominal clock tick time is 2.67 µs

### H.3.3 High-Speed 12-bit Message Implementation Examples

1) For messages limited to [7 7 7 6] the worst case message including the checksum is [7 6 7 5] with checksum = 15. For messages limited to [15 15 14] the worst case message including the checksum is [14 15 14] with checksum = 15. Length = 4 * 12 + 7 + 6 + 7 + 5 = 73 versus Length = 3 * 12 + 14 +15 + 14 = 79.

2) The nominal clock tick requirement is based on selecting the nominal clock tick so that the shortest clock tick period corresponds to the shortest clock tick allowed by the general SENT specification (i.e., 3 µs – 20% (2.4 µs)) and adjusting the nominal value accordingly. For example, a ±10% transmit clock tolerance yields a nominal clock tick of 2.4/0.9 = 2.67 µs and a longest clock tick of 2.4/0.9 *1.1 = 2.93 µs.

3) Worst case ±10% transmitter clock tolerance with no serial message channel information and no pause pulse is 493 µs. Including the pause pulse adds 12 ticks increasing the worst case message to 528 µs.

4) Worst case ±10% transmit clock tolerance with serial message channel information (maximum Status and Communication nibble value of 13) and no pause pulse is 531 µs. Including the pause pulse adds 12 ticks increasing the worst case message to 566 µs.

5) Transmitting the 12-bit value with four data nibbles. For 4 nibbles with a maximum of 3 non-zero bits the worst case message length is [7 7 7 7]. This is 5 ticks shorter than for 3 full nibbles Length = 4 * (12 + 7) = 76 versus Length = 3 * (12 + 15) = 81.

6) Using the reserved data ranges of Table E.1.3-1. Since data value of 4095 is only used if the sensor is in diagnostic mode, the receiver will not see a data value of 4095 limiting the message nibbles to [7 7 7 6].

## H.4 SECURE SINGLE SENSORS WITH 12-BIT FAST CHANNEL 1 AND 12-BIT SECURE SENSOR INFORMATION

The frame format for secure single sensors with 12-bit fast channel 1 for measurement data and 12-bit secure sensor information is defined as follows:

- Fast Channel 1
    - Data order:
        - data nibble 1: Channel 1 measurement data MSN
        - data nibble 2: Channel 1 measurement data MidN
        - data nibble 3: Channel 1 measurement data LSN
- Secure Sensor Information
        - 8- bit rolling counter 0 to 255 with rollover back to 0
        - Inverted copy of nibble 1 (calculated as 15 – nibble 1 value)
    - Data order:
        - data nibble 4: secure sensor counter MSN
        - data nibble 5: secure sensor counter LSN
        - date nibble 6: secure sensor inverted copy of nibble 1

- Error Flags
  - If the fast channel 1 measurement data is determined to be in error, bit 0 of the Status Nibble shall be set to 1 otherwise bit 0 shall be set to 0
- Pause pulse (optional)

Figure H.4-1 explains the allocation of the SENT data nibbles to the fast channel 1 secure sensor information and the status and communication nibble that conveys the serial message channel and the error flags.

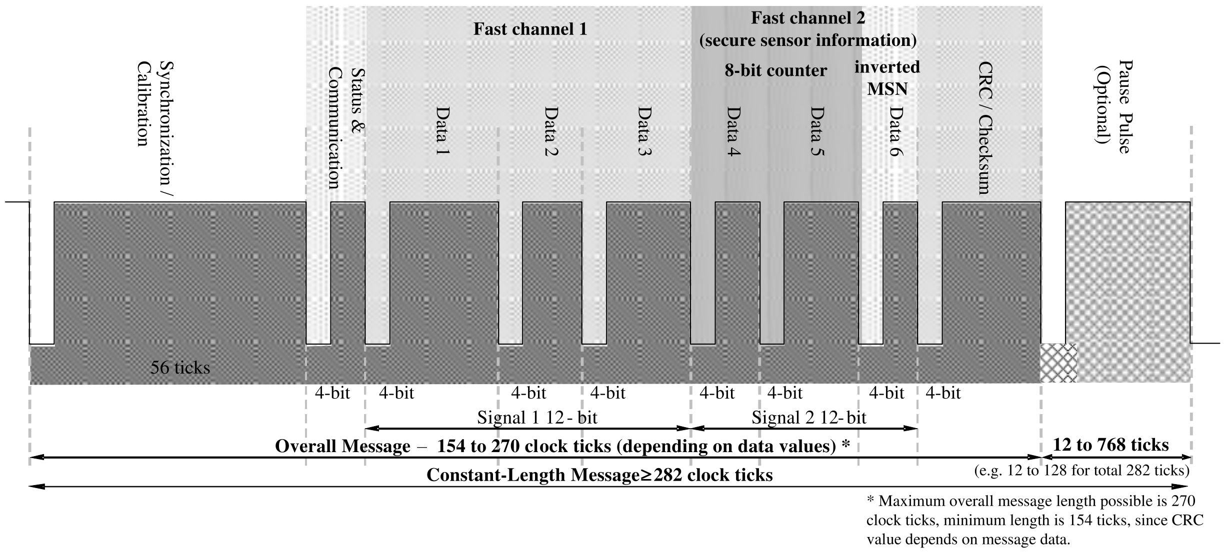



***Figure H.4-1 – Format and data channels of sensors with fast channel 1 and secure sensor information***

## H.5 SINGLE SENSORS

The frame format for single sensors with 12-bit fast channel 1 for measurement data and a 12-bit zero value is defined as follows (see Table H.5-1):

- Fast Channel 1
  - Data order:
    - data nibble 1: Channel 1 measurement data MSN
    - data nibble 2: Channel 1 measurement data MidN
    - data nibble 3: Channel 1 measurement data LSN
  - Error Flags
    - If the fast channel 1 measurement data is determined to be in error, bit 0 of the Status Nibble shall be set to 1 otherwise bit 0 shall be set to 0

- Empty nibbles
    - Data order:
        - data nibble 4: shall always be set to zero
        - data nibble 5: shall always be set to zero
        - date nibble 6: shall always be set to zero
- Pause pulse (optional)

***Table H.5-1 – Nibble and bit orders for single sensors with one 12-bit fast channel***

| Sensor Data | Bit Weight | SENT Nibble Bits | SENT Nibble |
|---|---|---|---|
| S&C [3] Serial Data Channel | - | S&C [3] | Status and Communication |
| S&C [2] Serial Data Channel | - | S&C [2] | |
| S&C [1] Channel 2 Indicator | - | S&C [1] | |
| S&C [0] Channel 1 Indicator | - | S&C [0] | |
| Channel 1 Data [11] | $2^{11}$ | Channel 1 MSN [3] | Data 1<br><br>Channel 1 MSN |
| Channel 1 Data [10] | $2^{10}$ | Channel 1 MSN [2] | |
| Channel 1 Data [9] | $2^{9}$ | Channel 1 MSN [1] | |
| Channel 1 Data [8] | $2^{8}$ | Channel 1 MSN [0] | |
| Channel 1 Data [7] | $2^{7}$ | Channel 1 MidN [3] | Data 2<br><br>Channel 1 MidN |
| Channel 1 Data [6] | $2^{6}$ | Channel 1 MidN [2] | |
| Channel 1 Data [5] | $2^{5}$ | Channel 1 MidN [1] | |
| Channel 1 Data [4] | $2^{4}$ | Channel 1 MidN [0] | |
| Channel 1 Data [3] | $2^{3}$ | Channel 1 LSN [3] | Data 3<br><br>Channel 1 LSN |
| Channel 1 Data [2] | $2^{2}$ | Channel 1 LSN [2] | |
| Channel 1 Data [1] | $2^{1}$ | Channel 1 LSN [1] | |
| Channel 1 Data [0] | $2^{0}$ | Channel 1 LSN [0] | |
| Always ‘0’ | - | Empty nibble LSN [3] | Data 4<br><br>Empty nibble LSN |
| Always ‘0’ | - | Empty nibble LSN [2] | |
| Always ‘0’ | - | Empty nibble LSN [1] | |
| Always ‘0’ | - | Empty nibble LSN [0] | |
| Always ‘0’ | - | Empty nibble MidN [3] | Data 5<br><br>Empty nibble MidN |
| Always ‘0’ | - | Empty nibble MidN [2] | |
| Always ‘0’ | - | Empty nibble MidN [1] | |
| Always ‘0’ | - | Empty nibble MidN [0] | |
| Always ‘0’ | - | Empty nibble MSN [3] | Data 6<br><br>Empty nibble MSN |
| Always ‘0’ | - | Empty nibble MSN [2] | |
| Always ‘0’ | - | Empty nibble MSN [1] | |
| Always ‘0’ | - | Empty nibble MSN [0] | |

Figure H.5-1 explains the allocation of the SENT data nibbles to the fast channel 1, the empty data nibbles and the status and communication nibble that conveys the serial message channel and the error flags.

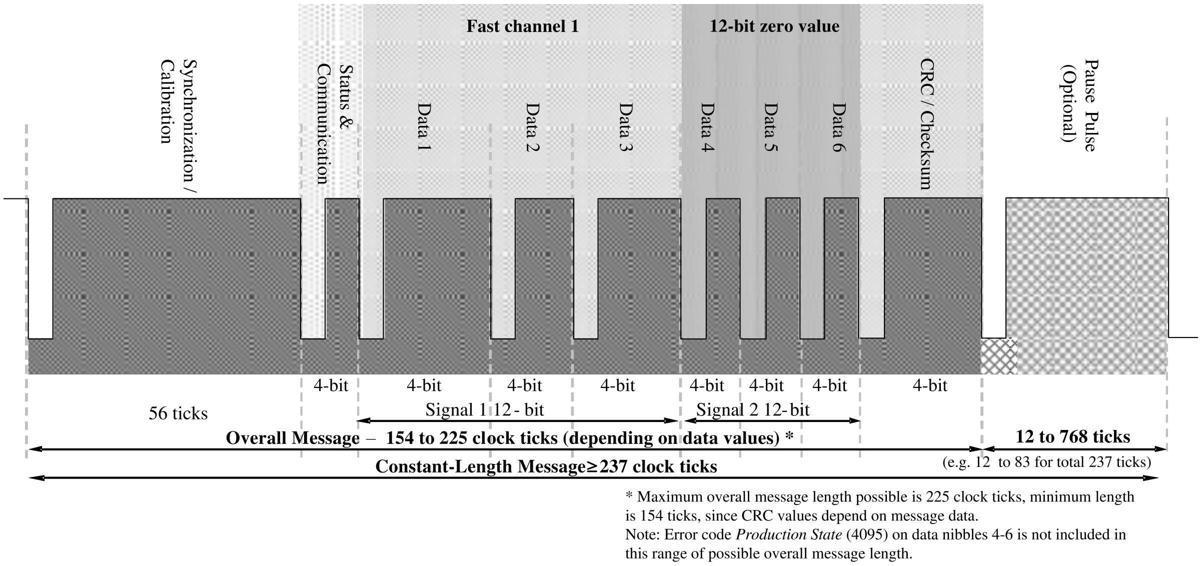



***Figure H.5-1 – Format and data channels of single sensors with one 12-bit fast channel***

## H.6 TWO FAST CHANNELS WITH 14-BIT FAST CHANNEL 1 AND 10-BIT FAST CHANNEL 2 (6 DATA NIBBLES)

The data channels of sensors with 14-bit fast channel 1 and 10-bit fast channel 2 are defined as follows (see Table H.6-1):

- Fast Channel 1
    - Data order:
        - data nibble 1: Channel 1 measurement data MSN
        - data nibble 2: Channel 1 measurement data MidMSN
        - data nibble 3: Channel 1 measurement data MidLSN
        - data nibble 4: Channel 1 measurement data LSBs 1, 0 (nibble bit 3 and bit 2)
    - Error Flags
        - If the fast channel 1 measurement data is determined to be in error, bit 0 of the Status Nibble shall be set to 1 otherwise bit 0 shall be set to 0 (see Table H.6-1)
- Fast Channel 2
    - Data order:
        - data nibble 4: Channel 2 measurement data LSBs 1, 0 (nibble bit 1 and bit 0)
        - data nibble 5: Channel 2 measurement data MidN
        - data nibble 6: Channel 2 measurement data MSN
    - The 2 LSBs are mapped to bit 1 and bit 0 of the data nibble 4. The data vector (ordered MSB to LSB) of the remaining 8-bits is first divided into 4-bit groups corresponding to the nibbles. The nibble order is then reversed and mapped to data nibble 5 and 6

- Error Flags

    - If the fast channel 2 measurement data is determined to be in error, bit 1 of the Status Nibble shall be set to 1 otherwise bit 1 shall be set to 0 (see Table H.6-1)

- If channel 2 is not used to transmit sensor data, data nibbles 4 (only nibble bit 1 and bit 0 ), 5 and 6 shall be all set to 0

- Pause pulse (optional)

***Table H.6-1 – Nibble and bit orders for sensors with 14-bit fast channel 1 and 10-bit fast channel 2***

| Sensor Data | Bit Weight | SENT Nibble Bits | SENT Nibble |
|---|---|---|---|
| S&C [3] Serial Data Channel | - | S&C [3] | Status<br>and Communication |
| S&C [2] Serial Data Channel | - | S&C [2] | |
| S&C [1] Channel 2 Indicator | - | S&C [1] | |
| S&C [0] Channel 1 Indicator | - | S&C [0] | |
| Channel 1 Data [13] | $2^{13}$ | Channel 1 MSN [3] | Data 1<br><br>Channel 1<br>MSN |
| Channel 1 Data [12] | $2^{12}$ | Channel 1 MSN [2] | |
| Channel 1 Data [11] | $2^{11}$ | Channel 1 MSN [1] | |
| Channel 1 Data [10] | $2^{10}$ | Channel 1 MSN [0] | |
| Channel 1 Data [9] | $2^{9}$ | Channel 1 MidMSN [3] | Data 2<br><br>Channel 1<br>MidMSN |
| Channel 1 Data [8] | $2^{8}$ | Channel 1 MidMSN [2] | |
| Channel 1 Data [7] | $2^{7}$ | Channel 1 MidMSN [1] | |
| Channel 1 Data [6] | $2^{6}$ | Channel 1 MidMSN [0] | |
| Channel 1 Data [5] | $2^{5}$ | Channel 1 MidLSN [3] | Data 3<br><br>Channel 1<br>MidLSN |
| Channel 1 Data [4] | $2^{4}$ | Channel 1 MidLSN [2] | |
| Channel 1 Data [3] | $2^{3}$ | Channel 1 MidLSN [1] | |
| Channel 1 Data [2] | $2^{2}$ | Channel 1 MidLSN [0] | |
| Channel 1 Data [1] | $2^{1}$ | Channel 1, 2 LSN [3] | Data 4<br><br>Channel 1, 2<br>LSN |
| Channel 1 Data [0] | $2^{0}$ | Channel 1, 2 LSN [2] | |
| Channel 2 Data [1] | $2^{1}$ | Channel 1, 2 LSN [1] | |
| Channel 2 Data [0] | $2^{0}$ | Channel 1, 2 LSN [0] | |
| Channel 2 Data [5] | $2^{5}$ | Channel 2 MidN [3] | Data 5<br><br>Channel 2<br>MidN |
| Channel 2 Data [4] | $2^{4}$ | Channel 2 MidN [2] | |
| Channel 1 Data [3] | $2^{3}$ | Channel 2 MidN [1] | |
| Channel 1 Data [2] | $2^{2}$ | Channel 2 MidN [0] | |
| Channel 2 Data [9] | $2^{9}$ | Channel 2 MSN [3] | Data 6<br><br>Channel 2<br>MSN |
| Channel 2 Data [8] | $2^{8}$ | Channel 2 MSN [2] | |
| Channel 2 Data [7] | $2^{7}$ | Channel 2 MSN [1] | |
| Channel 2 Data [6] | $2^{6}$ | Channel 2 MSN [0] | |

Figure H.6-1 explains the allocation of the SENT data nibbles to the fast channel 1, fast channel 2 and the status and communication nibble that conveys the serial message channel and the error flags.

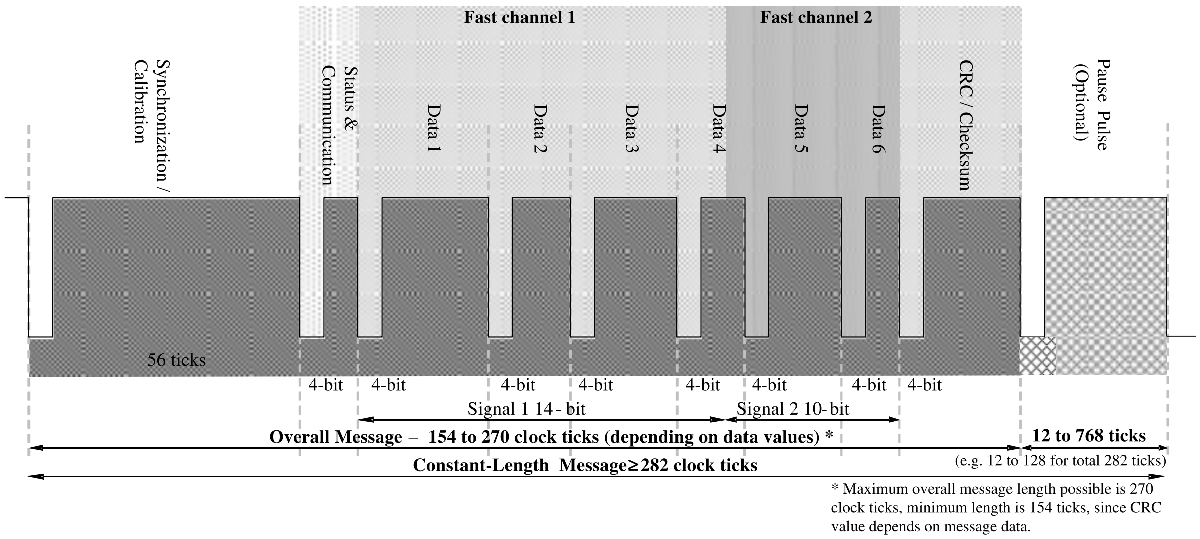



***Figure H.6-1 – Format and data channels of sensors with 14-bit fast channel 1 and 10-bit fast channel 2***

## H.7 TWO FAST CHANNELS WITH 16-BIT FAST CHANNEL 1 AND 8-BIT FAST CHANNEL 2 (6 DATA NIBBLES)

The data channels of sensors with 16-bit fast channel 1 and 8-bit fast channel 2 are defined as follows (see Table H.7-1):

- Fast Channel 1
  - Data order:
    - data nibble 1: Channel 1 measurement data MSN
    - data nibble 2: Channel 1 measurement data MidMSN
    - data nibble 3: Channel 1 measurement data MidLSN
    - data nibble 4: Channel 1 measurement data LSN
  - Error Flags
    - If the fast channel 1 measurement data is determined to be in error, bit 0 of the Status Nibble shall be set to 1 otherwise bit 0 shall be set to 0 (see Table H.7-1)
- Fast Channel 2
  - Data order:
    - data nibble 5: Channel 2 measurement data MSN
    - data nibble 6: Channel 2 measurement data LSN
  - Data vector (ordered MSB to LSB) is first divided into 4-bit groups corresponding to the nibbles. The nibble order of Channel 2 is then reversed.

- Error Flags
    - If the fast channel 2 measurement data is determined to be in error, bit 1 of the Status Nibble shall be set to 1 otherwise bit 1 shall be set to 0 (see Table H.7-1)
  - If channel 2 is not used to transmit sensor data, data nibbles 5 and 6 shall be all set to 0
- Pause pulse (optional)

***Table H.7-1 – Nibble and bit orders for sensors with 16-bit fast channel 1 and 8-bit fast channel 2***

| Sensor Data | Bit Weight | SENT Nibble Bits | SENT Nibble |
|---|---|---|---|
| S&C [3] Serial Data Channel | - | S&C [3] | Status and Communication |
| S&C [2] Serial Data Channel | - | S&C [2] | |
| S&C [1] Channel 2 Indicator | - | S&C [1] | |
| S&C [0] Channel 1 Indicator | - | S&C [0] | |
| Channel 1 Data [15] | $2^{15}$ | Channel 1 MSN [3] | Data 1<br><br>Channel 1 MSN |
| Channel 1 Data [14] | $2^{14}$ | Channel 1 MSN [2] | |
| Channel 1 Data [13] | $2^{13}$ | Channel 1 MSN [1] | |
| Channel 1 Data [12] | $2^{12}$ | Channel 1 MSN [0] | |
| Channel 1 Data [11] | $2^{11}$ | Channel 1 MidMSN [3] | Data 2<br><br>Channel 1 MidMSN |
| Channel 1 Data [10] | $2^{10}$ | Channel 1 MidMSN [2] | |
| Channel 1 Data [9] | $2^{9}$ | Channel 1 MidMSN [1] | |
| Channel 1 Data [8] | $2^{8}$ | Channel 1 MidMSN [0] | |
| Channel 1 Data [7] | $2^{7}$ | Channel 1 MidLSN [3] | Data 3<br><br>Channel 1 MidLSN |
| Channel 1 Data [6] | $2^{6}$ | Channel 1 MidLSN [2] | |
| Channel 1 Data [5] | $2^{5}$ | Channel 1 MidLSN [1] | |
| Channel 1 Data [4] | $2^{4}$ | Channel 1 MidLSN [0] | |
| Channel 1 Data [3] | $2^{3}$ | Channel 1 LSN [3] | Data 4<br><br>Channel 1 LSN |
| Channel 1 Data [2] | $2^{2}$ | Channel 1 LSN [2] | |
| Channel 1 Data [1] | $2^{1}$ | Channel 1 LSN [1] | |
| Channel 1 Data [0] | $2^{0}$ | Channel 1 LSN [0] | |
| Channel 2 Data [3] | $2^{3}$ | Channel 2 LSN [3] | Data 5<br><br>Channel 2 LSN |
| Channel 2 Data [2] | $2^{2}$ | Channel 2 LSN [2] | |
| Channel 2 Data [1] | $2^{1}$ | Channel 2 LSN [1] | |
| Channel 2 Data [0] | $2^{0}$ | Channel 2 LSN [0] | |
| Channel 2 Data [7] | $2^{7}$ | Channel 2 MSN [3] | Data 6<br><br>Channel 2 MSN |
| Channel 2 Data [6] | $2^{6}$ | Channel 2 MSN [2] | |
| Channel 2 Data [5] | $2^{5}$ | Channel 2 MSN [1] | |
| Channel 2 Data [4] | $2^{4}$ | Channel 2 MSN [0] | |

Figure H.7-1 explains the allocation of the SENT data nibbles to the fast channel 1, fast channel 2 and the status and communication nibble that conveys the serial message serial channel and the error flags.

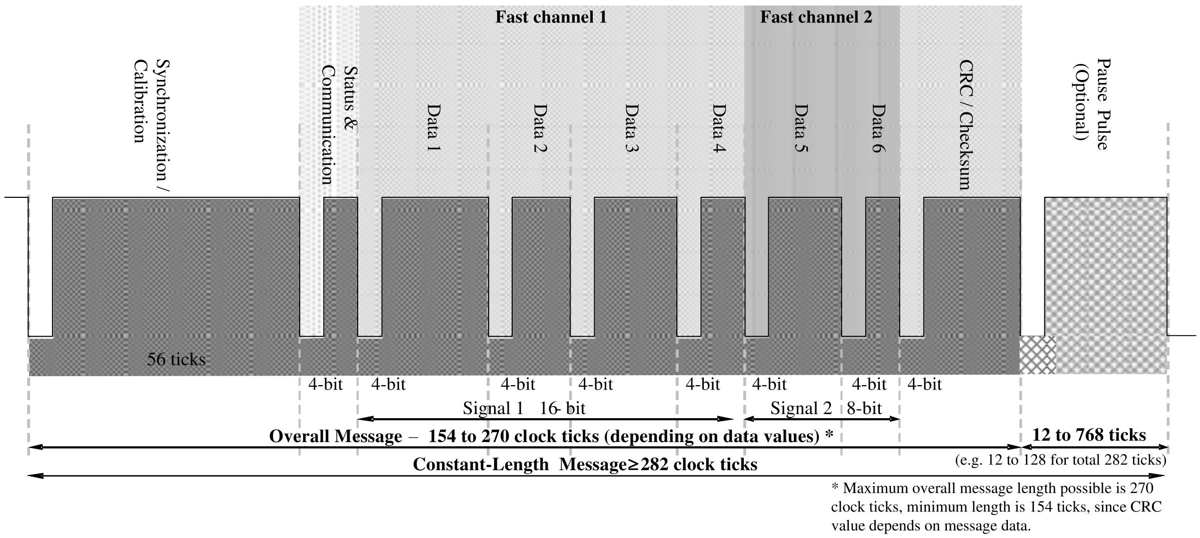



*Figure H.7-1 – Format and data channels of sensors with 16-bit fast channel 1 and 8-bit fast channel 2*

# APPENDIX I - ALTERNATIVE RECEIVER CIRCUIT

## I.1 ALTERNATIVE RECEIVER CIRCUIT

A circuit can be seen as equivalent to the reference receiver circuit (Figure 6.3.2-2), when the signal pulse shape of any transmitter connected to the alternative circuit, is nearly the same like it would be connected to the reference receiver. The signal pulse shape of the transmitter is defined by the parameters a, b, e1, f and h of Table 6.3.1-1.

***Table I.1-1 – Allowed pulse shape deviation between reference receiver and alternative circuit***

| Parameter | | Deviation | | Units |
|---|---|---|---|---|
| | | Min | Max | |
| a | $V_{OL}$ | -5 | 0 | % |
| b | $V_{OH}$ | 0 | +5 | % |
| e | $T_{FALL}$ | -5 | 0 | % |
| f | $T_{RISE}$ | -5 | 0 | % |
| h | $T_{STABLE}$ | 0 | +5 | % |

Since the signal pulse shape depends also on the receiver input impedance, the input impedance of the alternative receiver should match to the input impedance of the recommended circuit.

In the range of 0Hz to 100kHz pulse shapes (voltages) are compared according to Table I.1-1

Above 100kHz the impedance of the alternative circuit must match (real and imaginary part) within ±5%.

## I.2 HOW TO FIND AN ALTERNATIVE RECEIVER CIRCUIT

Since the behavior of the alternative circuit shall be similar to reference circuit for any transmitter circuit, a virtual worst case transmitter must be available. The two major classes of transmitters are voltage controlled pulse shapers and current controlled pulse shapers. Since the latter type is more dependent on the receiver input impedance, an ideal pulse current source with constant positive/negative output current during a SENT pulse rising/falling edge, and voltage limitation between $V_{OL}$ and $V_{OH}$ shall be used to create the worst case transmitter (model).

To determine the component values of the worst case transmitter, the worst case values for the reference circuit must be used: $R_{PULL\text{-}UP}$ min, $C_{INPUT}$ max, $R_{tau1}$ min, $C_{tau1}$ max, $R_f$ min, $C_f$ max. given in Table 6.3.1-2. For best case it is the opposite: $R_{PULL\text{-}UP}$ max, $C_{INPUT}$ min, $R_{tau1}$ max, $C_{tau1}$ min, $R_f$ max, $C_F$ min

Then the reference circuit must be connected to the transmitter (model) and the transmitter parameters must be adjusted till the pulse shape meets exactly the limits in Table 6.3.1-1 for parameters a, b, e, f, and h.

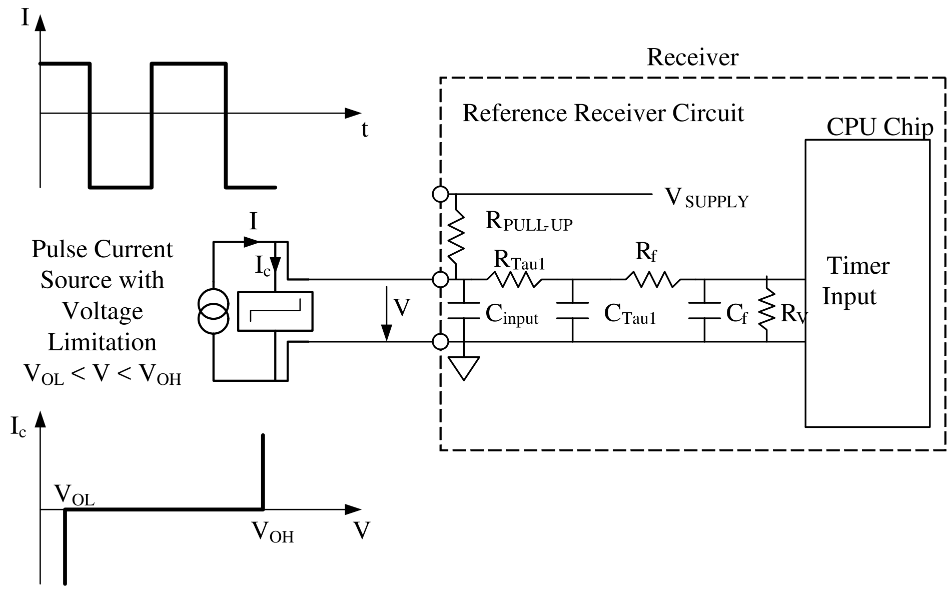



***Figure I.2-1 – Worst case transmitter with reference circuit topology***

After the worst case transmitter was created, it must be connected to the alternative circuit. The parameters of the alternative circuit must be adjusted in that way, that the worst case meets the requirements of Table 6.3.1-1 for parameters a, b, e, f, and h. By varying the parameters of the alternative circuit within the tolerance room (also best case), the pulse shape shall result in the same behavior (see Table I.1-1) as the reference circuit, when its components are varied in the tolerance parameter room of Table 6.3.2-2.

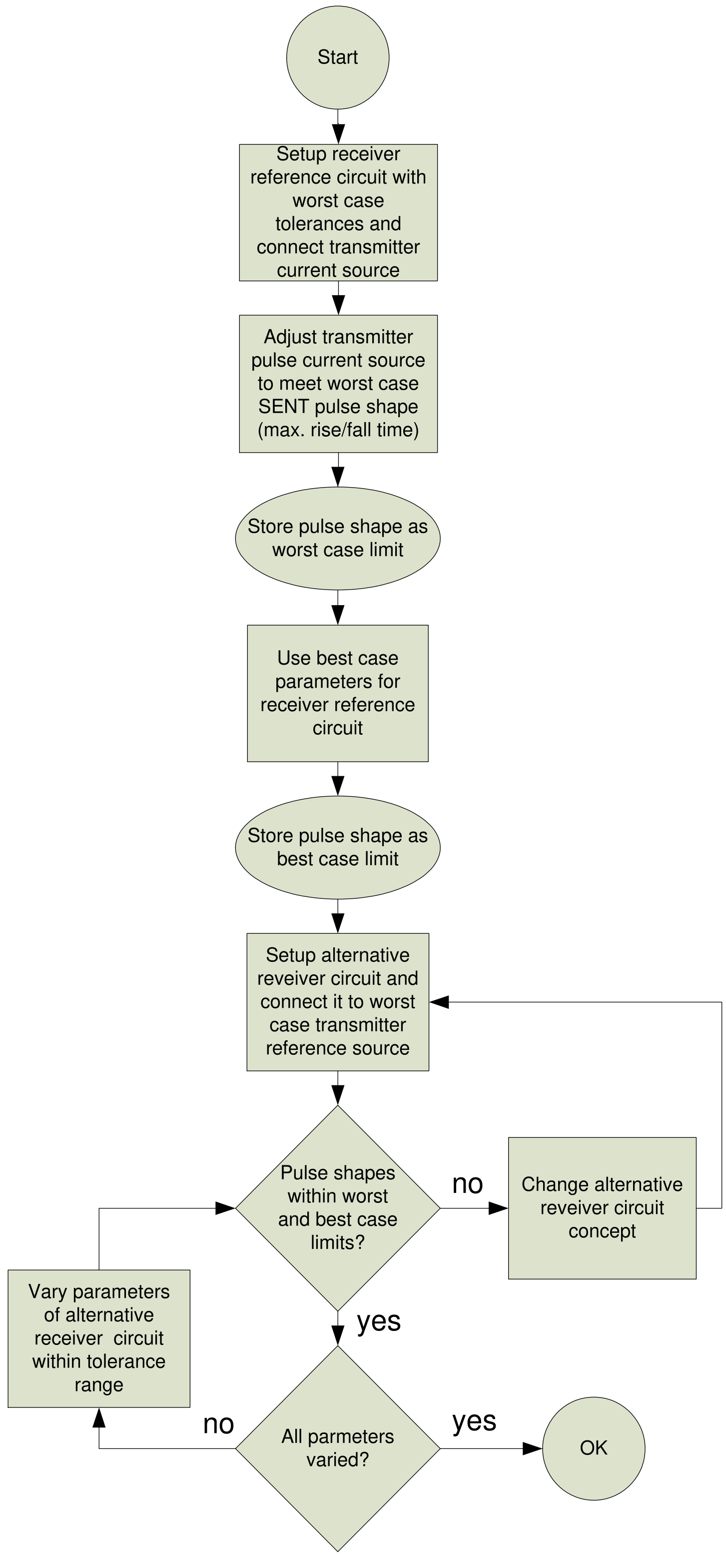



***Figure I.2-2 – Procedure to guarantee that pulse shapes of alternative circuit will always stay within the limits given by the tolerances of the reference circuit***