GEFRAN

CANopen GSH-S

Digital output

Single version
Semi-redundant version
Redundant version

Cod. 80716B Edit.07/2024 - ENG

SUMMARY

1. SAFETY INSTRUCTIONS

If total failure or malfunction of the sensor can cause danger or injury to the operator or damage to the machinery or equipment it is recommended that additional safety measures should be incorporated into the system.
Any alteration, reconstruction or extension of the sensor is not allowed.
Sensor must be operated only within values specified in the datasheet.
Connection to power supply must be performed in accordance with safety instructions for electrical facilities and performed only by trained staff.
Disregard of this advice can lead to malfunctions, damage to property or personal injury and releases the manufacturer from product liability.

Do not open sensor
Release of spring under tension can result in injury!

Do not snap cable
Uncontrolled cable retraction can break off cable fixing. Broken fixing and cable can result in injury. Also sensor will be damaged!

Do not travel over range
Uncontrolled cable retraction can result in injury. Also sensor will be damaged!

Special attention during mounting and operation of metal cable sensors
Risk of injury by the measuring cable!

Sensors without cover / housing (OEM sensors)
Risk of injury by moving parts. Mounting and operation of the sensor requires appropriate safety equipment that excludes any risk of personal injury!

Do not exceed maximum operating voltage listed in the catalog
Risk of injury. Sensor will be damaged!

Before connecting the sensor to the CANbus the devices have to be checked for correct bitrate and unique node-IDs.

After power-on the sensor will enter pre-operational state and send a boot-up message indicating that it is ready for configuration by Service Data Objects. Parameters configured by the user can be stored nonvolatile by SAVE command.

On receiving “NMT-Node-Start“, the sensor transits to operational state and starts process data transmission.

When “Auto-Start“ is configured the sensor will automatically transit to operational after boot-up without a need for the Node-Start message. Node monitoring is supported by Heartbeat protocol. The Heartbeat protocol provides automatic transmission of the node status (heartbeat message) by the slave within producer heartbeat time window

Do not damage cable!

Cable must not be oiled or lubricated!

Do not snap cable!

Do not travel over range!

Do not crack cable!

Cable travel should be axial to the cable outlet (no misalignment allowed!)

Do not drag cable along objects!

Precautions:

Do not let snap the cable
Uncontrolled retraction of cable may damage sensor.
No warranty will be granted for snapped cables.

Mounting hints for unfavorable conditions
If possible, fasten cable fixing with cable in retracted position.
For example, fit a mounting loop (see diagram) and put it around your wrist.
Do not remove the mounting loop before the cable is fastened.
The cable clip may be opened for easy attachment.

Mounting
To ensure proper operation, install the sensor only as described in this manual.

2. INTRODUCTION

The purpose of GSH-S position sensor is to transform position of a linear and guided movement into an electrical signal, specifications of measuring range, environment, handling and connections as specified in the catalogue, must be followed. Linear motion of the measuring cable (flexible stainless steel) is converted into rotation by means of a precision cable drum. A spring motor provides torque for the cable retraction. Special design assures precise and reproducible winding of the measuring cable. Cable extraction or retraction is transformed into an electrical signal.
The sensor is based on state-of-the-art contactless angle sensor (in HALL technology) implementing the functions of a CAN-BUS network slave device conforming to standard CANOpen protocol proposed by C.i.A. (Can in Automation) and described in the document entitled “CANOpen Application Layer and Communication Profile DS 301 v. 4.2” and in other documents mentioned below. Other reference documents used are C.i.A. DS-406 Device Profile for encoder and C.i.A. DSP-305 Layer Setting Services and Protocol V1.1.1.
This document describes the standard CANOpen implementations created. It is addressed to CANOpen network system integrators and to CANOpen device designers who already know the content of the above-mentioned standards defined by C.i.A..

Definition and Shortening

CAN: Controller Area Network.
Describes a serial communication bus that implements the “physical” level 1 and the “data link” level 2 of the ISO/OSI reference model.

CAL: CAN Application Layer.
Describes implementation of the CAN in the level 7 “application” of the ISO/OSI reference model from which CANOpen derives.

CMS: CAN Message Specification.
CAL service element. Defines the CAN Application Layer for the various industrial applications.

COB: Communication Object.
Unit of transport of data in a CAN network (a CAN message). A maximum of 2048 COBs may be present in a CAN network, each of which may transport from 0 to a maximum of 8 bytes.

COB-ID: COB Identifier.
Identifying element of a CAN message. The identifier determines the priority of a COB in case of multiple messages in the network.

D1 - D8: Data from 1 to 8.
Number of bytes in the data field of a CAN message.

DLC: Data Length code.
Number of data bytes transmitted in a single frame.

ISO: International Standard Organization.
International authority providing standards for various merchandise sectors.

NMT: Network Management.
CAL service element. Describes how to configure, initialize, manage errors in a CAN network.

PDO: Process Data Object.
Process data communication objects (with high priority).

RXSDO: Receive SDO.
SDO objects received from the remote device.

SDO: Service Data Object.
Service data communication objects (with low priority). The value of this data is contained in the “Objects Dictionary” of each device in the CAN network.

TXPDO: Transmit PDO.
PDO objects transmitted by the remote device.

TXSDO: Transmit SDO.
SDO objects transmitted by the remote device.

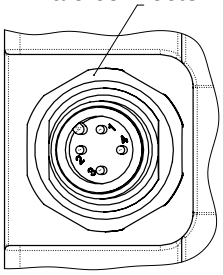
Note: The numbers followed by the suffix “h” represent a hexadecimal value, with suffix “b” a binary value, and with suffix “d” a decimal value. The value is decimal unless specified otherwise.

3. ELECTRICAL CONNECTIONS

For the connections refer to following images:

SINGLE VERSION **M-1-S**
REDUNDANT VERSION **M-1-R/ M-2-R**

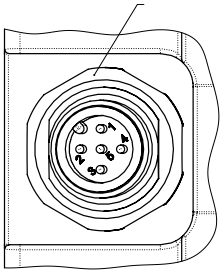
M12x1 4-pin
male connector



CAN
CONNECTIONS
1. +SUPPLY
2. GROUND
3. CAN-H
4. CAN-L

SINGLE VERSION **N-1-S**
REDUNDANT VERSION **N-1-R/ N-2-R**

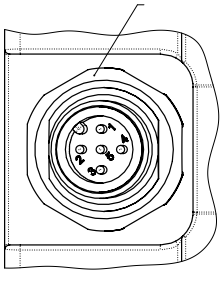
M12x1 5-pin
male connector



CAN
CONNECTIONS
1. n.c.
2. +SUPPLY
3. GROUND
4. CAN-H
5. CAN-L

SINGLE VERSION **P-1-S**
REDUNDANT VERSION **P-1-R P-2-R**
HALF-REDUNDANT VERSION **P-1-H**

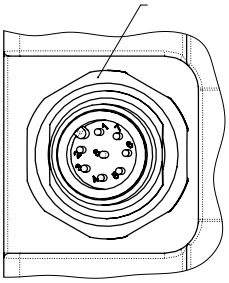
M12x1 5-pin
male connector



CAN
CONNECTIONS
1. GROUND
2. +SUPPLY
3. GROUND
4. CAN-H
5. CAN-L

REDUNDANT VERSION **O-1-R**

M12x1 8-pin
male connector

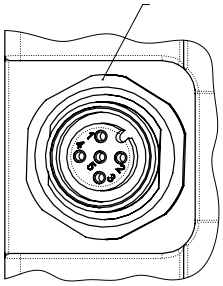


CAN CONNECTIONS

1. + SUPPLY CH1
2. GROUND CH1
3. CAN-H CH1
4. CAN-L CH1
5. + SUPPLY CH2
6. GROUND CH2
7. CAN-H CH2
8. CAN-L CH2

SINGLE / REDUNDANT / HALF-REDUNDANT VERSION IN-OUT VERSION **P-3-(S/R/H)**

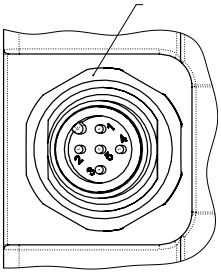
M12x1 5-pin
female connector



CAN CONNECTIONS

1. GROUND
2. +SUPPLY
3. GROUND
4. CAN-H
5. CAN-L

M12x1 5-pin
male connector

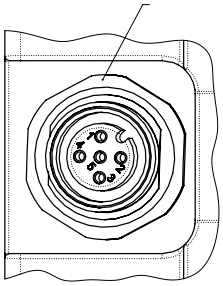


CAN CONNECTIONS

1. GROUND
2. +SUPPLY
3. GROUND
4. CAN-H
5. CAN-L

SINGLE / REDUNDANT / HALF-REDUNDANT VERSION IN-OUT VERSION **N-3-(S/R/H)**

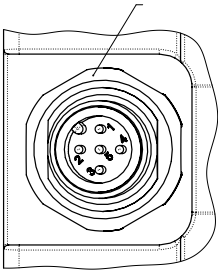
M12x1 5-pin
female connector



CAN CONNECTIONS

1. FLOATING
2. +SUPPLY
3. GROUND
4. CAN-H
5. CAN-L

M12x1 5-pin
male connector



CAN CONNECTIONS

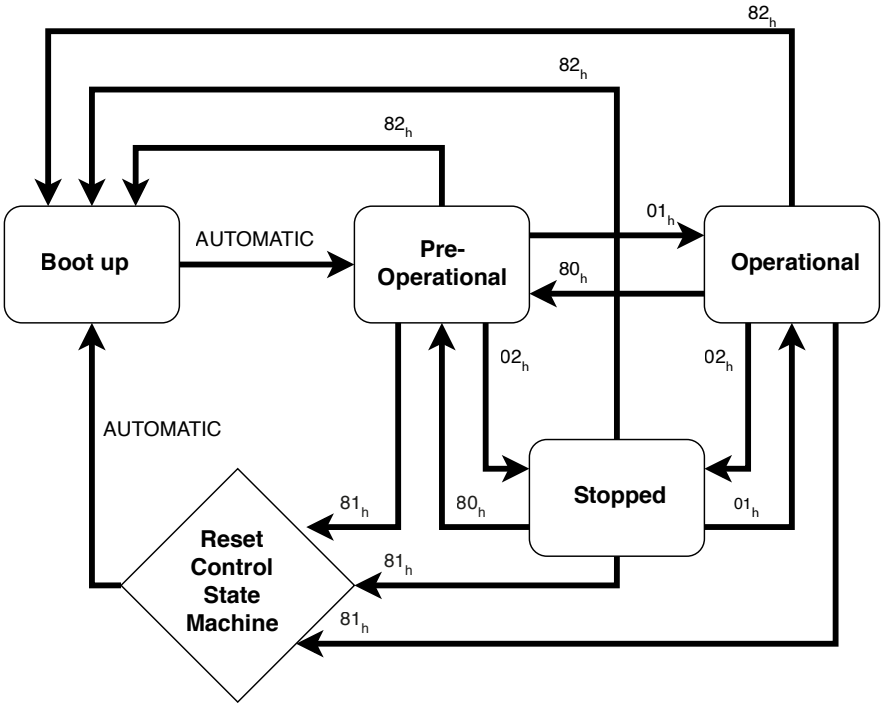
1. FLOATING
2. +SUPPLY
3. GROUND
4. CAN-H
5. CAN-L

PIN 1 male and female are internally connected

Note: please make sure that the CANbus is terminated.
The impedance measured between CAN H and CAN L must be 60 Ω that means the cable must be connected to a 120 Ω resistor on each ends of the bus line. Internally the transducer is not terminated with the resistor of 120 Ω.
Do not confuse the signal lines of the CAN bus, otherwise communication with the transducer is impossible.

4. NETWORK MANAGEMENT (NMT)

The device supports CanOpen network management functionality NMT Slave (Minimum Boot Up).



Every CanOpen device contains an internal Network Management server that communicates with an external NMT master. One device in a network, generally the host, may act as the NMT master.
Through NMT messages, each CANOpen device's network management server controls state changes within its built-in **Communication State Machine**.
This is independent from each node's operational state machine, which is device dependent and described in **Control State Machine**.

The **"Communication State Machine"** in all CanOpen devices, however, is identical as specified by the DS301. NMT messages have the highest priority. The 5 NMT messages that control the **Communication State Machine** each contain 2 data bytes that identify the node number and a command to that node's state machine.
Table 1 shows the 5 NMT messages supported, and Table 2 shows the correct message construction for sending these messages.

NMT Message	COB-ID	Data Bytes 1	Data Bytes 2
Start Remote Node	0	01h	Node-ID*
Stop Remote Node	0	02h	Node-ID*
Pre-operational State	0	80h	Node-ID*
Reset Node	0	81h	Node-ID*
Reset Communication	0	82h	Node-ID*

Table 1. NMT messages supported

** Node-ID = Drive address (from 01 to 7Fh)*

Arbitration Field		Data Field							
COB-ID	RTR	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
000h	0	See Table 1	See Table2	These bytes not sent					

Table 2. NMT message construction

5. BAUD RATE

Baud rate can be configurable via SDO communication object 0x20F2 and 0x20F3 (see communication examples at the end of this document).

The default Baud rate is 250kbit/s.

Important Note:
Changing this parameter can disturb the network! Only change this setting if a single device is connected to the network!

6. Node-ID

Node ID can be configurable via SDO communication object 0x20F0 and 0x20F1 (see communication examples at the end of this document).

The default Node-ID can vary depending on the version, as follows:

- **Single version** **0x7F (channel 1)**
- **Half-redundant version** **0x7F (channel 1 and 2)**
- **Redundant version** **0x7F (channel 1)**
  0x7E (channel 2)

Important Note:
Changing this parameter can disturb the network! Only change this setting if a single device is connected to the network!

7. PARAMETER SETTING

All object dictionary parameters (objects with marking PARA) can be saved in a special section of the internal EEPROM and secured by checksum calculation. The special LSS parameters (objects with marking LSS-PARA), also part of the object dictionary, will be also saved in a special section of the internal EEPROM and secured by checksum calculation. Due to the internal architecture of the microcontroller the parameter write cycles are limited to 100,000 cycles.

8. RESTORE DEFAULT PARAMETER

All object dictionary parameters (objects with marking PARA) can be restored to factory default values via SDO communication (index 0x1011).

9. HEARTBEAT

The heartbeat mechanism for this device is established through cyclic transmission of the heartbeat message done by the heartbeat producer. One or more devices in the network are aware of this heartbeat message. If the heartbeat cycle fails from the heartbeat producer, the local application on the heartbeat consumer will be informed about that event. The implementation of either guarding or heartbeat is mandatory.
The device supports **Heartbeat Producer** functionality. The producer heartbeat time is defined in object 0x1017.

Heartbeat Message:

COB-ID	**Byte**	**0**
700+Node-ID	Content	NMT State

10. ERROR HANDLING

Principle

Emergency messages (EMCY) shall be triggered by internal errors on device and they are assigned the highest possible priority to ensure that they get access to the bus without delay (**EMCY Producer**).
By default, the EMCY contains the error field with pre-defined error numbers and additional information.

Error Behavior (object 0x4000)

If a serious device failure is detected the object 0x4000 specifies, to which state the module shall be set:
0: pre-operational
1: no state change (default)
2: stopped

EMCY Message

The EMCY COB-ID is defined in object 0x1014. The EMCY message consists of 8 bytes. It contains an emergency error code, the contents of object 0x1001 and 5 byte of manufacturer specific error code. This device uses only the 1st byte as manufacturer specific error code.

Byte	**Byte1** **Byte2**	**Byte3**	**Byte4**	**Byte5**	**Byte6** **Byte7** **Byte8**
Description	Emergency Error Code	Error Register (object 0x1001) Always 0	Manufacturer specific error code (always 0x00)	Manufacturer specific error code (object 0x4001)	Manufacturer specific error code NOT IMPLEMENTED (always 0x00)

Error Code possible values:

Emergency **Error Code**	0x0000	Error Reset or no Error
	0x1000	Generic error

Supported Manufacturer Specific Error Codes (object 0x4001):

Manufacturer Specific Error Code (bit field) object 0x4001	**Description**
0bxxxxxxx1	Generic Error (e.g. potentiometric signal under/above limits, temperature under/above limits)
0bxxxxxx1x	Stroke below acceptable limits
0bxxxxx1xx	Stroke above acceptable limits

11. SDO COMMUNICATION AND READ/WRITE COMMANDS

The device fulfils the **SDO Server** functionality.

With Service Data Object (S.D.O.) the access to entries of a device Object Dictionary is provided. As these entries may contain data of arbitrary size and data type SDOs can be used to transfer multiple data sets from a client to a server and vice versa.

Write Access, Data Transfer from Host to Slave

Each access to the object dictionary is checked by the slave for validity. Any write access to non-existing objects, to read-only objects or with a non-corresponding data format are rejected and answered with a corresponding error message.

Structure of SDO-request by the Master:

COB-ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
600+Node-ID	8	CMD	Index		Sub-Index	Data	Data	Data	Data

CMD filed determines the direction of data transfer and the size of the data object:

23h Sending of 4-byte data (bytes 5...8 contain a 32-bit value)
2Bh Sending of 2-byte data (bytes 5, 6 contain a 16-bit value)
2Fh Sending of 1-byte data (byte 5 contains an 8-bit value)

Structure of SDO-answer by the Slave:

COB-ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
580+Node-ID	8	RES	Index		Sub-Index	Data	Data	Data	Data

RES field determines the correct/incorrect answer of the slave:

60h Data sent successfully
80h Error

Read Access, Data Transfer from Slave to Host

Each access to the object dictionary is checked by the slave for validity. Any read access to non-existing objects or to write-only are rejected and answered with a corresponding error message.

Structure of SDO-request by the Master:

COB-ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
600+Node-ID	8	CMD	Index		Sub-Index	Data	Data	Data	Data

CMD field determines the direction of data transfer for any size of the data object:

40h Read access of data (1-, 2- or 4-byte)

RES field determines the answer of the slave:

43h Sending of 4-byte data (bytes 5...8 contain a 32-bit value)
4Bh Sending of 2-byte data (bytes 5, 6 contain a 16-bit value)
4Fh Sending of 1-byte data (Byte 5 contains an 8-bit value)
80h Error

12. PDO COMMUNICATION AND LENGTH CALCULATION

Transmit PDO #0 and #1 – Length calculation GSH-S-xxx

The PDO transmits the length value of the position sensor, in particular:

- Single version, one PDO is transmitted containing length measurement of channel 1.
- Half-redundant version, one PDO is transmitted containing length measurement of channel 1 and channel 2.
- Redundant version, one PDO is transmitted for each channel. PDO #0 containing length measurement of channel 1 and PDO #1 containing length measurement of channel 2.

The PDOs shall be transmitted cyclically, the cyclic timer has default value equal to 100ms. However, cyclic timer of PDOs is programmable through object 0x1800 sub-index5 (and 0x1801 sub-index5, for redundant version). Values between 4ms and 65535ms shall be selectable by parameter settings.
The PDO #0 (and PDO #1, for redundant version) will be transmitted by entering the “Operational” state.

Single version, structure of PDO #0:

Byte 1,2 contain the length measurement of channel 1, signed integer 16-bit value with forward direction.

Byte1	**Byte2**	**Byte3**	**Byte4**	**Byte5, Byte6, Byte7**	**Byte8**
Position value Channel 1 LSB	Position value Channel 1 MSB	(0x00) Not in use	(0x00) Not in use	(0x00) Not in use	0x00 Error Code

Table 3. Structure of PDO #0, single version

Half-redundant version, structure of PDO #0:

Byte 1,2 contain the length measurement of channel 1, signed integer 16-bit value with forward direction,
Byte 3,4 contain the length measurement of channel 2, signed integer 16-bit value with backward direction.

Byte1	**Byte2**	**Byte3**	**Byte4**	**Byte5, Byte6, Byte7**	**Byte8**
Position value Channel 1 LSB	Position value Channel 1 MSB	Position value Channel 2 LSB	Position value Channel 2 MSB	(0x00) Not in use	0x00 Error Code

Table 4. Structure of PDO #0, half-redundant version

Redundant version, structure of PDO #0 and PDO #1:

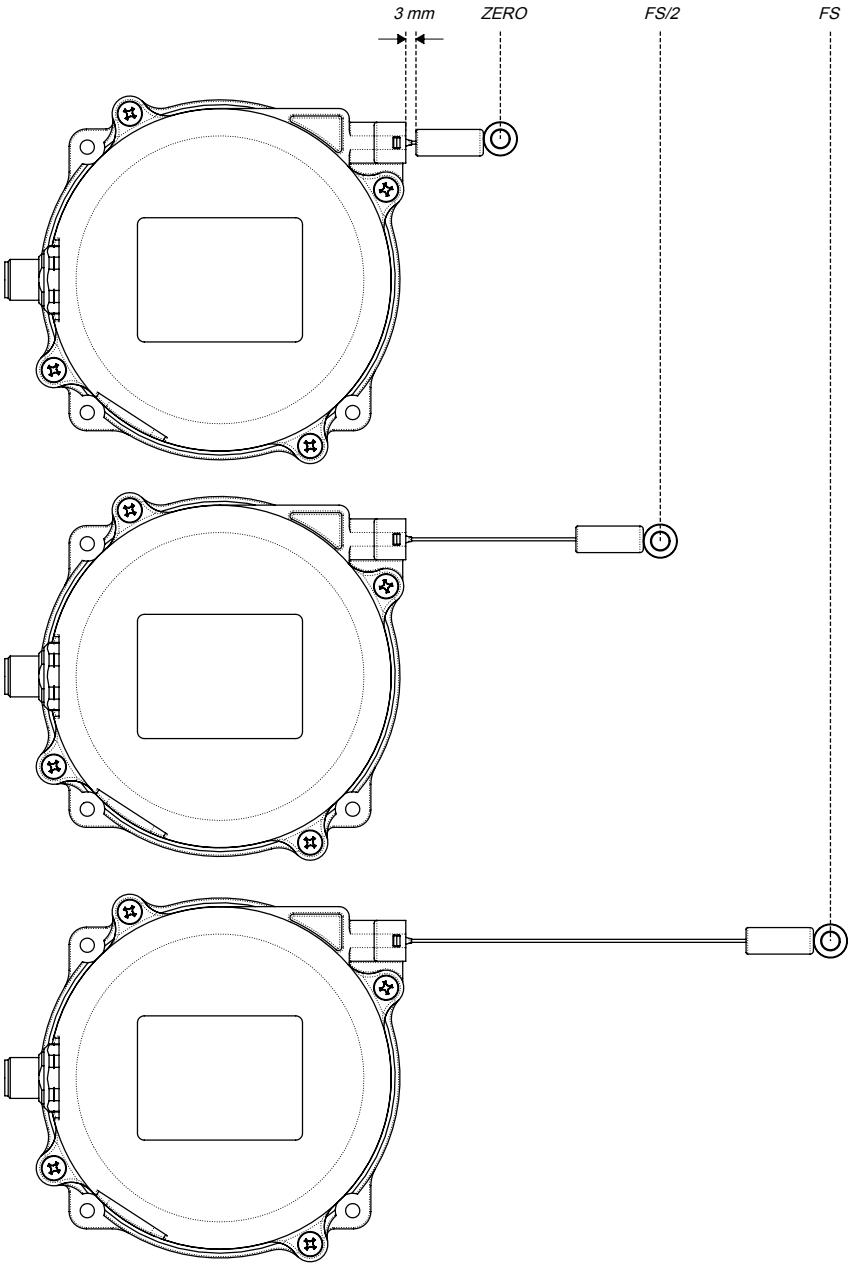
PDO #0, Byte 1,2 contain the length measurement of channel 1, signed integer 16-bit value with forward direction.
PDO #1, Byte 1,2 contain the length measurement of channel 2, signed integer 16-bit value with forward direction.

Byte1	**Byte2**	**Byte3**	**Byte4**	**Byte5, Byte6, Byte7**	**Byte8**
Position value Channel 1 LSB	Position value Channel 1 MSB	(0x00) Not in use	(0x00) Not in use	(0x00) Not in use	0x00 Error Code

Table 5. Structure of PDO #0, redundant version

Byte1	**Byte2**	**Byte3**	**Byte4**	**Byte5, Byte6, Byte7**	**Byte8**
Position value Channel 2 LSB	Position value Channel 2 MSB	(0x00) Not in use	(0x00) Not in use	(0x00) Not in use	0x00 Error Code

Table 6. Structure of PDO #1, redundant version



PDO EXAMPLES for different versions:

→ SINGLE VERSION

An example of a PDO is shown below with the following parameters:

- Node-ID = 7Fh
- Baude-rate = 250 kBaud
- Single version GSH-S
- Linear-encoder Cia406 setting as follow:
    I. Total Measuring Range (object 0x6002 sub-index 0) = 1800 mm (180 steps x 10 mm)
    II. Preset Value (object 0x6003 sub-index 0) = 0 mm (0 steps x 10^3 nm)
    III. Measuring Step (object 0x6005 sub-index 1) = 1 mm (1 000 000 steps x 10^3 nm)
    IV. Position Value (object 0x6004 sub-index 0) indicates the length measurement of channel 1 (forward)

• PDO #0 mapping, example 1:

ID	**Byte1**	**Byte2**	**Byte3**	Byte4	**Byte5**	**Byte6**	**Byte7**	**Byte8**
1FFh	00h	00h	00h	00h	00h	00h	00h	00h

Byte 1 (LSB) = 00h
Byte 2 (MSB) = 00h
=> Position Value = 0000h to decimal 0d (resolution 1 mm) = 0 mm

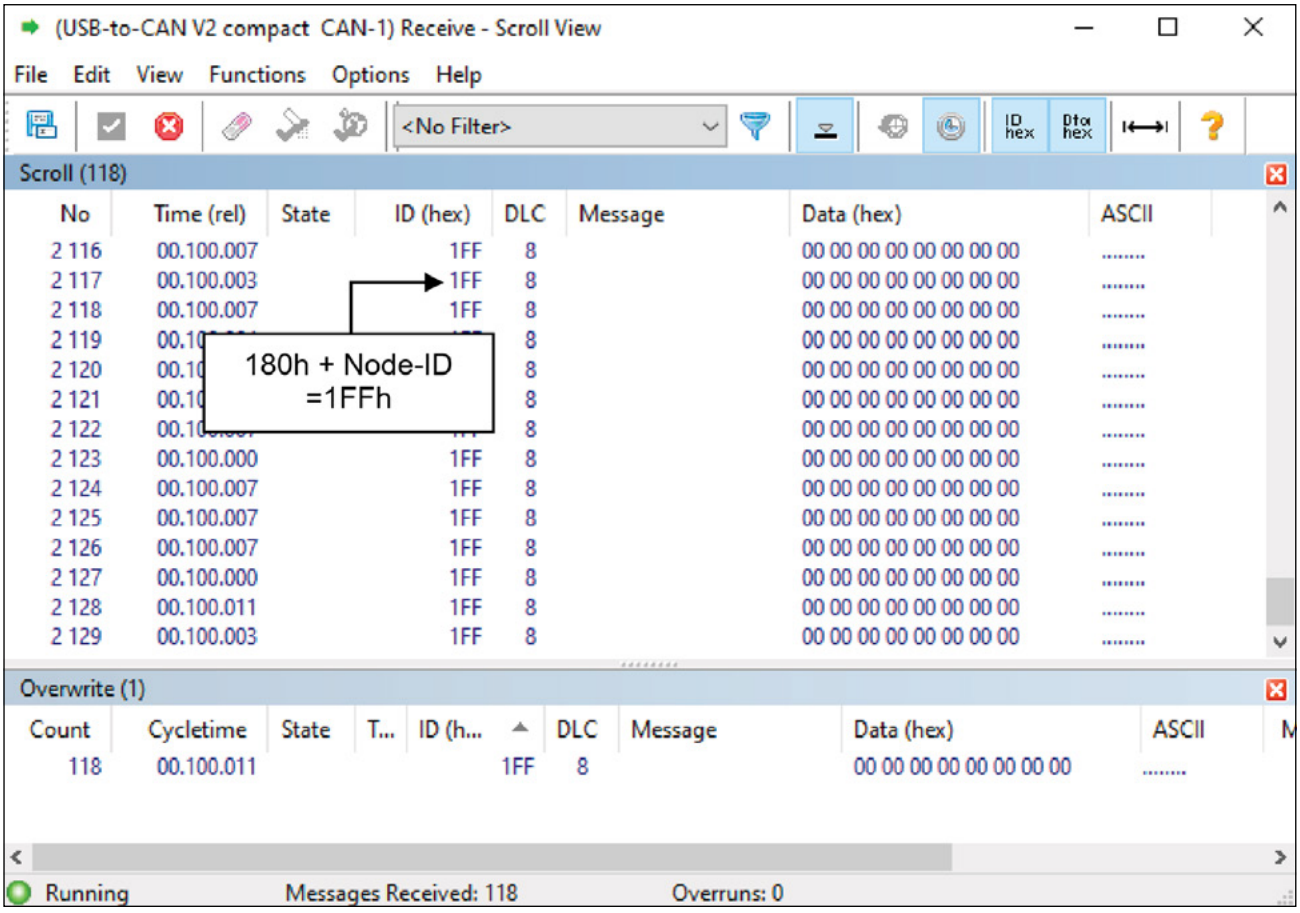


Figure 1. Single version PDO #0 mapping, example 1: CANbus sniffing by using a IXXAT USB-to-CAN V2 compact adapter and the canAnalyser3 software.

- ***PDO #0 mapping, example 2:***

ID	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
1FFh	48h	01h	00h	00h	00h	00h	00h	00h

Byte 1 (LSB) = 48h
Byte 2 (MSB) = 01h
=> Position Value = 0148h to decimal 329d (resolution 1 mm) = 329 mm

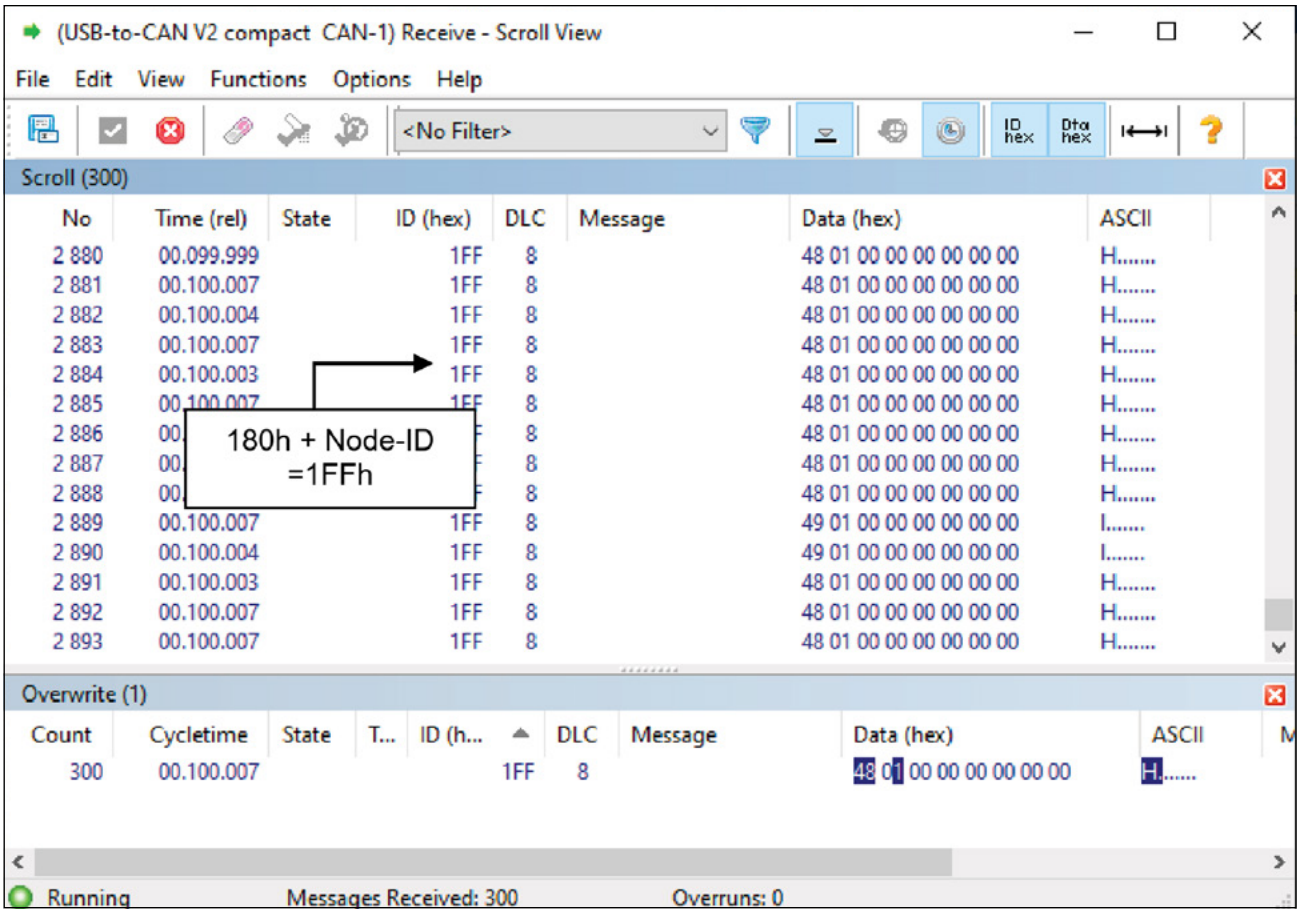


Figure 2. Single version PDO #0 mapping, example 2: CANbus sniffing by using a IXXAT USB-to-CAN V2 compact adapter and the canAnalyser3 software.

→ *SEMI-REDUNDANT VERSION*

In the following an example of PDO mapping is reported in the case of:

- Node-ID = 7Fh
- Baude-rate = 250 kBaud
- Half-redundant version GSH-S
- Linear-encoder Cia406 setting as follow:
    I. Total Measuring Range (object 0x6002 sub-index 0) = 1800 mm (180 steps x 10 mm)
    II. Preset Value (object 0x6010 sub-index 1 for channel 1, sub-index 2 for channel 2) = 0 mm (0 steps x 10^3 nm)
    III. Measuring Step (object 0x6005 sub-index 1) = 1 mm (1 000 000 steps x 10^3 nm)
    IV. Position Value A (object 0x6020 sub-index 5) indicates the length measurement of channel 1 (forward), Position Value B (object 0x6020 sub-index 8) indicates the length measurement of channel 2 (backward)

• *PDO #0 mapping, example 1:*

ID	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
1FFh	00h	00h	FFh	FFh	00h	00h	00h	00h

Byte 1 (LSB) = 00h
Byte 2 (MSB) = 00h
=> Position Value channel 1 = 0000h to decimal 0d (resolution 1 mm) = 0 mm

Byte 3 (LSB) = FFh
Byte 4 (MSB) = FFh
=> Position Value channel 2 = FFFFh – FFFFh = 0000h to decimal 0d (resolution 1 mm) = 0 mm

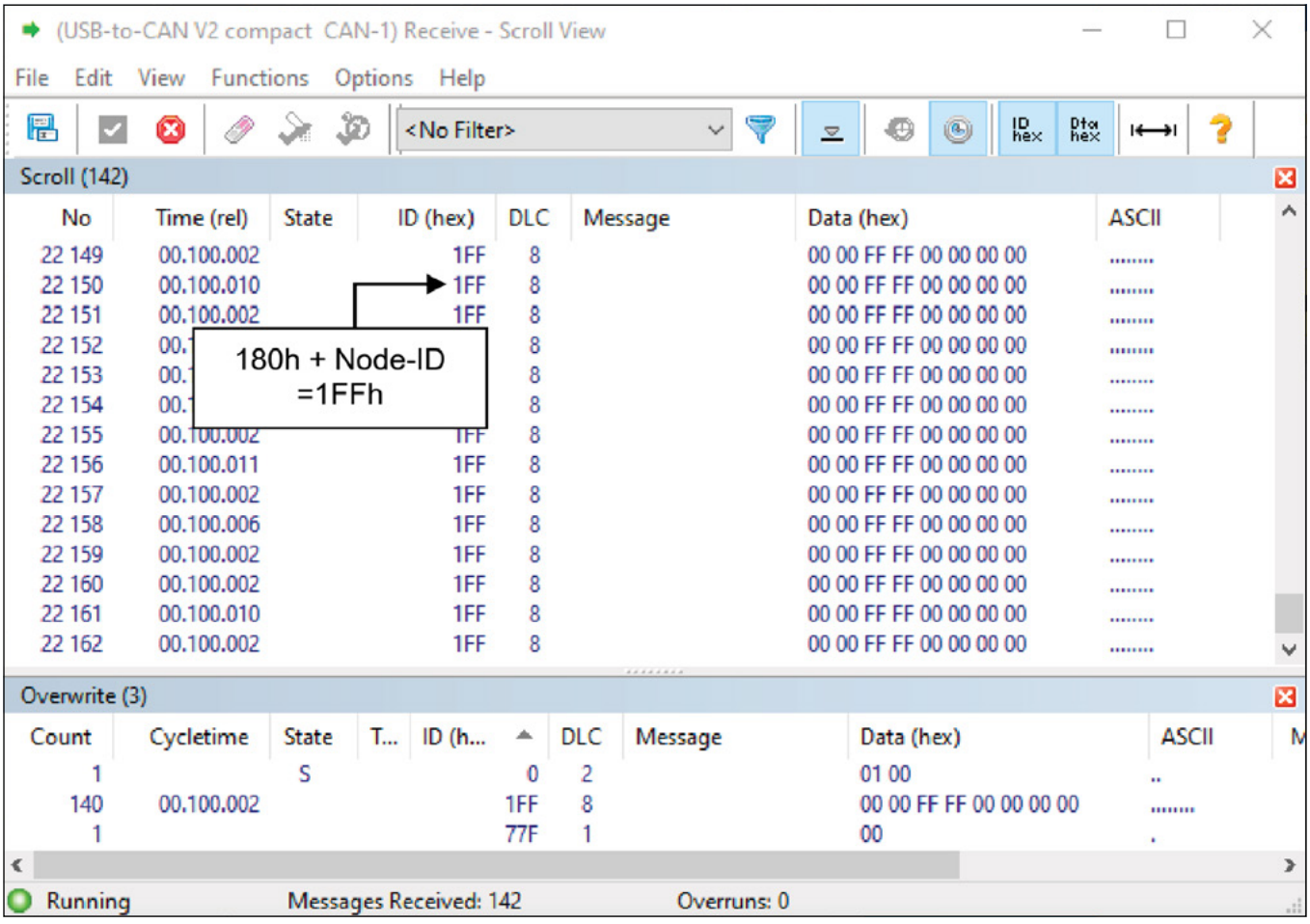


Figure 3. Half-redundant PDO #0 mapping, example 1: CANbus sniffing by using a IXXAT USB-to-CAN V2 compact adapter and the canAnalyser3 software.

- ***PDO #0 mapping, example 2:***

ID	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
1FFh	E4h	08h	1Fh	F7h	00h	00h	00h	00h

Byte 1 (LSB) = E4h
Byte 2 (MSB) = 08h
=> Position Value channel 1 = 08E4hex to decimal 2276d (resolution 1 mm) = 2276 mm

Byte 3 (LSB) = 1Fh
Byte 4 (MSB) = F7h
=> Position Value channel 2 = FFFFh –F71Fh = 2272h to decimal 0d (resolution 1 mm) = 2272 mm

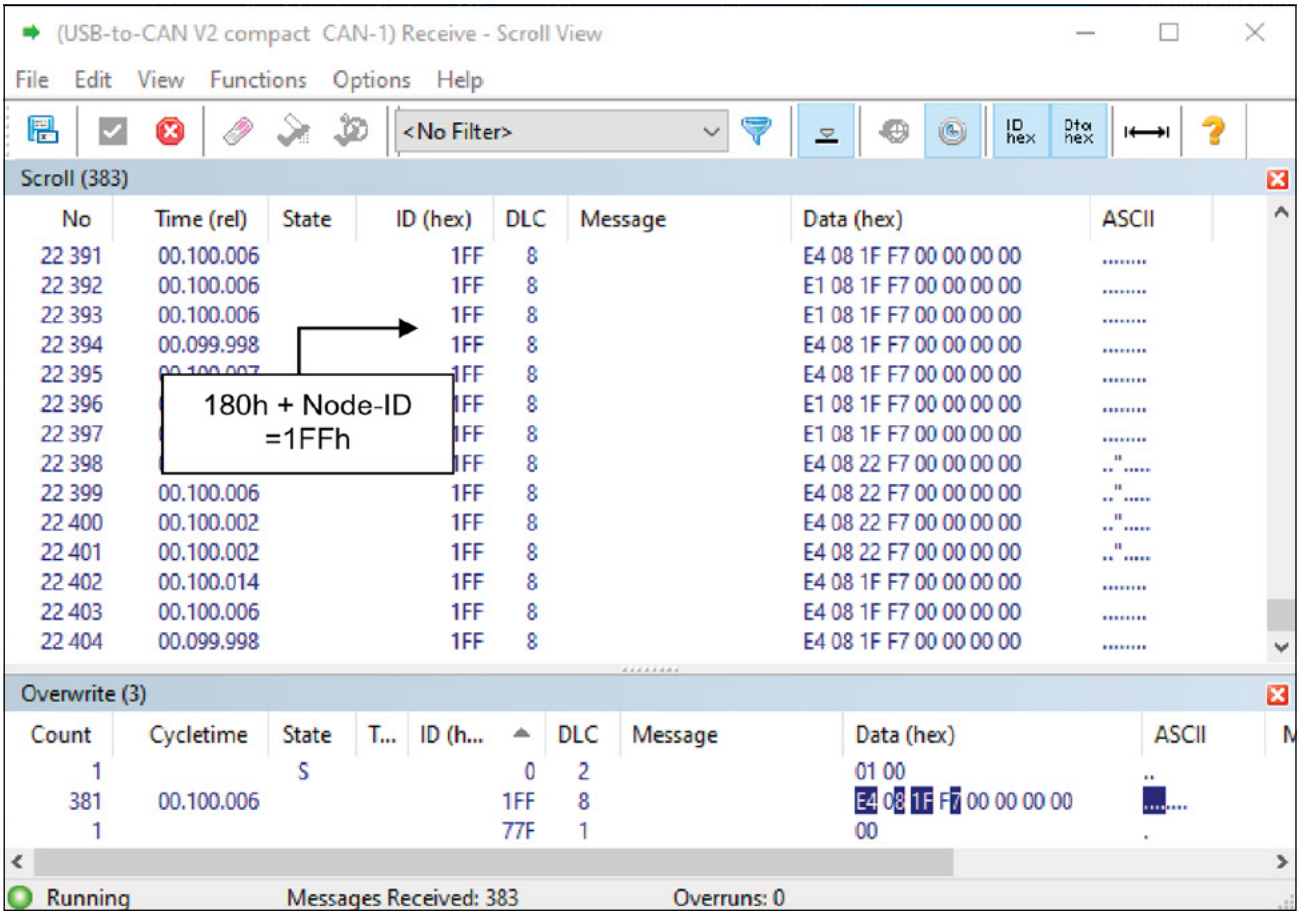


Figure 4. Half-redundant version PDO #0 mapping, example 2: CANbus sniffing by using a IXXAT USB-to-CAN V2 compact adapter and the canAnalyser3 software.

→ *REDUNDANT VERSION*

Please, consider that a redundant GSH-S is a sensor with two independent channels, each of them producing a measurement and transmitting it through PDO communication. Channel 1 and 2 can be considered as two independent sensors, each of them with his Node-ID.
In the following an example of PDO mapping is reported in the case of:

- Node-ID = 7Fh for channel 1, 7Eh for channel 2
- Baude-rate = 250 kBaud
- Redundant version GSH-S
- Linear-encoder Cia406 setting as follow:
    I. Total Measuring Range (object 0x6002 sub-index 0) = 4800 mm (480 steps x 10 mm)
    II. Preset Value (object 0x6003 sub-index 0) = 0 mm (0 steps x 10^3 nm)
    III. Measuring Step (object 0x6005 sub-index 1) = 1 mm (1 000 000 steps x 10^3 nm)
    IV. Position Value (object 0x6004 sub-index 0) indicates the length measurement

• *PDO #0 and PDO#1 mapping, example 1:*

ID	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
1FFh	A3h	0Fh	00h	00h	00h	00h	00h	00h

ID	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
1FEh	A3h	0Fh	00h	00h	00h	00h	00h	00h

Byte 1 (LSB) = A3h
Byte 2 (MSB) = 0Fh
=> Position Value channel 1 = 0FA3h to decimal 4003d (resolution 1 mm) = 4003 mm

Byte 3 (LSB) = A3h
Byte 4 (MSB) = 0Fh
=> Position Value channel 2 = 0FA3h to decimal 4003d (resolution 1 mm) = 4003 mm

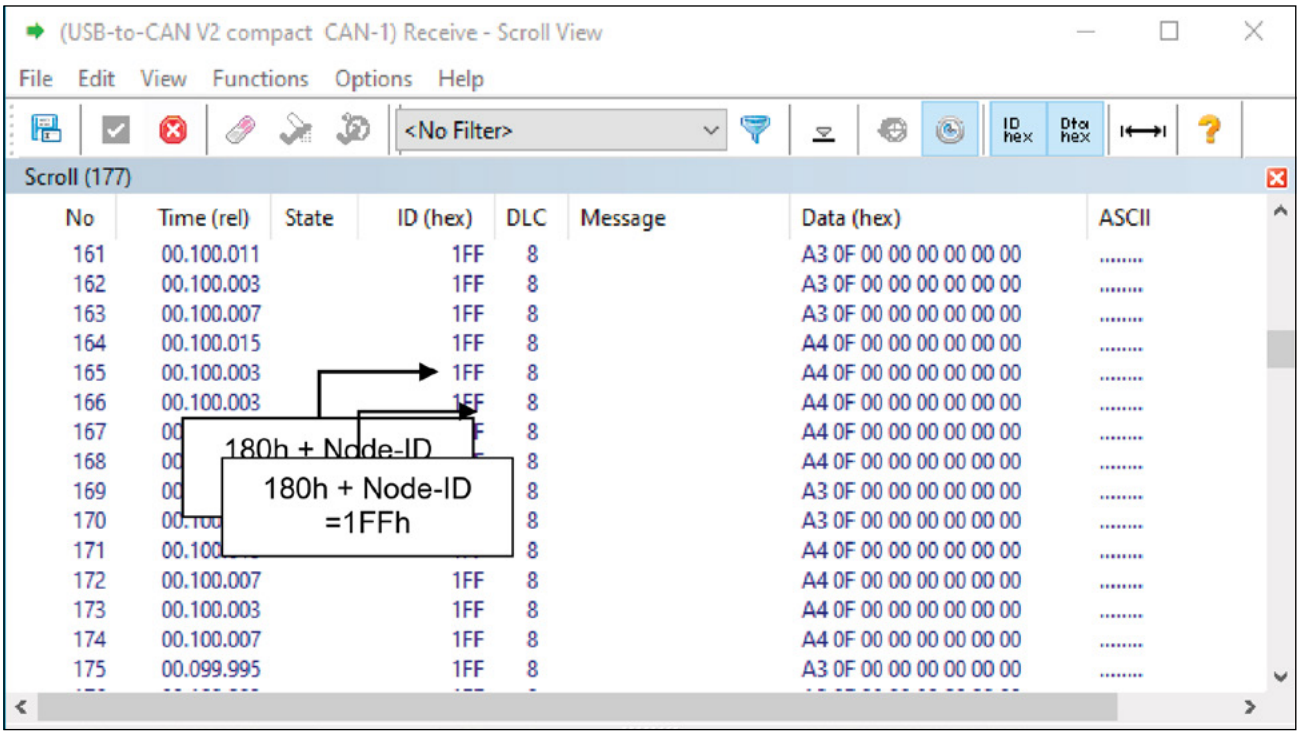


Figure 5. Redundant PDO #0 mapping, example 1: CANbus sniffing by using a IXXAT USB-to-CAN V2 compact adapter and the canAnalyser3 software.

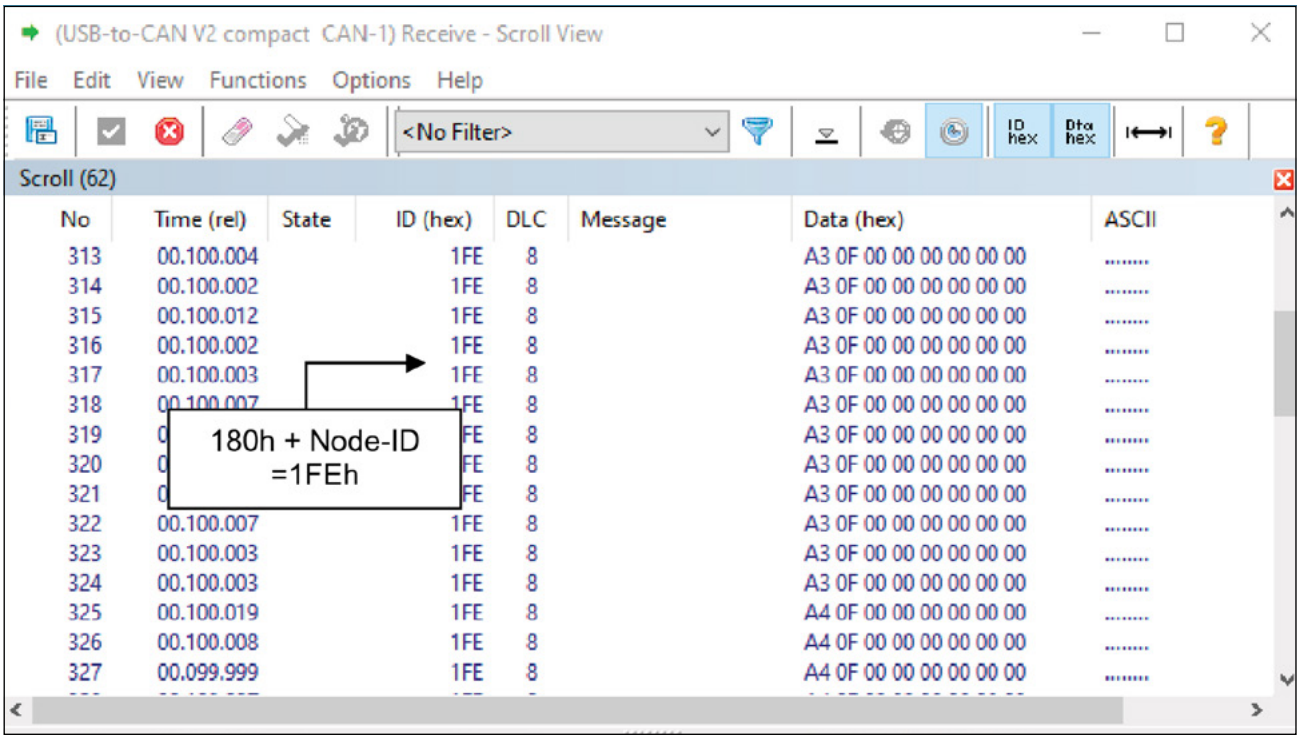


Figure 6. Redundant version PDO #1 mapping, example 1: CANbus sniffing by using a IXXAT USB-to-CAN V2 compact adapter and the canAnalyser3 software.

13. CANopen FEATURES SUMMARY

Communication profile

The parameters which are critical for communication are determined in the Communication profile. This area is common for all CanOpen devices.

Index	Sub-index	Name	Type	ACCESS	DEFAULT	COMMENT
0x1000		Device Type	Unsigned32	ro	0x00080196	Profile 406, absolute linear encoder with one cursor
0x1001		Error Register	Unsigned8	ro	0x00	Always 0.
0x1005		COB-ID SYNC	Unsigned32	rw	0x00000080	
0x1008		Manufacturer Device Name	String	const	“GSH-S ”	Refer to GEFRAN products catalogue: GSH-S
0x1009		Manufacturer Hardware Version	String	const	“1.xx”	“Maj.MinMin”
0x100A		Manufacturer Software Version	String	const	“1.xx”	“Maj.MinMin”
0x1010		Store Parameters				
	0x0	Number of Entries	Unsigned8	ro	4	
	0x1	Save all Parameters	Unsigned32	wo		“save” (0x65766173) to save all parameters
			Unsigned32			(objects with marking PARA).
	0x2	Communication parameters	Unsigned32	rw		“save”. Internally linked to object 0x1010-1.
	0x3	Application parameters	Unsigned32	rw		“save”. Internally linked to object 0x1010-1.
	0x4	Manuf. Specific parameters	Unsigned32	rw		“save”. Internally linked to object 0x1010-1.
0x1011		Restore Parameters				
	0x0	Number of Entries	Unsigned8	ro	4	
	0x1	Load all Parameters	Unsigned32	wo		“load” (0x64616F6C) to restore all parameters (objects with marking PARA and LSS-PARA).
	0x2	Communication parameters	Unsigned32	rw		“load”. Internally linked to object 0x1011-1.
	0x3	Application parameters	Unsigned32	rw		“load”. Internally linked to object 0x1011-1.
	0x4	Manuf. Specific parameters	Unsigned32	rw		“load”. Internally linked to object 0x1011-1.
0x1014		COB-ID EMCY	Unsigned32	ro	0x80 + Node-ID	
0x1017		Producer Heartbeat Time - PARA	Unsigned16	rw	0	0: not used heartbeat; 1..19 not accept; 20..65535 accept
0x1018		Identity Object				
	0x0	Number of Entries	Unsigned8	ro	4	
	0x1	Vendor ID	Unsigned32	ro	0x00000093	Refer to “Gefran Product Overview CANopen”.
						Gefran Vendor ID: 0x00000093
	0x2	Product Code	Unsigned32	ro	0x00000067	Refer to “Gefran Product Overview CANopen”.
	0x3	Revision Number	Unsigned32	ro	0x00000001	Major Revision Number: Bit16...31
			Unsigned32			Minor Revision Number: Bit0...15
	0x4	Serial Number	Unsigned32	ro	0x00000000	
0x1200		SDO #0 Server Parameter				
	0x0	Number of Entries	Unsigned8	ro	2	
	0x1	COB-ID Client->Server (Rx)	Unsigned32	ro	0x600 + Node-ID	

Index	Sub-index	Name	Type	ACCESS	DEFAULT	COMMENT
	0x2	COB-ID Server->Client (Tx)	Unsigned32	ro	0x580 + Node-ID	
0x1800		Tx PDO #0 Communication Parameter				
	0x0	Number of Entries	Unsigned8	ro	5	
	0x1	COB-ID	Unsigned32	ro	0x180 + Node-ID	
	0x2	Transmission Type	Unsigned8	rw	254	1..240: Synchronous Transmission
			Unsigned8			254: Asynchronous transmission, transmission event is manufacturer specific, i.e. event timer.
	0x3	Inhibit Time	Unsigned16	rw	0x0004	Minimum interval time for PDO #0
	0x4	Reserved	Unsigned8	ro	0	
	0x5	Event Timer - PARA	Unsigned16	rw	100	In the EDS file this default value must be set to 0, otherwise errors will be generated.
0x1801		Tx PDO #1 Communication Parameter				
	0x0	Number of Entries	Unsigned8	ro	5	
	0x1	COB-ID	Unsigned32	ro	0x280 + Node-ID	
	0x2	Transmission Type	Unsigned8	rw	254	1..240: Synchronous Transmission
						254: Asynchronous transmission, transmission event is manufacturer specific, i.e. event timer.
	0x3	Inhibit Time	Unsigned16	rw	0x0004	Minimum interval time for PDO #1
	0x4	Reserved	Unsigned8	ro	0	
	0x5	Event Timer - PARA	Unsigned16	rw	0	In the EDS file this default value must be set to 0, otherwise errors will be generated.
0x1A00		Tx PDO #0 Mapping Parameter				
	0x0	Number of Entries	Unsigned8	ro	2	
	0x1	1st Object	Unsigned32	rw	0x60040020	PDO mapping for the 1st application object
						Possible Value 0x60040020, 0x60200120, 0x60200320, 0x60200510, 0x60200710, 0x60200910, 0x60200B10
	0x2	2nd Object	Unsigned32	rw	0x00000000	PDO mapping for the 2nd application object
						Possible Value 0x60200220, 0x60200420, 0x60200610, 0x60200810, 0x60200A10, 0x60200C10, 0x00000000
0x1A01		Tx PDO #1 Mapping Parameter				
	0x0	Number of Entries	Unsigned8	ro	2	
	0x1	1st Object	Unsigned32	rw	0x60040020	PDO mapping for the 1st application object
						Possible Value 0x60040020, 0x60200120, 0x60200320, 0x60200510, 0x60200710, 0x60200910, 0x60200B10
	0x2	2nd Object	Unsigned32	rw	0x00000000	PDO mapping for the 2nd application object
						Possible Value 0x60200220, 0x60200420, 0x60200610, 0x60200810, 0x60200A10, 0x60200C10, 0x00000000

Manufacturer Specific Profile Objects

Index	Sub-index	Name	Type	ACCESS	DEFAULT	COMMENT
0x20F0		Setting of the Node ID - A	U8	rw	0x7F	The node ID used to access the sensor in the CANopen network
0x20F1		Setting of the Node ID - B	U8	rw	0x7F	The node ID used to access the sensor in the CANopen network
0x20F2		Setting of the Baud Rate - A	U8	rw	0x03	Baud rate of the CAN network Baud rate 0 = 1000 kBaud 1 = 800 kBaud 2 = 500 kBaud 3 = 250 kBaud 4 = 125 kBaud (default) 5 = 100 kBaud 6 = 50 kBaud 7 = 20 kBaud
0x20F3		Setting of the Baud Rate - B	U8	rw	0x03	Baud rate of the CAN network Baud rate 0 = 1000 kBaud 1 = 800 kBaud 2 = 500 kBaud 3 = 250 kBaud 4 = 125 kBaud (default) 5 = 100 kBaud 6 = 50 kBaud 7 = 20 kBaud
0x2100		Setting of the Baud Rate	U8	rw	0x03	Baud rate of the CAN network Baud rate 0 = 1000 kBaud 1 = 800 kBaud 2 = 500 kBaud 3 = 250 kBaud 4 = 125 kBaud (default) 5 = 100 kBaud 6 = 50 kBaud 7 = 20 kBaud
0x2101		Setting of the Node ID	U8	rw	0x7F	The node ID used to access the sensor in the CANopen network
0x3000		Setting of the Baud Rate	U8	rw	0x03	Baud rate of the CAN network Baud rate 0 = 1000 kBaud 1 = 800 kBaud 2 = 500 kBaud 3 = 250 kBaud 4 = 125 kBaud (default) 5 = 100 kBaud 6 = 50 kBaud 7 = 20 kBaud
0x3001		Setting of the Node ID	U8	rw	0x7F	The node ID used to access the sensor in the CANopen network
0x4000		Error Behavior - PARA	U8	rw	1	0: pre-operational 1: no state change 2: stopped
0x4001		Error Code	U8	ro	0	0: no error
0x5000		Automatic NMT Start after Power On – PARA	U8	rw	0	0: not activated 1: activated
0x599A		Serial Number - INTERNAL				
	0x0	Number of Entries	U8	ro	3	
	0x1	Serial Number Bit15 Bit0	U16	rw	0x0000	
	0x2	Serial Number Bit31 Bit16	U16	rw	0x0000	
	0x3	Save Serial Number	U32	wo		“save” (0x65766173) to save Serial Number

Manufacturer Specific Profile Objects (according to CiA DS-406)

In this section you will find the manufacturer specific profile indices for the transducer as LINEAR ENCODER.

Index	Sub-index	Name	Type	ACCESS	DEFAULT	COMMENT
0x6000		Operating Parameters – PARA	Unsigned16	ro	0x0000	
0x6002		Total Measuring Range	Unsigned32	ro	0x00000320	Measuring range in 10 mm steps
0x6003		Preset Value – PARA	Signed32	rw	0x00000000	The Preset function supports adaption of the GSH-S zero point to the mechanical zero point of the system. The output value is the position value decreased in the preset value stored in the GSH-S.
0x6004		Position Value	Unsigned32	ro	0x00000000	The object 6004h „Position value“defines the output position value for the communication objects 1800h.
0x6005		Position step settings – PARA				
	0x0	Number of Entries	UNSIGNED8	ro	1	
	0x1	Position step settings [nm/Step]	Unsigned32	rw	0x000F4240	Position step setting in nm. ANY VALUE is ALLOWED, it is used as a scale coefficient of the raw position measurement. Examples: - 100000 tenths of a millimeter; - 1000000 for millimeter; - 10000000 per centimeter; - 254000 to hundredths of an inch
0x6010		Preset Value for encoder – PARA				
	0x0	Number of Entries	UNSIGNED8	ro	2	
	0x1	Preset for encoder A [Step]	S32	rw	0x00000000	Present in steps for encoder A. Internally linked to object 0x6003-0.
	0x2	Preset for encoder B [Step]	S32	rw	0x00000000	Present in steps for encoder B
0x6020		Position Value for encoder				
	0x0	Number of Entries	UNSIGNED8	ro	8	
	0x1	Position encoder A [Step]	Signed32	ro	0x00000000	Position in steps for encoder A
	0x2	Position encoder B [Step]	Signed32	ro	0x00000000	Position in steps for encoder B
	0x3	Position encoder A [Step]	Unsigned32	ro	0x00000000	Position in steps for encoder A
	0x4	Position encoder B [Step]	Unsigned32	ro	0x00000000	Position in steps for encoder B
	0x5	Position encoder A [Step]	Signed16	ro	0x0000	Position in steps for encoder A
	0x6	Position encoder B [Step]	Signed16	ro	0x0000	Position in steps for encoder B
	0x7	Position encoder A c1 [Step]	Signed16	ro	0x0000	Position in steps for encoder A - ones' complement
	0x8	Position encoder B c1 [Step]	Signed16	ro	0x0000	Position in steps for encoder B - ones' complement
	0x9	Position encoder A [Step]	Unsigned16	ro	0x0000	Position in steps for encoder A
	0xA	Position encoder B [Step]	Unsigned16	ro	0x0000	Position in steps for encoder B
	0xB	Position encoder A c1 [Step]	Unsigned16	ro	0x0000	Position in steps for encoder A - ones' complement
	0xC	Position encoder B c1 [Step]	Unsigned16	ro	0x0000	Position in steps for encoder B - ones' complement

Note: The access to CanOpen variables can be ro (the parameter can be read only), rw (the parameter can be read and also written), wo (the parameter can be written only) or const (the parameter is always constant).

14. COMMUNICATION EXAMPLES

Example 1) How to change the Baud Rate Setting from 250 kbaud (current setting) to 500 kbaud

With Service Data Object (S.D.O.) the access to entries of a device Object Dictionary is provided. As these entries may contain data of arbitrary size and data type SDOs can be used to transfer multiple data sets from a client to a server and vice versa.

Structure of SDO-request by the Master:

COB-ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
600+Node-ID	8	CMD	Index		Sub-Index	Data	Data	Data	Data

CMD filed determines the direction of data transfer and the size of the data object:

23h Sending of 4-byte data (bytes 5...8 contain a 32-bit value)
2Bh Sending of 2-byte data (bytes 5, 6 contain a 16-bit value)
2Fh Sending of 1-byte data (byte 5 contains an 8-bit value)

Each access to the object dictionary is checked by the slave for validity. Any write access to non-existing objects, to read-only objects or with a non-corresponding data format are rejected and answered with a corresponding error message.

Structure of SDO-answer by the Slave:

COB-ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
580+Node-ID	8	RES	Index		Sub-Index	Data	Data	Data	Data

RES field determines the correct/incorrect answer of the slave:

60h Data sent successfully
80h Error

To change the Baud rate from the standard value 250 kBaud (03) to 500 kBaud (02), write the new Baud rate in the object 0x20F2 and 0x20F3 by using the following SDO message. Then send the Save all parameters SDO and perform a sensor reset.

SDO request 1:

COB-ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
67F	8	2Fh	F2h	20h	00h	02h	00h	00h	00h

The answers after successful storing you will receive is:

SDO answer 1:

COB-ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
5FFh	8	60h	F2h	20h	00h	00h	00h	00h	00h

SDO request 2:

COB-ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
67F	8	2Fh	F3h	20h	00h	02h	00h	00h	00h

The answers after successful storing you will receive is:

SDO answer 2:

COB-ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
5FFh	8	60	F3h	20h	00h	00h	00h	00h	00h

In order to save all new customized parameters, send the following SDO message. Then, turn-off and turn-on the power supply to le the sensor perform the reset.

SDO request 3 (SAVE):

COB-ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
67F	8	23h	10h	10h	01h	73h	61h	76h	65h

The answers after successful storing you will receive is:

SDO answer 3:

COB-ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
5FFh	8	60h	10h	10h	01h	00h	00h	00h	00h

Object:

20F2h	0	Setting of the Baude rate	Unsigned 8	rw	0x03 (250 kBaud)	Baud rate of the CAN network 0 = 1000 kBaud 1 = 800 kBaud 2 = 500 kBaud 3 = 250 kBaud (default) 4 = 125 kBaud 5 = 100 kBaud 6 = 50 kBaud 7 = 20 kBaud
20F3h	0	Setting of the Baude rate	Unsigned 8	rw	0x03 (250 kBaud)	Baud rate of the CAN network 0 = 1000 kBaud 1 = 800 kBaud 2 = 500 kBaud 3 = 250 kBaud (default) 4 = 125 kBaud 5 = 100 kBaud 6 = 50 kBaud 7 = 20 kBaud

The supported baud rate are listed in the following table:

Byte 5	**Baudrate**
07h	20 kBaud
06h	50 kBaud
05h	100 kBaud
04h	125 kBaud
03h	250 kBaud
02h	500 kBaud
01h	800 kBaud
00h	1000 kBaud

IMPORTANT NOTE:

A change of the Baud rate is only accepted if the entries 0x20F2 and 0x20F3 contain the same changed value. Values above 7 are not accepted; the existing setting remains valid. After setting the new entries a reset must be made so that the new entries become valid (switch off the module for a short time).

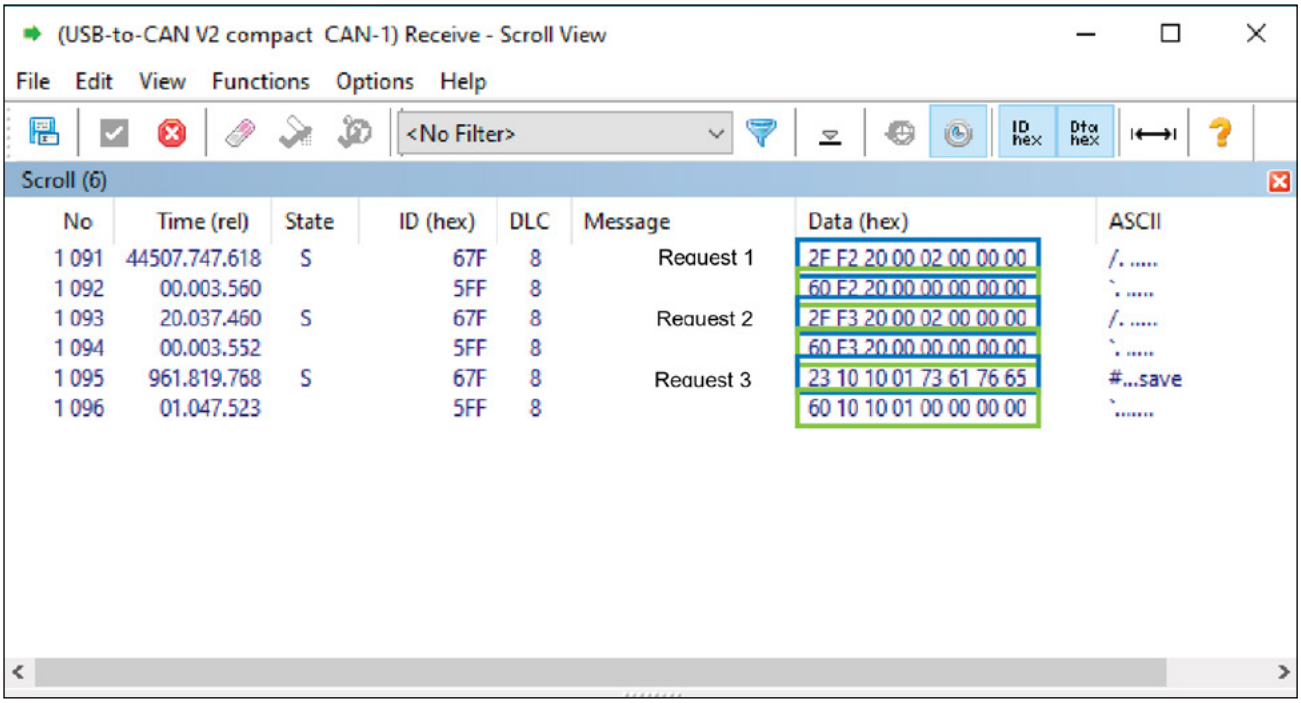


Figure 7. Example 1 How to change the Baud rate from 250 kbaud (default setting) to 500 kbaud

Example 2) How to change the Node-ID from 0x7Fh (127d, current setting) to 0x06h (06d)

With Service Data Object (S.D.O.) the access to entries of a device Object Dictionary is provided. As these entries may contain data of arbitrary size and data type SDOs can be used to transfer multiple data sets from a client to a server and vice versa.

Structure of SDO-request by the Master:

COB-ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
600+Node-ID	8	CMD	Index		Sub-Index	Data	Data	Data	Data

CMD filed determines the direction of data transfer and the size of the data object:

23h Sending of 4-byte data (bytes 5...8 contain a 32-bit value)
2Bh Sending of 2-byte data (bytes 5, 6 contain a 16-bit value)
2Fh Sending of 1-byte data (byte 5 contains an 8-bit value)

Each access to the object dictionary is checked by the slave for validity. Any write access to non-existing objects, to read-only objects or with a non-corresponding data format are rejected and answered with a corresponding error message.

Structure of SDO-answer by the Slave:

COB-ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
580+Node-ID	8	RES	Index		Sub-Index	Data	Data	Data	Data

RES field determines the correct/incorrect answer of the slave:

60h Data sent successfully
80h Error

To change the Node-ID from the standard value 7Fh to 06h, write the new Node-ID in the object 3001h by using the following SDO message. Then send the Save all parameters SDO and perform a sensor reset.

SDO request 1:

COB-ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
67F	8	2Fh	01h	30h	00h	06h	00h	00h	00h

The answers after successful storing you will receive is:

SDO answer 1:

COB-ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
5FFh	8	60	01h	30h	00h	00h	00h	00h	00h

In order to save all new customized parameters, send the following SDO message. Then, turn-off and turn-on the power supply to le the sensor perform the reset.

SDO request 2 (SAVE):

COB-ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
67F	8	23h	10h	10h	01h	73h	61h	76h	65h

The answers after successful storing you will receive is:

answer 2:

COB-ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
5FFh	8	60h	10h	10h	01h	00h	00h	00h	00h

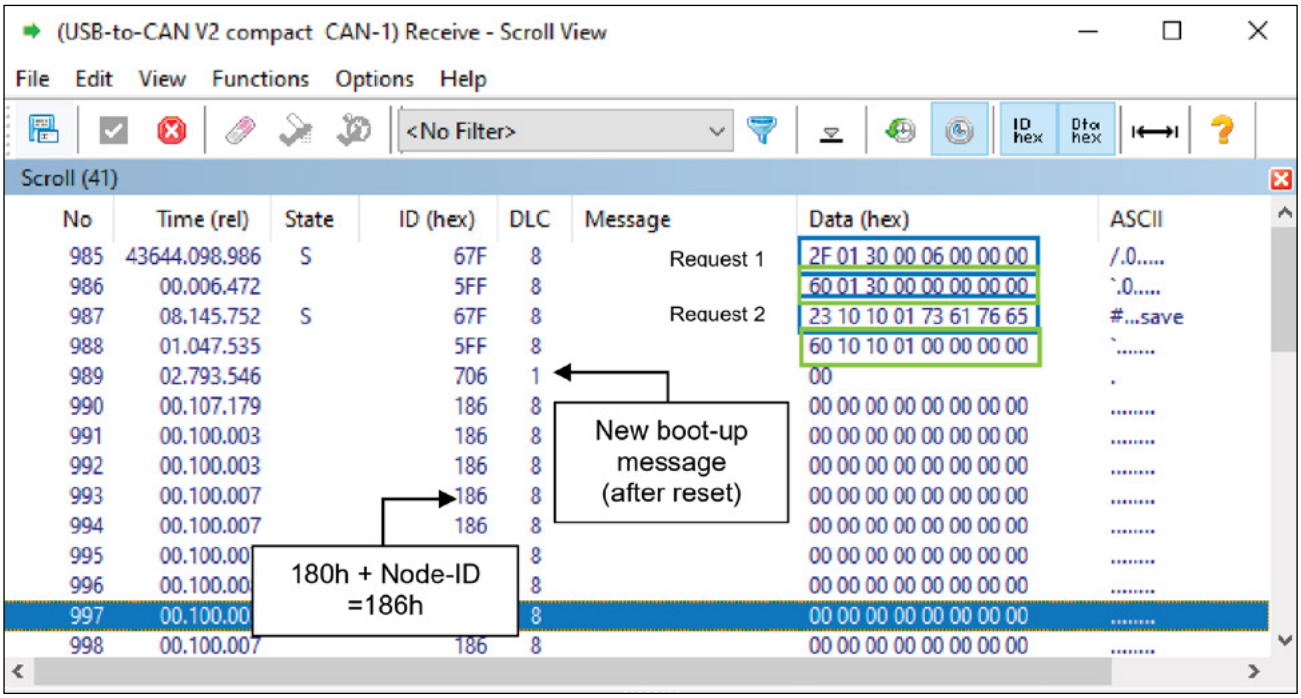


Figure 8. Example 2 How to change the Node-ID from 0x7Fh (127d, current setting) to 0x06h (06d)

Example 3) How to change the PDO rate (time interval) from 100 ms (current setting) to 1000 ms

With Service Data Object (S.D.O.) the access to entries of a device Object Dictionary is provided. As these entries may contain data of arbitrary size and data type SDOs can be used to transfer multiple data sets from a client to a server and vice versa.

Structure of SDO-request by the Master:

COB-ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
600+Node-ID	8	CMD	Index		Sub-Index	Data	Data	Data	Data

CMD filed determines the direction of data transfer and the size of the data object:

23h Sending of 4-byte data (bytes 5...8 contain a 32-bit value)
2Bh Sending of 2-byte data (bytes 5, 6 contain a 16-bit value)
2Fh Sending of 1-byte data (byte 5 contains an 8-bit value)

Each access to the object dictionary is checked by the slave for validity. Any write access to non-existing objects, to read-only objects or with a non-corresponding data format are rejected and answered with a corresponding error message.

Structure of SDO-answer by the Slave:

COB-ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
580+Node-ID	8	RES	Index		Sub-Index	Data	Data	Data	Data

RES field determines the correct/incorrect answer of the slave:

60h Data sent successfully
80h Error

To change the PDO event timer from the standard 100ms to 1000ms, write the new event timer in the object 1800h sub-index 5 by using the following SDO message. Then send the Save all parameters SDO and perform a sensor reset.

SDO request 1:

COB-ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
67F	8	2Bh	00h	18h	05h	E8h	03h	00h	00h

The answers after successful storing you will receive is:

SDO answer 1:

COB-ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
5FFh	8	60h	00h	18h	05h	00h	00h	00h	00h

In order to save all new customized parameters, send the following SDO message. Then, turn-off and turn-on the power supply to le the sensor perform the reset.

SDO request 2 (SAVE):

COB-ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
67F	8	23h	10h	10h	01h	73h	61h	76h	65h

The answers after successful storing you will receive is:

SDO answer 2:

COB-ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
5FFh	8	60h	10h	10h	01h	00h	00h	00h	00h

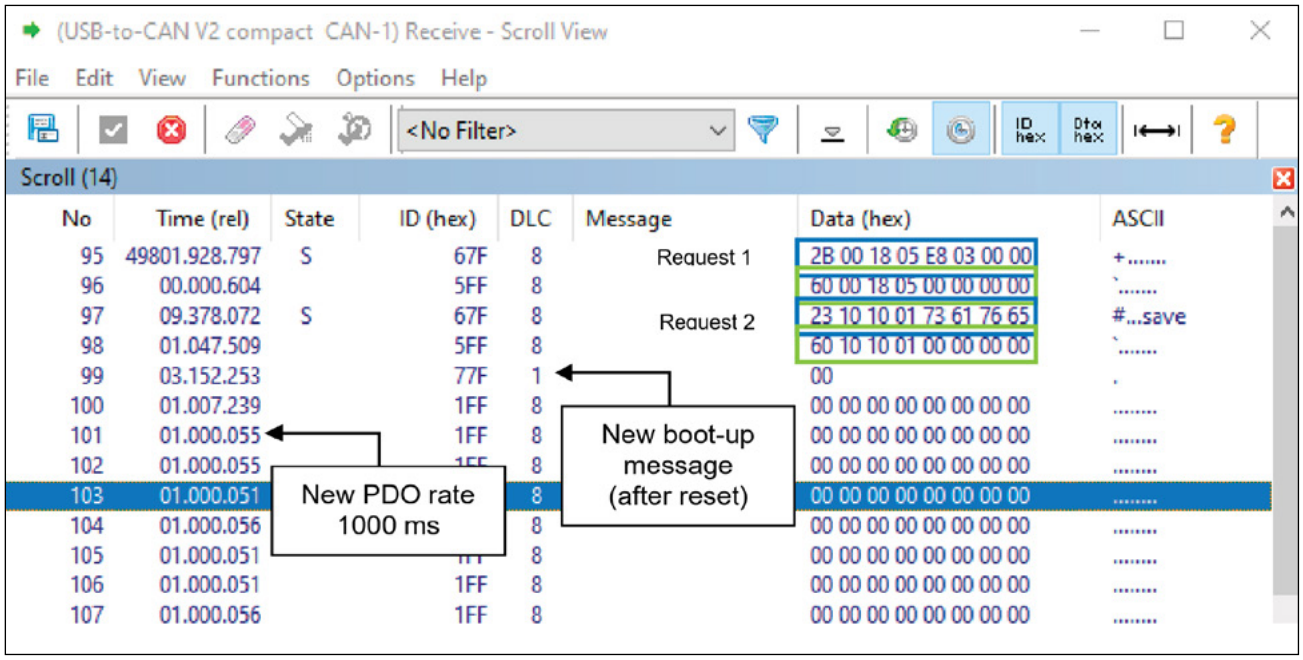


Figure 9. Example 3 How to change the PDO rate (time interval) from 100 ms (current setting) to 1000 ms

Example 4) How to de-activate an automatic NMT Start after Power ON (the PDO will not be sendt automatically after power ON)

With Service Data Object (S.D.O.) the access to entries of a device Object Dictionary is provided. As these entries may contain data of arbitrary size and data type SDOs can be used to transfer multiple data sets from a client to a server and vice versa.

Structure of SDO-request by the Master:

COB-ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
600+Node-ID	8	CMD	Index		Sub-Index	Data	Data	Data	Data

CMD filed determines the direction of data transfer and the size of the data object:

23h Sending of 4-byte data (bytes 5...8 contain a 32-bit value)
2Bh Sending of 2-byte data (bytes 5, 6 contain a 16-bit value)
2Fh Sending of 1-byte data (byte 5 contains an 8-bit value)

Each access to the object dictionary is checked by the slave for validity. Any write access to non-existing objects, to read-only objects or with a non-corresponding data format are rejected and answered with a corresponding error message.

Structure of SDO-answer by the Slave:

COB-ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
580+Node-ID	8	RES	Index		Sub-Index	Data	Data	Data	Data

RES field determines the correct/incorrect answer of the slave:

60h Data sent successfully
80h Error

To change the automatic NMT start after power on configuration, write the enable value (0=not activated, 1=activated) in the object 5000h sub-index 0 by using the following SDO message.
Then send the Save all parameters SDO and perform a sensor reset.

SDO request 1:

COB-ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
67F	8	2Fh	00h	50h	00h	00h	00h	00h	00h

The answers after successful storing you will receive is:

SDO answer 1:

COB-ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
5FFh	8	60h	00h	50h	00h	00h	00h	00h	00h

In order to save all new customized parameters, send the following SDO message. Then, turn-off and turn-on the power supply to le the sensor perform the reset.

SDO request 2 (SAVE):

COB-ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
67F	8	23h	10h	10h	01h	73h	61h	76h	65h

The answers after successful storing you will receive is:

SDO answer 2:

COB-ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
5FFh	8	60h	10h	10h	01h	00h	00h	00h	00h

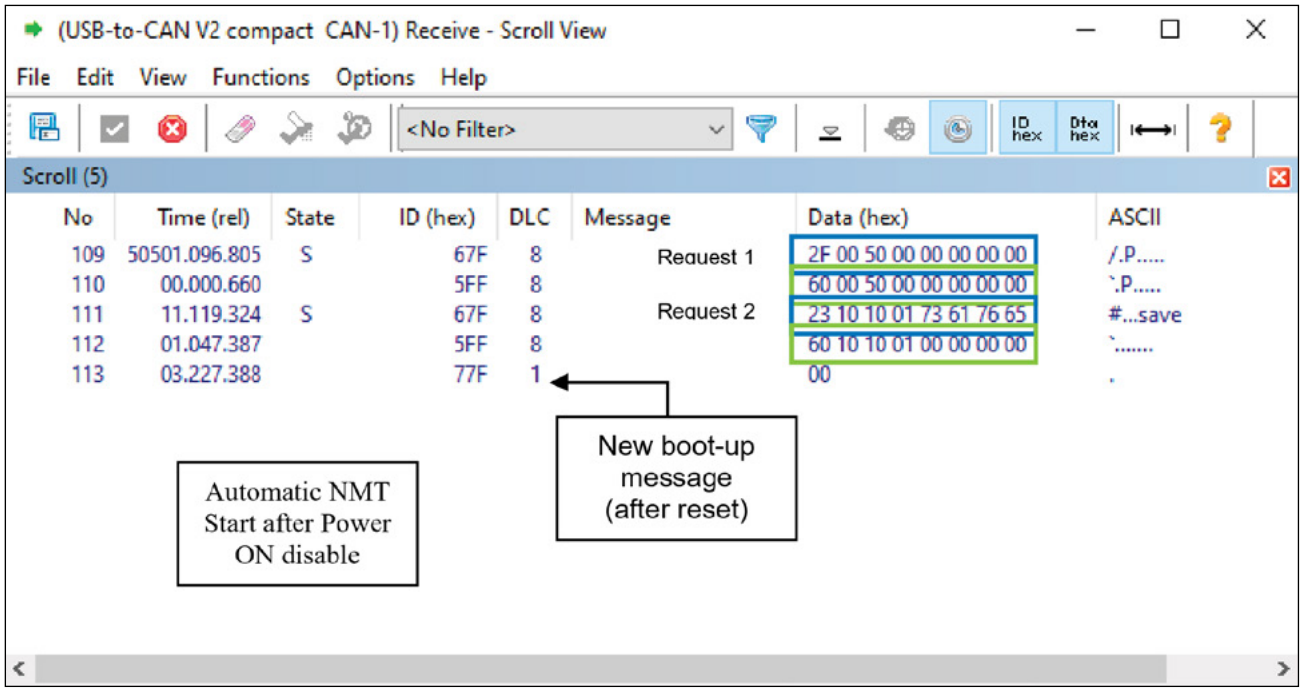


Figure 10. Example 4 How to de-activate an automatic NMT Start after Power ON

Example 5) How to change the position step setting (resolution)

With Service Data Object (S.D.O.) the access to entries of a device Object Dictionary is provided. As these entries may contain data of arbitrary size and data type SDOs can be used to transfer multiple data sets from a client to a server and vice versa.

Structure of SDO-request by the Master:

COB-ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
600+Node-ID	8	CMD	Index		Sub-Index	Data	Data	Data	Data

CMD filed determines the direction of data transfer and the size of the data object:

23h Sending of 4-byte data (bytes 5...8 contain a 32-bit value)
2Bh Sending of 2-byte data (bytes 5, 6 contain a 16-bit value)
2Fh Sending of 1-byte data (byte 5 contains an 8-bit value)

Each access to the object dictionary is checked by the slave for validity. Any write access to non-existing objects, to read-only objects or with a non-corresponding data format are rejected and answered with a corresponding error message.

Structure of SDO-answer by the Slave:

COB-ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
580+Node-ID	8	RES	Index		Sub-Index	Data	Data	Data	Data

RES field determines the correct/incorrect answer of the slave:

60h Data sent successfully
80h Error

To change the position step setting from the default 1 mm (1000000 nm) to 0,1 mm (100000 nm), write the new position step in the object 6005h sub-index 01 by using the following SDO message.
Then send the Save all parameters SDO and perform a sensor reset.

SDO request 1:

COB-ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
67F	8	23h	05h	60h	01h	A0h	86h	01h	00h

The answers after successful storing you will receive is:

SDO answer 1:

COB-ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
5FFh	8	60h	05h	60h	01h	00h	00h	00h	00h

In order to save all new customized parameters, send the following SDO message. Then, turn-off and turn-on the power supply to le the sensor perform the reset.

SDO request 2 (SAVE):

COB-ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
67F	8	23h	10h	10h	01h	73h	61h	76h	65h

The answers after successful storing you will receive is:

SDO answer 2:

COB-ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
5FFh	8	60h	10h	10h	01h	00h	00h	00h	00h

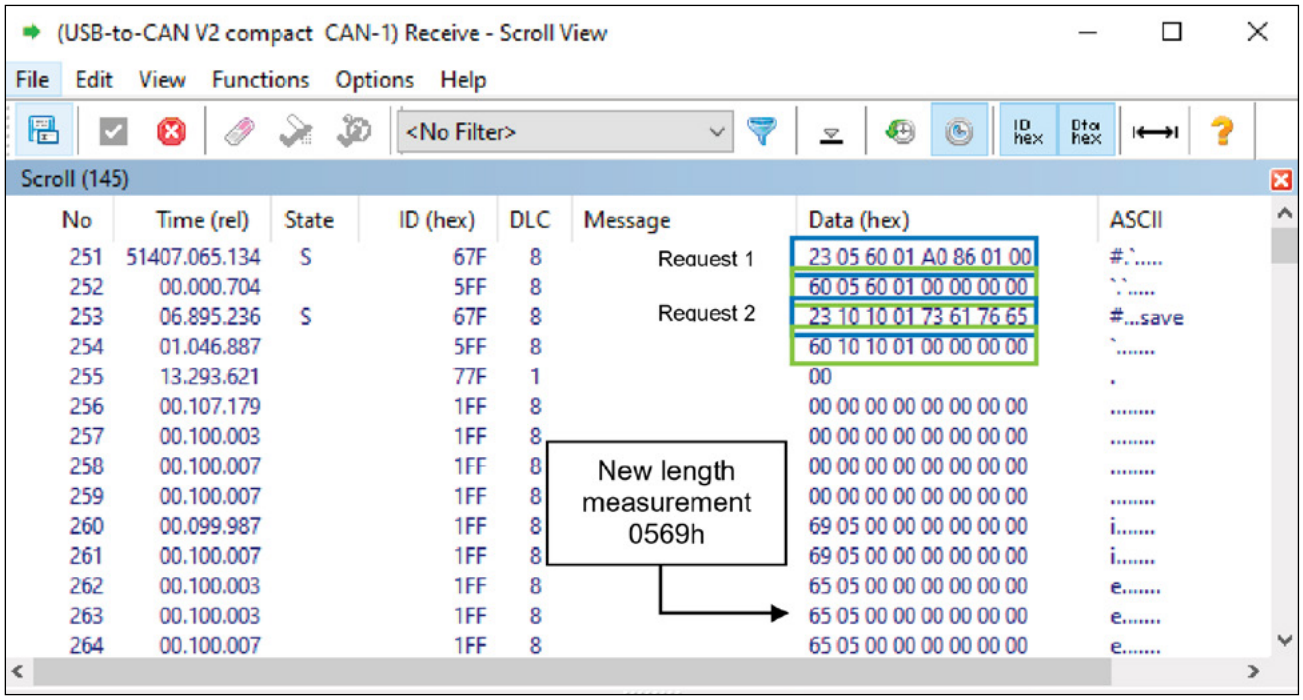


Figure 11. Example 5 How to change the position step setting (resolution)

Byte 1 (LSB) = 69h
Byte 2 (MSB) = 05h
=> Position Value = 0569h to decimal 1385d (resolution 0.1 mm) = 138.5 mm

Example 6) How to Preset the zero Position Value

With Service Data Object (S.D.O.) the access to entries of a device Object Dictionary is provided. As these entries may contain data of arbitrary size and data type SDOs can be used to transfer multiple data sets from a client to a server and vice versa.

Structure of SDO-request by the Master:

COB-ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
600+Node-ID	8	CMD	Index		Sub-Index	Data	Data	Data	Data

CMD filed determines the direction of data transfer and the size of the data object:

23h Sending of 4-byte data (bytes 5...8 contain a 32-bit value)
2Bh Sending of 2-byte data (bytes 5, 6 contain a 16-bit value)
2Fh Sending of 1-byte data (byte 5 contains an 8-bit value)

Each access to the object dictionary is checked by the slave for validity. Any write access to non-existing objects, to read-only objects or with a non-corresponding data format are rejected and answered with a corresponding error message.

Structure of SDO-answer by the Slave:

COB-ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
580+Node-ID	8	RES	Index		Sub-Index	Data	Data	Data	Data

RES field determines the correct/incorrect answer of the slave:

60h Data sent successfully
80h Error

To set the zero position, move the sensor to the desired position and then write the value 00000000h in the object 6010h sub-index 01.
Then send the Save all parameters SDO and perform a sensor reset.
To set the zero position for both encoders (i.e. semi-redundant version), move the sensor to the desired position and then write the value 00000000h in the object 6010h sub-index 01 and sub-index 02 by using the following SDO messages.

SDO request 1:

COB-ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
67F	8	23h	10h	60h	01h	00h	00h	00h	00h

The answers after successful storing you will receive is:

SDO answer 1:

COB-ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
5FFh	8	60h	10h	60h	01h	00h	00h	00h	00h

In case of semi-redundant version, an additional SDO request has to be send:

SDO request 2:

COB-ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
67F	8	23h	10h	60h	02h	00h	00h	00h	00h

The answers after successful storing you will receive is:

SDO answer 2:

COB-ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
5FFh	8	60h	10h	60h	02h	00h	00h	00h	00h

In order to save all new customized parameters, send the following SDO message. Then, turn-off and turn-on the power supply to le the sensor perform the reset.

SDO request 2/3 (SAVE):

COB-ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
67F	8	23h	10h	10h	01h	73h	61h	76h	65h

The answers after successful storing you will receive is:

SDO answer 2/3:

COB-ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
5FFh	8	60h	10h	10h	01h	00h	00h	00h	00h

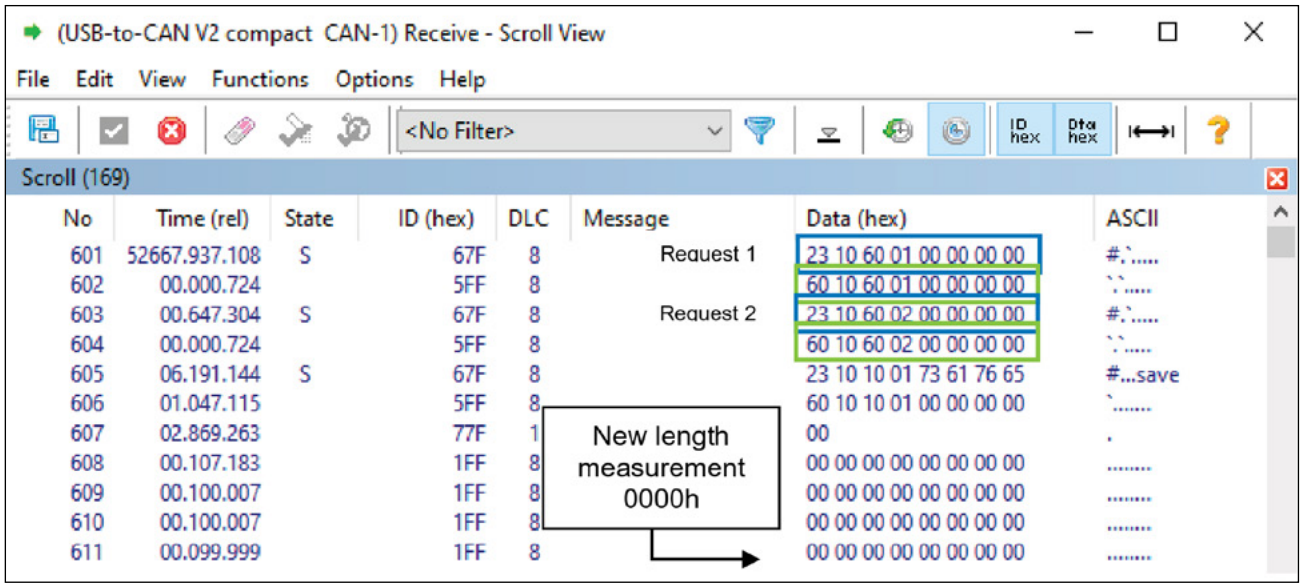


Figure 12. Example 6 How to Preset the zero Position Value (via object 0x6003.0) to 0 mm

Byte 1 (LSB) = 00h
Byte 2 (MSB) = 00h
=> Position Value = 0000h to decimal 0d (resolution 1 mm) = 0 mm
=> zero as be performed

Example 7) How to send the command RESTORE

With Service Data Object (S.D.O.) the access to entries of a device Object Dictionary is provided. As these entries may contain data of arbitrary size and data type SDOs can be used to transfer multiple data sets from a client to a server and vice versa.

Structure of SDO-request by the Master:

COB-ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
600+Node-ID	8	CMD	Index		Sub-Index	Data	Data	Data	Data

CMD filed determines the direction of data transfer and the size of the data object:

23h Sending of 4-byte data (bytes 5...8 contain a 32-bit value)
2Bh Sending of 2-byte data (bytes 5, 6 contain a 16-bit value)
2Fh Sending of 1-byte data (byte 5 contains an 8-bit value)

Each access to the object dictionary is checked by the slave for validity. Any write access to non-existing objects, to read-only objects or with a non-corresponding data format are rejected and answered with a corresponding error message.

Structure of SDO-answer by the Slave:

COB-ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
580+Node-ID	8	RES	Index		Sub-Index	Data	Data	Data	Data

RES field determines the correct/incorrect answer of the slave:

60h Data sent successfully
80h Error

To restore all the default parameters (replacing the customized user parameters with factory parameters), send the following SDO message.
Then, turn-off and turn on the power supply to le the sensor perform the reset.

SDO request:

COB-ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
67F	8	23h	11h	10h	01h	6Ch	6Fh	61h	64h

The answers after successful storing you will receive is:

SDO answer:

COB-ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
5FFh	8	60h	11h	10h	01h	00h	00h	00h	00h

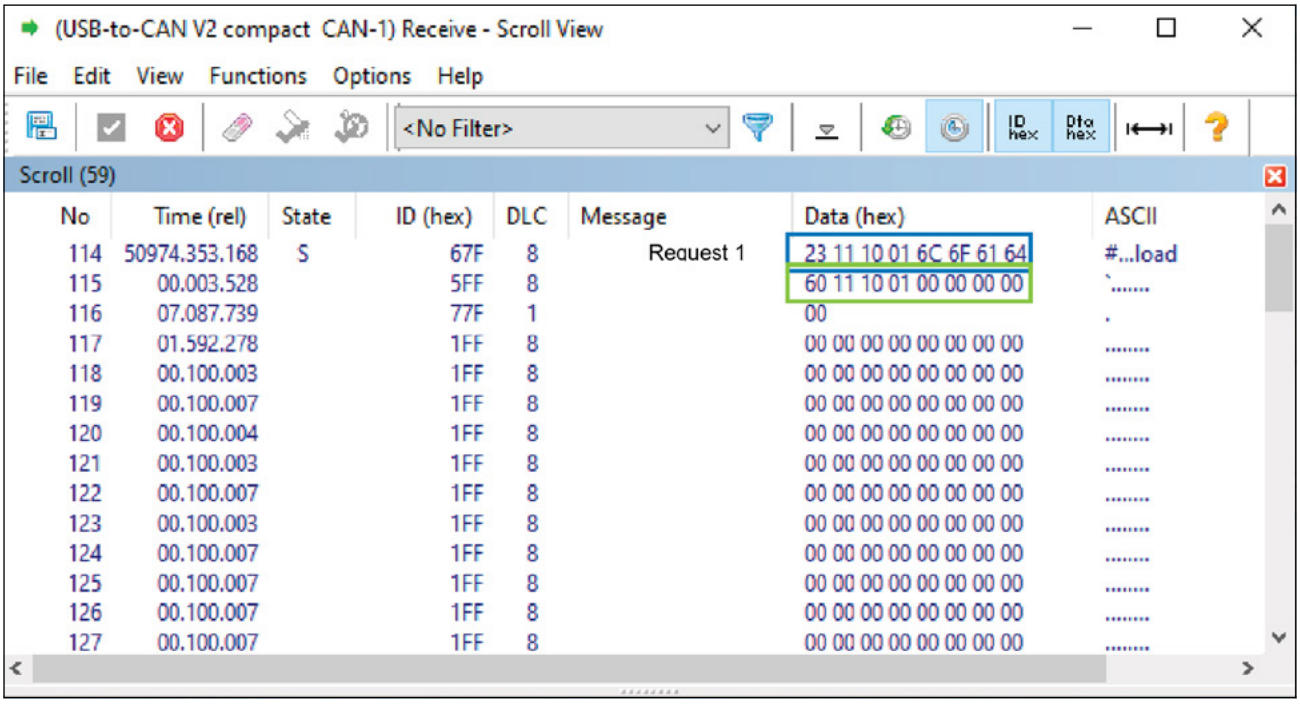


Figure 13. Example 7 How to send the command RESTORE

GEFRAN

GEFRAN spa
via Sebina, 74 - 25050 PROVAGLIO D'ISEO (BS) - ITALIA
tel. 0309888.1 - fax. 0309839063 Internet: http://www.gefran.com