SAE J2716 to Gateway

CAN bus and RS-232
Communication Protocol Specification

Changes

Date	Description	By
26.4.2019	Minimal Pause Pulse lowered to 282 ticks	Miroslav Machacek
3.4.2019	Pause Pulse clarified Communication examples added	Miroslav Machacek
7.1.2019	Configuration of analogue outputs CAN Alternative configuration removed	Miroslav Machacek
15.8.2018	Clarifications	Miroslav Machacek
14.2.2018	Initial Version	Miroslav Machacek

Table of Contents

1. Introduction

This document describes a communication protocol used by the SENT-CAN and SENT-RS232 gateways so that the user can integrate it into his system.

Each SENT interfaces 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.

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

The gateway features 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 – StartBit, BitLength, linear transfer function: Multiplier, Offset, Min/Max voltage limits.

2. Device Configuration

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

Once 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

Byte	Bit	Name	Length in Bits	Description	Range	Default Value
0	0	AutoStart	1	1 = enabled after 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	1 = 1 data nibble, 6 = 6 data nibbles	1 - 6	6
1	0	PulsePauseEnabled see *PulsePauseFramePeriod* *note: valid for TX only*	1	1 = transmit frames with pulse pause		0
	1,2	RxForwardMode / TxEchoMode	2	0 – Fast as possible / No echo (for Tx) 1 – Fixed 100ms 2 – On change + 1s		1
	3,4	SlowChannelMode	2	0 - Fast Channel Only 1 - Short Serial 2 - Enhanced Serial		0
	5	Reserved	1			
	6	Reserved	1			
	7	SPCEnabled	1	SPC mode 1=enabled, 0=disabled		0
2,3		UnitTime	16	Tick Time [hundreds of ns] 3us= 300	3 - 90 us	300
4,5		PulsePauseFramePeriod valid if *PulsePauseEnabled* is set *note: valid for TX only*	16	TX SENT frame length in us	see below	0

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*N <= FrameTime <= 848 + 12*N

N is number of data nibbles: 1 to 6

2.2. SENT Data Frame

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 for SENT TX frame and SwCRC: Field CRC is sent onto the SENT bus
	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	
	6	Slow Frame Type 0 = Short Serial 1 = Enhanced Serial	1	

	7	Enhanced Serial Format Configuration Bit 0 = 8-bit MessageId, 12-bit Data 1 = 4-bit MessageId, 16-bit Data Value of 0 also when Short Serial type	1	Valid for *Slow Frame Type*==1
4	0-5	CRC Calculated	6	

2.4. SENT Error Frame

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

2.5. SENT Slow Error Frame

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

2.6. 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 is configurable by the user. So is the linear transfer function and voltage Min+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
	6-7	SENT Channel	2	SENT channel selection	0

		0=Disabled (HighZ) 1=SENT 1 2=SENT 2			
1	0-4	Length bits	5		12
2,3		Offset	16	Intel coding (LSB first)	0
4,5		Multiplier	16	Intel coding (LSB first)	1024

Conversion:

Uout [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.

2.7. 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

3. Communication Protocol

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

The protocol consists of Message Id and Data. For RS-232, 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

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

RS-232 configuration is fixed: 115200 Baud, 8 data bits, no parity, 1 stop bit

STX (1B)	**LEN (1B)**	**ID (1B)**	**DATA (X B)**	**CHKSUM (1B)**	**ETX (1B)**
0x02	Length of the message in bytes 0 - 8	Message Id	… Number of bytes = LEN-1	Checksum - a byte sum of LEN+ID+(DATA0+DATA1…+DATAN)	0x03

3.2. CAN

The SENT Converter receives via CANID_RX a 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

SOF	ID	RTR	IDE	r	DLC	Data Bytes 0 - 8		ChkSUM	DEL	ACK	DEL	EOF
SOF	ID	RTR	IDE	r	DLC	Message Id	Data	ChkSUM	DEL	ACK	DEL	EOF
1	11 bit	1	1	1	4	8 bit	0 – 56 bit (0 – 7 bytes)	15 bit	1	1	1	7

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

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 1 = OK	
11	Read SENT 2 channel Configuration		1	Data	
12	Write SENT 2 channel Configuration 1	Configure channel SENT	8	0 = ERR 1 = OK	
21	Start SENT 1 channel		1	0 = ERR 1 = OK	
22	Stop SENT 1 channel		1	0 = ERR 1 = OK	
31	Start SENT 2 channel		1	0 = ERR 1 = OK	
32	Stop SENT 2 channel		1	0 = ERR 1 = OK	
41	Transmit SENT 1 frame	see 2.2 SENT Data Frame	3-5	0 = ERR 1 = OK	
42	Transmit SENT 1 slow frame	see 2.3 SENT Slow Data Frame	5	0 = ERR 1 = OK	
51	Transmit SENT 2 frame	see 2.2 SENT Data Frame	3-5	0 = ERR 1 = OK	
52	Transmit SENT 2 slow frame	see 2.3 SENT Slow Data Frame	5	0 = ERR 1 = OK	
60	LOAD_CONFIGURATION	Load configuration from non-volatile memory to RAM	1	0 = ERR 1 = OK	

61	SAVE_CONFIGURATION	save current configuration into non-volatile memory	1	0 = ERR 1 = OK	
62	LOAD_DEFAULT_CONFIGURATION				
70	Read AO1 Configuration		1	Data	
71	Write AO1 Configuration		7	0 = ERR 1 = OK	
72	Read AO1 Limits		1	Data	
73	Write AO1 Limits		5	0 = ERR 1 = OK	
74	Read AO2 Configuration		1	Data	
75	Write AO2 Configuration		7	0 = ERR 1 = OK	
76	Read AO2 Limits		1	Data	
77	Write AO2 Limits		5	0 = ERR 1 = OK	
80	Read CAN Configuration	*CAN variant only*	1	Data	
81	Write CAN Configuration	*CAN variant only*	8	0 = ERR 1 = OK	
84	Read CANID_RX	*CAN variant only* *Bit 31 (MSB) = Extended Id*	1	ID_RX 4 Bytes	
85	Write CANID_RX	*CAN variant only* *Bit 31 (MSB) = Extended Id*	5	0 = ERR 1 = OK	
86	Read CANID_TX	*CAN variant only* *Bit 31 (MSB) = Extended Id*	1	ID_TX 4 Bytes	
87	Write CANID_TX	*CAN variant only* *Bit 31 (MSB) = Extended Id*	5	0 = ERR 1 = OK	
90	READ_SER_NO	Serial No *Intel byte order*	1	Serial No 4 Bytes	
91	READ_HW_INFO	HWID/VariantId/Product Id	1	6 Bytes	
92	READ_SW_INFO	FW version (X.Y) Byte 0 = Y Byte 1 = X		2 Bytes	
93	READ_STATUS	byte 0: Bit 0 = Channel 1 running Bit 1 = Channel 2 running Byte 1 - 3: reserved		4 Bytes	
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 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 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	1		
255	Error		2	ErrorCode: 2 = Wrong Checksum 10 (dec) = Unknown Message Id	

5. Communication Examples

5.1. RS-232

Read Serial Number:

02 01 5A 5B 03

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

where FEFFFFFF (coded in Intel/Little-Endian) is a device serial number

5.2. CAN bus

5.2.1. SENT Channel 1 Configuration

Property	SENT 1
Auto Start	☑
Direction	Tx
Crc Mode	HwCrc
Nibble Count	5
Pulse Pause	☐
Pulse Pause Frame Period	0 Min: 527 Max: 2724
Rx Forward Mode / Tx Echo Mode	Fixed 100ms
Slow Channel mode	Fast Channel only
SPC Enabled	☐
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

5.2.2. Start Channel 1

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

5.2.3. Stop Channel 1

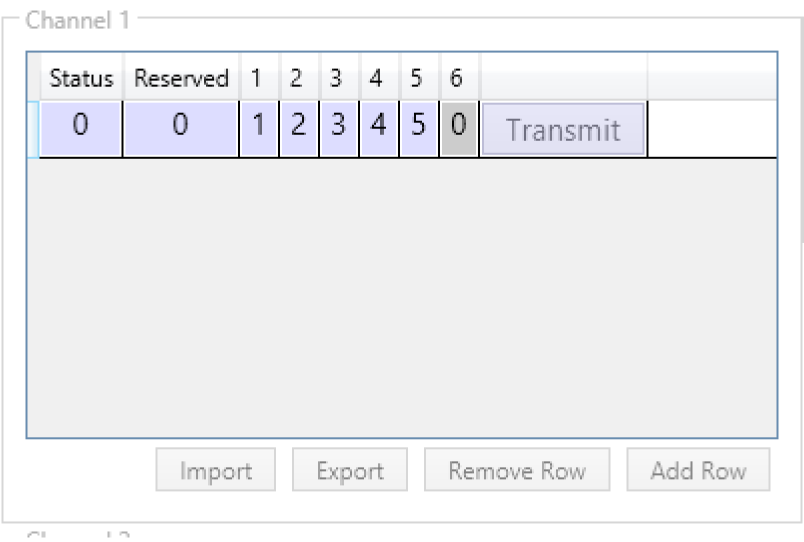
Frame Id	Frame Type	Data Length	Data
0x123	Std. Frame	1	0x16
0x321	Std. Frame	2	0x16 0x01

5.2.4. 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 request with id 0x5D

5.2.5. Transmit on Channel 1



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

6. Contact

MACH SYSTEMS s.r.o.

www.machsystems.cz

info@machsystems.cz

Czech Republic

Company Registration: 29413893

VAT no.: CZ29413893