ENGINEERING
TOMORROW

Operating guide

Wire position sensor
CANopen output
DST X800

ia.danfoss.com.

Table of Contents

1. General Information

1.1 Contact

Danfoss A/S
Industrial Automation
DK-6430 Nordborg
Denmark
www.ia.danfoss.com
E-mail: IA-Sensorglobaltechnicalsupport@danfoss.com

1.2 General

The document describes the standard CANopen implementations created. It is addressed to CANopen system integrators and to CANopen device designers who already know the content of standards designed by C.i.A. (CAN in Automation).

1.3 Abbreviations and terms

Abbreviation/term	Definition
CAN	Controller Area Network Describes a serial communication bys 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 level 7 "application" of the ISO/OSI reference model form which CANopen derives.
CMS	CAN Message Specification CAL service element. Defines the CAN Apllication Layer for the various industrial applications.
COB	Communication Object Unit of transport of data in a CAN network (aCAN message). A maximum of 2,048 COBs may be present i 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 data 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.

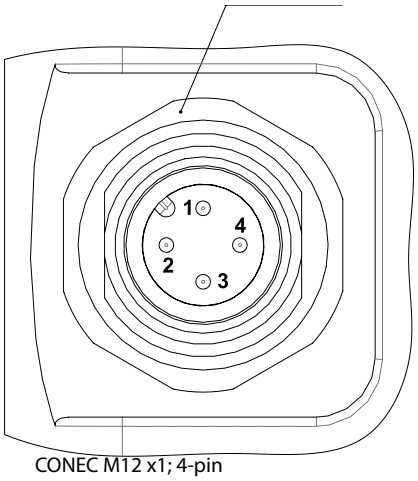
NOTE:
The numbers followed by the suffix "h" represent a hexadecimal value, with suffix "b" a binary value, and with suffix "d" a decimal value. The value is decimal unless specified otherwise.

2. Electrical connections

2.1 M12 x 1

Single version

CONNECTIONS
1: + SUPPLY
2: GROUND
3: OUPUT
4: n.c.:

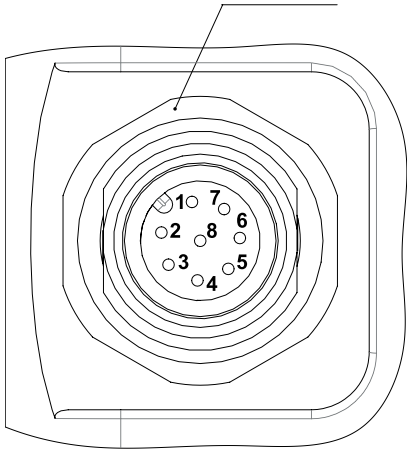


CONEC M12 x1; 4-pin
43-01088 connector

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

Redundant version

CONNECTIONS
1: + SUPPLY
2: GROUND
3: OUPUT1
4: n.c.:
5: + SUPPLY
6: GROUND
7: OUTPUT 2
8: n.c.



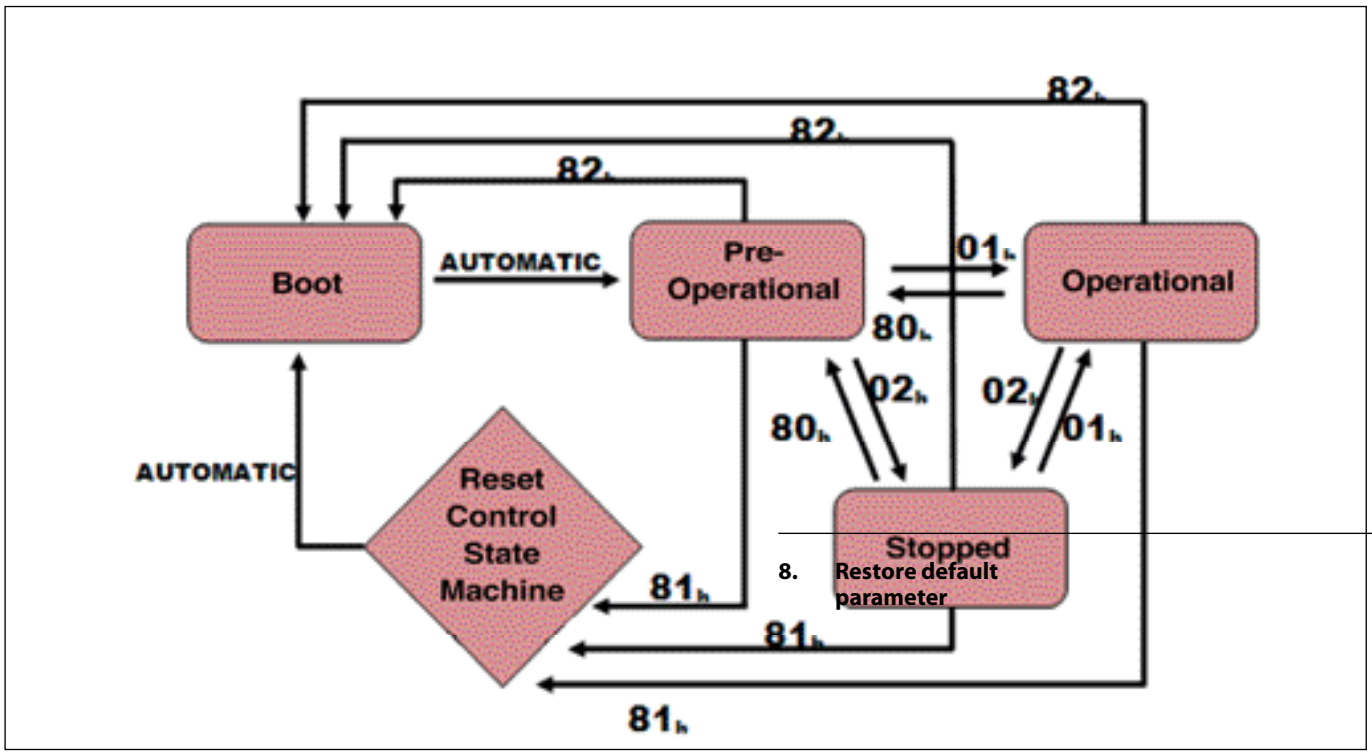
CONEC M12 x1; 8-pin
43-01100 connector

CONNECTIONS
1: + SUPPLY
2: GROUND
3: CAN-H 1
4: CAN-L 1
5: + SUPPLY
6: GROUND
7: CAN-H 2
8: CAN-L 2

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

3. Network Management (NMT)

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



Every CANopen device contains an international 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 dependant and described in **Control State Machine.**

It is important to distinguish a CANopen device's operational state machine from its Communication State Machine.
CANopen sensors and I/O modules, for example, have completely different operational state machines than servo drives. 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 date bytes that identify the node number and a command to that node's state machine.
Table 1 shows the 5 NMT messages surpported, and Table 2 shows the correct message for sending these messages.

NMT Message	COB-ID	Data Byte 1	Data Byte 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'
* Node-ID = Drive address (from 1 to 7Fh)			

Table 1

8. Restore default parameter

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 table 2	These bytes are not sent					

Table 2

4. Baud rate

Node-ID can be configurable via SDO communication object =x20F2 and 020F3 (see communication examples at the end of this coument).
The default Baud rate is 250kbit/s.

Important Note:
Changing this parameter can disturb the network! Use the service only if one device is connected to the network!

5. Node-ID and resolution

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

The default Node-ID is 7F.

Important note:
Changing this parameter can disturb the network! Use the service only if one device is connected to the network!

6. Parameter settings

All object dictionary parameters (object 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 LL-PARA), also part of the objec 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.

7. Restore default parameters

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

8. Heartbeat

The heartbeat mechanism for this device isestablished 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 herartbeat 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

9 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: Mo 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. The device uses only the 1st byte as manufacturer specific error code.

Byte	Byte 1 Byte 2	Byte 3	Byte 4	Byte 5	Byte 6 Byte 7 Byte 8
Description	Emergency Error code [1]	Error Register (object 0x1001 [2])	Manufacturer specific error code (always 0x00)	Manufacturer specific error code (object 0x4001)	Manufacturer specific error code NOT IMPLEMENTED (always 0x00)
[1] Error code		0x0000 Error Reset on no ERrror (Error Register = 0) 0x1000 Generic error			
[2] Always 0					

Supported Manufacturer Specific Error Codes (object 0x4001)

Manufacturer Specific Error Code (bit field)	Description
0bxxxxxxx1	(e.g. potentiometric signal under/above limits, temperature under/above limits)
0bxxx1xxxx	Program checksum error
0bxxx1xxxx	Program checksum error
0bxx1xxxxx	Flash limit reached - error
0bx1xxxxxx	LSS Parameter checksum error

10. 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 typ 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

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

Write Access, Data Transfer from Host to Slave

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

CMD determines the direction of data transfer and the size of the data object:
23 hex Sending of 4-byte data
(bytes 5 - 8 contian a 32 bit value)
2B hex Sending of 2-byte data
(bytes 5, 6 contain a 16-bit value
2F hex Sending of 1-byte data
(byte 5 contians an 8-bit value)

The Slave answers:

RES response of the slave:
60 hex Data sent successfully
80 hex Error

Read Access, Data Transfer form Slave to Host

Any read access to non-existing objects is answered with an error message.

CMD determines the direction of data transfer:
40 hex read access (in any case)

The Slave answers:

RES Response of the slave:
42 hex Bytes used by node when replying to read command with 4 or less data
43 hex Bytes 5 - 8 contain a 32-bit value
4B hex Bytes 5, 6 contain a 16-bit value
4F hex Byte 5 contains an 8-bit value
80 hex Error

11. PDO communication and Length calculation

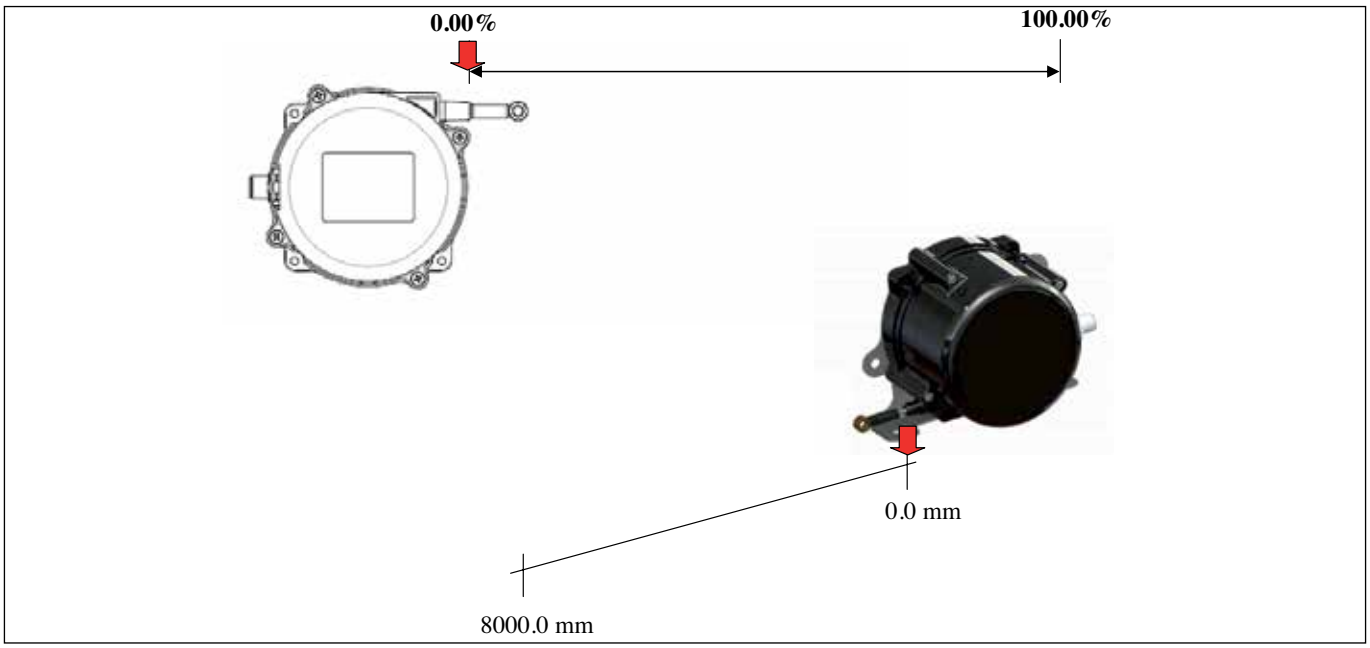
Transmit PDO #0
This PDO transmits length value of the position sensor.The Tx PDO#0 shall be transmitted cyclically, if the cyclic timer (object 0x1800.5) is programmed > 0. Values between 4 ms and 65535 ms shall be selectable by parameter settings. The Tx PDO#0 will be transmitted by entering the "Operational" state.

Byte	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5 Byte 6 Byte 7 Byte 8
Description	Position value object (0x6004) Low-Byte LSB	Posxition value object (0x6004)	Posxition value object (0x6004)	Position value object (0x6004) High-Byte MSB	(0x00)

11.1 Example 1 : TPDO #0 length 0.0 mm

Below an example of PDO mapping is reported in the case of:
- Node-ID = 7Fh
- Baud rate = 250 kBaud

Linear-encoder Cia406 setting as follow:
1. Total measuring range (object 0x6002.0) = 8000 mm (800 steps x 10 mm)
2. Preset value (object 0x6003.0) = 0 mm (0 steps x 10^3 nm)
3. Measuring step (object 0x6005.0) = 1 mm (500 steps x 10^3 nm)
4. Position value (object 0x6004.0)



ID	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
1FFh	00h	00h	00h	00h	00h	00h	00h	00h

Position value:
Byte 1 LSB (00h) = 00h
Byte 2 = 00h
Byte 3 = 00h
Byte 4 (MSB) = 00h

Position value = 00000000h to decimal 0d (resolution 1 mm) = 0 mm

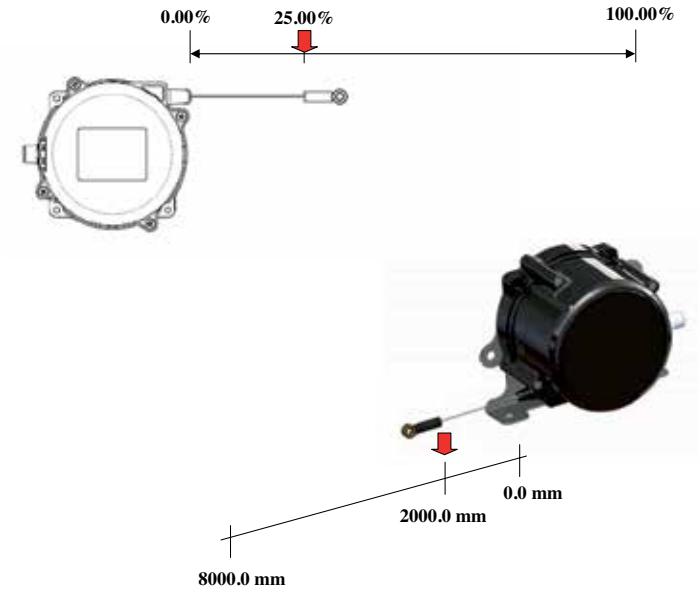
11.2 Example 2 : TPDO #0 length 2000.0 mm

Below an example of PDO mapping is reported in the case of:

- Node-ID = 7Fh
- Baud rate = 250 kBaud

Linear-encoder Cia406 setting as follow:

1. Total measuring range (object 0x6002.0) = 8000 mm (800 steps x 10 mm)
2. Preset value (object 0x6003.0) = 0 mm (0 steps x 10^3 nm)
3. Measuring step (object 0x6005.0) = 1 mm (500 steps x 10^3 nm)
4. Position value (object 0x6004.0)



ID	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
1FFh	A0h	0Fh	00h	00h	00h	00h	00h	00h

Position value:
Byte 1 LSB) = A0h
Byte 2 = 0Fh
Byte 3 = 00h
Byte 4 (MSB) = 00h

Position value = 00000FA0h to decimal 4000d (resolution 1 mm) = 2000.0

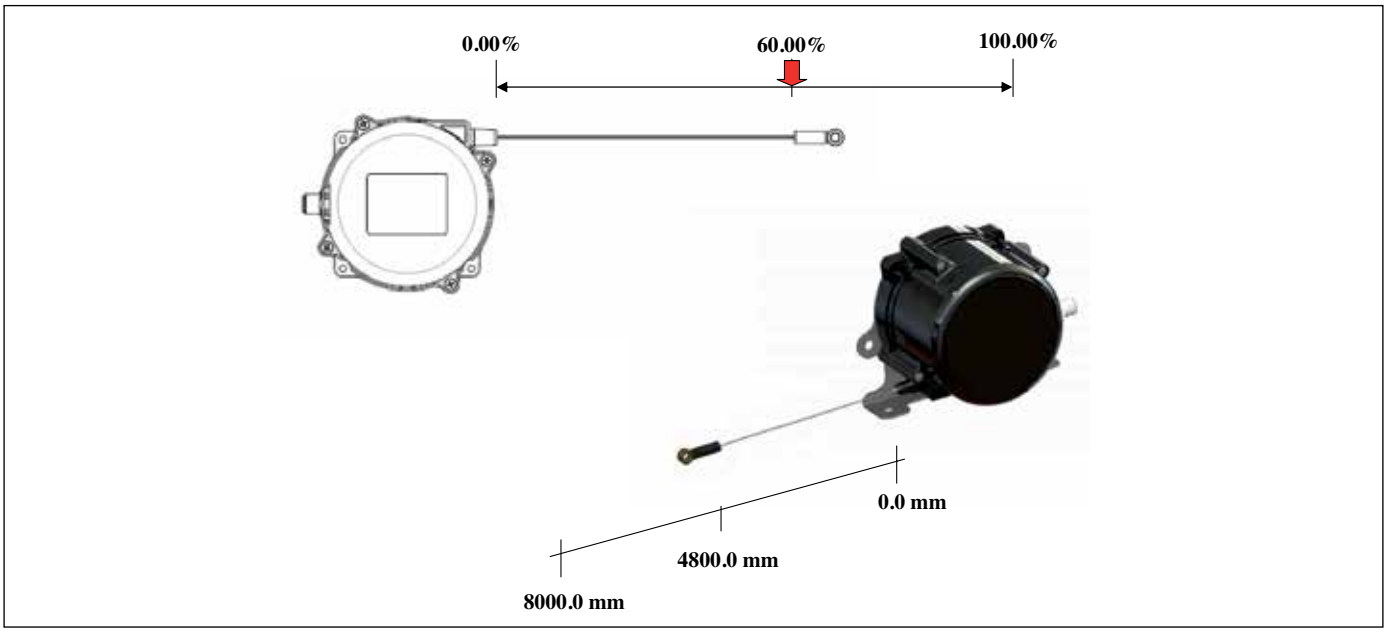
11.3 Example 1 : TPDO #0 length 4800.0 mm

Below an example of PDO mapping is reported in the case of:

- Node-ID = 7Fh
- Baud rate = 250 kBaud

Linear-encoder Cia406 setting as follow:

1. Total measuring range (object 0x6002.0) = 8000 mm (800 steps x 10 mm)
2. Preset value (object 0x6003.0) = 0 mm (0 steps x 10^3 nm)
3. Measuring step (object 0x6005.0) = 1 mm (500 steps x 10^3 nm)
4. Position value (object 0x6004.0)



ID	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
1FFh	80h	25h	00h	00h	00h	00h	00h	00h

Position value:
Byte 1 LSB) = 80h
Byte 2 = 25h
Byte 3 = 00h
Byte 4 (MSB) = 00h

Position value = 00002580h to decimal 9600d (resolution 1 mm) = 4800.0

12. 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 value	Comments
1000h		Device Profile	Unsigned 32	Ro	0x0008019A	Profile 410: Device profile for inclinometer (not fully implemented)
1001h		Error Register	Unsigned 8	Ro	0x00	Always ZERO
1005h		COB-ID SYNC	Unsigned 32	Rw	0x00000080	Always ZERO
1008h		Manufacturer Device Name	String	Const	"GIB"	Refer to Danfoss data sheet: DST X800 Wire position sensor
1009h		Manufacturer Hardware version	String	Const	"1.00"	
100Ah		Manufacturer Software version	String	Const	"1.10	
1010h	0	Number of entries	Unsigned 8	Ro	1	"save" (0x65766173) to store all parameters (objects with marking PARA)
	1	Save all parameters	Unsigned 32	Wo		
1011h	0	Restore default parameters	Unsigned 8	Ro	"1"	"load" (0x64616F6C) to restore all parameters (objects with marking PARA and LSS-PARA)
	1	Restore all parameters	Unsigned 32	Rw		
1014h	0	Emergency ID	Unsigned 32	Rw	0x80 + Node-ID	
1017h	0	Producer time/ Heart beat	Unsigned 16	Rw	0	Min. = 0 & Max. = 65536 with unit = 1 ms If 0: NOT USED From 1 - 19 NOT ACCEPTED From 20 to 65535 ACCEPTED
1018h	0	Identity object	Unsigned 8	Ro	4	Refer to Vendor ID:0x0000093
	1	Vendor ID	Unsigned 32	Ro	0x0000093	
	2	Product code	Unsigned 32	Ro	0x0000064	
	3	Revision number	Unsigned 32	Ro	0x0000001	
	4	Serial number	Unsigned 32	Ro	0x0000000	
SDO Server Parameter						
1200h	0	Number of entries	Unsigned 8	Ro	2	
	1	COB-ID Client to Server (Rx)	Unsigned 32	Ro	0x600 + Node-ID	
	2	COB-ID Server to to Server (Tx)	Unsigned 32	Ro	0x580 + Node-ID	
1800h	0	1st Transmit PDO Parameter	Unsigned 8	Ro		
	1	COB-ID	Unsigned 32	Ro	180h + Node-ID	
	2	Transmission Type Trans PDO-PARA	Unsigned 8	Rw	254 (0xFE)	0x01 - 0xf0 = synch cyclic Outputs are only updated after "n" synch objects. n = 0c01 (1) - 0xF0 (240 0xFC not implemented 0xFS not implemented 0xFE = asynchronous 0xFF = not implemented
	5	Event Timer Trans PDO-PARA	Insigned 16	Rw	100 (0x65)	0 = Inactive Min. = 4 & Max. = 65535 with unit = 1 ms
1802h	0	Tx PD0 #2 Parameter	Unsigned 8	RO	5	
	1	COB-ID Trans PDO	Unsigned 32	Ro	3280h + Node-ID	0x01 - 0xf0 = synch cyclic Outputs are only updated after "n" synch objects. n = 0c01 (1) - 0xF0 (240 0xFC not implemented 0xFS not implemented 0xFE = asynchronous 0xFF = not implemented 0 = Inactive Min = 4; Max = 65535 with unit = 1ms
	2	Trans type Trans PDO	Unsigned 8	Rw	254 (0xFE)	
	5	Event Timer Trans PDO-PARA	Unsigned 16	Rw	100 (0x64)	

Tx PDO [X] 0 Mapping Parameter						
1A00h	0	Number of entries	Unsigned 8	Ro	1	Wire length is indicated in Idx 6004
	1	1st Mapped Object	Unsigned 32	Ro	0x60040020	
Tx PDO [X] Mapping parameter						
1A02h	0	Number of entries	Unsigned 8	Ro	1	Wire length is indicated in Idx 6300
	1	1st Mapped Object	Unsigned 32	Ro	0x63000108	

Manufacturer Specific Profile Objects
In this section you will find the manufacturer specific profile indices for transducer.

"Setting the Node-ID"

Index	Sub Index	Name	Type	Access	Default value	Comments
20F0h	0	Setting of the Node-ID	Unsigned 8	Rw	0x7F (=127d)	The node ID used to access the sensor in the CANopen network
20F1h	0	Setting of the Node-ID	Unsigned 8	Rw	0x7F (=127d)	

A change of the Node ID is only accepted if the entries 20F0 and 20F1 contain the same changed value. Values below 1/above 127 are not accepted; the existing setting remains valid. After setting the new entries a reset must be made so that the new entries become vlaid (switch off the module for a short time).

"Setting the Baud Rate"

Index	Sub Index	Name	Type	Access	Default value	Comments
20F2h	0	Setting the Baud 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 the Baud 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

A change of the Baud rate is only accepted if the entries 20F2 and 20F3 contain the same changed value. Values above 7 are not accepted; the existing setting remains valid. After setting new entries a reset must be made so that the new entries become valid (switch off the module for a short time).

Index	Sub Index	Name	Type	Access	Default value	Comments
4000h		Error Behavior - PARA	Unsigned 8	Rw	1	0: Pre-operational 1: no state change 2: stopped Min = 0 & Max = 255
4001h		Error code	Unsigned 8	Ro	0	0: no error Min = 0 & Max = 255
5000h		Automatic NMT Start after Power-On-PARA	Unsigned 8	Rw	0	0 = not activated 1: activated Min = 0 & Max = 1
5001h		PDO coding Used - PARA	Unsigned 8	RW	1	0: Big Endian 1: Little Endian

Manufacturer Specific Profile Objects (according to CIA DS-410)

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

Index	Sub Index	Name	Type	Access	Default value	Comments
6000h	0	Operating Parameters - PARA	Unsigned 16	Ro	0x00(0d)	Contain the functions for code sequence commissioning diagnostic control and scaling function contorl (*). Operational functions NOT activated in standard version (always 0x00)
6002h	0	Total measuring range	Unsigned 32	Ro		Measuring range in 10 mm steps. Example: Measuring rang 8000 mm Total measuring range = 0x0320 (00d)
6003h	0	Preset value	Unsigned 32	Rw	0x00 (0d)	The preset function supports adaption of the GSF zero point to the mechanical zero point of the system. The output position value is set to the parameter “Preset value” and the offset from the position value is calculated and stored in the GSF.
6004h	0	Position value	Unsigned 32	Ro		The object 6004h “Position value” defines the output position value for the communication objects 1800 h.
6005h	0	Measuring steps	Unsigned 32	Rw	0x000003E8 (1000d) 10^3 mm	The parameter “Linear encoder measuring step settings” defines the measuring step settings for position value. 10^3 - 10^6 mm

In this section you will find the manufacturer specific profile indices for transducer. as CAM (optional functions NOT activated in standard version)

Index	Sub Index	Name	Type	Access	Default value	Comments
6300h	0	CAM State register	Unsigned 8	Ro	0x00(0d)	The parameter "CAM state register" defines the status bit of the cam in a cam channel. The status bit set to 1 defines "cam active". The status bit set to 0 defines "cam inactive". If the polarity bit of a cam is set (refer to index 302h) the actual cam state will be inverted.
6301h	0	CAM enable register	Unsigned 8	Rw	0x00 (0d)	Each Cam_polarity_channel contains the actual settings for a maximum of 8 cam's for one position channel. If the enabled bit is set to 1, the cam state will be calculated by the device. In the other case the cam state of the related cam will be set permanently to 0.
6302h	0	CAM Polarity Register	Unsigned 8	Rw	0x00(0d)	Each Cam_enable_channel contains the calculation state for maximum of 8 cam's for one position channel. If the polarity bit is set to 1, the cam state of an active CAM will signal by setting the related cam state bit to zero. In the other case the cam state of the related cam will not be inverted.
6310h	0	CAM 1 _ LOW LIMIT	Unsigned 32	Rw	0x00(0d)	Each CAM_low_limit channel contians the switch point for the lower limit setting for a max. of 8 cam's for one position channel.
6311h	0	CAM 2 _ LOW LIMIT	Unsigned 32	Rw	0x00(0d)	Each CAM_low_limit channel contians the switch point for the lower limit setting for a max. of 8 cam's for one position channel.
6312h	0	CAM 3 _ LOW LIMIT	Unsigned 32	Rw	0x00(0d)	Each CAM_low_limit channel contians the switch point for the lower limit setting for a max. of 8 cam's for one position channel.
6313h	0	CAM 4 _ LOW LIMIT	Unsigned 32	Rw	0x00(0d)	Each CAM_low_limit channel contians the switch point for the lower limit setting for a max. of 8 cam's for one position channel.
6314h	0	CAM 5 _ LOW LIMIT	Unsigned 32	Rw	0x00(0d)	Each CAM_low_limit channel contians the switch point for the lower limit setting for a max. of 8 cam's for one position channel.
6315h	0	CAM 6 _ LOW LIMIT	Unsigned 32	Rw	0x00(0d)	Each CAM_low_limit channel contians the switch point for the lower limit setting for a max. of 8 cam's for one position channel.
6316h	0	CAM 7 _ LOW LIMIT	Unsigned 32	Rw	0x00(0d)	Each CAM_low_limit channel contians the switch point for the lower limit setting for a max. of 8 cam's for one position channel.
6317h	0	CAM 8 _ LOW LIMIT	Unsigned 32	Rw	0x00(0d)	Each CAM_low_limit channel contians the switch point for the lower limit setting for a max. of 8 cam's for one position channel.
6320h	0	CAM 1 _ HIGH LIMIT	Unsigned 32	Rw	0x00(0d)	Each CAM_high_limit channel contians the switch point for the higher limit setting for a max. of 8 cam's for one position channel.
6321h	0	CAM 2 _ HIGH LIMIT	Unsigned 32	w	0x00(0d)	Each CAM_high_limit channel contians the switch point for the higher limit setting for a max. of 8 cam's for one position channel.

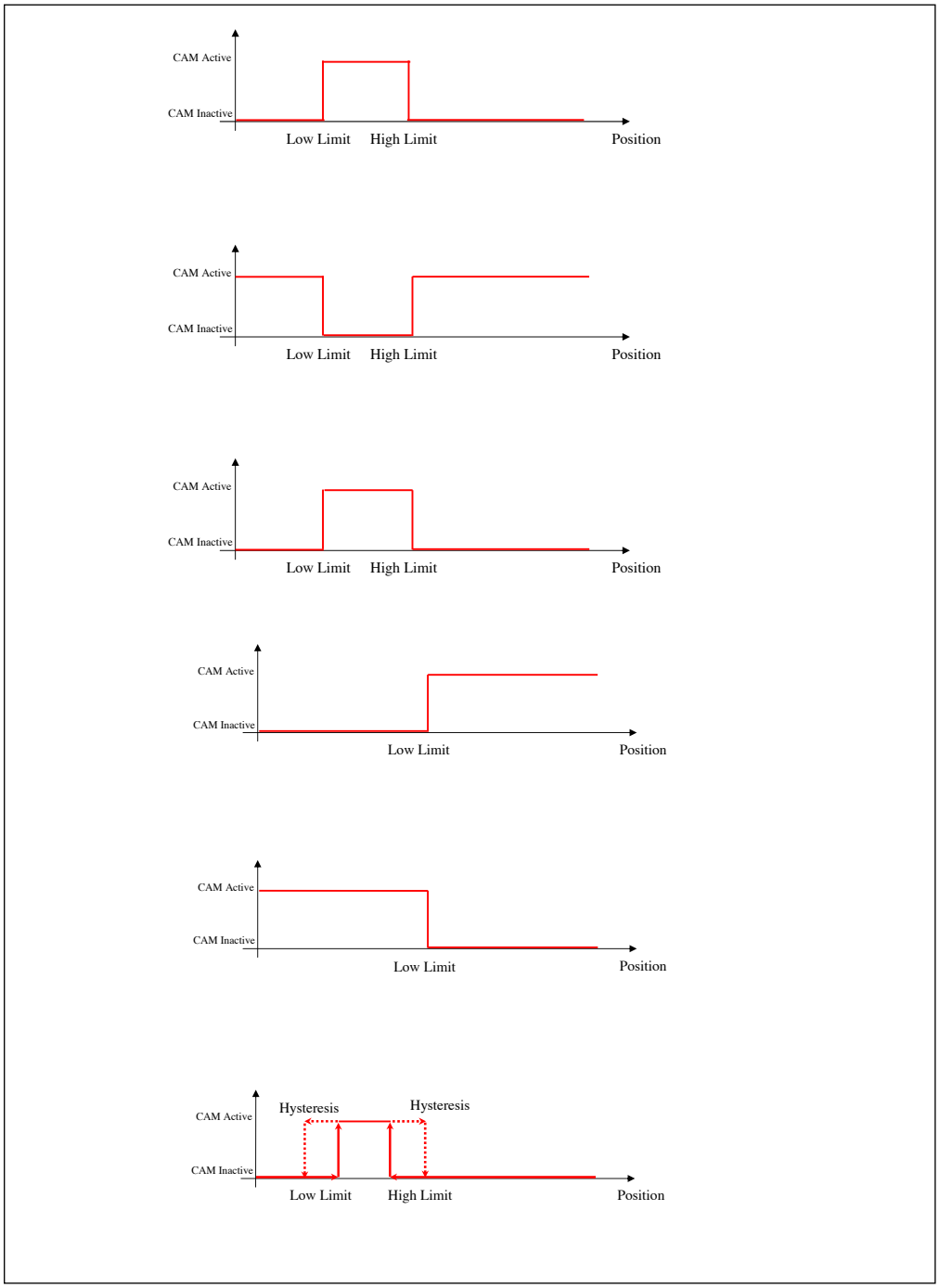
Index	Sub Index	Name	Type	Access	Default value	Comments
6322h	0	CAM 3_ HIGH LIMIT	Unsigned 8	Ro	0x00(0d)	Each CAM_high_limit channel contians the switch point for the higher limit setting for a max. of 8 cam's for one position channel.
6323h	0	CAM 4_ HIGH LIMIT	Unsigned 8	Rw	0x00 (0d)	Each CAM_high_limit channel contians the switch point for the higher limit setting for a max. of 8 cam's for one position channel.
6324h	0	CAM 5_ HIGH LIMIT	Unsigned 8	Rw	0x00(0d)	Each CAM_high_limit channel contians the switch point for the higher limit setting for a max. of 8 cam's for one position channel.
6325h	0	CAM 6 _ HIGH LIMIT	Unsigned 32	Rw	0x00(0d)	Each CAM_high_limit channel contians the switch point for the higher limit setting for a max. of 8 cam's for one position channel.
6326h	0	CAM 7 _ HIGH LIMIT	Unsigned 32	Rw	0x00(0d)	Each CAM_high_limit channel contians the switch point for the higher limit setting for a max. of 8 cam's for one position channel.
6327h	0	CAM 8 _ HIGH LIMIT	Unsigned 32	Rw	0x00(0d)	Each CAM_high_limit channel contians the switch point for the higher limit setting for a max. of 8 cam's for one position channel.
6330h	0	CAM 1 HYSTERESIS	Unsigned 32	Rw	0x00(0d)	Each CAM_hystertesis channel contians the delay setting of switch points for a max. of 8 cam's for one position channel. For illustration of the hysteresis functionality refer below.
6331h	0	CAM 2 HYSTERESIS	Unsigned 32	Rw	0x00(0d)	Each CAM_hystertesis channel contians the delay setting of switch points for a max. of 8 cam's for one position channel. For illustration of the hysteresis functionality refer below.
6332hh	0	CAM 3 HYSTERESIS	Unsigned 32	Rw	0x00(0d)	Each CAM_hystertesis channel contians the delay setting of switch points for a max. of 8 cam's for one position channel. For illustration of the hysteresis functionality refer below.
6333h	0	CAM 4 HYSTERESIS	Unsigned 32	Rw	0x00(0d)	Each CAM_hystertesis channel contians the delay setting of switch points for a max. of 8 cam's for one position channel. For illustration of the hysteresis functionality refer below.
6334h	0	CAM 5 HYSTERESISIT	Unsigned 32	Rw	0x00(0d)	Each CAM_hystertesis channel contians the delay setting of switch points for a max. of 8 cam's for one position channel. For illustration of the hysteresis functionality refer below.
6335h	0	CAM 6 HYSTERESIS	Unsigned 32	Rw	0x00(0d)	Each CAM_hystertesis channel contians the delay setting of switch points for a max. of 8 cam's for one position channel. For illustration of the hysteresis functionality refer below.
6336h	0	CAM 7 HYSTERESIS	Unsigned 32	Rw	0x00(0d)	Each CAM_hystertesis channel contians the delay setting of switch points for a max. of 8 cam's for one position channel. For illustration of the hysteresis functionality refer below.
6337h	0	CAM 8 HYSTERESIS	Undesigned 32	Rw	0x00(0d	Each CAM_hystertesis channel contians the delay setting of switch points for a max. of 8 cam's for one position channel. For illustration of the hysteresis functionality refer below.

(*) Operating parameters (Object 0x6000)

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
-	-	-	-	-	-	-	-	-	-	-	-	MD	SF	-	-
MSB	***	***	***	***	***	***	***	***	***	***	***	***	***	***	LSB

MD = 0/1 Measuring direction UP/DOWN
SF = 0/1 Scaling function DISABLE/ENABLE

GSF Cams functionality (optional functions NOT activated in standard version)
Each Cam has parmeters for the minimum switch point, the maximum switch point and setting a hesteresis to the switch points.
Possible usage of cam's and switch points.



13. Status LED

The integrated two color status LED signals the recent device state (Run LED, green) as well as CAN communication errors that moight have occured (Error LED), red). The color and the flashing frequency of the LED distinguish the different device states as shown below.

Status LED		
Run LED	LED State	Description
	Off	No power supply is connected
	Blinking	The device is in state Pre-Operational
	Single Flash	The device is in state Stopped
	ON	The device is in state Operational

Error LED		
Run LED	LED State	Description
	Off	The device is in working condition
	Single Flash	CAN Warning Limit reached
	On	The device is in state Bus-Off
	Red/Green On	Limit Angles reached (110% FS or ±87°)

Legend

	LED green OFF
	LED green ON
	LED red OFF
	LED red ON
	LEDs red & green ON together
	LED green blinking (200 ms ON/OFF)
	LEDs green single flash (500 ms ON/OFF)

14. Communication examples

Example 1: How to change the Baud Rate Setting from 250 kbaud 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			

***CMD** determines the direction of data transfer and the size of the data object:*
23 hex Sending of 4-byte data (bytes 5 - 5 contain a 32 bith value)
2B hex Sending of 2-byte data (bytes 5, 6 contain a 16-bit value)
2F hex 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				

RES Response of Slave:
60 hex Data sent successfully
80 hex Error

A change of the Baud rate is only accepted if the entries 0x20F2 and 0x20F3 contain the same changed value. With the aim to change the baud rate from 250 kBaud (0x03) to 500 kBaud (0x02) write a second SDO (in the example the Node-ID = 0x7F9

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

A change of the Baud rate is only accepted if the entries 0x20F2 and 0x20F3 contain the same changed value. With the aim to change the baud rate from 250 kBaud (0x03) to 500 kBaud (0x02) write a second SDO (in the example the Node-ID = 0x7F9

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

Object:

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

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 kBbaud
02h	500 kBbaud
01h	800 kBbaud
00h	1000 kBbaud

The answer after successful storing you will receive is:

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

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

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. Afer setting the new entries a reset must be made so that the new entries becom valid (switch off the module for a short time).

Example 2: How to change the ID-Node from 0x7Fh (127d) (Current setting) to 0x06h (6d)

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				

***CMD** determines the direction of data transfer and the size of the data object:*
23 hex Sending of 4-byte data (bytes 5 - 5 contain a 32 bith value)
2B hex Sending of 2-byte data (bytes 5, 6 contain a 16-bit value)
2F hex 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				

RES Response of Slave:
60 hex Data sent successfully
80 hex Error

A change of the Node-ID is only accepted if the entries 0x20F0 and 0x20F1 contain the same changed value. With the aim to change the Node-ID from 127 (0x7F) to 6 (0x06) write a firat SDO (in the example the Node-ID = 0x7F)

ID	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
67Fh	2Fh	F0h	20h	00h	06h	00h	00h	00h

A change of the Node-ID is only accepted if the entries 0x20F0 and 0x20F1 contain the same changed value. With the aim to change the Node-ID from 127 (0x7F) to 6 (0x06) write a second SDO (in the example the Node-ID = 0x7F)

ID	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
67Fh	2Fh	F1h	20h	00h	06h	00h	00h	00h

Object:

20F0h	0	Setting of the Node-ID	Unsigned 8	Rw	0x7F (0127d)	The Node-ID used to access the sensor in the CANopen
20F1h	0	Setting of the Node-ID	Unsigned 8	Rw	0x7F (0127d)	The Node-ID used to access the sensor in the CANopen

The supported Node_ID are 0x01 to 0x7F. The answer after the succesful storing is:

ID	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
5FFh	60h	F0h	20h	00h	00h	00h	00h	00h

ID	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
5FFh	60h	F1h	20h	00h	00h	00h	00h	00h

IMPORTANT NOTE:
A change of the Node_ID is only accepted if the entries 0x20F0 and 0x20F1 contain the same changed value. Values below 1 / above 127 are not accepted; the existing setting remains valid. Afer setting the new entries a reset must be made so that the new entries become valid (switch off the module for a short time).

Example 3: How to change the PDO rate (time interval) from 100 ms (current setting) to 20 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			

***CMD** determines the direction of data transfer and the size of the data object:*
23 hex Sending of 4-byte data (bytes 5 - 5 contain a 32 bith value)
2B hex Sending of 2-byte data (bytes 5, 6 contain a 16-bit value)
2F hex 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				

RES Response of Slave:
60 hex Data sent successfully
80 hex Error

With the aim to change the PDO rate from 100 ms (0x64) to 20 ms (0x14)
Write (in the example the Node-ID = 0x7F)

ID	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
67Fh	2Bh	00h	18h	05h	14h	00h	00h	00h

Object:

1800h	0	1st Transmit PDO Parameter	Unsigned 8	Ro		
	1	COB-ID	Unsigned 32	Ro	180h + Node-ID	
	2	Transmission Type	Unsigned 8	Rw	254	Asynchronous transmission
	3	Inhibit Time	Unsigned 16	Ro	0	Min. = 0 & Max. = 65535 with unit = 1 ms
	4	Reserved	//	//		
	5	Timer	Unsigned 16	Rw	100 (64)	Min. = 0 & Max. = 65535 with unit = 1 ms

The answer after successful storing you will receive is:

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

With the aim to save functionality write the "save" command as below:
Write (in the example the Node-ID = 0x7F)

ID	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
67Fh	23h	10h	10h	01h	73h	61h	76h	65h

Note: save command is given by sending the code:

73h	61h	76h	65h

Where:
73h = ASCII code "s"
61h = ASCII code "a"
76h = ASCII code "v"
65h = ASCII code "e"

The answer after successful storing you will receive is:

The answer after successful storing you will receive is:

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

IMPORTANT NOTE:
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).

Example 4: How to activate an automatic NMT Start after Power ON (the PDO will be send 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			

***CMD** determines the direction of data transfer and the size of the data object:*
23 hex Sending of 4-byte data (bytes 5 - 5 contain a 32 bith value)
2B hex Sending of 2-byte data (bytes 5, 6 contain a 16-bit value)
2F hex 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	CMD	Index		Sub-Index	Data			

RES Response of Slave:
60 hex Data sent successfully
80 hex Error

With the aim to activate an automatic NMT Start after power ON write (in the example the Node-ID = 0x7F

ID	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
67Fh	2Fh	00h	50h	00h	01h	00h	00h	00h

Object:

5000h	0	Automatic NMT Start after Power ON - PARA	Unsigned 8	Rw	1	0 = not activated 1= activated Min. = 0 & Max. = 1

The answer after successful storing you will receive is.

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

With the aim to save functionality write the "save" command as below:
Write (in the example the Node-ID = 0x7F)

ID	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
67Fh	23h	10h	10h	01h	73h	61h	76h	65h

Note: save command is given by sending the code:

73h	61h	76h	65h

Where:
73h = ASCII code "s"
61h = ASCII code "a"
76h = ASCII code "v"
65h = ASCII code "e"

The answer after successful storing you will receive is:

The answer after successful storing you will receive is:

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

IMPORTANT NOTE:
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).

Example 5: How to Preset the Position value (via object 0x6003.0) to 0 mm
The value "Preset Value" (Idx 6003.0) affects the display of the Position value. The value entered in "Preset value" immediately correct the measured value of the sensor cell at the instan tacc. A typical application is the compensation of display errors du to mountine (e.g. sensor zeroing).
The senosor must first to brought to a defined position.

With Service data object (S.D.O) the access to entries of a device Object Dictionary is provided. As these entries may contain data or arbitray 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			

***CMD** determines the direction of data transfer and the size of the data object:*
23 hex Sending of 4-byte data (bytes 5 - 5 contain a 32 bith value)
2B hex Sending of 2-byte data (bytes 5, 6 contain a 16-bit value)
2F hex 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	CMD	Index		Sub-Index	Data			

RES Response of Slave:
60 hex Data sent successfully
80 hex Error

Consider the actual reading value **(Node-ID = 0x7F)** is:

ID	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
1FFh	05h	01h	00h	00h	00h	00h	00h	00h

Position value:
Byte 1 (LSB) = 05h
Byte 2 = 01h
Byte 3 = 00h
Byte 4 (MSB) = 00h

Position value = 00000105h to decimal 261d
(measuring steps 1 mm) = 261 mm

With the aim to PRESET the reading value to 0 mm write (in the example the Node_ID = 0x7F)

ID	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
67Fh	23h	03h	60h	00h	05h	01h	00h	00h

Object:

6003h	0	Preset Value	Signed 32	Rw	0x00 (0d)	The preset value function supports adaption of the GSF zero point to mechanical zero point of the system (user offset).

The answer after successful storing you will receive is.

ID	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
5FFh	60h	03h	60h	00h	00h	00h	00h	00h

With the aim to save functionality write the "save" command as below:
Write (in the example the Node-ID = 0x7F)

ID	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
67Fh	23h	10h	10h	01h	73h	61h	76h	65h

Note: save command is given by sending the code:

73h	61h	76h	65h

Where:
73h = ASCII code "s"
61h = ASCII code "a"
76h = ASCII code "v"
65h = ASCII code "e"

The answer after successful storing you will receive is.

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

IMPORTANT NOTE:
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).

Example 6: 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			

***CMD** determines the direction of data transfer and the size of the data object:*
23 hex Sending of 4-byte data (bytes 5 - 5 ontain a 32 bith value)
2B hex Sending of 2-byte data (bytes 5, 6 contain a 16-bit value)
2F hex 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	Date			

RES Response of Slave:
60 hex Data sent successfully
80 hex Error

With the aim to restore all parameters to default write
(in exam. the Node-ID = 0x7F)

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

Object:

1011h	1	Load all parameters	Unsigned 8	Wo		"load" (0x64616F6C) to restore all parameters (objects with marking PARA and LSSPARA)

The answer after successful storing you will receive is.

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

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).

Example 7: How to disable the Asynchronous Transmission (Asynchronous TPDO inactive)

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		

***CMD** determines the direction of data transfer and the size of the data object:*
23 hex Sending of 4-byte data (bytes 5 - 5 ontain a 32 bith value)
2B hex Sending of 2-byte data (bytes 5, 6 contain a 16-bit value)
2F hex 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				

RES Response of Slave:
60 hex Data sent successfully
80 hex Error

With the aim to disable the asynchronous transmission write the SDO **(in exam. the Node-ID = 0x7F)**

ID	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
67Fh	2Bh	00h	18h	05h	00h	00h	00h	00h

Object:

1800h	0	1st Transmit PDO Parameter	Unsigned 8	Ro		
	1	COB-ID Trans PDO	Unsigned 32	Ro	180+ Node-ID	
	2	Transmission Type Trans PDO - PARA	Unsigned 8	Rw	254 (0xFE)	0x01 - 0xF0 = synch cyclic Outputs are only updated after "n" synch objects n = 0x01 (1) - 0xF0 (240) 0xFC not impelemented 0xFD not implemented 0xFE = asynchronous 0xFF = not implemented
	5	Event Timer PDO - PARA	Unsigned 16	Rw	100 (0x64)	0 = inactive Min. = 4 & Max. = 65535 with unit = 1ms

The answer after successful storing you will receive is.

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

With the aim to save functionality write the "save" command as below:
Write (in the example the Node-ID = 0x7F)

ID	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
67Fh	23h	10h	10h	01h	73h	61h	76h	65h

Note: save command is given by sending the code:

73h	61h	76h	65h

Where:
73h = ASCII code "s"
61h = ASCII code "a"
76h = ASCII code "v"
65h = ASCII code "e"

The answer after successful storing you will receive is.

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

IMPORTANT NOTE:
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).

Example 8: How to enable the Synchronous Transmission (Synchronous TPDO active after 1st sync message)
)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			

***CMD** determines the direction of data transfer and the size of the data object:*
23 hex Sending of 4-byte data (bytes 5 - 5 ontain a 32 bith value)
2B hex Sending of 2-byte data (bytes 5, 6 contain a 16-bit value)
2F hex 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				

RES Response of Slave:
60 hex Data sent successfully
80 hex Error

With the aim to disable the synchronous transmission with TPDO active after 1st sync message write the SDO **(in exam. the Node-ID = 0x7F)**

ID	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
67Fh	2Fh	00h	18h	02h	01h	00h	00h	00h

Object:

1800h	0	1st Transmit PDO Parameter	Unsigned 8	Ro		
	1	COB-ID Trans PDO	Unsigned 32	Ro	180+ Node-ID	
	2	Transmission Type Trans PDO - PARA	Unsigned 8	Rw	254 (0xFE)	0x01 - 0xF0 = synch cyclic Outputs are only updated after "n" synch objects n = 0x01 (1) - 0xF0 (240) 0xFC not impelemented 0xFD not implemented 0xFE = asynchronous 0xFF = not implemented
	5	Event Timer PDO - PARA	Unsigned 16	Rw	100 (0x64)	0 = inactive Min. = 4 & Max. = 65535 with unit = 1ms

The answer after successful storing you will receive is.

ID	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
5FFh	60h	00h	18h	02h	00h	00h	00h	00h

With the aim to save functionality write the "save" command as below:
Write (in the example the Node-ID = 0x7F)

ID	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
67Fh	23h	10h	10h	01h	73h	61h	76h	65h

Note: save command is given by sending the code:

73h	61h	76h	65h

Where:
73h = ASCII code "s"
61h = ASCII code "a"
76h = ASCII code "v"
65h = ASCII code "e"

The answer after successful storing you will receive is.

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

IMPORTANT NOTE:
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).

ENGINEERING
TOMORROW

Danfoss A/S
Industrial Automation
DK-6430 Nordborg
Denmark