Technical Manual

Absolute Singleturn/Multiturn
Encoders

Series F36X8 USF

optical scanning
electronic multiturn stage

CANopen®

8.F58X8.XX2X.212X
Firmware Release V2.6 USF

Table of Contents

1 General

CANopen Singleturn Encoder Series F36X8 F = Optical scanning principle with Multiturn stage

The CANopen encoders of Series F36X8 support the latest CANopen communication profile according to **DS 301 V4.2** In addition, device-specific profiles such as the encoder profile **DS 406 V3.2.16** are available.

The following operating modes can be selected: Polled Mode, Cyclic Mode, Sync Mode. Moreover, scale factors, preset values, limit switch values and many other additional parameters can be programmed via the CAN-Bus. At Power ON all parameters are loaded from a Flash Memory, which had previously been saved to protect them in case of power failure. The following output values may be freely combined as **PDO** (PDO Mapping): **position, speed**, as well as the status of the two limit switches.

A 2-colour LED on the backside indicates the operating and error status of the CAN bus, as well as the condition of internal diagnostics. CANopen encoders are available both in shaft and hollow shaft versions; their IP 65 protection level allows using them even in tough industrial environments.

CANopen Communication Profile DS 301 V4.2.0

CANopen represents a unified user interface and thus allows for a simplified system structure with a wide variety of devices. CANopen is optimized for the fast exchange of data in real-time systems and possesses a number of different device profiles that have been standardized. The CAN in Automation (CiA) manufacturers and users group is in charge of the creation and standardization of the relevant profiles.

CANopen offers

- user-friendly access to all device parameters
- auto-configuration of the network and of the devices
- device synchronization within the network
- cyclic and event-driven process data exchange
- simultaneous reading and writing of data

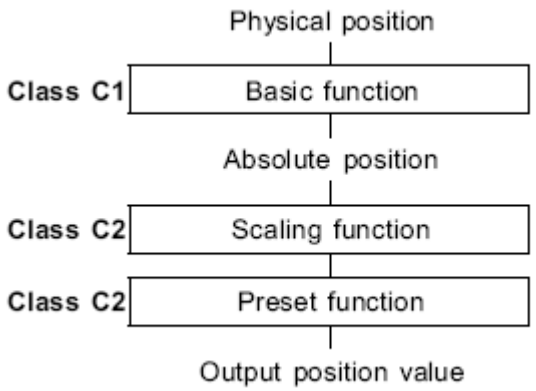
CANopen uses four communication objects (COB) with different properties

- Process Data Objects (PDO) for real-time data
- Service Data Objects (SDO) for transmitting parameters and programs
- Network Management (NMT, Life-Guarding, Heartbeat)
- Predefined Objects (for synchronisation, time-Stamp, emergency)

All device parameters are filed in an **Object dictionary**. This Object Dictionary contains the description, data type and structure of the parameters, as well as the address (Index).
The dictionary is divided into a communication profile section, a section covering the device profile as well as a section specific to the manufacturer.

Encoder Device Profile DS 406

This profile describes a **vendor-independent** mandatory definition of the interface with regard to encoders.
The profile determines which CANopen functions are to be used as well as how they are to be used.
This standard thus makes possible an open vendor-independent bus system.
The device profile is broken down into two Object classes:



• **Class C1** describes all the basic functions that the encoder must include

• **Class C2** contains numerous extended functions, which must either be supported by encoders of this class (Mandatory) or which are optional. Class 2 devices thus contain all C1 and C2 mandatory functions, as well as additional optional functions dependent on the manufacturer. An address range is also defined in the profile to which the manufacturer's own special functions can be assigned.

2 LSS services DS 305 V2.0

CiA DSP 305 CANopen Layer Setting Service and Protocol (LSS) were created to enable the following parameters to be read and changed through the network:

- Node address
- Baud rate
- LSS address

These features increase the "plug–and-play" compatibility of the device, while the configuration possibility has been significantly simplified. The LSS Master is responsible for configuring these parameters on one or more slaves on the network.

Data transmission

With CANopen data is transferred via two different communication types (COB=Communication Object) with different properties:

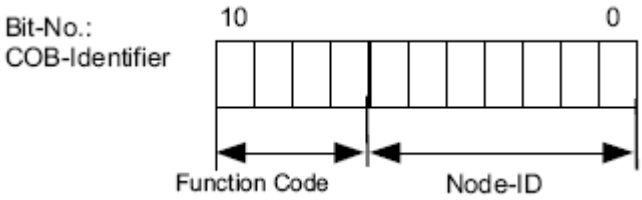
- **Process Data Objects (PDO – real-time capable)**
- **Service Data Objects (SDO)**

The Process Data Objects **(PDO)** provide highly dynamic exchange of real-time data (e.g. encoder position, speed, comparative position status) with a maximum length of 8 byte. This data is transmitted with a high priority (low COB-Identifier). PDOs are broadcast messages and provide their real-time data simultaneously to all desired receivers. PDOs can be mapped, i.e. 4 byte of position and 2 byte of speed can be combined in one 8 byte data word.

The Service Data Objects **(SDO)** form the communication channel for the transfer of device parameters (e.g. encoder resolution programming). As these parameters are transmitted acyclically (e.g. only once during boot-up of the network), the SDO objects have a low priority (high COB-Identifier).

Objects and Function Code in the Predefined Connection Set

For easier management of the Identifiers, CANopen uses the "Predefined Master/Slave Connection Set", where all identifiers are defined with standard values in the object dictionary. These identifiers can however be changed and customized via SDO access.



The 11-bit Identifier is made up of a **4-bit function code** and a 7-bit **node number**.

The higher the value of the COB-Identifier, the lower is its priority!

Broadcast (network-wide) Objects

object	function code (binary)	resulting COB-ID	Communication Parameters at Index
NMT	0000	0	-
SYNC	0001	128 (80h)	1005h, 1006h, 1007h
TIME STAMP	0010	256 (100h)	1012h, 1013h

Peer-To Peer (device-to-device) Objects

object	function code (binary)	Resulting COB-IDs	Communication Parameters at Index
EMERGENCY	0001	129 (81h) – 255 (FFh)	1014h, 1015h
PDO1 (tx)	0011	385 (181h) – 511 (1FFh)	1800h
PDO1 (rx)	0100	513 (201h) – 639 (27Fh)	1400h
PDO2 (tx)	0101	641 (281h) – 767 (2FFh)	1801h
PDO2 (rx)	0110	769 (301h) – 895 (37Fh)	1401h
PDO3 (tx)	0111	897 (381h) – 1023 (3FFh)	1802h
PDO3 (rx)	1000	1025 (401h) – 1151 (47Fh)	1402h
PDO4 (tx)	1001	1153 (481h) – 1279 (4FFh)	1803h
PDO4 (rx)	1010	1281 (501h) – 1407 (57Fh)	1403h
SDO (tx)	1011	1409 (581h) – 1535 (5FFh)	1200h
SDO (rx)	1100	1537 (601h) – 1663 (67Fh)	1200h
NMT Error Control	1110	1793 (701h) – 1919 (77Fh)	1016h, 1017h

Restricted, reserved Objects

COB-ID	used by object
0 (000h)	NMT
1 (001h)	reserved
257 (101h) – 384 (180h)	reserved
1409 (581h) – 1535 (5FFh)	default SDO (tx)
1537 (601h) – 1663 (67Fh)	default SDO (rx)
1760 (6E0h)	reserved
1793 (701h) – 1919 (77Fh)	NMT Error Control
2020 (780h) – 2047 (7FFh)	reserved

3 Process Data Transmission

The **4 PDO services** PDO1 (tx) ,PDO2 (tx) … PDO4 (tx) are available for process data transmission. A PDO transmission can be triggered by a variety of events
(see Object Dictionary Index 1800h):

- **asynchronously** (event-driven) by an internal cyclic event timer or by a change in the process value of the sensor data
- **synchronously** as a response to a SYNC telegram; (a SYNC command will cause all CANopen nodes to store their values synchronously, after which they are transferred in succession to the bus according to their set priority)
- **as a response to a** RTR **telegram** (per Remote Frame=recessive RTR-bit, exactly that message with the communicated identifier will be requested)

Standard setting for the **Mapping of the transmit PDOs1-4**:

Mapping	TPDO1 1800h	TPDO2 1801h	TPDO3 1802h	TPDO4 1803h
Mapping object	1A00h	1A01h	1A02h	1A03h
Entry	0x60040020	0x60040020	0x60300110	0x21600020 0x21620010
Object	6004h	6004h	6030h	2160h 2162h
Subindex	00	00	01	00
Data length	20h (32 bits)	20h (32 bits)	10h (16 bits)	20h (32 bits)
	Asynchronous	Synchronous	Asynchronous	Asynchronous

Transmit PDO 1 (1800h) Position asynchronous

Default COB-ID is 180 + Node number: Example 180h + 3Fh = 1BFh

Message	Byte 0	Byte 1	Byte 2	Byte 3
1BF	Position LSB	XX	XX	Position MSB

The position values can have a maximum value of 0 – 2^{32} bits.

Transmit PDO2 (1801h) Position synchronous

Default COB-ID is 280 + Node number: Example 280h + 3Fh = 2BFh

Message	Byte 0	Byte 1	Byte 2	Byte 3
2BF	Position LSB	XX	XX	Position MSB

The position values can have a maximum value of 0 – 2^{32} bits.

Transmit PDO3 (1802h) Speed asynchronous

Default COB-ID is 380 + Node number: Example 380h + 3Fh = 3BFh

Message	Byte 0	Byte 1
3BF	Speed LSB	Speed MSB

The speed value is signed and can have values in the range
0 – 1A00h or 0 – E600h.

4 Extended functionality CANopen Encoder Profile V3.2.16

In addition, the transmission can be set so as to include an additional **CRC check sum** with the raw position data with the help of mapping Object (1A03h or 1A04h). * (optional)

Transmit PDO4 (1803h) **Raw position data from the sensor (32 bits) without scaling**
Raw position data CRC16

Default COB-ID is 480 + Node number: Example 480h + 3Fh = 4BFh

Message	Byte 0	Byte 1	Byte 2	Byte 3	Byte 5	Byte 6
4BF	LSB	...	....	MSB	CRC_L	CRC_H

The position values can have a maximum value of 0 – 2^{32} bits.

5 Service Data Transmission

SDO-COB-ID
The following identifiers are available as a standard for the SDO services:

SDO (tx) (Encoder→Master): 580h (1408) + node number
SDO (rx) (Master→Encoder): 600h (1536) + node number
The SDO identifiers cannot be modified!

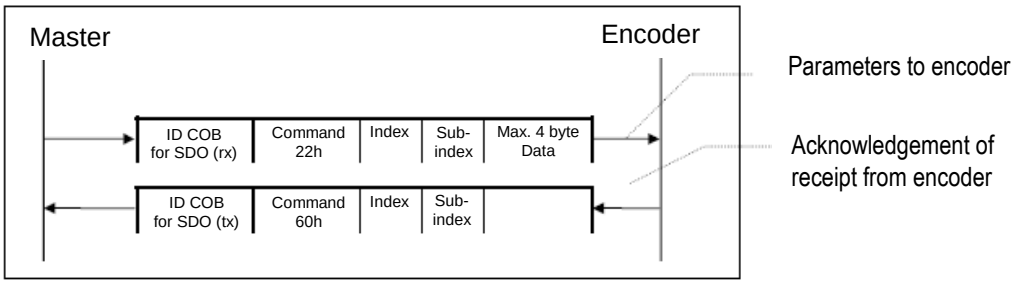
The command byte describes the type of the SDO message:

Command (Expedited Protocol)	Type	Function
22h	SDO(rx), Initiate Download Request	Send parameters to encoder (max. data length 4 byte)
23h	SDO(rx), Initiate Download Request	Send parameters to encoder (data length 4 byte)
2Bh	SDO(rx), Initiate Download Request	Send parameters to encoder (data length 2 byte)
2Fh	SDO(rx), Initiate Download Request	Send parameters to encoder (data length 1 byte)
60h	SDO(tx), Initiate Download Response	Acknowledgement of receipt by Master
40h	SDO(rx), Initiate Upload Request	Request of parameters from encoder
43h	SDO(tx), Initiate Upload Response	Parameters to Master, data length = 4 byte (unsigned 32)
4Bh	SDO(tx), Initiate Upload Response	Parameters to Master, data length = 2 byte (unsigned 16)
4Fh	SDO(tx), Initiate Upload Response	Parameters to Master, data length = 1 byte (unsigned 8)
80h	SDO(tx), Abort Domain Transfer	Encoder sends an error code to Master

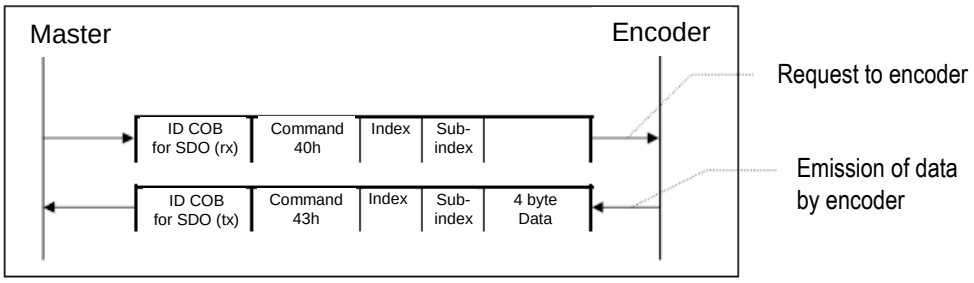
An error message (Command 80h) replaces the normal acknowledgement (response) in case of an error. The error message includes as well communication protocol errors (e.g. wrong command byte) as object dictionary access errors (e.g. wrong index, write attempt on a read-only object, wrong data length, etc.).

The error codes are described in the CANopen profile (DS 301) and in the device profile (DSP 406).

Example of a service data transmission to and from the encoder



Transmission of parameters from Master en encoder



Request for parameters from Master to encoder

6 LSS services

LSS Hardware requirements (LSS address)

All LSS slaves must have a valid entry in the object dictionary for the Identity Object [1018h] in order to be able to carry out a selective configuration of the node. This Object is made of the following subindices:

- **Vendor-ID (numerical number)**
- **Product-Code (numerical number)**
- **Revision-Number (major an minor revision as numerical number)**
- **Serial-Number (numerical number)**
- **LSS-Master CAN-ID 2021**
- **LSS-Slave CAN-ID 2020**

A product code, a revision number and a serial number are set by the manufacturer. The LSS address must be univocal in the network.

LSS operative restrictions
In order to ensure a perfect LSS functionality, all devices in the network must support the LSS services. There can only be one LSS master. All nodes must be started up wit the same baud rate.
LSS communication can only take place in "Stop" mode or in "Pre-Operational" mode.

7 CAN-Bus connection cable outlet

Abbreviation	Name	color
CG	**CAN Ground**	**grey**
CL	**CAN_L (-)**	**yellow**
CH	**CAN_H (+)**	**green**
0V	**0-Volt Power Ground**	**white**
+V	**+UB Power VCC**	**Brown**
Shield	**CAN_SHLD**	

***Termination is activated by default**

Table 1 – Terminology for connectors

Signal description	Notation
CAN_L bus line (dominant low)	CAN_L or CAN_{low} or CAN-
CAN_H bus line (dominant high)	CAN_H or CAN_{high} or CAN+
CAN ground	CAN_GND or CAN_{GND} or Ground or GND
Optional CAN shield	CAN_SHLD or CAN_{SHIELD} or Shield or SHLD
Optional CAN external positive supply	CAN_V+ or CAN_{V+} or V+ or UC or U_{CAN}
Optional ground	OPT_GND or GND_{opt} or V- or 0 V

8 First start-up - General settings on the device

Baud rate

The baud rate can be modified either with a **CANopen software on Object 2100h** or via the corresponding **LSS service**.

Default setting: 250 kBit/s (Entry 5)

Value	Baud rate in KBit/s
0	**10**
1	**20**
2	**50**
3	**100**
4	**125**
5	**250**
6	**500**
8	**1000**

To be considered for the corresponding baud rate

The selected cycle time (see Object 1800h, subindex 5 Event timer) must be larger than the duration of the bus transmission in order to allow an error-free PDO transmission!

For baud rate 10 KBaud: Minimum cycle time 14 ms
For baud rate 20 KBaud: Minimum cycle time 10 ms
For baud rate 50 KBaud: Minimum cycle time 4 ms

Node number

The node number can also be modified **by software on Object 2101h** or via the corresponding LSS service.

Default setting: 0x3F (63 decimal).

Node number 0 is reserved and may not be used by any node.
The resulting node numbers lie in the range **1...7Fh** hexadecimal (1...127 decimal)

The acceptance of a new node number only becomes effective when the encoder is rebooted (Reset/Power-on) or by means of an **NMT-Reset Node** command.
All settings in the Object table are reset to their standard/default value.

Save All Bus Parameters (2105h)

This parameter (**Object 2105h**) stores the desired bus parameters (Object 2100h, 2101h, 2102h) permanently in the Flash memory. This object is an additional protection against unintentional baud rate or node address modifications.

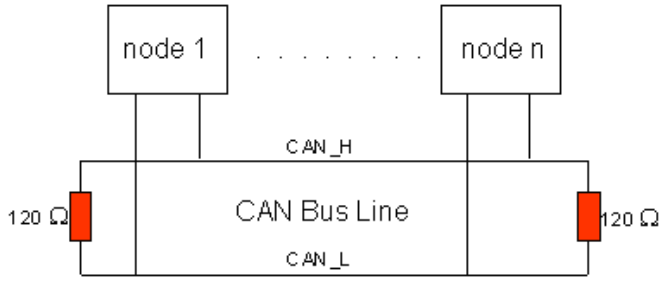
Only the targeted storage using parameter "save" **(hexadecimal 0x65766173)** saves the bus parameters **baud rate, node address and termination** permanently.

Important: **After the modification of the bus parameters and the subsequent storage with command 2105h, the application parameters must be re-programmed and saved again using Object 1010h.**

Object 1010h does not save the bus parameters.

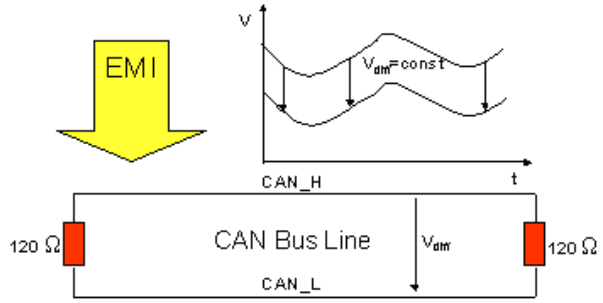
Termination

Default setting: 0x01 (bus termination active)



CAN is a 2-wire bus system to which all participants are connected in parallel (i. e. with short stub lines. The bus must be terminated at both ends with a bus termination resistor of 120 (or 121) ohms in order to avoid reflections. This is also necessary even for very short line lengths!

The **CAN bus termination** must be set **by software on Object 2102h**.



Since the CAN signals are represented on the bus as differential levels, the CAN line is comparatively insensitive to interference (EMI). Interferences always affect both lines, and therefore they almost do not modify the differential level.

Bus length

With CAN, the maximum bus length is mainly limited by the signal propagation time. The multimaster bus access procedure (arbitration) requires an almost simultaneous presence of the signals (for the duration of one before scanning) at all nodes. Since the signal propagation time is almost constant in the CAN connections (transceiver, optocoupler, CAN controller), the line length must be adapted to the baud rate.

Baud rate	Bus length
1 MBit/s	< 20 m*
500 kBit/s	< 100 m
250 kBit/s	**< 250 m**
125 kBit/s	< 500 m
50 kBit/s	< 1000 m
20 kBit/s	< 2500 m
10 kBit/s	< 5000 m

*) the length of 40 m for 1 MBit/s is often quoted in the literature for CAN. However, this does not apply to networks with optodecoupled Can controllers. The worst-case calculation with optocouplers gives, for 1 MBit/s, a maximum bus length of 5 m - experience shows however that 20 m can be reached without problem.

For bus lengths exceeding 1,000 m, the use of repeaters may become necessary.

9 Layer Setting Services (LSS services)

Exactly two conditions must be met for devices to be connected to a CANopen network - all devices must have the same baud rate and every device must have a unique node address within the network. The condition for a use under LSS is that there is a 1:1 CAN connection with the device. A special dialogue mode then allows modifying the baud rate and the node address. COB-ID 0x7E5 is sent from the master to the slave, the slave answers with COB-ID 0x7E4.
LSS messages are always 8-byte long. Unused bytes are reserved and should be filled with 0.

A "Switch Mode Global" command is sent to switch a device in LSS configuration mode:

0x04	0x01	reserved

Unfortunately, this command is not confirmed, and only a visual check of the LEDs allows making sure that the device really switched to this mode.

The following command calls for the "Inquire Node-ID" service.

0x5E	reserved

If the command was carried out successfully, the slave answers with:

0x5E	Node ID	reserved

If no feedback answer is received from the device, the LSS service may not be supported, or the baud rate may be incorrect.

The "Configure Node-ID" command is used to re-configure the node address:

0x11	Node ID	reserved

The error code is included in the answer of the slave device:

0x11	Error code	Error extension	reserved

Error code 0 means that the command has been accepted - Error code 1 indicates an invalid Node ID.
The remaining error codes are reserved. The error extension includes manufacturer-specific information can only be used in case of error code 0xFF.

The baud rate is activated with the command "Configure Bit Timing Parameters".

0x13	Bit timing	Table entry	reserved

The following baud rates are standardised by CiA:

Baudrate table 0x00	
Table index	Baud rate
0	1000 kBit/s
1	800 kBit/s * not supported
2	500 kBit/s
3	250 kBit/s
4	125 kBit/s
5	reserved
6	50 kBit/s
7	20 kBit/s
8	10 kBit/s

Again, the device answers with an error code:

0x13	Error code	Error extension	reserved

Error code 0 means that the command has been accepted - Error code 1 indicates an invalid Node ID.
The remaining error codes are reserved. The error extension includes manufacturer-specific information can only be used in case of error code 0xFF.

Both network-specific parameters have now been modified, and the new parameters must be saved using "Store Configuration":

0x17	reserved

Again, the device answers with an error code:

0x17	Error code	Error extension	reserved

Error code 0 means that the command has been accepted - Error code 1 indicates an invalid Node ID.
The remaining error codes are reserved. The error extension includes manufacturer-specific information can only be used in case of error code 0xFF.

To complete the LSS service, the device is switched back from the LSS configuration mode to the Preoperational mode with the command "Switch Mode Global":

0x04	0x00	reserved

The device performs automatically a new boot-up (Reset node), after which all new settings become valid.

10 Default settings on delivery

Devices with cable outlet

Description	Setting	Software
Baud rate	**250 kBit/s**	Object 2100h = 05h
Node address	**63**	Object 2101h = 3Fh
Termination	**on**	Object 2102h = **01h**

Communication parameters

Index (hex)	Name	Standard value
1005h	COB-ID Sync	80h
100Ch	Guard Time	0
100Dh	Life Time Factor	0
1012h	COB-ID Time stamp	100h
1013h	High Resolution time stamp	0
1016h	Consumer heartbeat time	Node-ID 0,Time=0
1017h	Producer heartbeat time	0
1029h	Error Behaviour	0 = Comm Error
		1 = Device specific
		1 = Manufacturer Err.
1800h	**TPDO1** Communication Parameter	
01h	COB-ID	180h + node number
02h	Transmission Type	255 (asynchronous)
03h	Inhibit Time	0 [step 100 µs]
05h	Event timer	0 [step 1 ms]
1801h	**TPDO2** Communication Parameter	
01h	COB-ID	280h + node number
02h	Transmission Type	1 (synchronous)
03h	Inhibit Time	0 [step 100 µs]
05h	Event timer	0 [step 1 ms]
1802h	**TPDO3** Communication Parameter	
01h	COB-ID	30h + node number
02h	Transmission Type	255 (asynchronous)
03h	Inhibit Time	0 [step 100 µs]
05h	Event timer	0 [step 1 ms]
1803h	**TPDO4** Communication Parameter	
01h	COB-ID	101h -13Fh
02h	Transmission Type	255 (asynchronous)
03h	Inhibit Time	0 [step 100 µs]
05h	Event timer	0 [step 1 ms]
1A00h	TPDO1 Mapping	
01h	1.Mapped Object	0x60040020
1A01h	TPDO2 Mapping	
01h	1.Mapped Object	0x60040020

1A02h	TPDO3 Mapping	
01h	1.Mapped Object	0x60300110
1A03h	TPDO4 Mapping	
01h	1.Mapped Object	0x21600020
	2. Mapped Object	0x21620010

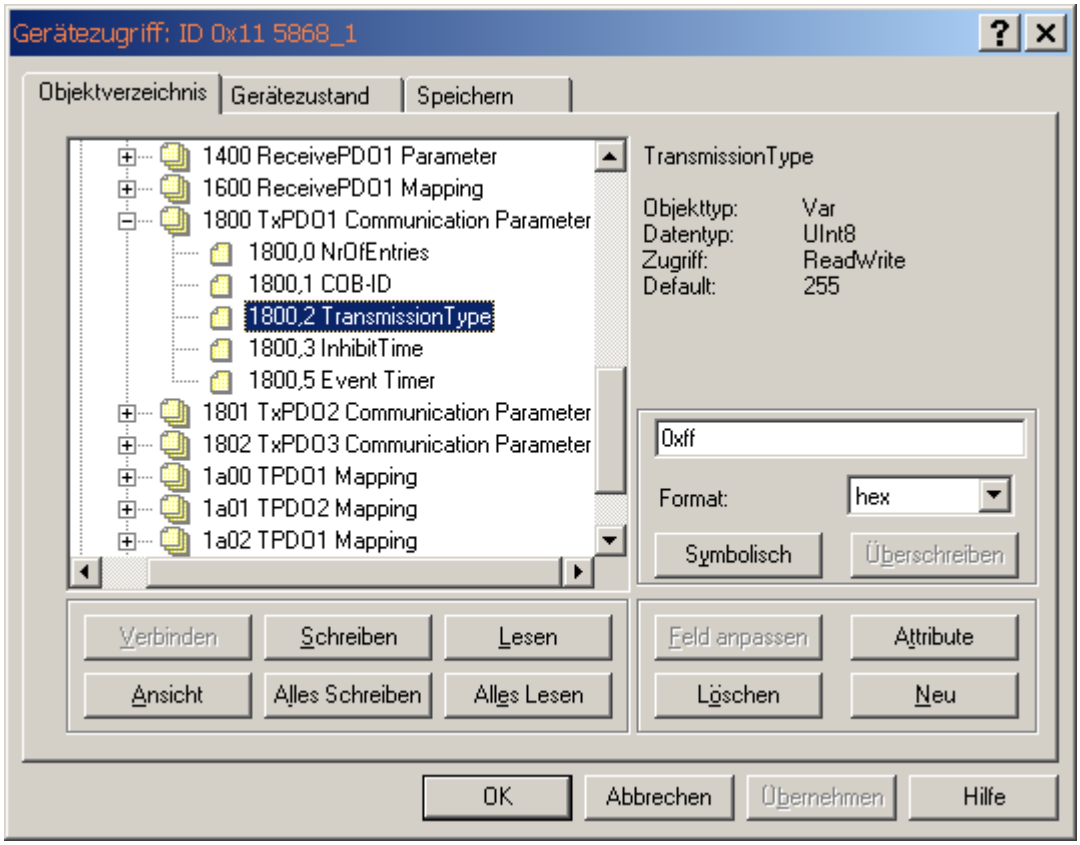
Index (hex)	Name	Standard value
	Encoder Profile	
6000h	Operating Parameter	0x00 Scaling off
6001h	Measuring Units per Revolution	**2^16 bits**
6002h	Total Measuring Range	**2^32 bits** (Singleturn 2^16)
6003h	Preset value	0
6200h	Cyclic Timer (see TPDO1 Comm.Par)	0
6031H	Speed Parameter	
	Speed Source Selector	2
	Speed Integration Time	100
	Speed Calc Multiplier	1
	Speed Calc Divisor	1
6401h	Work area low limit	0
6402h	Work area high limit	Max.Resolution
650Dh	Absolute Accuracy	13 bits
650EH	Device Capability	3
2100h	Baud rate	5
2101h	Node address	0x3F
2102h	Termination	1
2105h	Save All Bus Parameters	0x65766173
300Ah	Password-protected Area	
300Bh	Serial Number	

The original standard values (default values on delivery) can be reloaded using Object **1011h** **(Restore the parameters)**
with the indication "load" **(0x6C6F6164)**.

In order to save the modified parameters against power failures, they must imperatively be transferred in the FLASH MEMOY using Object **1010h**.
This operation will overwrite all data previously stored in the FLASH MEMORY!

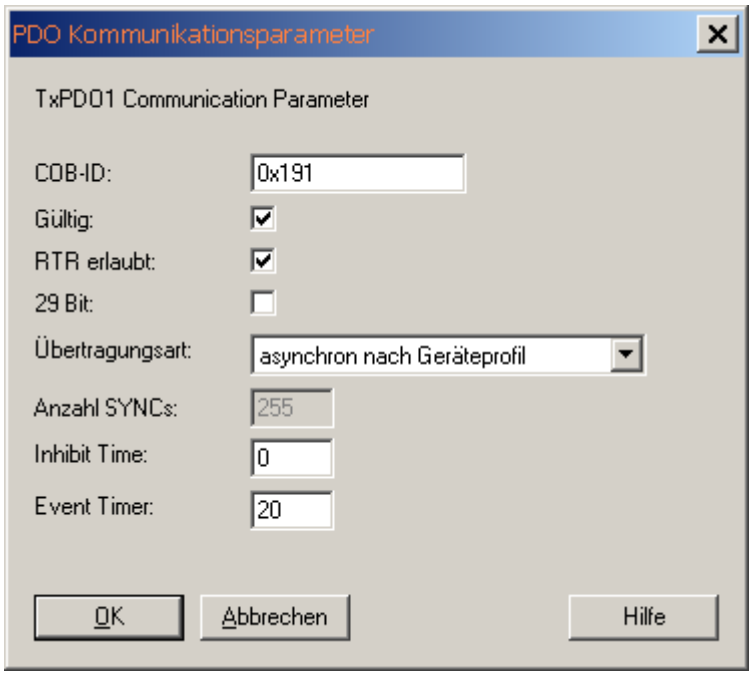
11 Communication parameters

The COB-ID and the transmission type for **PDO1** are defined in the object dictionary index **1800h**.

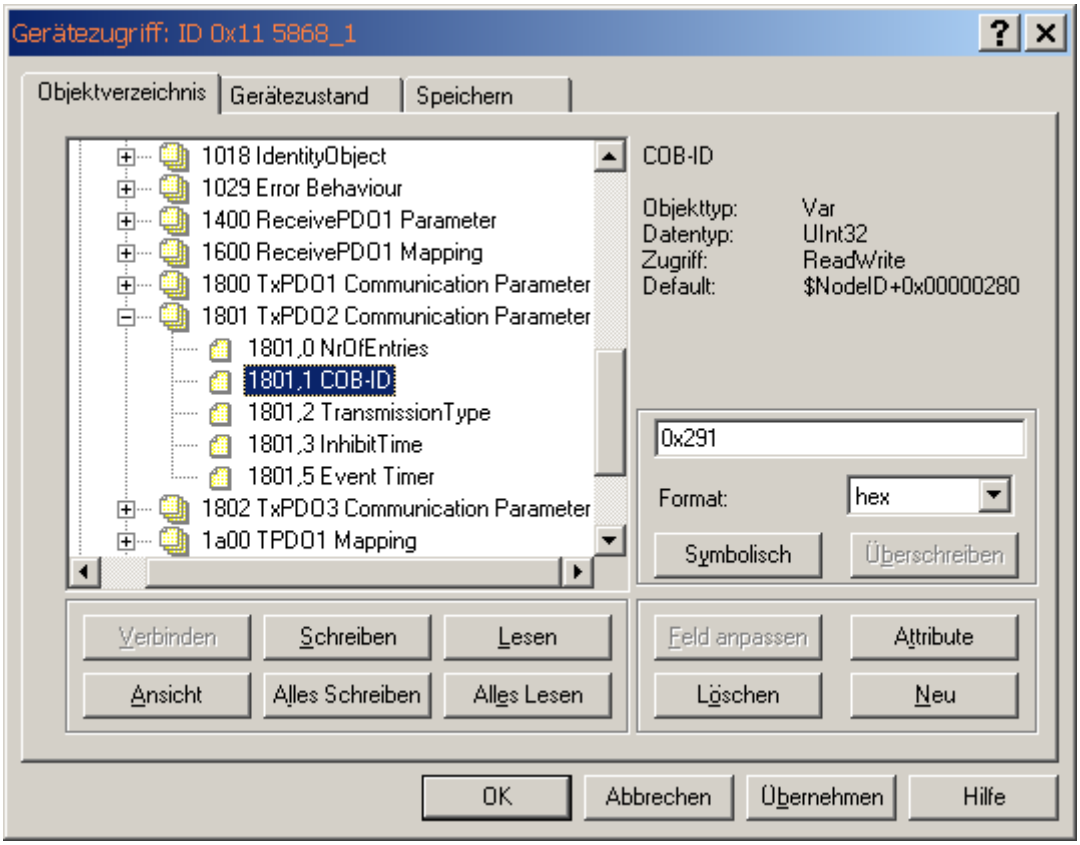


Default settings:

Release:	**PDO valid (enabled)**	**RTR enabled**
COB-ID:	**180h + set node number (here 11h)**	
Transmission type:	**255 = asynchronous according to device profile**	
Event Timer:	**20 ms**	

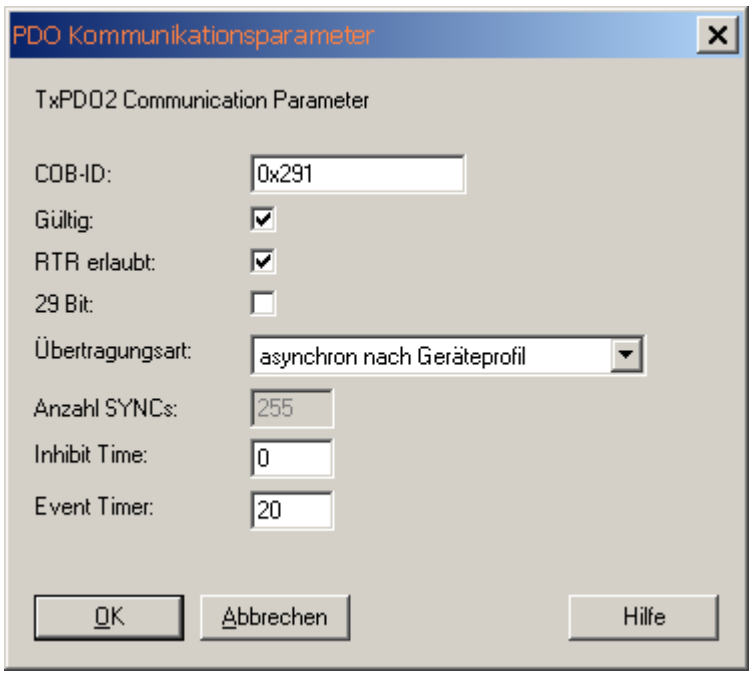


The COB-ID and the transmission type for **PDO2** are defined in the object dictionary index 1801h.



Default settings:

Release:	**PDO valid (enabled)**	**RTR enabled**
COB-ID:	**280h + set node number (here 11h)**	
Transmission type:	**255 = asynchronous according to device profile**	
Event Timer:	**20 ms**	



Definition of the transmission type (Transmission type) of the PDOs

transmission type	PDO transmission				
	cyclic	acyclic	synchronous	asynchronous	RTR only
0		X	X		
1-240	X		X		
241-251	- reserved -				
252			X		X
253				X	X
254				X	
255				X	

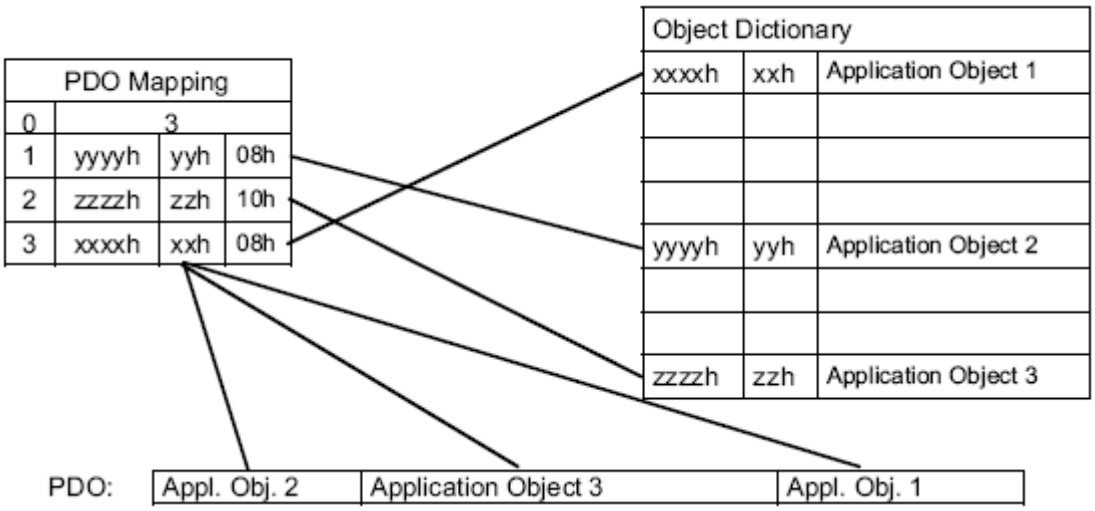
A value between 1 ... 240 means that the PDO is sent **synchronously and cyclically**. The number of the transmission type indicates the **number of SYNC pulses** required to send the PDOs.
Transmission types 252 and 253 indicate that the PDO is only sent upon request through RTR.

Type 254 indicates an application-dependent triggering of the event, while number 255 indicates a device profile-dependent triggering. In addition, a time-controlled **EventTimer** may be used for numbers 254/255. The value range for the timer reaches from **1 ms ... 65535 ms**.

Variable PDO mapping

Variable mapping of the various objects means that the user can configure the content of the transmit PDOs according to his application.

Example of an entry in the mapping table:
The mapped PDO includes 3 application object entries with different lengths:

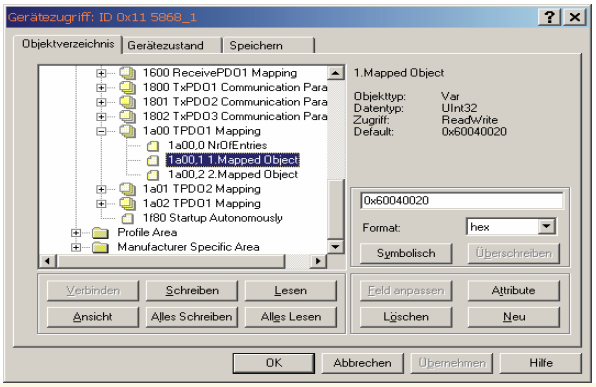


Application object 2 occupies the first byte (08h) in the transmit PDO. It is followed with the 16-bit long application object 3 (10h = 2 bytes), and finally the 1-byte long application object 1. A total of 32 bits are occupied in this PDO.

12 Example of a variable mapping entry

Mapping Object 1A00h

Mapping object **1A00h describes the first transmit PDO**. Objects may be mapped until the maximum data length of **8 bytes** is reached. Similarly, object **1A01h describes PDO2** and **1A02h describes PDO3**.



Mapping	TPDO1 Mapping	TPDO1 Mapping	TPDO1 Mapping
Subindex	00	01	02
Entry	Nbr of entries	1st mapped object	2nd mapped object
Object	2	6004h	6030h
Subindex		00	01
Data length	Byte	20h(32 Bit)	10h(16 Bit)
		Asynchron	Asynchron

The following objects can be mapped: (indicated in blue)

Device-**specific objects**

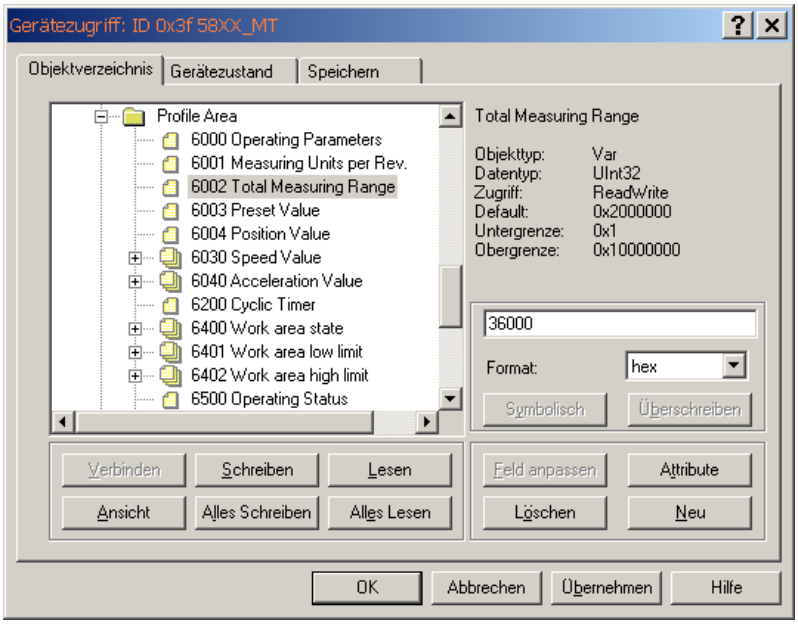
INDEX (hex)	Object Symb.	ATTRIB	Name	M/O C2	TYPE
6000	VAR	RW	Operating parameters	M	unsigned16
6001	VAR	RW	Measuring Units p.Revolution (MUR)	M	unsigned32
6002	VAR	RW	Total Measuring Range (TMR)	M	unsigned32
6003	VAR	RW	Preset value	M	unsigned32
6004	**VAR**	**RO**	**Position value**	**M MAP**	**unsigned32**
6030	**ARRAY**	**RO**	**Speed Value**	**O MAP**	**signed16**
6040	ARRAY	RO	Acceleration Value	O	Signed16
6200	VAR	RW	Cyclic Timer	M	unsigned16
6400	**ARRAY**	**RO**	**Working Area state**	**O MAP**	**unsigned 8**
6401	ARRAY	RW	Working Area Low Limit	O	Unsigned32
6402	ARRAY	RW	Working Area High Limit	O	Unsigned32
6500	VAR	RO	Operating Status	M	unsigned16
6501	VAR	RO	Measuring Step (Singleturn)	M	unsigned32
6502	VAR	RO	Number of revolutions	M	unsigned16
6503	**VAR**	**RO**	**Alarms**	**M MAP**	**unsigned16**
6504	VAR	RO	Supported alarms	M	unsigned16
6505	**VAR**	**RO**	**Warnings**	**M MAP**	**unsigned16**

In this example, 2 process data objects, 6004h and 6030h, are mapped.

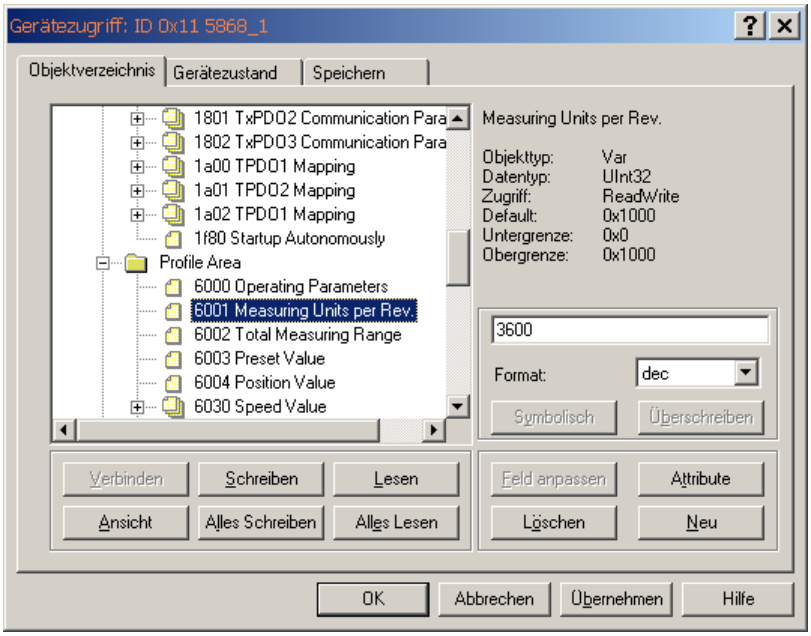
Programming example of an application:
Objects set-up

- Limit Total Measuring Range to 36000
- Measuring Units per Revolution must be set to 3600 steps per revolution
- Position value must be set to 0
- TPDO1 (Position) must be sent by 10 ms events
- TPDO2 (Speed) must be sent by 20 ms events
- Producer Heartbeat must be reduced to 500 ms
- Work area Limit values are 1000 and 35000
- The new parameters must be saved in the **FLASH MEMORY**

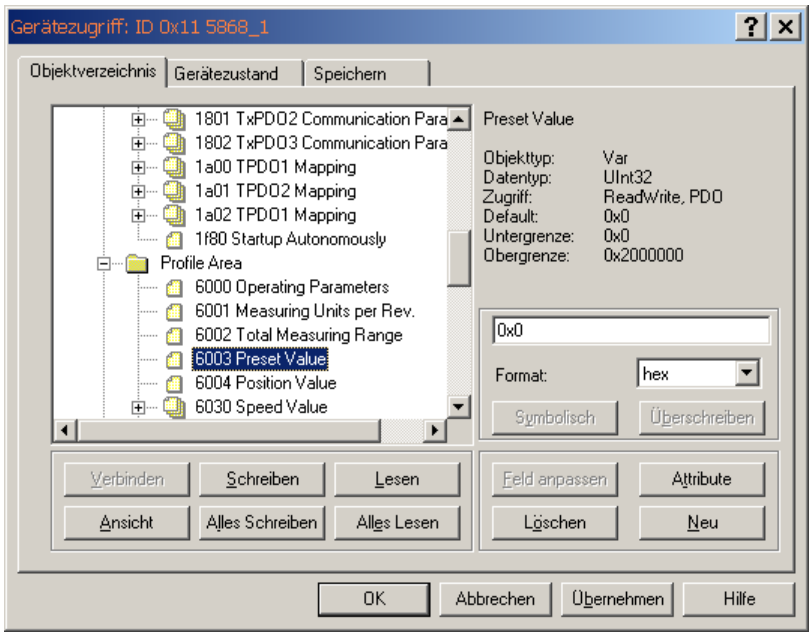
Limiting Total Measuring Range to 36000



Limiting Measuring Units per Revolution to 3600

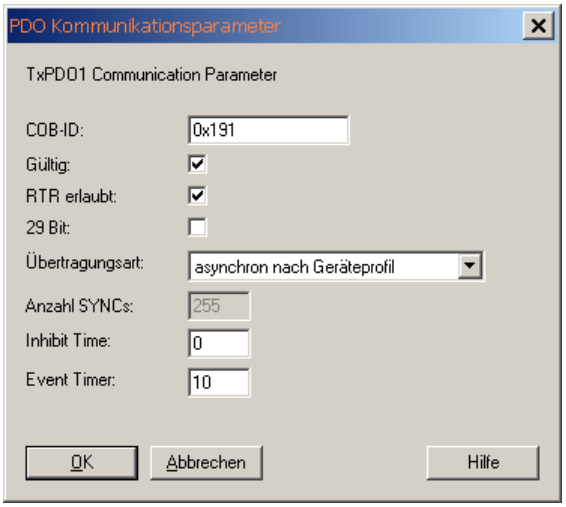


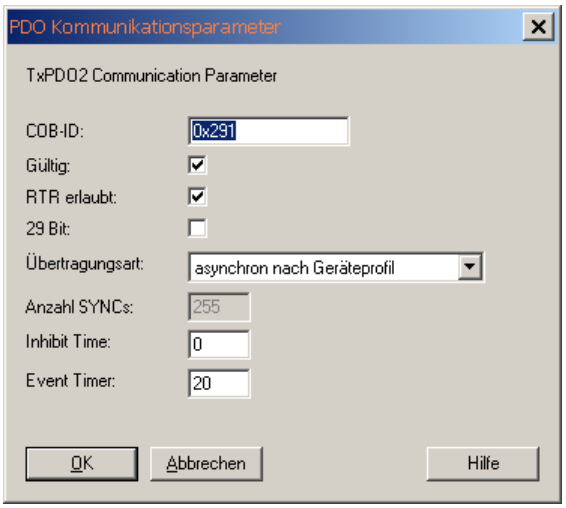
Preset value to 0



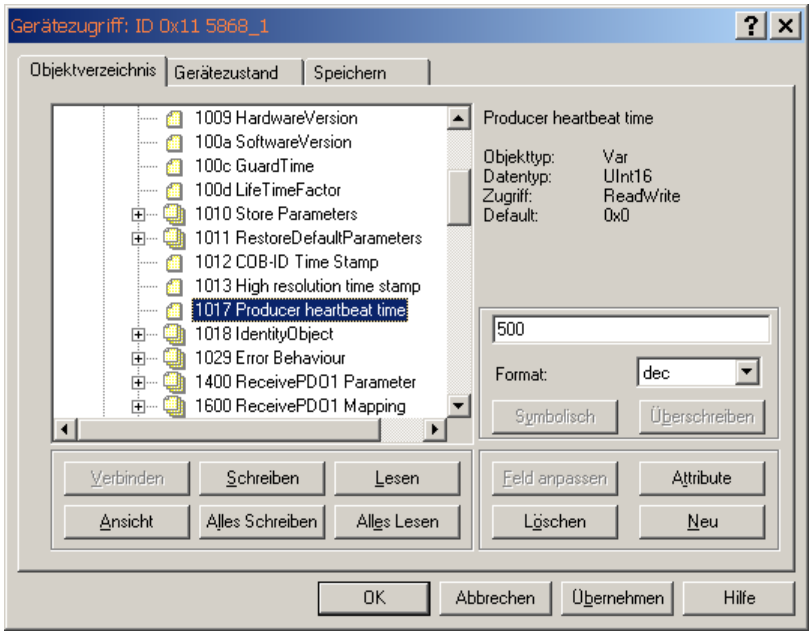
Definition of the values of transmit parameters TPDO1 and TPDO2

Type 254 indicates an application-dependent triggering of the event, while number 255 indicates a **device profile-dependent** triggering. In addition, a time-controlled **EventTimer** may be used for numbers 254/255. The value range for the timer reaches from
1 ms ... 65535 ms.

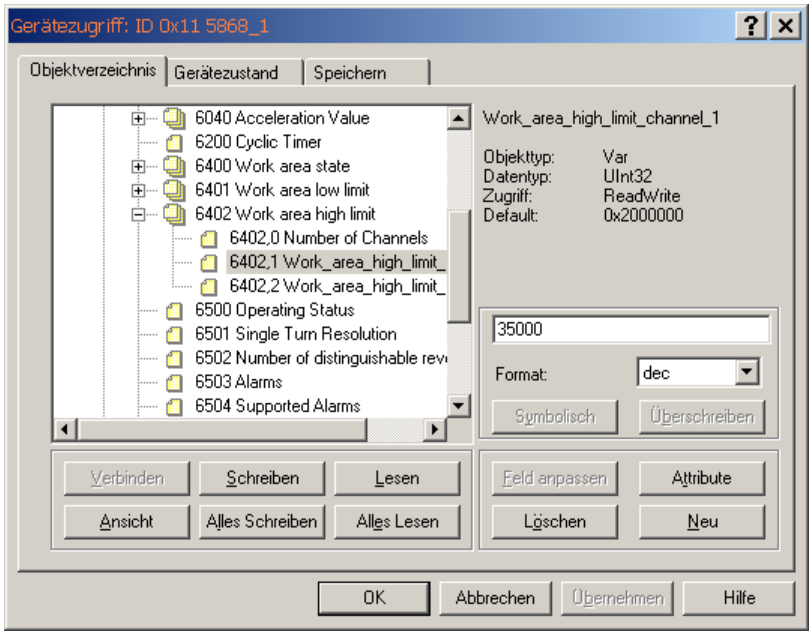




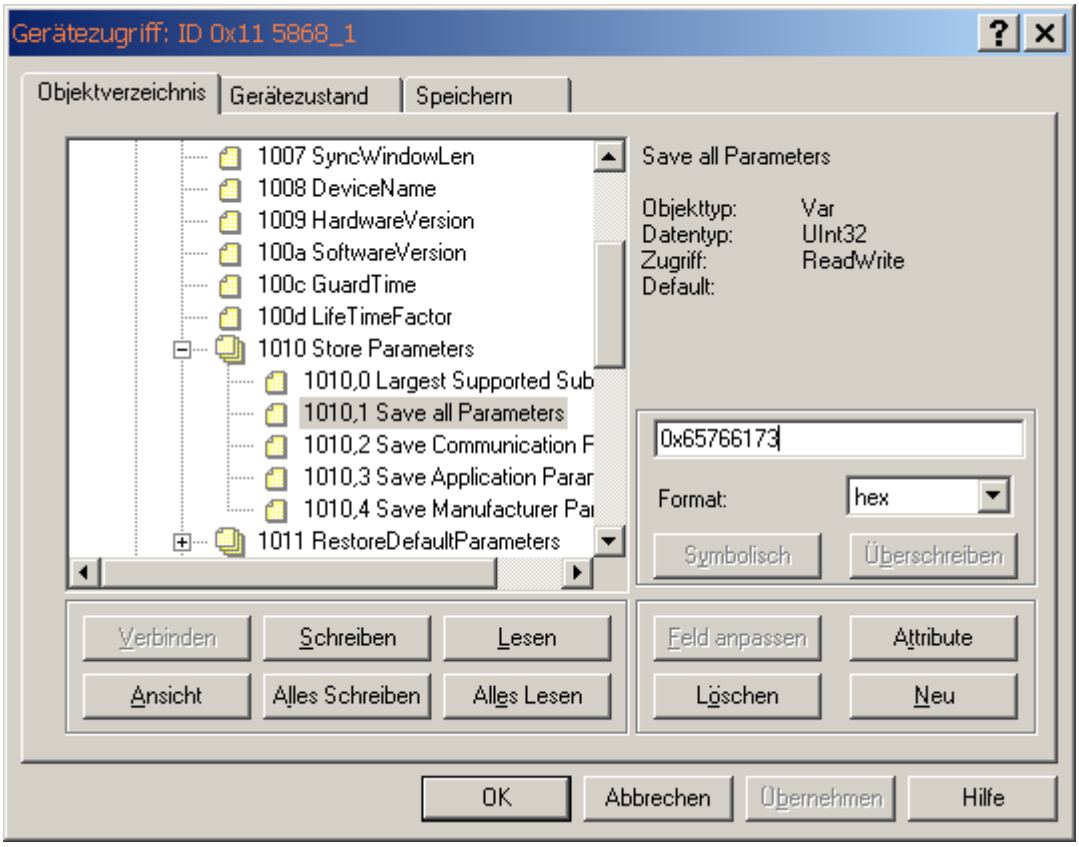
Setting Producer Heartbeat to 500 ms



Defining the Work area Low and High Limit values



Saving all modified parameters in the Flash memory
Store Parameters 1010h



Object 1010h Store Parameters

Command "save" under subindex 1h (save all Parameters) stores the parameters in the non-volatile memory (FLASH MEMORY).
This sub-item stores all communication objects, application objects and manufacturer-specific objects. ***This operation requires about 14 ms.***

In order to avoid any unintentional storage, the command is only carried out when the string "save" is entered as code word in this subindex.
A read access to subindex 1h provides information about the memory functionality.

Byte 0: 73h (ASCII-Code for "s")
Byte 1: 61h (ASCII-Code for "a")
Byte 2: 76h (ASCII-Code for "v")
Byte 3: 65h (ASCII-Code for "e")

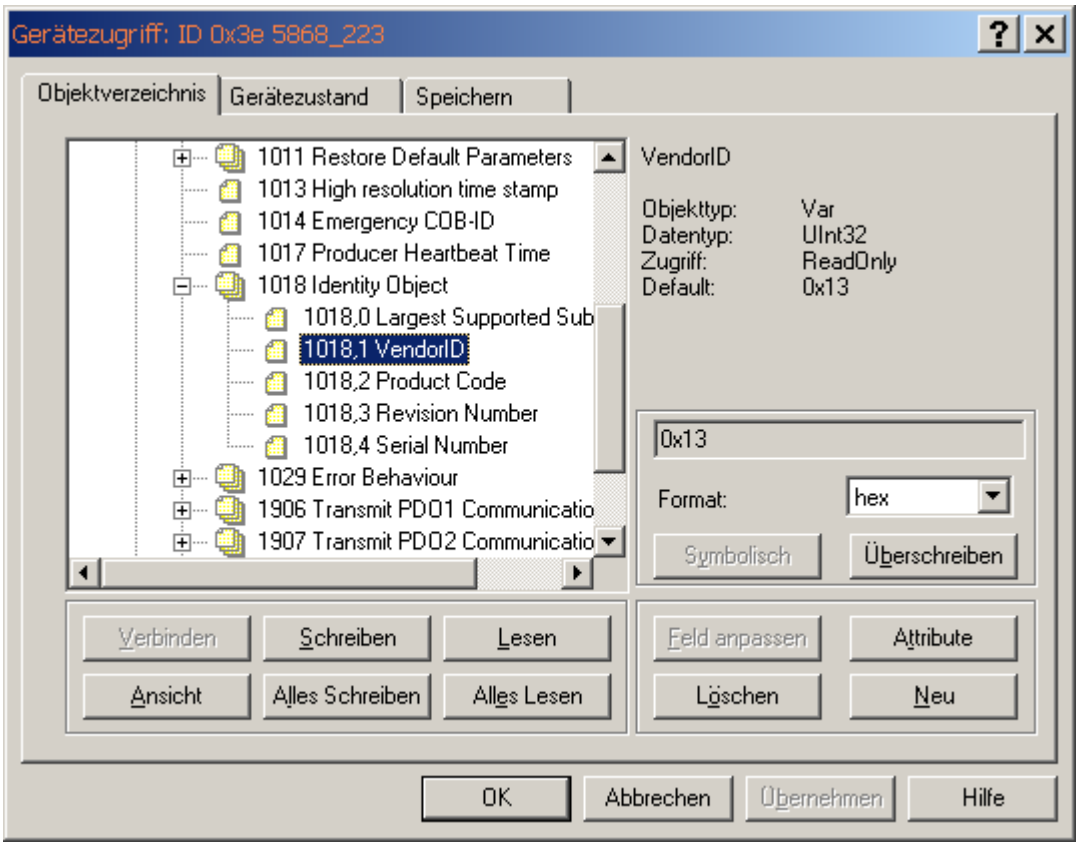
Object 1011h: Load standard values

The command "load" under subindex 1h resets all parameters to their standard values. In order to avoid any unintentional loading of the standard values, the command is only carried out when the string "load" is entered as code word in this subindex.

Byte 0: 6Ch (ASCII-Code for "l")
Byte 1: 6Fh (ASCII-Code for "o")
Byte 2: 61h (ASCII-Code for "a")
Byte 3: 64h (ASCII-Code for "d")

Communication profile - further objects
Object 1018h: Identity Object

Information about the manufacturer and the device:



1018 RECORD Device – Identification read only

Sub-Index 0h : Number of Subindices“
returns the value 4
Sub-Index 1h: "read” only
returns the Vendor-ID (000000013h) Fritz Kübler GmbH
Subindex 2h: returns the Product Code
(e. g. 0x56682001 CANopen encoder)
Sub-Index 3h: "read” only
returns the software revision number
(e. g. 108)
Sub-Index 4h: "read” only
returns the 11-digit **Serial number** of the encoder

13 Speed output configuration

The rotational speed of the encoder shaft is determined as the value difference between two physical (unscaled) position values with a dynamic time interval of 1ms, 10 ms or 100ms, or 200ms ***(Object 6031h).***
To adapt the speed measurement to the concerned application, the operator can use 3 parametrisable objects in the manufacturer-specific area. In case of high speeds, the integration time of the measurement can be reduced to achieve accordingly high dynamics. The number of mean values especially affects the dynamics of the measurements; it must be determined according to the application.

Speed measurement accuracy
The accuracy of the measurement depends mainly of the following parameters:

- actual speed
- temporal speed change (internal dynamics)

Object 6031h: Speed gating Time (values for speed measurement)

The speed is calculated according to the following formula:

$$\textbf{Speed} = \frac{\textbf{Position change}}{\textbf{Integration time}} \times \textbf{Units factor}$$

For the units factor, parameters are available under **Object 6031, sub3 (Speed Calculation Multiplier)** and under **Object 6031,sub4 (divisor)**. Speed is output either in **RPM** or as a number of **steps per second** in **Object 6000h Bit 13**.

Parameter **Object 6031,sub3** (Speed Measuring Multiplier) can contain e. g. the periphery of a measuring wheel in order to affect the speed.

Important: Under **Objects 6031, sub3/4**, only the speed output with the **unit [unit/sec]** may be affected, the **RPM** output **is not** parametrisable.

14 Emergency message

Emergency Objects arise in case of error situations within a CAN network and are triggered depending on the event and transmitted over the bus with a **high priority**.
Important: an Emergency Object is only triggered **once per "Event"**. No new object is generated as long as the error is active. Once the error is eliminated, a new Emergency Object with the content 0 (Error Reset or No Error) is generated and transmitted over the bus.

Byte	0	1	2	3	4	5	6	7
Content	Emergency Error Code (see Table 21)		Error register (Object 1001H)	Manufacturer specific Error Field				

Figure 34: Emergency Object Data

The behaviour in case of an error is described in **Object 1029h Error Behavior**
Example of a message for overtemperature:

Transfer Data	**00**	**42**	**09**	**80**	**56**	**20**	**50**	**2E**

[Errcode]		**4200 Sensor temperature threshold value exceeded**
[Error Register]	**09**	**Error register**
[ManufacturerSpecific1]	**80**	ICLG Error register
[ManufacturerSpecific2]	**56**	ICLG momentary temperature
[ManufacturerSpecific3]	**20**	ICLG current threshold value lower range
[ManufacturerSpecific3]	**50**	ICLG current threshold value higher range
[ManufacturerSpecific5]	**2E**	ICLG Version register

15 Implemented error codes

Error Code	Error register	BYTE 3	Byte 4	BYTE 5	Byte 6	Byte 7	Remarks
5200	01	09	81	45	00	00	**ICLG Optic Failure**
		81					ICLG Error Mask Register
		45					ICLG Error Register
4200	01	07	81	A8	20	A2	**System Temperature Error**
		81					ICLG Error Register
		A8					ICLG Temperature Register
		20					ICLG Temperature Lower Reg
		A2					ICLG Temperature Upper Reg
3200	01	02	6A	01	00	00	**System Battery Low Error**
		6A					Battery low voltage (LSB)
		01					Battery Low voltage (MSB)
3200	05	02	**00**	**00**	00	FF	**Battery cable damage**
8110	11	00					CAN Overrun Error
8120	11	00					CAN Passive Error Mode
8130	01	00					**Life Guard or Heartbeat Error**
6200	00/01	00	XX	XX	XX	XX	**USF new Offset value**

16 Heartbeat Consumer Protocol

Object 1016h: Consumer Heartbeat Time

If this function is to be activated, a valid node ID to be monitored, with a corresponding time, must be entered in Object 1016h,Subindex 1 and/or 1016h,Subindex 2.
The time entered must always be larger than the Heartbeat Producer time that is to be monitored.
The function is active upon a boot-up cycle, provided the entered data has been saved.
(Store parameters object 1010h).

Monitoring is started **after the first occurrence of a Heartbeat** with the corresponding node ID. If a **0** ms time is entered, the function is inactive.
Valid settings: **1ms up to max. 65535 ms**

	MSB		LSB
Bits	31-24	23-16	15-0
Value	reserved (value: 00h)	Node-ID	heartbeat time
Encoded as	-	UNSIGNED8	UNSIGNED16

Figure 62: Structure of Consumer Heartbeat Time entry

OBJECT DESCRIPTION

INDEX	1016h
Name	Consumer Heartbeat Time
Object Code	ARRAY
Data Type	UNSIGNED32
Category	Optional

ENTRY DESCRIPTION

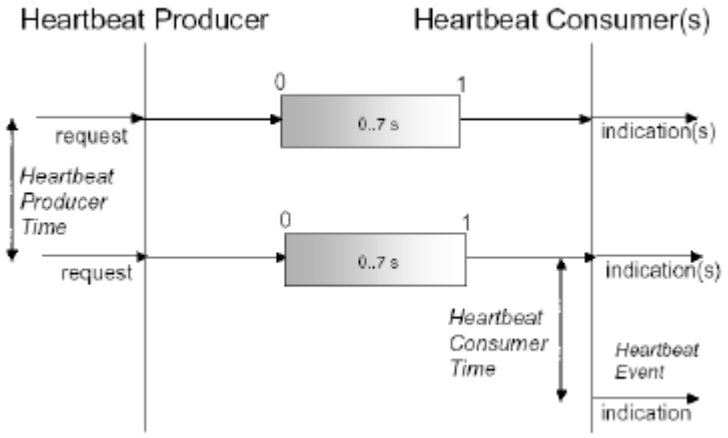
Sub-Index	0h
Description	number entries
Entry Category	Mandatory
Access	ro
PDO Mapping	No
Value Range	1 – 127
Default Value	No

Sub-Index	1h
Description	Consumer Heartbeat Time
Entry Category	Mandatory
Access	rw
PDO Mapping	No
Value Range	UNSIGNED32 (Figure 62)
Default Value	0

Sub-Index	2h – 7Fh
Description	Consumer Heartbeat Time
Entry Category	Optional
Access	rw
PDO Mapping	No
Value Range	UNSIGNED32 (Figure 62)
Default Value	No

Two nodes with node ID and corresponding time setting are supported.

Entries with different times for **one node ID** and changes of the entries **without previously erasing the values with 0** are answered with an abort code **0604 0043h** (General parameter incompatibility reason).



One or several **Heartbeat Consumers** receive the Producer's message. If this message is missing for any reason after the time set for the Consumer has elapsed, a "**Heartbeat Event**" is generated.

The Heartbeat Consumer device activates an emergency message with an error code "**8130 Lifeguard or heartbeat**". Depending on the setting of Error behaviour **Objects1029h,Sub 1**, the Consumer switches back to **Preoperational state** when this error appears. **The behaviour is therefore determined by Object 1029h Subindex 1 "Communication Error".**
(0 = switch to Preop, 1= no state change)

Configuration example:

Object 1016, 1 h: Consumer Heartbeat

Time

	MSB		LSB
Bits	31-24	23-16	15-0
Value	reserved (value: 00h)	Node-ID	heartbeat time
Encoded as	-	UNSIGNED8	UNSIGNED16

Figure 62: Structure of Consumer Heartbeat Time entry

Sample string:	00	07	1F4	= 0x000701F4

Monitored device **Node 07** **Time = 500 ms**

In case of an error, the following emergency message is generated:

Transfer Data	30	81	11	00	00	00	00	00

[Errcode]	**8130**	**Life Guard or Heartbeat error**
[Error Register]	**11**	**Error Register**
[ManufacturerSpecific1]	**00**	ICLG error register

A NMT "Reset-Node Command"* of the Consumer device or a new input of data in Object 1016h activates the Supervisor function again.
(*Only if the storage has been carried out previously with Object 1010h.)

17 Heartbeat Producer Protocol

Object 1017h: Producer Heartbeat Time

The Producer heartbeat time defines the heartbeat cycle. If this function is not required, input a **0 time**. This function is activated with a time starting from **1 ms**. (max. 65535 ms)

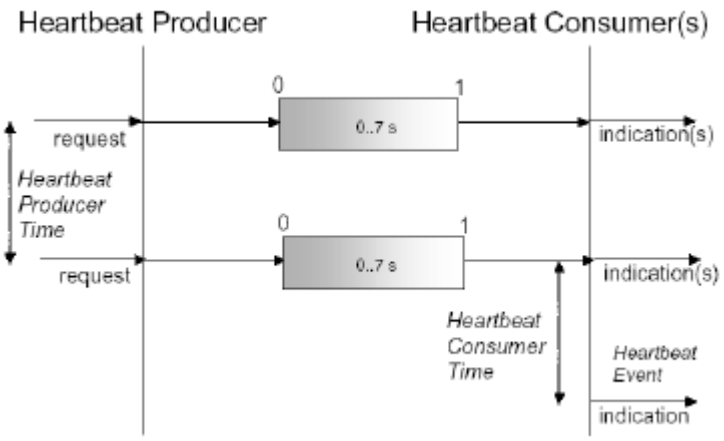
OBJECT DESCRIPTION

INDEX	1017h
Name	Producer Heartbeat Time
Object Code	VAR
Data Type	UNSIGNED16
Category	Conditional; Mandatory if guarding not supported

ENTRY DESCRIPTION

Access	rw
PDO Mapping	No
Value Range	UNSIGNED16
Default Value	0

A "Heartbeat Producer" **transfers the message cyclically with the set time.** The content of the data byte corresponds to the CAN node status. (Pre-op, Operational, Stopped)



18 CANopen Object Dictionary

Index (hex)	Subindex (hex)	Object	Name	Type	Attr.	M/O

Index: 16-bit address of the record
Subindex: 8-bit pointer to subentries;
used only for complex data structures (e. g. arrays);
if there is no subentry: subindex = 0

Object:	NULL	Entry without data
	DOMAIN	High, variable quantity of data, e. g. programme code
	DEFTYPE	Data type definition, e. g. boolean, float, unsigned 16, etc.
	DEFSTRUCT	Definition of an entry, e. g. PDO mapping structure
	VAR	Individual value, e. g. boolean, float, unsigned 16, string, etc.
	ARRAY	Array of data of the same type, e. g. unsigned 16 data
	RECORD	Field of data of different types

Name : Short description of the function
Type : Data type, e. g. boolean, float, unsigned 16, integer, etc.
Attr. : Attribute that defines the access rights to the object :

	rw	Read/write
	ro	Read only
	const	Read only, the value is a constant
M/O	M	Mandatory: the objet must be implemented in the device
	O	Optional: the objet does not have to be implemented in the device

Structure of the whole object dictionary

Index (hex)	Object
0000	not used
0001 - 001F	static data types
0020 - 003F	complex data types
0040 - 005F	manufacturer-specific data types
0060 - 0FFF	reserved
1000 - 1FFF	Communication profile
2000 - 5FFF	manufacturer-specific profile
6000 - 9FFF	standardised device profile
A000 - FFFF	reserved

19 CANopen communication profile DS 301

Communication objects

INDEX (hex)	OBJECT SYMBOL	ATTRIB	Name	M/O	TYPE
1000	VAR	CONST	Device Type	M	Unsigned32
1001	VAR	RO	Error Register	M	Unsigned8
1002	VAR	RO	Manufacturer Status	O	Unsigned32
1003	RECORD	RO	Predefined Error Field	O	Unsigned32
1004	ARRAY	RO	Number of PDO supported	O	Unsigned32
1005	VAR	RW	COB-ID Sync message	O	Unsigned32
1006	VAR	RW	Communication cycle period	O	Unsigned32
1007	VAR	RW	synchr.window length	O	Unsigned32
1008	VAR	CONST	Manufacturer Device Name	O	visible string
1009	VAR	CONST	Manufacturer Hardware Version	O	visible string
100A	VAR	CONST	Hardware Version	O	visible string
100B	VAR	RO	Node-ID	O	Unsigned32
100C	VAR	RW	Guard Time	O	Unsigned32
100D	VAR	RW	LifeTime Factor	O	Unsigned32
1010	VAR	RW	Store parameters (Device Profile)	O	Unsigned32
1011	VAR	RW	Restore parameters (Device Profile)	O	Unsigned32
1014	VAR	RO	COB_ID Emcy	O	Unsigned32
1015	VAR	RW	Inhibit Time Emcy	O	Unsigned32
1016	ARRAY	RW	Consumer Heartbeat time	O	Unsigned32
1017	VAR	RW	Producer Heartbeat time	O	Unsigned16
1018	RECORD	RO	Identity Object	M	PDOComPar
1029	ARRAY	RW	Error Behaviour	O	Unsigned8
1800	**RECORD**		**1st transmit PDO Comm. Par.**	**O**	**PDOComPar**
1801	**RECORD**		**2nd transmit PDO Comm. Par.**	**O**	**PDOComPar**
1802	**RECORD**		**3rd transmit PDO Comm. Par.**	**O**	**PDOComPar**
1803	**RECORD**		**4th transmit PDO Comm. Par.**	**O**	**PDOComPar**
1A00	**ARRAY**		**1st transmit PDO Mapping Par.**	**O**	**PDOMapping**
1A01	**ARRAY**		**2nd transmit PDO Mapping Par.**	**O**	**PDOMapping**
1A02	**ARRAY**		**3rd transmit PDO Mapping Par.**	**O**	**PDOMapping**
1A03	**ARRAY**		**4th transmit PDO Mapping Par.**	**O**	**PDOMapping**

Manufacturer-specific objects

2100	VAR	RW	Baud Rate	O	Unsigned 8
2101	VAR	RW	Node number	O	Unsigned 8
2102	VAR	RW	CAN Bus Termination	O	Unsigned 8
2103	VAR	RO	Firmware Flash Version	O	Unsigned16
2105	VAR	RW	Save All Bus Parameters	O	Unsigned32
2125	VAR	RO	Battery voltage	O	Unsigned16
2140	Array	RW	Customer Memory	O	Unsigned32
2160	VAR	RO	Raw position data	O	Unsigned32
2161	VAR	RO	Complemented position data	O	Unsigned32
2162	VAR	RO	Raw position data CRC16	O	Unsigned16

CANopen Encoder Device Profile DS 406 V3.1

Device-specific objects

INDEX (hex)	Object Symb.	ATTRIB	Name	M/O C2	TYPE
6000	VAR	RW	Operating parameters	M	unsigned16
6001	VAR	RW	Measuring Units p. Revolution (MUR)	M	unsigned32
6002	VAR	RW	Total Measuring Range (TMR)	M	unsigned32
6003	VAR	RW	Preset value	M	unsigned32
6004	VAR	RO	Position value	M	unsigned32
6030	ARRAY	RO	Speed Value	O	Unsigned16
6031	Array	RW	Speed Calculation Parameter	M	Unsigned16
6040	ARRAY	RO	Acceleration Value	O	Signed16
6200	VAR	RW	Cyclic Timer	M	unsigned16
6400	ARRAY	RO	Working Area state	O	Unsigned 8
6401	ARRAY	RW	Working Area Low Limit	O	Unsigned32
6402	ARRAY	RW	Working Area High Limit	O	Unsigned32
6500	VAR	RO	Operating Status	M	unsigned16
6501	VAR	RO	Measuring Step (Singleturn)	M	unsigned32
6502	VAR	RO	Number of revolutions	M	unsigned16
6503	VAR	RO	Alarms	M	unsigned16
6504	VAR	RO	Supported alarms	M	unsigned16
6505	VAR	RO	Warnings	M	unsigned16
6506	VAR	RO	Supported warnings	M	unsigned16
6507	VAR	RO	Profile and SW version	M	unsigned32
6508	VAR	RO	Operating time	M	unsigned32
6509	VAR	RO	Offset value (calculated)	M	signed32
650A	VAR	RO	Module Identification	M	signed32
650B	VAR	RO	Serial Number	M	unsigned32
650D	VAR	RO	Absolute Accuracy	M	Unsigned8
650E	VAR	RO	Device Capability	M	Unsigned8

VAR = Variable
ARRAY = Array of variables
RW = Read/Write RO = Read only const = Constant Name = Name of the Object M/O = Mandatory or Optional.

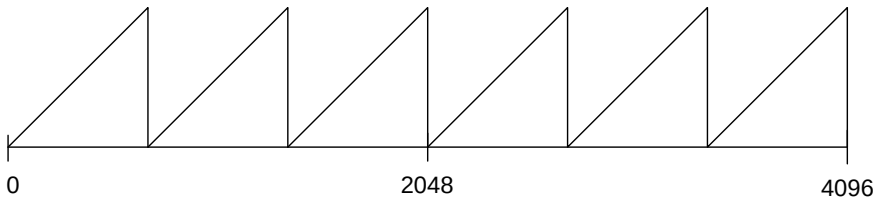
Range end problems with encoders with limited multiturn value

An error appears at the end of the physical resolution of an encoder, when **scaling is enabled**, if the division of the **physical boundaries (GP_U)** by the programmable **total resolution (TMR)** is not an **integer**.

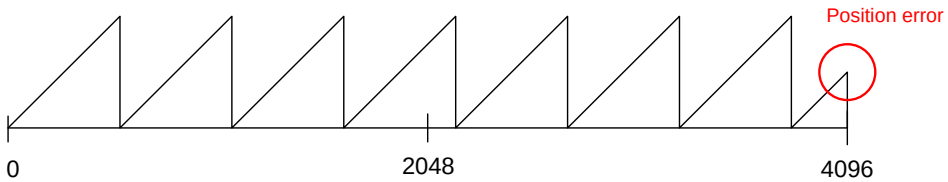
k = GP_U / TMR k <> integer

At the end of the multi-turn range during clockwise rotation, the encoder outputs **position zero** again. The same error appears **immediately**, if the encoder is set back **to zero** with a preset and the maximum multiturn value is approached after.

Example multiturn value 4096



Scaling with binary divider



Scaling with decimal divider

Example with binary divider: Entry Object 6001h MUR = 16384

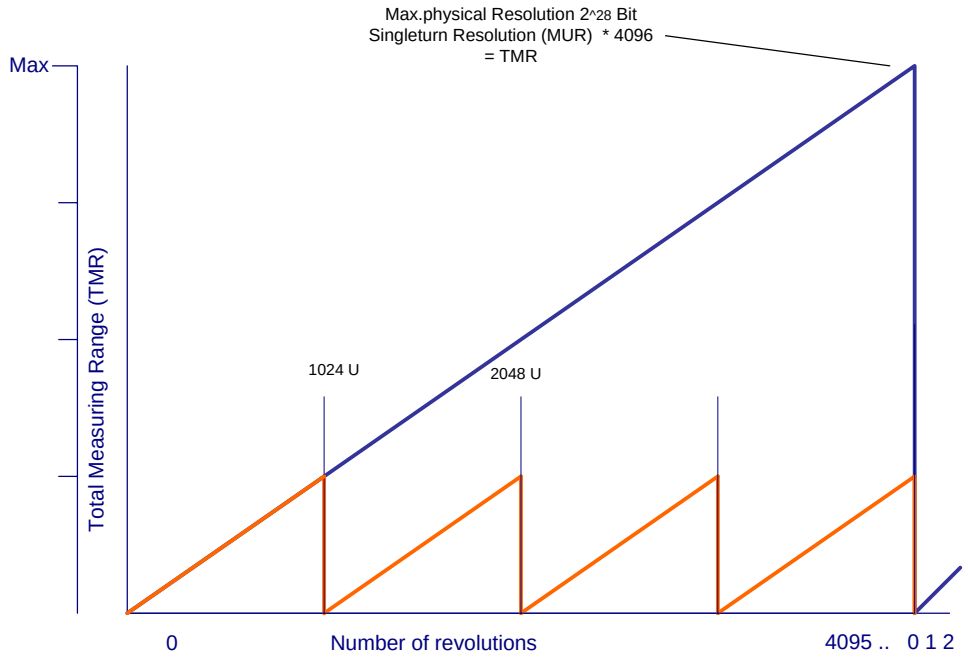
Total position $_{Scaled}$ = ((GP_U / STA_U) * MUR)

Total position $_{Scaled}$ = ((GP_U / STA_U) * 16384)
Total position $_{Scaled}$ = (4096 * 16384)
TMR = 67.108.864

k = GP_U/ TMR k= integer
k = 2^{28} /67.108.864 = 4 -> no position error with multiturn carry-over

*Abbreviations at the end of the document

Example with binary divider:



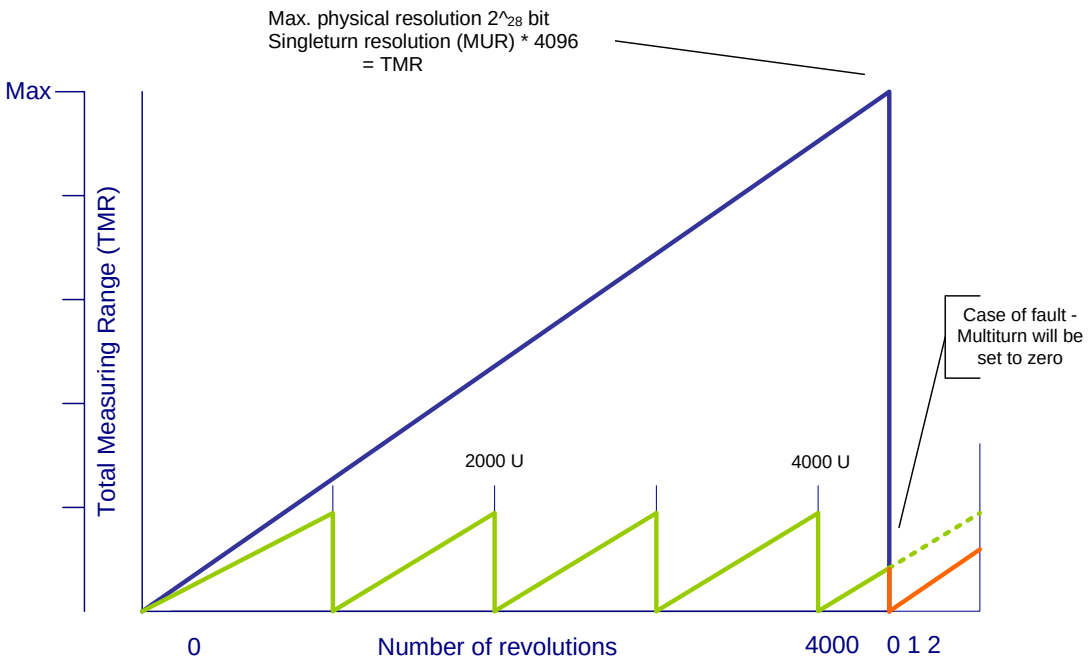
Example with decimal divider:

Entry Object 6001h MUR= 65000
Entry Object 6002h TMR= 65.000.000

Calculated number of MT rotations = 1000

k = GP_U/ TMR k= integer

Error $k = 2^{28}/65.000.000 = 4,1297$



21 Solution with USF

In order to have a "**safe**" area with sufficient reserve both in the positive and in the negative direction of rotation of the absolute position area of an encoder, the ***zero point*** of the sensor must be offset. First, the physical position of the sensor is separated from the position that is sent via CANbus. This creates two layers called *physical* and *virtual* layer.

Physical layer
On this level, the absolute raw position data of the mounted sensor is read and used internally as the basis for the calculation of the virtual position.

Virtual layer
The *virtual* level contains the ***corrected*** position data of the encoder. This position data is forwarded with the current MUR/TMR settings ***via the bus***. The correction positions the ***zero point*** of the encoder in the ***centre*** of the actual measuring range, which avoids the critical areas at the ends. This allows correcting the position error in case of negative direction of rotation of the shaft. After the ***zero point offset***, the boundaries of the critical areas must be defined, so as to allow detecting whether the position is already close to a faulty position overflow.

In order to achieve a good compromise between safety and flexibility, the boundary for the positive critical area has been set to **1/4** of the whole ***unscaled multiturn resolution (MTA_U)***.
A step width must be defined in order to avoid simultaneously a fault in case of a negative direction of rotation. This step width represents the distance at which the position ramps of the scaled encoder position repeat themselves. This step width must be ***smaller*** than the positive boundary.

It follows from the calculated step width that the negative boundary must be closer to the zero point than the positive boundary. The boundary must be set to **1/8 MTA_U** in order to make sure that the boundary is detected before reaching the end of the negative critical area.

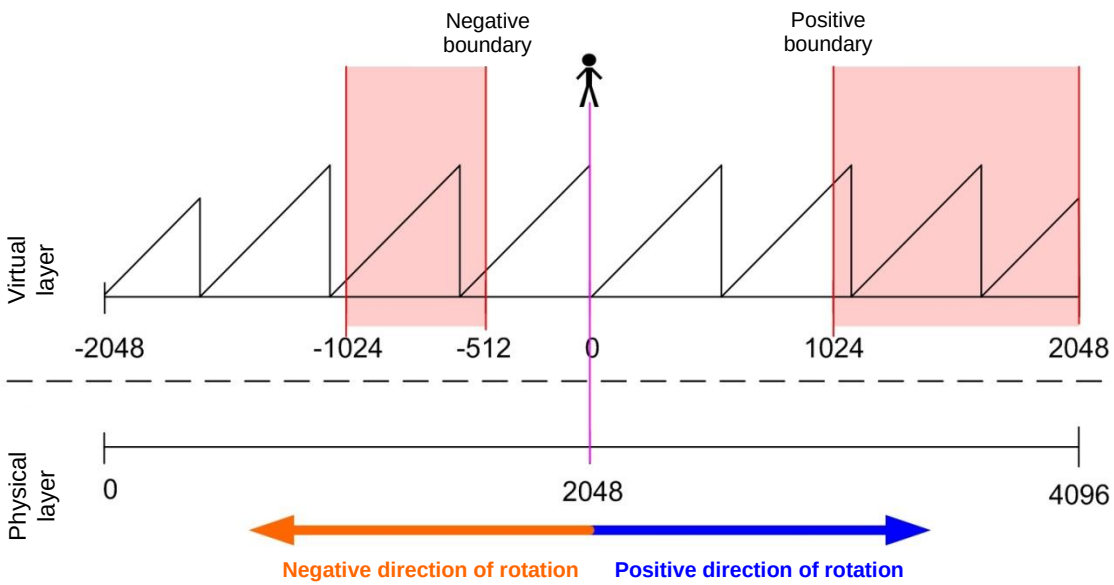


Fig. 2 - User position, directions of rotation and areas

If, during the operation of the system, one of these critical areas is reached, a correction must take place, which will shift the current position on the virtual layer so that the **physical position** is located again outside of the critical area. This will create a virtual position on the user layer, and the encoder will always be in the safe area.

22 Basic Activation of USF

Proceed as follows in order to activate the **USF** in the encoder:

1. Entry of the gear factor

Object 6001h: Measuring units per revolution (MUR) (resolution)
Object 6002h: Total measuring range (TMR)

If the ratio TMR/MUR < MTA/4 is not observed, this may also result in a miscalculation and the **functionality** of the USF cannot be guaranteed.
False entry and range excess are rejected in both objects 6001h and 6002h.

2. Activation of the function with bit 12 Object 6000h Operating Parameters

12	**Universal Scaling Function**	**Disabled**	**Enabled**	**o**	**o**

Example: Value in 6000h 0x1004 Scaling active, USF active

3. New encoder referencing wit the Preset function

Object 6003h: Preset value entry

Value range: 1....max. physical resolution (268435456) 28 bits
Default setting: 0

When entering the preset value, the system checks automatically whether the point is located within the activated scaling of the total measuring range, otherwise the entry is rejected.

4. USF is active and can be used

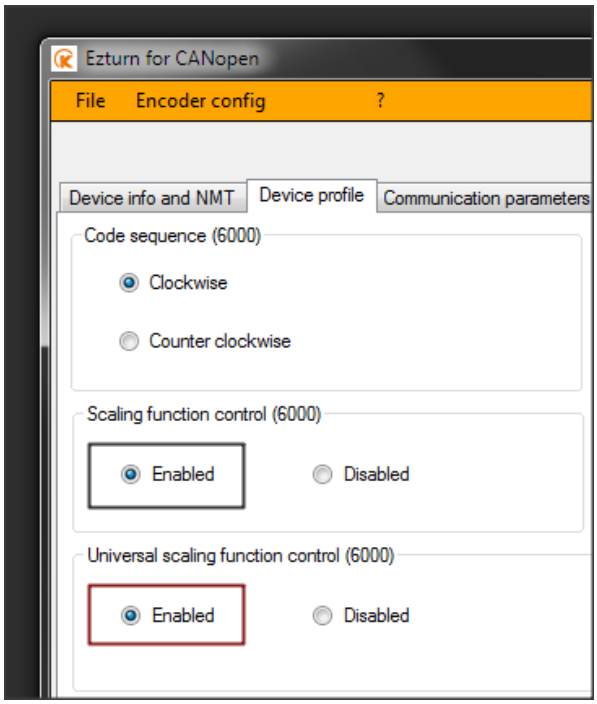
23 Error Messages during USF operation

Emergency Objects arise in case of error situations within a CAN network and are triggered depending on the event and transmitted over the bus with a **high priority**.

Important:
An Emergency Object is only triggered **once per "Event"**. No new object is generated as long as the error is active. Once the error is eliminated, a new Emergency Object with the content 0 (Error Reset or No Error) is generated and transmitted over the bus.

If an error is detected within the USF calculation, the encoder generates an error message with the code ***6200h User Software*** and the ***yellow DIAG LED*** flashes.

24 Activating USF with the Ezturn software



In the Ezturn software, the USF functionality can be selected with a radio button.
The system checks whether the boundary conditions are respected (it may happen that the TMR and MUR values have been set in such a way, with de-activated **USF**, that the TMR/MUR ratio does not lie below the prescribed limit of MTA_U/4). An attempt to start USF in these conditions will lead to the detection of this wrong ratio, an error message will then be generated and the option will not be activated.

If all values are in the allowed range, the function is activated, otherwise a warning message is generated.

25 Error Messages during USF operation

1. **The *yellow/red LED* flashes cyclically after Power On.**

   The saved offset value cannot be verified any more, and the device requires a ***Preset*** operation.
   An ***Emergency Message 6200 User Software*** is triggered.
   Preset is executed - ***yellow/red LED*** **flashes cyclically.**

2. **The *green and yellow LEDs* blink alternately after Power On or during operation.**

   The data storage of the ***offset value*** cannot be guaranteed. The FRAM did not pass the memory test.
   An ***Emergency Message 5100 Hardware error*** is triggered. The device should be sent for servicing.

26 Encoder Profile DS 406

Object 6000h Operating Parameters

Bit 0: Code sequence:	0 = increasing for clockwise rotation (cw)
	1 = increasing for counter-clockwise rotation (cw)
Bit 2: Scaling function:	0 = disable, 1 = enable; Standard: Bit = 0 (see Object 6001,6002)
Bit12: USF:	0 = turned off, 1 = activated
Bit13: Speed Format:	0 = revolutions/min, 1 = units/second
Bit14: Startup Mode:	0 = Boot-up after Pre-operational, 1 = Boot-up after operational
Bit15: Event Mode:	0 = position output acc. to TPDO 1800h, 1 = every change in position is output

Defaults: 0x00 (cw,scaling off,USF off,rpm,normal bootup)

Bit	Function	Bit = 0	Bit = 1	C1	C2
0	**Code sequence**	**CW**	**CCW**	**m***	**m***
1	**Commissioning Diagnostic Control**	Disabled	Enabled	o	o
2	**Activate scaling**	Disabled	Enabled	o	m
3	**Measuring direction**	Forward	Reward	o**	o**
4..11	Reserved for further use				
12	**Universal Scaling Function**	Disabled	Enabled	o	o
13	**Speed Format**	RPM	Units/sec	o	o
14	**Start-up automatic in OP-Mode**	Disabled	Enabled	o	o
15	**Event Mode Position**	Disabled	Enabled	o	o

*m = function has to be supported o = optional

Object 6001h: Measuring units per revolution (MUR) (resolution)

This parameter adjusts the desired resolution per revolution. The encoder internally calculates the corresponding scaling factor. The calculated scaling factor MURF (which multiplies the physical position value), is calculated by using the following formula:

MURF = Measuring units per revolution (6001h) / phys. resolution singleturn (6501h)

Byte 0	Byte 1	Byte 2	Byte 3
$2^7 \dots 2^0$	$2^{15} \dots 2^8$	$2^{23} \dots 2^{16}$	$2^{31} \dots 2^{24}$

value range: 1....max. physical resolution (65536) 16 bits

It may happen that the TMR and MUR values have been set in such a way, with de-activated **USF**, that the TMR/MUR ratio does not lie below the prescribed limit of MTA_U/4. An attempt to start USF in these conditions will lead to the detection of this wrong ratio, an error message will then be generated and the option will not be activated.

Object 6002h: Total measuring range (TMR)

This parameter adjusts the total measurement range of **singleturn and multiturn**. The maximum physical resolution affected with a factor. The factor is always < 1. After the scaled overall position of measurement units, the encoder resets **to zero (with limitations)***.

Data content:

Byte 0	Byte 1	Byte 2	Byte 3
$2^7 \dots 2^0$	$2^{15} \dots 2^8$	$2^{23} \dots 2^{16}$	$2^{31} \dots 2^{24}$

Value range: 1....max. physical resolution 32 bits

If **TMR/MUR is altered**, the ratio between TMR and MUR is also checked. An attempt to set a value for TMR that would result in an invalid ratio will trigger an error message and the new value will be dismissed. The old value will remain in the encoder.

An attempt to activate **USF** without active scaling will be detected and an error message will be output. The USF Option will not be activated.

Object 6003h: Preset value

The position value of the encoder is set to the entered preset value.
This will allow e. g. to align the zero position of the encoder with the machine zero point.

Data content:

Byte 0	Byte 1	Byte 2	Byte 3
$2^7 \dots 2^0$	$2^{15} \dots 2^8$	$2^{23} \dots 2^{16}$	$2^{31} \dots 2^{24}$

Value range: 1....max. physical resolution 32 bits
Default setting: 0

When entering the preset value, the system checks automatically assessed whether the point lies within the activated scaling or total measurement range. If this is not the case, the entry is rejected.

Object 6004h: Position value

The encoder outputs the current position value (possibly calculated with a scaling factor).

Data content:

Byte 0	Byte 1	Byte 2	Byte 3
$2^7 \dots 2^0$	$2^{15} \dots 2^8$	$2^{23} \dots 2^{16}$	$2^{31} \dots 2^{24}$

Value range: 1....max. physical resolution 32 bits
Default setting: current position

Current position output = ((GP_U / STA_U) * MUR) % TMR (Modulo Division)

Object 6030h: Speed Value

The encoder outputs the current calculated speed (possibly with a scaling factor) as a 16-bit value. The speed depends on the settings of **Object 6031h**. These values affect the calculation and the result.

Data content:

Byte 0	Byte 1
$2^7 \dots 2^0$	$2^{15} \dots 2^8$

Value range: 0....+/- maximum speed 15000 RPM (signed value)

For values larger than 12,000 RPM for multiturn, a warning message is generated and the warning bit "Speed exceeded bit 0" is set in Object Warnings 6505h.

For values larger than 4000 RPM for singleturn, a warning message is generated and the warning bit "Speed exceeded bit 0" is set in Object Warnings 6505h. In addition, the output value is limited to 4,000 RPM.

The parameters also affecting this object are mentioned in Object 6031h.

Object 6040h: Acceleration Value

The encoder outputs the current calculated acceleration (signed) as a 16-bit value. The acceleration is calculated from the speed changes and thus also depends indirectly on the **settings of Object 6031h**. These values affect the calculation and the result.

Data content:

Byte 0	Byte 1
$2^7 \dots 2^0$	$2^{15} \dots 2^8$

Value range: 0.... +/- maximum acceleration

Negative values mean a negative acceleration (the speed decreases)

An average acceleration **a** is the change in time of speed **v** and can thus be described formally from the derivative of speed versus time **t**, this calculation results in an **average acceleration** from the difference between speeds Δv at 2 different moments Δt (t2-t1).

a = Δv /Δt or a = v2- v1 / t2-t1

Object 6200h: Cycle timer

Defines the cycle time with which the current position is output using PDO 1 (see Object 1800h). The timer-controlled output becomes active as soon as a cycle time >0 is entered.

This Object is only present for reasons of compatibility with earlier profile versions. Instead of this Object, please use the Event Timer Subindex (05h) in the current Transmit PDO.

Data content:

Byte 0	Byte 1
$2^7 \dots 2^0$	$2^{15} \dots 2^8$

Value range: 0 ... FFFFh (65535) gives a cycle time in milliseconds
Standard value = 0h

Object 6500h: Display Operating Status

This Object displays the status of the programmed settings of Object 6000h.

Data content:

Byte 0	Byte 1
$2^7 \dots 2^0$	$2^{15} \dots 2^8$

Data content: see Object 6000h

Object 6502h: Number of Multiturn revolutions

This Object shows the current number of revolutions. The value depends on the encoder type and any value between 4096 (12 bits) and 65535 (16 bits) could occur. This value only affects the number of revolutions. It does not affect the resolution.

Data content:

Byte 0	Byte 1
00	10h

Value range: 1 …65535
Default setting corresponds to 65536 for MT **1 for ST**

Object 6503h: Alarms

In addition to the errors that are signalled via emergency messages,
Object 6503h provides for further error messages. The corresponding error bit is set to 1 for as long as the error condition applies.

Data content:

Byte 0	Byte 1
$2^7 \dots 2^0$	$2^{15} \dots 2^8$

Bit no.	Description	Value = 0	Value = 1
Bit 0	**Position error**	Position value valid	Position error
Bit 1	**Hardware check**	No error	Error
Bit 2..15	**Not used**		

In both cases, if an alarm occurs, an emergency message **(ID=80h+node number)** is sent together with the error code **1000h (Generic error)**.

Object 6504h: Supported Alarms

This Object is used to display which alarm messages are supported by the encoder (see Object 6503h).

Data content:

Byte 0	Byte 1
$2^7 \dots 2^0$	$2^{15} \dots 2^8$

Range of values: see Object 6503h
The alarm message is supported when the bit is set to 1.
Example:
Bit 0 = 1 Position error display is supported

Object 6505h: Warnings

Warning messages show that tolerances of internal encoder parameters have been exceeded. With a warning message – unlike with an alarm message or emergency message – the measured value can still be valid. The corresponding error bit is set to 1 for as long as the tolerance is exceeded or the warning applies.

Data content:

Byte 0	Byte 1
$2^7 \dots 2^0$	$2^{15} \dots 2^8$

Bit no.	Description	Value = 0	Value = 1
Bit 0	**Overspeed**	none	exceeded Limits: 4000 rpm ST Limits: 12000 rpm MT
Bit 1	**not used**		
Bit 2	**Watchdog Status**	System OK	Reset carried out
Bit 3	**Operating time**	Below < 100000h	> 100000h
Bit 4	**Battery warning**	Battery > 2,4 VDC	**Battery < 2,4 VDC***
Bit 15	**Temperature error**	Temperature OK	**Overtemperature**

When Bit 0 is active, an emergency message (ID=80h+node number) is sent together with **Error code 4200h** (Device specific).

*When Bit 4,15 is active, an emergency message (ID=80h+node number) is sent together with **Error code 5200h** (Device Hardware).

Object 6506h: Supported Warnings

This Object is used to display which warning messages are supported by the encoder (see Object 6505h).

Data content:

Byte 0	Byte 1
$2^7 \dots 2^0$	$2^{15} \dots 2^8$

Range of values: see Object 6505h
The warning is supported when the bit is set to 1.

Object 6400h: Working Area State Register 2 values

This Object contains the current state of the encoder position with respect to the programmed limits. The flags are either set or reset depending on the position of both limit values. The comparison with both limit values takes place in "real time" and can be used for real-time positioning or for limit switching.

Name	Bit	Value	Definition
out of range	0	0_b	Position between minimum and maximum value (refers to module identification, object $650A_h$)
		1_b	Position (refers to minimum and maximum value in module identification, object $650A_h$) is reached or exceeded
range overflow	1	0_b	No range overflow
		1_b	Position is higher than the position value set in work area high limit (object 6402_h)
range underflow	2	0_b	No range underflow
		1_b	Position is lower than the position value set in work area low limit (object 6401_h)

Range of values 8 bits Data content see bits 0...7

The Bitposition for the second limit value (6401h,6402h) is starting at Bit 3,4, and 5.

Object 6401h: Working Area Low Limit 2 values
Object 6402h: Working Area High Limit 2 values

These two parameters configure the working area. The state inside and outside this area can be signalled by means of Flag bytes **(Object 6400h Working Area State)**. These area markers can also be used as software limit switches.

Data content:

Byte 0	Byte 1	Byte 2	Byte 3
$2^7 \dots 2^0$	$2^{15} \dots 2^8$	$2^{23} \dots 2^{16}$	$2^{31} \dots 2^{24}$

Value range: 1....max. physical resolution ($2^{\wedge 32}$) 32 bits
Default setting: 33554432 (25 bits) Working Area High Limit for MT 65565 for ST
0 Working Area Low Limit

Object 2100h: Baud rate

This Object is used to change the baud rate via software. The default setting is FFh, which means that the hardware setting for the baud rate has priority. If the value is set between 1..9 and the parameter saved with **Object 2105h Save All Bus Parameters**, then on the next Power ON or with a reset node, the device will boot up with the modified baud rate.

Byte 0
$2^7 \dots 2^0$

Data content:

Value range **1 ...8 (see Baud rate table)**
Default setting: 0x05h 250 Kbit/s

The acceptance of a new node number only becomes effective when the encoder is rebooted (Reset/Power-on) or by means of an **NMT-Reset Node** command. All other settings within the Object table remain unchanged.

Object 2101h: Node address

This Object is used to change the node address via software. The default setting is 0xFFh, which means that the hardware setting for the node address has priority. If the value is set between 1..127 and the parameter saved with **Object 2105h Save All Bus Parameters**, then on the next Power ON or with a reset node, the device will boot up with the modified node address.

Data content:

Byte 0
$2^7 \dots 2^0$

Value range 1 ...127 or 1..7Fh
Default setting: 0x3Fh Address 63

Node number 0 is reserved and may not be used by any node.
The resulting node numbers lie in the range **1...7Fh** hexadecimal or (1...127)

The acceptance of a new node number only becomes effective when the encoder is rebooted (Reset/Power-on) or by means of an **NMT-Reset Node** command. All other settings within the Object table remain unchanged.

Object 2102h: CAN bus termination OFF/ON

This Object is used to set the bus termination via software. The default setting is 1.

Data content:

Byte 0
$2^7 \dots 2^0$

Value range 0..1
Default setting: 1 ***for devices with cable outlet and CAN connection = 1**

Object 2103h: Firmware Flash version

This object is used to display the current firmware version as a 16-bit hexadecimal value.
This value serves to verify that the device is to the latest revision.

Data content:

Byte 0	Byte 1
$2^7 \dots 2^0$	$2^{15} \dots 2^8$

Value range up to FFFFh
Example: **4FA6h current firmware**

Object 2105h: Save All Bus Parameters

This parameter stores the desired bus parameters (Object 2100h, 2101h, 2102h) permanently in the Flash memory. This object is an additional protection against unintentional baud rate or node address modifications.
Only the targeted storage using parameter **"save" (hexadecimal 0x65766173)** saves the bus parameters **baud rate, node address and termination** permanently.

Data content:

Byte 0	Byte 1	Byte 2	Byte 3
$2^7 \dots 2^0$	$2^{15} \dots 2^8$	$2^{23} \dots 2^{16}$	$2^{31} \dots 2^{24}$

Value range: **„save“ in hexadecimal 0x65766173**

Object 2110h: Sensor Configuration Data

This Object is used to get information about the current configuration of the sensor.
The array is displayed as a byte-hexadecimal value.

Byte 0	Byte 1	Byte 2
$2^7 \dots 2^0$	$2^{15} \dots 2^8$	$2^{23} \dots 2^{16}$

Value range up to FF, FFh…….
Can only be used for service purposes.

Object 2120,4h: Actual temperature Position-Sensor *

This Object is used to display the current temperature inside the sensor as a 16-bit hexadecimal value. This value allows determining the momentary device temperature.

Byte 0
$2^7 \dots 2^0$

Value range up to 00…FFh
Example: **0x59 means about 25°C**

The following temperature values can be used as reference values:

-20°C	**means**	**0x2Ch**
0°C	**means**	**0x40h**
100°C	**means**	**0xA4h**

Example: Read value 0x71h **from Object 2120,4h**
0x71h – 0x40h = 0x31h corresponds to 49°C decimal

*** The temperature value can be mapped as an 8-bit value in the process data, where it will be updated all 60 seconds. Accuracy is ± 6°C, the measurement takes place within the encoder electronics.**

Object 2120,2h: Actual temperature lower limit Position-Sensor
Object 2120,3h: Actual temperature upper limit Position-Sensor

This object is used to display the upper/lower temperature limit as a 8-bit hexadecimal value. **This value is used to determine the triggering threshold for the Emergency message**.

Byte 0
$2^7 \dots 2^0$

Value range up to 00…FFh
Example: **0x20 means about -32°C**

The following temperature values can be used as reference values:

-20°C	**means**	**0x2Ch**
0°C	**means**	**0x40h**
100°C	**means**	**0xA4h**

If this temperature threshold is exceeded in either direction, an **Emergency message** is output (see below) and a corresponding reaction is triggered.

value range: 0x20h .. 0xACh
Default setting: 0xA2h Temperature High Limit
0x20h Temperature Low Limit

Object 2125h: Battery Voltage (Multiturn only)

This object is used to display the current battery voltage as a 16-bit hexadecimal value.
This value is used to check the integrated battery and shows directly the exact voltage in VDC.

Data content:

Byte 0	Byte 1
$2^7 \dots 2^0$	$2^{15} \dots 2^8$

Value range up to 0 ..500
Example: **361 current battery voltage is 3.61 VDC**

Limit: **corresponds to 2.4 V**
Error: **0x3FF no battery or battery discharged**

The current battery voltage is 3.62 V.

Object 2140h: Customer Memory (16 Bytes)

These 4 parameters constitute a memory area for the user. **4 data words with a maximum of 4 bytes can be stored**. This area is not checked for content, which means that any format may be stored.

Byte 0	Byte 1	Byte 2	Byte 3
$2^7 \dots 2^0$	$2^{15} \dots 2^8$	$2^{23} \dots 2^{16}$	$2^{31} \dots 2^{24}$

Value range: Numeric, alphanumeric
Default setting:0

Object 2160h: RAW position value data

In addition, the position data can be output as raw data. The data is transmitted as 32-bit value, logically correct and complemented, *a CRC for the position data can also be added to the mapping*.* Object **1803h** may be used to that purpose.

The encoder outputs the current **original position value** directly from the sensor.

Data content:

Byte 0	Byte 1	Byte 2	Byte 3
$2^7 \dots 2^0$	$2^{15} \dots 2^8$	$2^{23} \dots 2^{16}$	$2^{31} \dots 2^{24}$

Value range: 1....maximum physical resolution ($2^{\wedge 32}$) 32 bits for MT ($2^{\wedge 16}$) ST

Object 2161h: Complemented RAW position value data

The encoder outputs the **complemented** current **original position value**.

Data content:

Byte 0	Byte 1	Byte 2	Byte 3
$2^7 \dots 2^0$	$2^{15} \dots 2^8$	$2^{23} \dots 2^{16}$	$2^{31} \dots 2^{24}$

Value range: 1....maximum physical resolution ($2^{\wedge 32}$) 32 bits for MT ($2^{\wedge 16}$) ST

Example for the transmission of RAW and complemented position data:

Message	Byte 0	Byte 1	Byte 2	Byte 3	
2160h	15	3A	7F	01	Raw position data
2161h	EA	C5	80	FE	Complemented raw data
XOR	FF	FF	FF	FF	Result of the check

When an "exclusive OR" function is applied to both position data values, the result must always be a logical "1".
If transmission errors occur for any reason, this can be detected immediately by obtaining a logical "0" after this check.

Object 2162h: Raw position data CRC16

Current Object 2160h Position raw data generates a standard CRC16.
CRC-CCITT (CRC-16) $x^{16} + x^{12} + x^5 + 1$ (Polynomial 0x1021)

The implementation performs a polynomial division when the start value used is *0000...* This corresponds to a polynomial division if the first *n* bits of the data flow are complemented. A start value different from *0000...* should be preferred, since otherwise missing bits within leading zeros would not be detected in the data flow (just like in a usual division, the leading zeros are not taken into consideration for a polynomial division).

Start value (seed value) used here = 0xFFFF
The polynomial is displayed as a 16-bit hexadecimal value.
This value serves to verify the current raw position date of the device.

Data content:

Byte 0	Byte 1
$2^7 \dots 2^0$	$2^{15} \dots 2^8$

Value range up to FFFFh
Example: **4FA6h current CRC16 for the position data**

Object 1029h Error Behavior

If a serious error is detected, the device should automatically switch to **Pre-Operational** mode. The settings in this Object can be used to determine how the device is to behave when an error arises. The following error classes are covered:

1029h,Subindex 1 Communication Errors

- Bus Off state of the CAN interface
- Life guarding event has occurred
- Heartbeat monitoring has failed

1029h,Subindex 2 Device Profile Specific

- Sensor error and Controller error
- Temperature error

1029h,Subindex 3 Manufacturer Specific

- internal error

The value of the Object classes is put together as follows:

Byte 0
$2^7 \dots 2^0$

Value range 8 bits

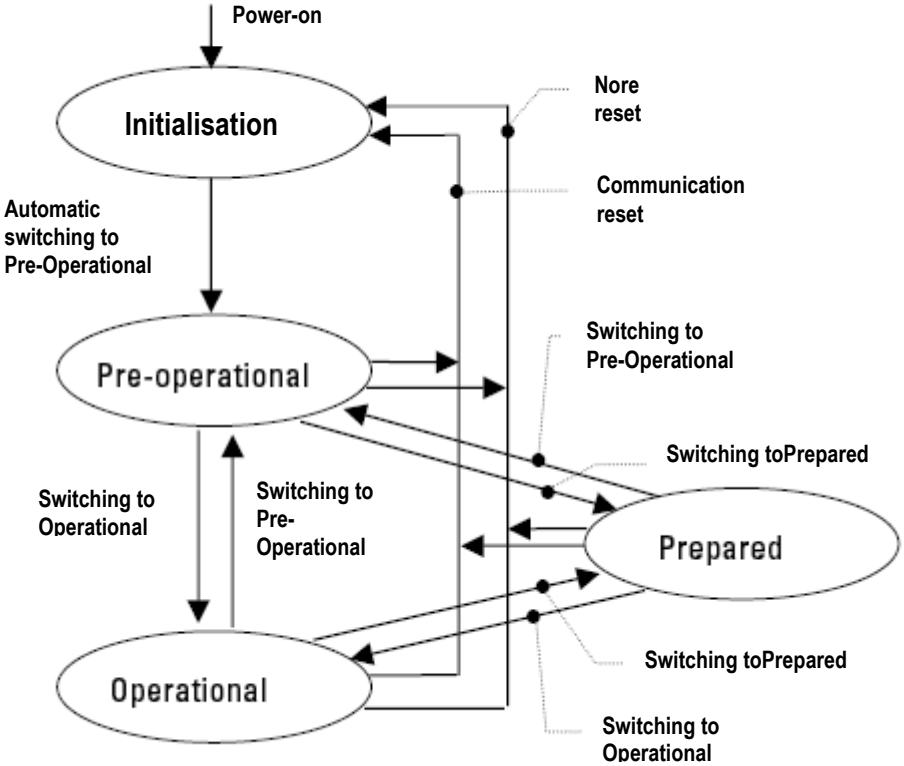
- **0 Pre-Operational Mode (only if Operational Mode was active before)**
- **1 no mode change**
- **2 Stopped mode**
- **3 .. 127 reserved**

Objects not mentioned

All Objects not mentioned here serve as additional information and can be found in **Encoder profile DS 406 V3.2**.

27 Network Management

The encoder supports the simplified Network Management as defined in the profile for "minimum capability devices" (minimum boot up).
The following function state diagram acc. to DS 301 shows the various node states and the corresponding network commands (controlled by the Network Master via NMT services):



Initialisation: this is the initial state after the power supply is applied, following a device Reset or Power ON. The node automatically enters the Pre-operational state once it has run through the Reset and Initialization routines. The LEDs display the momentary status.

Pre-operational: The CAN node can now be addressed via SDO messages or with NMT commands under the standard identifier. Then follows the programming of the encoder or communication parameters.

Operational: The node is active Process values are transmitted through the PDOs. All NMT commands can be evaluated.

Prepared or **Stopped:** In this state the node is no longer active, which means that neither SDO nor PDO communications are possible. The node can be set to either the Operational or
Pre-operational state by means of NMT commands.

28 NMT Commands

All NMT commands are transferred as an unconfirmed NMT Object. Because of the broadcast (network-wide) communication model, the NMT commands are recognized by each station.
A NMT Object is structured as follows:

COB ID = 0

Byte 0	Byte 1
$2^7 \dots 2^0$	$2^{15} \dots 2^8$

Byte 0 = Command byte
Byte 1 = Node number

The COB ID of the NMT Object is always 0
The node is addressed via the node numbers. With node number 0 all nodes are addressed.

Kommandobyte (hex)	Beschreibung
01h	Start_Remote_Node: Wechsel zu Operational
02h	Stop_Remote_Node: Wechsel zu Prepared
80h	Enter_Pre-Operational_State: Wechsel zu Pre-operational
81h	Reset_Node: Reset Knoten[1]
82h	Reset_Communication: Reset Kommunikation[2]

[1] All the parameters in the whole Object Dictionary will have their values set to Power-On values.
[2] Only the parameters in the section Communication Profile of the Object Dictionary will have their values set to Power-On values.

29 LED Displays during operation

green LED = BUS Status

red LED = ERR Display

Display	LED	Meaning	Cause of error	Addition
Bus off		No connection to master [2]	Data line interrupted Wrong baud rate Swapped data line	Watch out for combination with ERR LED If ERR-LED is also turned off, check power supply [3]
Bus Flashing about 250 ms		Connection to master Pre-operational status		SDO communication
Bus Flashing about 1 sec.		Connection to master Stopped status		SDO communication not possible Only NMT commands
Bus on		Connection to master Operational status		PDO transfer is active
ERR off		Device runs error-free		Watch out for combination with BUS LED
ERR Flashing		Connection to master interrupted	Combination with bus-status	BUS LED flashes green or is on, depending on Object 1029h error behaviour
ERR on		BUS OFF status	Bus short circuit or wrong baud rate	

.
The individual LED messages can of course also appear in combination.

[2] Master can be PLC or 2nd communication partner
[3] Operating voltage

LED combinations during operation

Display	LED	Meaning	Cause of error	Addition
ERR Flashing		Red LED flashing Red LED flashes briefly, duration 3 sec.	Temperature overflow Sensor monitoring Single step error Sensor LED current monitoring	Device on CAN bus Connection to master OK + additional cause of error

Error display upon switching on

Display	LED	Meaning	Cause of error	Addition
ERR + BUS Flashing		Alternating quick flashing of green and red LED	Data connection to sensor is faulty Sensor is defective	Device needs to be sent to manufacturer for service
ERR + BUS Flashing		Alternating quick flashing of green and red LED (300 ms)	Watchdog error	Device needs to be sent to manufacturer for service
ERR Flashing		Connection to master interrupted		No CAN Bus available
Bus +Err flashing fast		LSS Layer Service active Global mode activated	Encoder waiting for configuration	LSS mode

30 Definitions

Explanation of symbols

This symbol indicates text sections that are to be specially taken into consideration in order to ensure the good operation of the device and exclude any risk.
This symbol indicates important information for the correct handling of the encoder. Non-compliance with this information might lead to failures of the encoder or its environment.

This symbol indicates a particular feature

Factory default setting of the parameters

31 Abbreviations used

CAL	CAN Application Layer. Application Layer (Layer 7) in the CAN communication model
CAN	Controller Area Network
CiA	CAN in Automation. International society of users and manufacturers of CAN products
CMS	CAN Message Specification. Service element of CAL
COB	Communication Object. Transportation unit within CAN network (CAN message). Data is sent within a COB via the network.
COB ID	COB identifier. Unique identification of a CAN message. The identifier defines the priority of the COB within the network.
DBT	Distributor. Service element of CAL, responsible for the dynamic assignment of identifiers.
DS	Draft Standard
DSP	Draft Standard Proposal
ID	Identifier, see COB ID
LMT	Layer Management. Service element of CAL, responsible for the configuration of the parameters within the single layers of the communication model.
LSB	Least Significant Bit/Byte
MSB	Most Significant Bit/Byte
NMT	Network Management. Service element of CAL, responsible for the initialisation, configuration and errors handling within the network.
MT	**Multiturn encoder**
OSI	Open Systems Interconnection. Layer model to describe areas of activity in a data-communication system.
PDO	Process Data Object. Object for process data exchange.
RTR	Remote Transmission Request
SDO	Service Data Object; communication object, through which the master can access the object dictionary of a node.
SYNC	Synchronisation telegram. Bus devices reply with their process value to the SYNC command
ST	Singleturn encoder

32 Decimal-Hexadecimal conversion table

In case of numbers, decimal values are indicated as figures without extension (e. g. 1408), binary values are followed by b (e. g.1101b) and hexadecimal values are followed by h (e. g. 680h) behind the figures.

Dez	**Hex**	Dez	**Hex**	Dez	**Hex**	Dez	**Hex**
0	**00**	32	**20**	64	**40**	96	**60**
1	**01**	33	**21**	65	**41**	97	**61**
2	**02**	34	**22**	66	**42**	98	**62**
3	**03**	35	**23**	67	**43**	99	**63**
4	**04**	36	**24**	68	**44**	100	**64**
5	**05**	37	**25**	69	**45**	101	**65**
6	**06**	38	**26**	70	**46**	102	**66**
7	**07**	39	**27**	71	**47**	103	**67**
8	**08**	40	**28**	72	**48**	104	**68**
9	**09**	41	**29**	73	**49**	105	**69**
10	**0A**	42	**2A**	74	**4A**	106	**6A**
11	**0B**	43	**2B**	75	**4B**	107	**6B**
12	**0C**	44	**2C**	76	**4C**	108	**6C**
13	**0D**	45	**2D**	77	**4D**	109	**6D**
14	**0E**	46	**2E**	78	**4E**	110	**6E**
15	**0F**	47	**2F**	79	**4F**	111	**6F**
16	**10**	48	**30**	80	**50**	112	**70**
17	**11**	49	**31**	81	**51**	113	**71**
18	**12**	50	**32**	82	**52**	114	**72**
19	**13**	51	**33**	83	**53**	115	**73**
20	**14**	52	**34**	84	**54**	116	**74**
21	**15**	53	**35**	85	**55**	117	**75**
23	**17**	55	**37**	87	**57**	119	**77**
24	**18**	56	**38**	88	**58**	120	**78**
25	**19**	57	**39**	89	**59**	121	**79**
26	**1A**	58	**3A**	90	**5A**	122	**7A**
27	**1B**	59	**3B**	91	**5B**	123	**7B**
28	**1C**	60	**3C**	92	**5C**	124	**7C**
29	**1D**	61	**3D**	93	**5D**	125	**7D**
30	**1E**	62	**3E**	94	**5E**	126	**7E**
31	**1F**	63	**3F**	95	**5F**	127	**7F**

33 Glossary

Baudrate
The baud rate is the transmission speed. It is related to the nominal bit timing. The maximum possible baud rate depends on many factors, which affect the propagation time of the signals on the bus. There is a significant link between the maximum baud rate, the bus length and the cable type. Various baud rates between 10 kbit/s and 1 Mbit/s are defined in CANopen.

CANopen
CANopen is a CAN-based protocol that has been developed originally for industrial control systems. The specifications include as well various device profiles as the framework for specific applications. CANopen networks are also used in off-road vehicles, marine electronics, medical appliances and trains. The very flexible application layer and the numerous optional features are ideal for customised solutions. There is furthermore a wide variety of configuration tools. The user can, on this base, define application-specific device profiles. You can find further information about CANopen in the Internet under **www.can-cia.org**.

EDS file
The EDS (Electronic Data Sheet) file is supplied by the manufacturer of a CANopen device. It has a standardised format for the description of devices. The EDS file includes information about:

- File description (name, version, creation date, among others)
- General device information (manufacturer name and code)
- Device name and type, version, LMT address
- Supported baud rates and Boot-up ability
- Description of the supported Objects and their attributes.

Node number
Within a CANopen network, every device is identified with its node number (node ID). The usable node numbers are in the range between 1-127 and can appear only once within a network.

Network management
In a distributed system, there are various tasks in relation with the configuration, initialisation and monitoring of the network participants. The service element »Network management (NMT)« defined in CANopen makes these functionalities available.

PDO
The process data objects (PDO) are the real transportation means for the transfer of process data (application objects). A PDO is sent by a Producer and can be received by one or several Consumers.

PDO mapping
The size of a PDO may reach 8 byte. It may be used to transport several application objects. The PDO mapping describes the definition of the arrangement of the application objects within the data field of the PDO.

SDO
The service data objects (SDO) are used for the confirmed transfer of data of any length between two network participants. Data transfer takes place in Client-Server mode.

Release- Information

Date	Version	Modifications	Approved
12.01.2013	1.0	Initial Version	Hk
26.02.2013	1.1	Modified USF descritption	Hk
09.04.2013	1.2	Working state Object 6400h adapted to new Encoder Profile 3.2.16	Hk
06.05.2013	1.3	Release Information Cover page modification Order number Firmware Version	Hk
3.4.2014	1.4	Termination default	Hk
17.8.2014	1.5	Default adjustments	Hk