

# SAE J2716 Gateway

CAN bus, RS-232 and USB  
Communication Protocol Specification



## Changes

| Date       | Firmware Version | Description                                                                                                                                                                                        | By                |
|------------|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|
| 29.12.2025 | 1.11             | Tick time range fixed in 2.1                                                                                                                                                                       | Miroslav Machacek |
| 10.4.2025  | 1.11             | SPC field removed                                                                                                                                                                                  | Miroslav Machacek |
| 24.11.2023 | 1.11             | Clarifications – CRC, Pulse Pause in RX                                                                                                                                                            | Karel Hevessy     |
| 8.8.2022   | 1.9              | Possibility to transmit a rolling counter (RCNT) in Fast messages.                                                                                                                                 | Karel Hevessy     |
| 15.6.2021  | 1.7              | Possibility of CRC fault injection for transmission of slow messages (2.1).<br>Data nibble order for analogue outputs can be set (2.7).<br>Tick time (UT) can be set down to 0.5 us                | Miroslav Machacek |
| 9.9.2020   | 1.6              | SENT-USB gateway added.<br>Serial line baud rate can be changed for SENT-USB and SENT-RS232 (3.1.1).<br>Data nibbles can be swapped within a byte (see <i>SwapFastDataNibbles</i> in 2.1 and 2.2). | Miroslav Machacek |
| 4.2.2020   | 1.5              | Cyclic message buffers for transmission of Slow Messages (2.6).                                                                                                                                    | Miroslav Machacek |

|           |     |                                                                                 |                   |
|-----------|-----|---------------------------------------------------------------------------------|-------------------|
|           |     | SENT-CAN: possible to transmit SENT RX messages over separated CAN Ids (3.2.1). |                   |
| 26.8.2019 | 1.4 | Negative values for Analogue Multiplier and Offset are clarified                | Miroslav Machacek |
| 26.4.2019 | 1.3 | Minimal Pause Pulse lowered to 282 ticks                                        | Miroslav Machacek |
| 3.4.2019  | 1.2 | Pause Pulse clarified<br>Communication examples added                           | Miroslav Machacek |
| 7.1.2019  | 1.1 | Configuration of analogue outputs<br>CAN Alternative configuration removed      | Miroslav Machacek |
| 15.8.2018 | 1.0 | Clarifications                                                                  | Miroslav Machacek |
| 14.2.2018 | 1.0 | Initial Version                                                                 | Miroslav Machacek |

## Table of Contents

|                                            |    |
|--------------------------------------------|----|
| 1. Introduction.....                       | 3  |
| 1.1. Nibble Endianness and Start Bit ..... | 4  |
| 2. Configuration.....                      | 5  |
| 2.1. SENT Channel Configuration .....      | 5  |
| 2.2. SENT Data Frame .....                 | 6  |
| 2.3. SENT Slow Data Frame .....            | 7  |
| 2.4. SENT Error Frame .....                | 8  |
| 2.5. SENT Slow Error Frame.....            | 8  |
| 2.6. SENT Slow Message TX Buffers .....    | 9  |
| 2.7. Analogue Output Configuration .....   | 9  |
| 2.8. Analogue Output Limits.....           | 10 |
| 2.9. Rolling Counter Configuration.....    | 10 |
| 3. Communication Protocol.....             | 11 |
| 3.1. RS-232 and USB .....                  | 11 |
| 3.1.1. Serial Line Baud Rate .....         | 11 |
| 3.2. CAN.....                              | 11 |
| 3.2.1. CAN Configuration.....              | 12 |
| 3.2.2. CAN IDs for SENT RX Messages .....  | 12 |
| 4. Messages.....                           | 13 |
| 4.1. To Gateway.....                       | 13 |
| 4.2. From Gateway .....                    | 15 |
| 5. Command Examples .....                  | 16 |
| 5.1. Transmission over SENT Channel .....  | 16 |
| 5.1.1. Fast Message .....                  | 16 |
| 5.1.2. Slow Message .....                  | 16 |
| 5.1.3. Slow Messages - Multiplexing .....  | 16 |

|                                          |    |
|------------------------------------------|----|
| 6. Communication Examples .....          | 17 |
| 6.1. RS-232 and USB .....                | 17 |
| 6.2. CAN bus .....                       | 17 |
| 6.2.1. SENT Channel 1 Configuration..... | 17 |
| 6.2.2. Start Channel 1 .....             | 18 |
| 6.2.3. Stop Channel 1.....               | 18 |
| 6.2.4. Start Both Channels.....          | 18 |
| 6.2.5. Transmit on Channel 1 .....       | 18 |
| 7. Contact .....                         | 19 |

## 1. Introduction

This document describes the communication protocol used by the SENT-CAN, SENT-RS232 and SENT-USB gateways so that the device can be integrated into a user's system.

The SENT gateway features two independent SENT channels and allows the user to configure SENT parameters, receive and transmit SENT frame including Short Serial and Enhanced Serial formats via the communication protocol. The device can also automatically transmit Slow messages with different Message Id over a SENT channel with the help of message buffers for slow message.

The SENT-CAN and SENT-RS232 gateway feature two analogue output channels (12-bit DAC) with precise internal voltage reference (range 0-4.095 V). Each analogue channel (AO1, AO2) can be mapped either on SENT1 or SENT2 channel. The conversion settings is configurable by the user – start bit, bit length, nibble order, linear transfer function: Multiplier, Offset, Min/Max voltage limits.

The SENT-CAN device can be configured to transmit SENT Fast and Slow Data frames over separated CAN identifiers, allowing a measurement system to easily process the data.

There are two types of messages - configuration and data. Configuration can be saved and load into a non-volatile memory (EEPROM).

Advanced functions:

- TX buffers for multiple slow messages
- Slow message CRC fault injection
- Rolling counter transmission in Fast messages

### 1.1. Nibble Endianness and Start Bit

Throughout this document the nibble endianness specifies the nibble order (E.g. not bit order). The following example depicts the endianness on 6 data nibbles when there is a variable *FOO*: Length=12-bit, Start bit=0.

Note that the Start bit order also differs between little and big-endian formats.

#### Little-endian

| Nibble          | 0 |   |   |   | 1 |   |   |   | 2 |   |    |    | 3  |    |    |    | 4  |    |    |    | 5  |    |    |    |
|-----------------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit Position    | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 |
| <i>FOO bits</i> | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |    |    |    |    |    |    |    |    |    |    |    |    |

#### Big-endian

| Nibble          | 0  |    |    |    | 1  |    |    |    | 2  |    |    |    | 3 |   |    |    | 4 |   |   |   | 5 |   |   |   |
|-----------------|----|----|----|----|----|----|----|----|----|----|----|----|---|---|----|----|---|---|---|---|---|---|---|---|
| Bit Position    | 20 | 21 | 22 | 23 | 16 | 17 | 18 | 19 | 12 | 13 | 14 | 15 | 8 | 9 | 10 | 11 | 4 | 5 | 6 | 7 | 0 | 1 | 2 | 3 |
| <i>FOO bits</i> |    |    |    |    |    |    |    |    |    |    |    |    | 8 | 9 | 10 | 11 | 4 | 5 | 6 | 7 | 0 | 1 | 2 | 3 |

## 2. Configuration

The gateway is configured by the communication protocol described below. The configuration is split into virtual registers that are stored in RAM.

If the user wants to save the configuration from RAM into a non-volatile memory, he transmits SAVE\_CONFIGURATION message to the device. Similarly, LOAD\_CONFIGURATION message is used to load a configuration into registers in RAM. If a valid configuration is present in a non-volatile memory, it is automatically loaded on power-up.

The registers / data structures are described below.

### 2.1. SENT Channel Configuration

| Byte | Bit | Name                                                                                                                                                                          | Length in bits | Description                                                                      | Range       | Default value |
|------|-----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|----------------------------------------------------------------------------------|-------------|---------------|
| 0    | 0   | AutoStart                                                                                                                                                                     | 1              | 1 = Channel is started on power-up                                               |             | 1             |
|      | 1   | Reserved                                                                                                                                                                      | 1              |                                                                                  |             |               |
|      | 2   | Direction                                                                                                                                                                     | 1              | 0=TX, 1=RX                                                                       |             | 1             |
|      | 3,4 | CrcMode                                                                                                                                                                       | 2              | 0 = hwCRC off, 1 = hwCRC on, 2 =swCRC                                            |             | 1             |
|      | 5-7 | DataNibbleCount                                                                                                                                                               | 3              | Number of data nibbles                                                           | 1 - 6       | 6             |
| 1    | 0   | PulsePauseEnabled<br><i>see PulsePauseFramePeriod<br/>note: For RX, when enabled, framing errors that may originate from detecting two successive Sync Pulses are ignored</i> | 1              | 1 = Transmit frames with pulse pause                                             |             | 0             |
|      | 1,2 | RxForwardMode / TxEchoMode                                                                                                                                                    | 2              | 0 = Fast as possible / No echo (for Tx)<br>1 = Fixed 100ms<br>2 = On change + 1s |             | 1             |
|      | 3,4 | SlowChannelMode                                                                                                                                                               | 2              | 0 = Fast Channel Only<br>1 = Short Serial<br>2 = Enhanced Serial                 |             | 0             |
|      | 5   | Reserved                                                                                                                                                                      | 1              |                                                                                  |             |               |
|      | 6   | SlowChannelTx_crcFault<br><i>relevant when SlowChannelMode is 1 or 2</i>                                                                                                      | 1              | TX Slow message CRC fault injection<br>0 = Disabled<br>1 = Enabled               |             | 0             |
|      | 7   | Reserved                                                                                                                                                                      | 1              |                                                                                  |             |               |
|      | 2,3 | UnitTime                                                                                                                                                                      | 16             | Tick Time [hundreds of ns] 3us= 300                                              | 0.5 - 90 us | 300           |
| 4,5  |     | PulsePauseFramePeriod<br>valid if <i>PulsePauseEnabled</i> is set<br><i>note: valid for TX only</i>                                                                           | 16             | TX SENT frame length in us                                                       | see below   | 0             |

|   |   |                                                                                         |   |                                                                                  |  |   |
|---|---|-----------------------------------------------------------------------------------------|---|----------------------------------------------------------------------------------|--|---|
| 6 | 0 | SwapFastDataNibbles<br>(Data nibble order within a byte – nibble endianness)<br>See 2.2 | 1 | 0 = Data nibble order is unchanged<br>1 = Data nibbles are swapped within a byte |  | 0 |
|---|---|-----------------------------------------------------------------------------------------|---|----------------------------------------------------------------------------------|--|---|

When CrcMode is set to hwCRC off (value 0), incorrect CRC in received Fast Frame does not generate an error. CRC and CRC Calculated fields in the forwarded frame are still valid (CRC is calculated by the specification).

A SENT channel has to be stopped before writing configuration.

TX Pause Pulse Range:

| Number of Data Nibbles | Min. Frame Time [ticks] | Max. Frame Time [ticks] |
|------------------------|-------------------------|-------------------------|
| 1                      | 147                     | 860                     |
| 2                      | 174                     | 872                     |
| 3                      | 201                     | 884                     |
| 4                      | 228                     | 896                     |
| 5                      | 255                     | 908                     |
| 6                      | 282                     | 920                     |

FrameTime = Tframe (us) / Ttick

$120 + 27 \cdot N \leq \text{FrameTime} \leq 848 + 12 \cdot N$

N is number of data nibbles: 1 to 6

## 2.2. SENT Data Frame

Depending on SENT channel configuration field *SwapFastDataNibbles*, the data nibbles can be swapped within a byte. Swapping is useful when the user needs to parse 12-bit values based on a DBC file.

It should be noted that *SwapFastDataNibbles* configuration bit is considered for both directions. E.g. not only when a SENT frame is forwarded onto CAN/RS-232/USB but also when the user writes TX buffers for SENT transmission (Message Ids 41 and 51).

When *SwapFastDataNibbles* is 0:

| Byte   | Bit | Name            | Length in bits | Description                                                                                                            |
|--------|-----|-----------------|----------------|------------------------------------------------------------------------------------------------------------------------|
| 0      | 0-3 | StatusNibble    | 4              |                                                                                                                        |
|        | 4-7 | DataNibbleCount | 4              | Number of following data nibbles                                                                                       |
| 1      | 0-3 | DataNibble 0    | 4              |                                                                                                                        |
|        | 4-7 | DataNibble 1    | 4              |                                                                                                                        |
| 2      | 0-3 | DataNibble 2    | 4              | Byte not sent if NibbleCount<3                                                                                         |
|        | 4-7 | DataNibble 3    | 4              |                                                                                                                        |
| 3      | 0-3 | DataNibble 4    | 4              | Byte not sent if NibbleCount<5                                                                                         |
|        | 4-7 | DataNibble 5    | 4              |                                                                                                                        |
| 4(2,3) | 0-3 | CRC             | 4              | Byte 2 if NibbleCount<3, byte 3 if NibbleCount<5<br><br>for SENT TX frame:<br>- SwCRC: Field CRC is sent onto the SENT |
|        | 4-7 | CRC Calculated  | 4              |                                                                                                                        |

|  |  |  |  |                                            |
|--|--|--|--|--------------------------------------------|
|  |  |  |  | bus<br>- Otherwise: This field is ignored. |
|--|--|--|--|--------------------------------------------|

When *SwapFastDataNibbles* is 1:

| Byte   | Bit | Name            | Length in bits | Description                                                                                                                                                       |
|--------|-----|-----------------|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0      | 0-3 | StatusNibble    | 4              |                                                                                                                                                                   |
|        | 4-7 | DataNibbleCount | 4              | Number of following data nibbles                                                                                                                                  |
| 1      | 0-3 | DataNibble 1    | 4              |                                                                                                                                                                   |
|        | 4-7 | DataNibble 0    | 4              |                                                                                                                                                                   |
| 2      | 0-3 | DataNibble 3    | 4              | Byte not sent if NibbleCount<3                                                                                                                                    |
|        | 4-7 | DataNibble 2    | 4              |                                                                                                                                                                   |
| 3      | 0-3 | DataNibble 5    | 4              | Byte not sent if NibbleCount<5                                                                                                                                    |
|        | 4-7 | DataNibble 4    | 4              |                                                                                                                                                                   |
| 4(2,3) | 0-3 | CRC             | 4              | Byte 2 if NibbleCount<3, byte 3 if NibbleCount<5<br><br>for SENT TX frame:<br>- SwCRC: Field CRC is sent onto the SENT bus<br>- Otherwise: This field is ignored. |
|        | 4-7 | CRC Calculated  | 4              |                                                                                                                                                                   |

### 2.3. SENT Slow Data Frame

| Byte | Bit | Name                                                                                                                                                                 | Length in Bits | Description                          |
|------|-----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|--------------------------------------|
| 0    | 0-7 | MessageId                                                                                                                                                            | 8              |                                      |
| 1,2  |     | Data                                                                                                                                                                 | 16             | Intel coding (LSB first)             |
| 3    | 0-5 | CRC Received                                                                                                                                                         | 6              | ignored for TX                       |
|      | 6   | Slow Frame Type<br>0 = Short Serial<br>1 = Enhanced Serial                                                                                                           | 1              |                                      |
|      | 7   | Enhanced Serial Format Configuration Bit<br>0 = 8-bit MessageId, 12-bit Data<br>1 = 4-bit MessageId, 16-bit Data<br>Value of 0 also for the <i>Short Serial type</i> | 1              | Valid for <i>Slow Frame Type</i> ==1 |
| 4    | 0-5 | CRC Calculated                                                                                                                                                       | 6              | ignored for TX                       |

## 2.4. SENT Error Frame

| Byte | Bit | Name      | Length in Bits | Description                                                                                                                                                                                                              |
|------|-----|-----------|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0    | 0-3 | ErrorCode | 4              | 1 = Status Nibble / Sync error<br>2 = Data Nibble 0 error<br>3 = Data Nibble 1 error<br>4 = Data Nibble 2 error<br>5 = Data Nibble 3 error<br>6 = Data Nibble 4 error<br>7 = Data Nibble 5 error<br>8 = CRC Nibble error |
|      | 4-5 | ErrorType | 2              | 0 = OK<br>1 = CRC<br>2 = FRAMING<br>3 = SYNC                                                                                                                                                                             |

Note that when CrcMode in channel configuration is set to hwCRC off (value 0), incorrect CRC in received Fast Frame does not generate an error (see above).

## 2.5. SENT Slow Error Frame

| Byte | Bit | Name      | Length in Bits | Description                                  |
|------|-----|-----------|----------------|----------------------------------------------|
| 0    | 0-3 |           | 4              | unused                                       |
|      | 4-5 | ErrorType | 2              | 0 = OK<br>1 = CRC<br>2 = FRAMING<br>3 = SYNC |



## 2.6. SENT Slow Message TX Buffers

The gateway allows for a cyclic transmission of Slow messages with different Message Ids. This is useful for sensor simulations when the user needs to multiplex between several Slow messages.

By enabling several message buffers, the particular SENT TX channel will automatically transmit the Slow messages in a cyclic order. The *SlowChannelMode* in SENT channel configuration has to be set to either Short Serial or Enhanced serial mode.

| Byte | Bit | Name              | Length in Bits | Description                                                                       |
|------|-----|-------------------|----------------|-----------------------------------------------------------------------------------|
| 0    | 0-4 | Buffer Index      | 5              |                                                                                   |
|      | 5   | Buffer Enabled    | 1              | Enable or disable the buffer                                                      |
|      | 6   | EnhancedConfigBit | 1              | Relevant for Enhanced Serial Format<br><i>Not relevant when Buffer Enabled==0</i> |
|      | 7   | Reserved          |                |                                                                                   |
| 1    |     | Message Id        | 8              | Slow message identifier<br><i>Not relevant when Buffer Enabled==0</i>             |
| 2,3  |     | Data              | 16             | Intel coding (LSB first)<br><i>Not relevant when Buffer Enabled==0</i>            |

A message buffer can be enabled or disabled even whilst a SENT channel is running. Transmitting a single slow message (Message Id 42 and 52) will switch back to single slow message buffer mode even when at least one message buffer above is enabled.

All message buffers are cleared when a SENT channel is stopped. The user can configure the buffers before a SENT channel is started so that a cyclic transmission begins once the first Fast message is sent (Message Ids 41 and 51).

## 2.7. Analogue Output Configuration

An analogue output channel (12-bit, 0-4.095 V) can be mapped on any RX SENT channel. Bit position and bit length within the SENT data nibbles, and nibble order (endianness) are configurable. So is the linear transfer function and voltage Min and Max limits.

| Byte | Bit | Name                                                       | Length in Bits | Description                                                                                   | Default Value |
|------|-----|------------------------------------------------------------|----------------|-----------------------------------------------------------------------------------------------|---------------|
| 0    | 0-4 | Start bit                                                  | 5              | Start bit in the Data field of a SENT frame                                                   | 0             |
|      | 5   | Nibble order (endianness)                                  | 1              | 0 = Big endian (MSN first)<br>1 = Little endian (LSN first)                                   | 0             |
|      | 6-7 | SENT channel<br>0=Disabled (HighZ)<br>1=SENT 1<br>2=SENT 2 | 2              | SENT channel selection;<br><i>E.g. what SENT channel is this analogue output mapped onto.</i> | 0             |
| 1    | 0-4 | Length                                                     | 5              | Bit length in bits                                                                            | 12            |
| 2,3  |     | Offset                                                     | 16             | <b>Signed</b> value<br>Intel coding (LSB first)                                               | 0             |
| 4,5  |     | Multiplier                                                 | 16             | <b>Signed</b> value<br>Intel coding (LSB first)                                               | 1024          |

The Nibble order specifies what order are the nibbles from a SENT data frame taken in. E.g. how a raw value is calculated out of the nibbles. Apparently, the Nibble order value is relevant when Length >= 5 bits. See 0 for endianness description.

Please be aware the **Start bit** also depends on the endianness.

### Conversion:

$$U_{out} [mV] = (RawValue * Multiplier / 1024) + Offset$$

*Note: Apart from the physical range of the DAC (which is 0-4.095 V), the voltage range can further be limited by software. See next paragraph for voltage limits.*

## 2.8. Analogue Output Limits

This limits the range of the analogue channel.

| Byte | Bit | Name                 | Length in Bits | Description              | Default Value |
|------|-----|----------------------|----------------|--------------------------|---------------|
| 0,1  |     | Minimum voltage [mV] | 16             | Intel coding (LSB first) | 0             |
| 2,3  |     | Maximum voltage [mV] | 16             | Intel coding (LSB first) | 4095          |

## 2.9. Rolling Counter Configuration

A transmission of a rolling counter (RCNT) in the data nibbles of Fast message can be configured. The device then automatically transmits the RCNT and increments its value every time a Fast message has been transmitted. The RCNT configuration has the priority of its data field over a standard Transmit Fast Frame. Hence Transmit Fast Frame does not overwrite the RCNT data field when the RCNT is enabled.

SENT channel number relevant for the RCNT is determined by the Message Id (see 4.1). The RCNT can be enabled or disabled on-the-fly by its Enable bit.

| Byte | Bit | Name         | Length in Bits | Description                                                 |
|------|-----|--------------|----------------|-------------------------------------------------------------|
| 0    | 0-4 | Start bit    | 5              | Start bit in the Data field of a SENT frame                 |
|      | 5   | Nibble order | 1              | 0 = Big endian (MSN first)<br>1 = Little endian (LSN first) |
|      | 6   | Enable       | 1              | Enable or disable the counter                               |
| 1    | 0-4 | Length       | 5              | Bit length of the counter                                   |

Please be aware the **Start bit** also depends on the endianness. See 0 for endianness description.

### 3. Communication Protocol

The communication between the SENT gateway and other system is based upon a binary protocol. The same message structure is used for both directions - to and from a device.

The communication protocol consists of Message Id and Data. For RS-232 and USB product variants, the protocol is encapsulated by Start, Length, Checksum and End. For CAN bus, the protocol is placed into the data bytes of a CAN frame.

#### 3.1. RS-232 and USB

The protocol contains delimiters for start and end of a message, a Message Id, Data with variable length and a Checksum.

| STX (1B) | LEN (1B)                                | ID (1B)    | DATA (X B)                     | CHKSUM (1B)                                                   |  | ETX (1B) |
|----------|-----------------------------------------|------------|--------------------------------|---------------------------------------------------------------|--|----------|
| 0x02     | Length of the message in bytes<br>0 - 8 | Message Id | ...<br>Number of bytes = LEN-1 | Checksum<br>- a byte sum of<br>LEN+ID+(DATA0+DATA1...+DATA N) |  | 0x03     |

LEN value is the length of ID + DATA

##### 3.1.1. Serial Line Baud Rate

8 data bits, no parity, 1 stop bit. Baud rate is user selectable, default 115200.

| Baud rate Selector Value | Baud rate | SENT-RS232 | SENT-USB |
|--------------------------|-----------|------------|----------|
| 0                        | 19200     | +          | +        |
| 1                        | 115200    | +          | +        |
| 2                        | 230400    | +          | +        |
| 3                        | 460800    |            | +        |
| 4                        | 921600    |            | +        |

User can select different baud rate by writing command with Message ID 65 (0x41) and Baud rate Selector value in Data[0]. Both SENT channels have to be disabled before calling this command. Gateway acknowledges command and then changes baud rate settings.

#### 3.2. CAN

The SENT Gateway receives via CANID\_RX and transmits over CANID\_TX. Both CAN identifiers can be changed per device - see Message Ids 84 - 87.

##### Default configuration:

CANID\_RX = 0x123 Std Id.

CANID\_TX = 0x321 Std Id.

CAN Baud = 500 KBaud

## CAN Frame

| CAN    | Data Bytes 0 - 8 |                          |
|--------|------------------|--------------------------|
| ID     | Message Id       | Data                     |
| 11 bit | 8 bit            | 0 – 56 bit (0 – 7 bytes) |

Data Byte 0 is always used as Message Id (just like in RS-232), the rest of the data bytes carry the message content.

*Note: Grey parts are automatically generated by a CAN controller.*

### 3.2.1. CAN Configuration

| Byte | Bit | Name             | Length in Bits | Description    | Range   |
|------|-----|------------------|----------------|----------------|---------|
| 0-3  |     | BaudRate         | 32             | default 500000 |         |
| 4    |     | Sample point [%] | 8              | default 80     | 70 - 90 |

### 3.2.2. CAN IDs for SENT RX Messages

For some data acquisition systems, it is easier to receive SENT data messages over a CAN frame with a CAN identifier different from the one used for the Communication Protocol (see 3.2 CAN). This will help the measurement system to process incoming CAN frames without checking the *Message Id* at Data[0] of each CAN frame.

This feature is enabled for a SENT channel by Message Id 82 and 83. When enabled, the gateway will transmit Fast RX SENT frames (2.2 and 2.4) and Slow RX SENT frames (2.3 and 2.5) for the particular SENT channel over CAN messages with separated CAN identifiers. *RxForwardMode* is still applied.

| Byte | Name                                                                                                                                                                                                                                                                                                                                                                                                                                             | Length in Bits | Description                                                                                                   |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|---------------------------------------------------------------------------------------------------------------|
| 0-3  | CAN Id of SENT Fast Data Frame<br><i>Notes:</i> <ul style="list-style-type: none"> <li><i>CAN Id of SENT Fast Error Frame is configured implicitly to "(CAN Id of SENT Fast Data Frame) + 1"</i></li> <li><i>CAN Id of SENT Slow Data Frame is configured implicitly to "(CAN Id of SENT Fast Data Frame) + 2"</i></li> <li><i>CAN Id of SENT Slow Error Frame is configured implicitly to "(CAN Id of SENT Fast Data Frame) + 3"</i></li> </ul> | 32             | 0 = Disabled (default)<br>Otherwise:<br>Bit 0 - 28 = CAN Id (Intel coding)<br>Bit 31 (MSB) = Extended Id Flag |

*Note: Even when this feature is enabled, the gateway will still transmit these messages over the main CAN identifiers (Message Ids 100 - 103). This means the information will be doubled.*

## 4. Messages

### 4.1. To Gateway

| PC to Device |                                                     |                                                     |           |                   |  |
|--------------|-----------------------------------------------------|-----------------------------------------------------|-----------|-------------------|--|
| Message Id   | Name                                                | Description                                         | LEN value | Response          |  |
| 1            | Read SENT 1 channel Configuration                   |                                                     | 1         | Data              |  |
| 2            | Write SENT 1 channel Configuration 1                | Configure channel SENT                              | 8         | 0 = ERR<br>1 = OK |  |
|              |                                                     |                                                     |           |                   |  |
| 11           | Read SENT 2 channel Configuration                   |                                                     | 1         | Data              |  |
| 12           | Write SENT 2 channel Configuration 1                | Configure channel SENT                              | 8         | 0 = ERR<br>1 = OK |  |
|              |                                                     |                                                     |           |                   |  |
| 21           | Start SENT 1 channel                                |                                                     | 1         | 0 = ERR<br>1 = OK |  |
| 22           | Stop SENT 1 channel                                 |                                                     | 1         | 0 = ERR<br>1 = OK |  |
|              |                                                     |                                                     |           |                   |  |
| 31           | Start SENT 2 channel                                |                                                     | 1         | 0 = ERR<br>1 = OK |  |
| 32           | Stop SENT 2 channel                                 |                                                     | 1         | 0 = ERR<br>1 = OK |  |
|              |                                                     |                                                     |           |                   |  |
| 41           | Transmit SENT 1 Fast frame                          | See 2.2                                             | 3-5       | 0 = ERR<br>1 = OK |  |
| 42           | Write SENT 1 Slow frame buffer (no multiplexing)    | See 2.3                                             | 5         | 0 = ERR<br>1 = OK |  |
| 43           | Write SENT 1 Slow frame buffers (with multiplexing) | See 2.6                                             | 4         | 0 = ERR<br>1 = OK |  |
| 44           | Configure rolling counter transmission for SENT1    | See 2.9                                             | 3         |                   |  |
|              |                                                     |                                                     |           |                   |  |
| 51           | Transmit SENT 2 Fast frame                          | See 2.2                                             | 3-5       | 0 = ERR<br>1 = OK |  |
| 52           | Write SENT 2 Slow frame buffer (no multiplexing)    | See 2.3                                             | 5         | 0 = ERR<br>1 = OK |  |
| 53           | Write SENT 2 Slow frame buffers (with multiplexing) | See 2.6                                             | 4         | 0 = ERR<br>1 = OK |  |
| 54           | Configure rolling counter transmission for SENT2    | See 2.9                                             | 3         |                   |  |
|              |                                                     |                                                     |           |                   |  |
| 60           | LOAD_CONFIGURATION                                  | Load configuration from non-volatile memory to RAM  | 1         | 0 = ERR<br>1 = OK |  |
| 61           | SAVE_CONFIGURATION                                  | Save current configuration into non-volatile memory | 1         | 0 = ERR<br>1 = OK |  |

|    |                                               |                                                                   |   |                      |  |
|----|-----------------------------------------------|-------------------------------------------------------------------|---|----------------------|--|
| 62 | LOAD_DEFAULT_CONFIGURATION                    | Load                                                              |   |                      |  |
| 65 | Write Serial Line Baud rate                   | See 3.1.1<br><i>RS232 and USB variant only</i>                    | 1 | 0 = ERR<br>1 = OK    |  |
| 70 | Read AO1 Configuration                        |                                                                   | 1 | Data                 |  |
| 71 | Write AO1 Configuration                       |                                                                   | 7 | 0 = ERR<br>1 = OK    |  |
| 72 | Read AO1 Limits                               |                                                                   | 1 | Data                 |  |
| 73 | Write AO1 Limits                              |                                                                   | 5 | 0 = ERR<br>1 = OK    |  |
| 74 | Read AO2 Configuration                        |                                                                   | 1 | Data                 |  |
| 75 | Write AO2 Configuration                       |                                                                   | 7 | 0 = ERR<br>1 = OK    |  |
| 76 | Read AO2 Limits                               |                                                                   | 1 | Data                 |  |
| 77 | Write AO2 Limits                              |                                                                   | 5 | 0 = ERR<br>1 = OK    |  |
|    |                                               |                                                                   |   |                      |  |
| 80 | Read CAN Configuration                        | See 3.2.1<br><i>CAN variant only</i>                              | 1 | Data                 |  |
| 81 | Write CAN Configuration                       | See 3.2.1<br><i>CAN variant only</i>                              | 8 | 0 = ERR<br>1 = OK    |  |
| 82 | Write SENT 1 Alternative CAN Id Configuration | See 3.2.2<br><i>CAN variant only</i>                              | 8 | 0 = ERR<br>1 = OK    |  |
| 83 | Write SENT 2 Alternative CAN Id Configuration | See 3.2.2<br><i>CAN variant only</i>                              | 8 | 0 = ERR<br>1 = OK    |  |
| 84 | Read CANID_RX                                 | <i>CAN variant only</i><br><i>Bit 31 (MSB) = Extended Id Flag</i> | 1 | ID_RX<br>4 Bytes     |  |
| 85 | Write CANID_RX                                | <i>CAN variant only</i><br><i>Bit 31 (MSB) = Extended Id Flag</i> | 5 | 0 = ERR<br>1 = OK    |  |
| 86 | Read CANID_TX                                 | <i>CAN variant only</i><br><i>Bit 31 (MSB) = Extended Id</i>      | 1 | ID_TX<br>4 Bytes     |  |
| 87 | Write CANID_TX                                | <i>CAN variant only</i><br><i>Bit 31 (MSB) = Extended Id</i>      | 5 | 0 = ERR<br>1 = OK    |  |
|    |                                               |                                                                   |   |                      |  |
| 90 | READ_SER_NO                                   | Serial No<br><i>Intel byte order</i>                              | 1 | Serial No<br>4 Bytes |  |
| 91 | READ_HW_INFO                                  | HWID/VariantId/Product Id                                         | 1 | 6 Bytes              |  |
| 92 | READ_SW_INFO                                  | FW version (X.Y)<br>Byte 0 = Y<br>Byte 1 = X                      |   | 2 Bytes              |  |
| 93 | READ_STATUS                                   | byte 0:<br>Bit 0 = Channel 1 running<br>Bit 1 = Channel 2 running |   | 4 Bytes              |  |

|    |              |                        |   |  |  |
|----|--------------|------------------------|---|--|--|
|    |              | Byte 1 - 3: reserved   |   |  |  |
| 99 | RESTART_BOOT | Reboot into bootloader | 1 |  |  |
|    |              |                        |   |  |  |

#### 4.2. From Gateway

| Gateway to PC |                            |                                                                                 |           |                                                                   |  |
|---------------|----------------------------|---------------------------------------------------------------------------------|-----------|-------------------------------------------------------------------|--|
| Message Id    | Name                       | Description                                                                     | LEN value | Response                                                          |  |
| 100           | SENT 1 Frame Received      | See 2.2 SENT Data Frame                                                         | 3-5       |                                                                   |  |
| 101           | Slow SENT 1 Frame Received | Short Serial Message or Enhanced Serial Message<br>See 2.3 SENT Slow Data Frame | 4         |                                                                   |  |
| 102           | SENT 1 Error Received      | See 2.4 SENT Error Frame                                                        | 2         |                                                                   |  |
| 103           | SENT 1 Slow Error Received | See 2.5 SENT Slow Error Frame                                                   | 2         |                                                                   |  |
| 110           | SENT 1 Tx Echo             | See 2.2 SENT Data Frame                                                         | 3-5       |                                                                   |  |
|               |                            |                                                                                 |           |                                                                   |  |
| 200           | SENT 2 Frame Received      | See 2.2 SENT Data Frame                                                         | 3-5       |                                                                   |  |
| 201           | Slow SENT 2 Frame Received | Short Serial Message or Enhanced Serial Message<br>See 2.3 SENT Slow Data Frame | 4         |                                                                   |  |
| 202           | SENT 2 Error Received      | See 2.4 SENT Error Frame                                                        | 2         |                                                                   |  |
| 203           | SENT 2 Slow Error Received | See 2.5 SENT Slow Error Frame                                                   | 2         |                                                                   |  |
| 210           | SENT 2 Tx Echo             | See 2.2 SENT Data Frame                                                         | 3-5       |                                                                   |  |
| 250           | Boot Up                    | Device was powered up or restarted                                              | 1         |                                                                   |  |
|               |                            |                                                                                 |           |                                                                   |  |
| 255           | Error                      |                                                                                 | 2         | ErrorCode:<br>2 = Wrong Checksum<br>10 (dec) = Unknown Message Id |  |

## 5. Command Examples

### 5.1. Transmission over SENT Channel

#### 5.1.1. Fast Message

Transmit a fast message over SENT 1 channel.

| Command   | Description                                                       | Message Id |
|-----------|-------------------------------------------------------------------|------------|
| Stop      | A SENT channel has to be stopped before writing its configuration | 22         |
| Configure | Write channel configuration                                       | 2          |
| Start     | Start Channel                                                     | 21         |
| Transmit  | Transmit a Fast Message                                           | 41         |
|           | The gateway will begin to transmit the fast message continuously. |            |

#### 5.1.2. Slow Message

Transmit a fast message and slow message over SENT 1 channel.

| Command          | Description                                                                                         | Message Id |
|------------------|-----------------------------------------------------------------------------------------------------|------------|
| Stop             | A SENT channel has to be stopped before writing its configuration                                   | 22         |
| Configure        | Write channel configuration (make sure you enable the Slow Channel mode by <i>SlowChannelMode</i> ) | 2          |
| Start            | Start Channel                                                                                       | 21         |
| Write Slow frame | Prepare the Slow Message buffer                                                                     | 42         |
| Transmit         | Transmit a Fast Message                                                                             | 41         |
|                  | The gateway will begin to transmit the fast and slow message continuously.                          |            |

#### 5.1.3. Slow Messages - Multiplexing

Transmit a fast message and multiple slow messages over SENT 1 channel.

| Command                    | Description                                                                                                                                           | Message Id |
|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| Stop                       | A SENT channel has to be stopped before writing its configuration                                                                                     | 22         |
| Configure                  | Write channel configuration (make sure you enable the Slow Channel mode by <i>SlowChannelMode</i> )                                                   | 2          |
| Start                      | Start Channel                                                                                                                                         | 21         |
|                            |                                                                                                                                                       |            |
| Write Slow message buffers | Enable and configure Slow Message buffers                                                                                                             | 43         |
| Write Slow message buffers | ..... up to 32 slow message buffers                                                                                                                   | 43         |
| Transmit                   | Transmit a Fast Message                                                                                                                               | 41         |
|                            | The gateway will begin to transmit the fast and slow messages continuously whilst the configured slow message buffers will be transmitted cyclically. |            |



## 6. Communication Examples

### 6.1. RS-232 and USB

#### Read Serial Number:

02 01 5A 5B 03

Gateway Response: 02 05 5A **FF FF FF FE** 5A 03

where 0xFEFFFFFF (coded in Intel/Little-Endian) is the Gateway's serial number

### 6.2. CAN bus

#### 6.2.1. SENT Channel 1 Configuration

| Property                          | SENT 1                              |
|-----------------------------------|-------------------------------------|
| Auto Start                        | <input checked="" type="checkbox"/> |
| Direction                         | Tx                                  |
| Crc Mode                          | HwCrc                               |
| Nibble Count                      | 5                                   |
| Pulse Pause                       | <input type="checkbox"/>            |
| Pulse Pause Frame Period          | 0<br>Min: 527 Max: 2724             |
| Rx Forward Mode /<br>Tx Echo Mode | Fixed 100ms                         |
| Slow Channel mode                 | Fast Channel only                   |
| SPC Enabled                       | <input type="checkbox"/>            |
| Unit Time                         | 3                                   |

| Frame Id | Frame Type | Data Length | Data                               |
|----------|------------|-------------|------------------------------------|
| 0x123    | Std. Frame | 7           | 0x02 0xA9 0x02 0x2C 0x01 0x00 0x00 |
| 0x321    | Std. Frame | 2           | 0x02 0x01                          |

### 6.2.2. Start Channel 1

| Frame Id | Frame Type | Data Length | Data      |
|----------|------------|-------------|-----------|
| 0x123    | Std. Frame | 1           | 0x15      |
| 0x321    | Std. Frame | 2           | 0x15 0x01 |

### 6.2.3. Stop Channel 1

| Frame Id | Frame Type | Data Length | Data      |
|----------|------------|-------------|-----------|
| 0x123    | Std. Frame | 1           | 0x16      |
| 0x321    | Std. Frame | 2           | 0x16 0x01 |

### 6.2.4. Start Both Channels

| Frame Id | Frame Type | Data Length | Data                     | Other Properties |
|----------|------------|-------------|--------------------------|------------------|
| 0x123    | Std. Frame | 1           | 0x15                     |                  |
| 0x321    | Std. Frame | 2           | 0x15 0x01                |                  |
| 0x123    | Std. Frame | 1           | 0x1F                     |                  |
| 0x321    | Std. Frame | 2           | 0x1F 0x01                |                  |
| 0x123    | Std. Frame | 1           | 0x5D                     |                  |
| 0x321    | Std. Frame | 5           | 0x5D 0x03 0x00 0x00 0x00 |                  |

Recommended practice is to check channel status after channel start. This can be done easily by sending a request with Id=0x5D

### 6.2.5. Transmit on Channel 1

Channel 1

| Status | Reserved | 1 | 2 | 3 | 4 | 5 | 6 |          |  |
|--------|----------|---|---|---|---|---|---|----------|--|
| 0      | 0        | 1 | 2 | 3 | 4 | 5 | 0 | Transmit |  |

| Frame Id | Frame Type | Data Length | Data                     |
|----------|------------|-------------|--------------------------|
| 0x123    | Std. Frame | 5           | 0x29 0x00 0x21 0x43 0x00 |
| 0x321    | Std. Frame | 2           | 0x29 0x01                |

## 7. Contact

**MACH SYSTEMS s.r.o.**

[www.machsystems.cz](http://www.machsystems.cz)

[info@machsystems.cz](mailto:info@machsystems.cz)

Czech Republic



Company Registration: 29413893

VAT no.: CZ29413893