R67028.0002 - Index 2

English

Manual

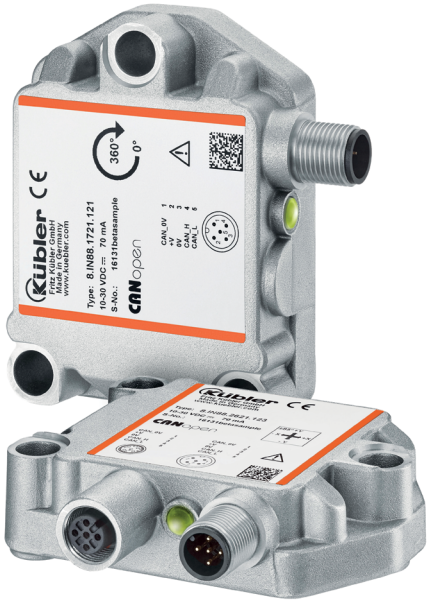
Inclinometer IN88

Inclinometer 1-dimensional

Order code: 8.IN88.1721.12X

Inclinometer 2-dimensional

Order code: 8.IN88.2621.12X



pulses for automation

Manual CANopen

IN88 Inclinometer 1-dimensional
IN88 Inclinometer 2-dimensional

Publisher	Kübler Gruppe, Fritz Kübler GmbH
Technical support	Phone +49 (0) 7720 3903-0 Fax +49 (0) 7720 21564 servicecenter@kuebler.com
Document no.	R67028.0002 - Index 2
Document name	Manual - CANopen IN88 inclinometer 1-dimensional, order code 8.IN88.1721.12X IN88 inclinometer 2-dimensional, order code 8.IN88.2621.12X
Language version	English (ENG) - German is the original version
Issue date	11/2016

Table of Contents

These operating instructions do not contain information about the installation.
You will find these in separate installation instructions.

1. Technical details and characteristics

Working temperature range
-40…+85°C

Supply voltage and current consumption
10…30 VDC
70 mA at 10 VDC
30 mA at 24 VDC
26 mA at 30 VDC

Hardware characteristics

2-dimensional sensor: Measuring range per axis	± 85°
1-dimensional sensor: Measuring range per axis	0°… 360°
Internal cycle	20 ms

CANopen Interface **Transceiver according to ISO 11898**
Function display and diagnostics by means of LEDs (red/green)

Supported standards and protocols
CiA Standard 301 Communication Profile 4.2.0
CiA Standard 305 Layer Setting Services 2.2
CiA Standard 410 Device Profile for Inclinometers 1.3.0

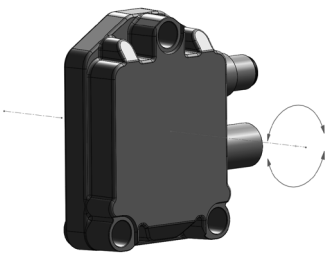
Supported standards and protocols

The CANopen inclinometers support the latest CANopen communication profile according to **DS 301 V4.2.0.** In addition, device-specific profiles such as the inclinometer profile **DS 410 V1.3.0** are available.
The additional integrated **DS 305** LSS services allow performing node number allocation and CAN bit rate configuration directly via the CAN bus. The LSS functionalities allow simple and fast detection and addressing of new unconfigured appliances.

Operating modes
The available operating modes are Polled Mode, Cyclic Mode, Sync Mode. Moreover, scaling, preset values and many other additional parameters can be programmed via the CANbus. When switching the appliance on, all parameters are loaded from a flash memory. These parameters have previously been stored in a zero-voltage secure manner. The output values, e.g. **measuring axes angle and temperature**, can be combined very variably as a **PDO** (PDO mapping).

Error status
A 2-color LED signals the operating and error status of the CAN bus.

Inclination direction
1-dimensional



2-dimensional

The CANopen communication profile DS 301 V4.2.0

CANopen offers a uniform user interface and therefore allows a simplified system structure with very different appliances. CANopen is optimized for fast data exchange in real-time systems and has various device profiles that have been standardized. The CAN in Automation (CiA) manufacturers and users assocation is in charge of establishing and standardizing the corresponding profiles.

CANopen offers
- comfortable 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 traffic
- simultaneous input or output of data

CANopen uses four communication objects (COB) with various characteristics
- Process Data Objects (PDO) for real-time data,
- Service Data Objects (SDO) for parameters and program transmission
- Network Management (NMT, Life-Guarding, Heartbeat)
- Predefined objects (for synchronization, emergency)

All device parameters are saved 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 structured in a communication profile section, a device profile-related section and a manufacturer-specific section.

The inclinometer device profile DS 410

This profile describes a manufacturer-independent and binding specification of the interface for inclinometers. The profile defines which CANopen functions are used and how they are to be used. This standard allows for an open and manufacturer-independent bus system.

The manufacturer profile is subdivided in four classes.
- **Inclinometer 1 measuring axis 15 bits measuring data + 1bit sign**
- **Inclinometer 2 measuring axes 15 bits measuring data + 1bit sign**

LSS services DS 305 V2.0

CiA DSP 305 CANopen Layer Setting Service and Protocol (LSS) were created to read and modify the following parameters via the network:
- Node address
- Baud rate
- LSS address

These abilities increase the "plug-and-play" compatibility of the device and the configuration possibilities have been considerably simplified. The LSS master is responsible for the configuration of these parameters for one or several slaves in the network.

Data transmission

With CANopen, data is transferred using two communication types (COB=Communication Object) with different characteristics:

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

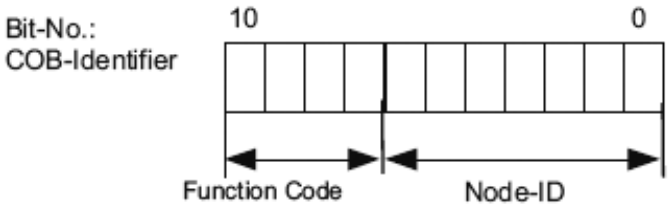
The Process Data Objects **(PDO)** are used for highly-dynamic exchange of real-time data (e.g. angular information, temperature) with a maximum length of 8 byte. This data is transferred with high priority (low COB identifier). PDOs are broadcast messages and make their real-time data available simultaneously to all desired recipients. PDO can be mapped, i.e. 4 bytes of angular information and 2 bytes of temperature can be combined in one 8-bit data word.

The Service Data Objects **(SDO)** are the communication channel for the transmission of device parameters (e.g. programming of the measurement resolution). As these parameters are transmitted acyclically (e.g. only once when starting the network up), the SDO objects have a low priority (high COB identifier).

Objects and function code in the Predefined Connection Set

For an easier identifier management, CANopen uses the "Predefined Master/Slave Connection Set". All identifiers are defined with standard values in the objects dictionary. These identifiers can however be modified on a customer-specific basis by SDO access.

The 11-bit identifier consists in a **4-bit function code** and a **7-bit node number**.



Broadcast (nework-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 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)	1111	897 (381h) - 1023 (3FFh)	1802h
PDO3 (rx)	1000	1025 (401h) - 1151 (47Fh)	1502h
PDO4 (tx)	1001	1153 (481h) - 1279 (4FFh)	1803h
PDO4 (rx)	1010	1281 (501h) - 1407 (57h)	1403h

SDO (tx)	1011	1409 (581h) - 1535 (56FFh)	1200h
SDO (rx)	1100	1537 (601h) - 1663 (67Fh)	1200h
NMT Error Control	1110	1793 (701h) - 1919 (77Fh)	1016h, 1017h

Restricted, reserved objects

COB	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

Process data transmission

The **2 PDO services** PDO1 (tx) … PDO2(tx) are available for process data transfer. A PDO transfer can be triggered by various events (see object dictionary Index 1800h):

- **Asynchronously** (event-driven) by an internal cyclic event timer or in case of a process value change of the sensor data
- **Synchronously** as an answer to a SYNC telegram; (a SYNC instruction prompts all CANopen nodes to save synchronously their values, and to send them subsequently one after the other on the bus according to the set priority)
- **Answer** to a RTR request **is not supported**

Standard setting for the mapping of transmit PDO1-2 → 2-dimensional
The PDO transmission type defines how the sending of the PDO is triggered:

Mapping	TPDO1 1800h		
Mapping object	1A00_01h	1A00_02h	1A00_03h
Entry	0x60100010	0x60200010	0x50000010
Process	Slope long axis	Slope lateral axis	Temperature
Object	3010h	6020h	5000h
Subindex	00	00	00
Data length	10h (16 bits)	10h (16 bits)	10h (16 bits)
	Asynchronous	**Asynchronous**	**Asynchronous**

Mapping	TPDO01 1801h		
Mapping object	1A01_01h	1A01_02h	1A01_03h
Entry	0x60100010	0x60200010	0x50000010
Process	Slope long axis	Slope lateral axis	Temperature
Object	3010h	6020h	5000h
Subindex	00	00	00
Data length	10h (16 bits)	10h (16 bits)	10h (16 bits)
	Synchronous	**Synchronous**	**Synchronous**

Transmit PDO1 (1800h) Position asynchronous
Default COB-ID is 180 + node number: Example 180h + 3Eh = 1BEh

Message	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
1BE	Slope long axis LSB	Slope long axis MSB	Slope lateral axis LSB	Slope lateral axis MSB	Temp. MSB	Temp. MSB

All values are INT16 signed
Transmit PDO2 (1801h) Position synchronous
Default COB-ID is 280 + node number: Example 280h + 3Eh = 2BEh

Message	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
2BE	Slope long axis LSB	Slope long axis MSB	Slope lateral axis LSB	Slope lateral axis MSB	Temp. MSB	Temp. MSB

Standard setting for the mapping of transmit PDO1-2 → 1-dimensional

Mapping	TPDO1 1800h	
Mapping object	1A00_01h	1A00_02h
Entry	0x60100010	0x50000010
Process	Slope long axis	Temperature
Object	6010h	5000h
Subindex	00	00
Data length	10h (16 bits)	10h (16 bits)
	Asynchronous	**Asynchronous**

Mapping	TPDO1 1801h	
Mapping object	1A01_01h	1A01_02h
Entry	0x60100010	0x50000010
Process	Slope long axis	Temperature
Object	6010h	5000h
Subindex	00	00
Data length	10h (16 bits)	10h (16 bits)
	Synchronous	**Synchronous**

Transmit PDO1 (1800h) Position asynchronous
Default COB-ID is 180 + node number: Example 180h + 3Eh = 1BEh

Message	Byte 0	Byte 1	Byte 2	Byte 3
1BE	Slope long axis LSB	Slope long axis MSB	Temp. LSB	Temp. MSB

All values are INT16 signed
Transmit PDO2 (1801h) Position synchronous
Default COB-ID is 280 + node number: Example 280h + 3Eh = 2BEh

Message	Byte 0	Byte 1	Byte 2	Byte 3
2BE	Slope long axis LSB	Slope long axis MSB	Temp. LSB	Temp. MSB

Service data transmission

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

SDO (tx) (sensor –> master): 580h (1408) + node number
SDO (rx) (master –> sensor): 600h (1536) + node number

The command byte describes the type of the SDO message:

Command (Expedited Protocol)	Type	Function
22h	SDO(rx), Initiate Download request	Send parameters to slave (Data length max. 4 byte)
23h	SDO(rx), Initiate Download request	Send parameters to slave (Data length max. 4 byte)
2Bh	SDO(rx), Initiate Download request	Send parameters to slave (Data length max. 2 byte)
2Fh	SDO(rx), Initiate Download request	Send parameters to slave (Data length max. 1 byte)
60h	SDO(tx), Initiate Download response	Confirmation of the acceptance to master
40h	SDO(rx), Initiate Upload request	Request parameters from slave
43h	SDO(tx), Initiate Upload response	Parameters to master (Data length = 4 byte UINT32)
4Bh	SDO(tx), Initiate Upload response	Parameters to master (Data length = 2 byte UINT16)
4Fh	SDO(tx), Initiate Upload response	Parameters to master (Data length = 1 byte UINT8)
80h	SDO(tx), Abort Domain transfer	Slave sends error code to master

Error messages

In case of an error, an error message (command 80h) replaces the normal confirmation (response).
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 read-only object, wrong data length, etc).

The error codes are described in the CANopen profile (DS 301) or in the device profile (DS 410).

Abort code	Description
0503 0000$_c$	Toggle bit not alternated
0504 0000$_h$	SDO protocol timed out
0504 0001$_h$	Client/server command specifier not valid or unknown
0504 0002$_h$	Invalid block size (block mode only)
0504 0003$_h$	Invalid sequence number (block mode only)
0504 0004$_h$	CRC error (block mode only)
0504 0005$_h$	Out of memory
0601 0000$_h$	Unsupported access to an object
0601 0001$_h$	Attempt to read a write only object
0601 0002$_h$	Attempt to write a read only object
0602 0000$_h$	Object does not exist in the object dictionary
0604 0041$_h$	Object cannot be mapped to the PDO
0604 0042$_h$	The number and length of the objects to be mapped would exceed PDO length
0604 0043$_h$	General parameter incompatibility reason
0604 0047$_h$	General internal incompatibility in the device
0606 0000$_h$	Access failed due to a hardware error
0607 0010$_h$	Data type does not match, length of service parameter does not match
0607 0012$_h$	Data type does not match, length of service parameter too high
0607 0013$_h$	Data type does not match, length of service parameter too low
0609 0011$_h$	Sub-index does not exist
0609 0030$_h$	Invalid value for parameter (download only)

Abort code	Description
0609 0031$_c$	Value of parameter written too high (download only)
0609 0032$_h$	Value of parameter written too low (download only)
0609 0036$_h$	Maximum value is less then minimum value
060A 0023$_h$	Resource not available: SDO connection
0800 0000$_h$	General error
0800 0020$_h$	Data cannot be transferred or stored to the application
0800 0021$_h$	Data cannot be transferred or storred to the application because of local control
0800 0022$_h$	Data cannot be transferred or stored to the application because of the present device state
0800 0023$_h$	Object dictionary dynamic generation fails or no object dictionary is present (e.g. object dictionary is generated from file and generation fails because of a file error)
0800 0024$_h$	No data available

Additional error codes of the DS 410 profile

Error code	Subindex	Error description
4200h		**Temperature sensor error**
	01h	Error when initializing the temperature sensor
	02h	Error when reading the temperature sensor
5010h		**Error SELFTEST**
FF00h		**Sensor error**
	01h	Error when initializing the sensor interface
	02h	Error when initializing the sensor module
	03h	Error during cyclic reading of the position values
FF20h		**Error when writing into the non-volatile memory**

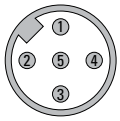
2. Electrical installation / Supply voltage and CANbus

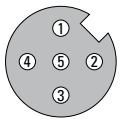
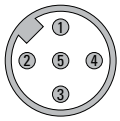
This chapter contains information about the electrical installation, configuration and startup of the CANopen inclinometer.

Electrical installation
Switch off the plant!
Make sure that the whole plant remains switched off during the electrical installation.

Electrical installation requires connectors or connection cables (see data sheet)

Terminal assignment

Interface	Type of connection	1 M12 connector, 5-pin					
2	1		Bus IN				
		Signal	+ V	0 V	CAN_GND	CAN_H	CAN_L
		Pin	2	3	1	4	5

Interface	Type of connection	2 M12 connectors, 5-pin					
2	3		Bus OUT				
		Signal	+ V	0 V	CAN_GND	CAN_H	CAN_L
		Pin	2	3	1	4	5
			Bus IN				
		Signal	+ V	0 V	CAN_GND	CAN_H	CAN_L
		Pin	2	3	1	4	5

Connect the shield to the inclinometer housing
Comply with the maximum cable lengths for stub lines and for the total length of the CANbus.
If possible, mount all cables with traction relief.
Check the maximum supply voltage on the device.

Function and status LED

The device is equipped with **two** LEDs for displaying status and error messages

Green = CANopen BUS status
Red = CANopen ERR display

Display	LED	Meaning	Error cause	Addition
All LEDs off	● ● ●	No connection to the master	Data line interruption Wrong baudrate Interchanged data lines **No voltage**	Observe the combination with the ERR LED If ERR LED is also off, please check voltage
Bus Flashing about 250 ms	●	Connection to the master **Pre-operational Status**		SDO communication
Bus Flashing about 1 sec.	●	Connection to the master **Stopped Status**		SDO communication impossible Only NMT commands
Bus On	●	Connection to the master **Operational Status**		PDO transfer is active
ERR off	● ●	Device operates without errors		Watch combination with BUS LED
ERR flashing	●	Connection to the master interrupted **Red** LED (1000 ms)	Combination with bus status	Bus LED flashing green or on - depends on object 1029h Error Behaviour
ERR On	●	BUS OFF status	Bus short-circuit or wrong baud rate	

The various LED messages can of course be displayed in combination.

[2]Master may be the PLC or a 2nd communication partner
[3]Operating voltage

CANopen LED combinations during operation

Display	LED	Meaning	Error cause	Addition
ERR flashing **Bus** on or flashing	● ●	**Red LED flashing** **Red LED flashes briefly duration 3 sec.**	Temperature overrun sensor monitoring	Device on CAN bus connection with master ok + additional error cause

CANopen error display after powering

Display	LED	Meaning	Error cause	Addition
ERR + BUS flashing	● ●	**Fast alternate flashing of green and red LED**	Data connection with sensor faulty Sensor faulty	Device must be sent back to manufacturer for repair
ERR + **BUS** flashing	● ●	**Simultaneous fast flashing of green and red LED (300 ms)**	Watchdog error	Device must be sent back to manufacturer for repair
ERR flashing	●	Connection to the master interrupted **red LED (1000 ms)**		No CAN bus present
BUS + **ERR** Fast flashing 50 ms	● ●	LSS layer service active Global mode active	Encoder waiting for configuration	LSS mode

3. Quick Start Guide – General settings on the appliance

- Carry out the electrical installation (voltage supply, bus connection)
- **Switch the appliance on**
- Set the bus parameters via LSS services or directly via the objects table
- Setting of the required baud rate — Object 2100h Baud rate
- Setting of the node address — Object 2101h Node address
- Setting of the termination — Object 2102h Termination
- Saving the bus parameters — Object 2105h Save all bus parameters
- **Appliance on/off cycle**

Object 2100h **Baud rate default setting: 250 kbit/s (Input 5)**
The baud rate can be modified **with a CANopen software on Object 2100h** or via the corresponding **LSS service**.

Value	Baudrate in kBit/s
0	10
1	20
2	50
3	---
4	125
5	250
6	500
7	800
8	1000

To be taken into consideration for the corresponding baudrate

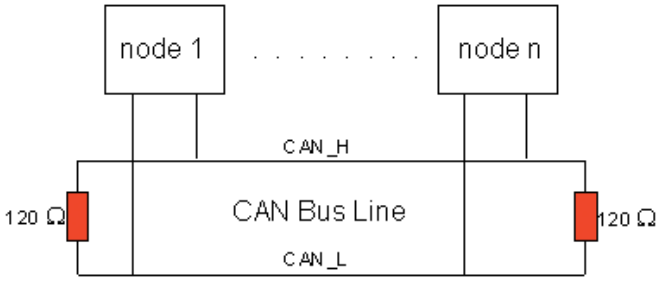
The chosen cycle time (see Object 1800h,Subindex 5 Event Timer) must be longer than the bus transfer time to ensure error-free PDO delivery!

For all baudrates, general cycle time at least **20ms**

Object 2101h Node address default setting: 0x3E (62 decimal).
The node number can also be modified **per software on Object 2101h** or via the corresponding **LSS service**.
Node number 0 is reserved and shall not be used by any node.
The resulting node numbers are in the range **1...7Fh** hexadecimal (1...127 decimal).

A new node number is only taken over at the following booting (reset/power-on) of the encoder or via a **NMT-Reset Node** command.

Object 2102h Termination default setting 0x1 (on)



CAN is a 2-wire bus system in which all participants are connected in parallel (that is to say with short stub lines). The bus must be terminated at both ends with a 120 (or 121) ohm terminating resistor in order to prevent reflections. This is necessary even in case of very short line lengths!

The **CAN bus termination** must be modified **per Software on Object 2102h**.

Since the CAN signals are represented on the bus as differential levels, the CAN line is relatively insensitive to interference (EMI). As always both lines are concerned, the interference almost does not affect the difference level.
With CAN, the maximum bus length is mainly limited by the signal propagation time. The multi-master bus access procedure (arbitration) requires that the signals are applied to all nodes quasi simultaneously (before the scanning within one bit time). As the signal propagation times of the CAN modules (transceiver, optocoupler, CAN controller) are nearly constant, the line length has to be adjusted according to the baudrate.

Baud rate	Bus length
1 MBit/s	< 20 m*
800 kBit/s	< 50 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

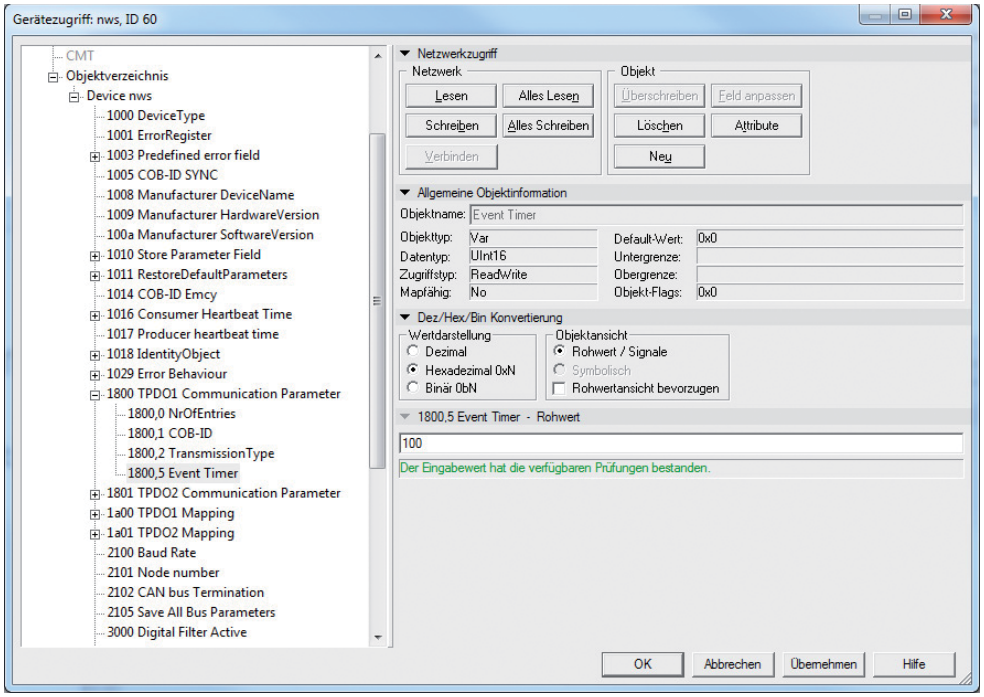
*) The value of 40 m for 1 MBit/s is often found for CAN in the literature. This however does not apply to networks with opto-decoupled CAN controllers. The worst-case calculation with optocouplers gives for 1 MBit/s a maximum bus length of 5m - experience however showed that 20 m can be reached without problems.

For bus lengths exceeeding 1000 m repeaters may be necessary.

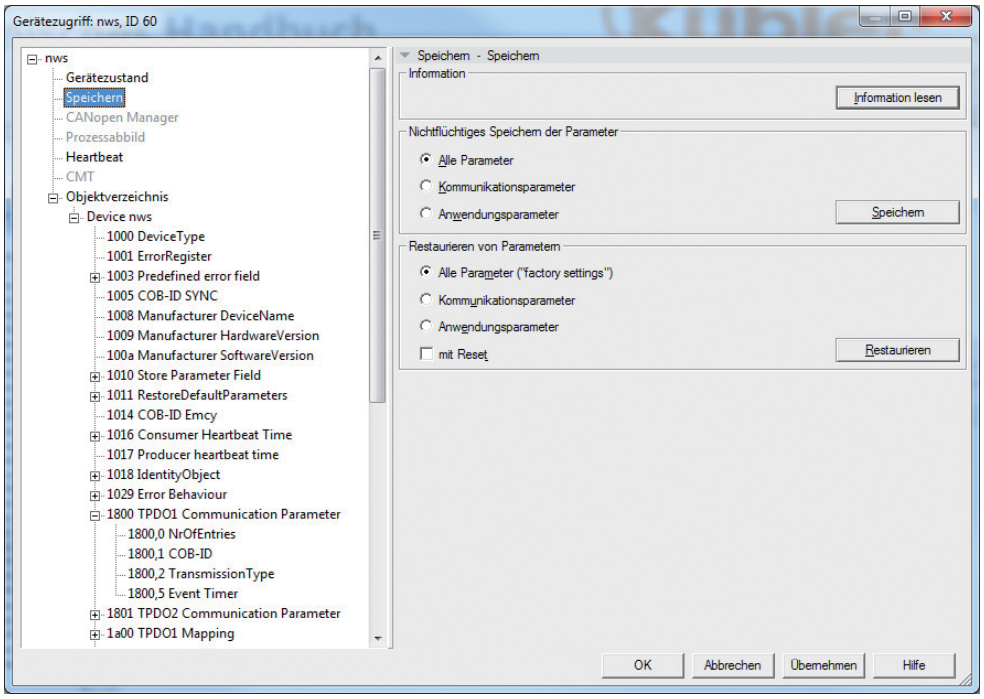
Object 2105h Save All Bus Parameters
This parameter **(Object 2105h)** saves the desired bus parameters (Object 2100h, 2101h, 2102h) **permanently** in the Flash memory. This object serves as an additional protection against accidental changes of the baudrate and node address.

Only targeted saving with parameter "save“ (hexadecimal 0x65766173) will save permanently the bus parameters **baudrate, node address and termination**.

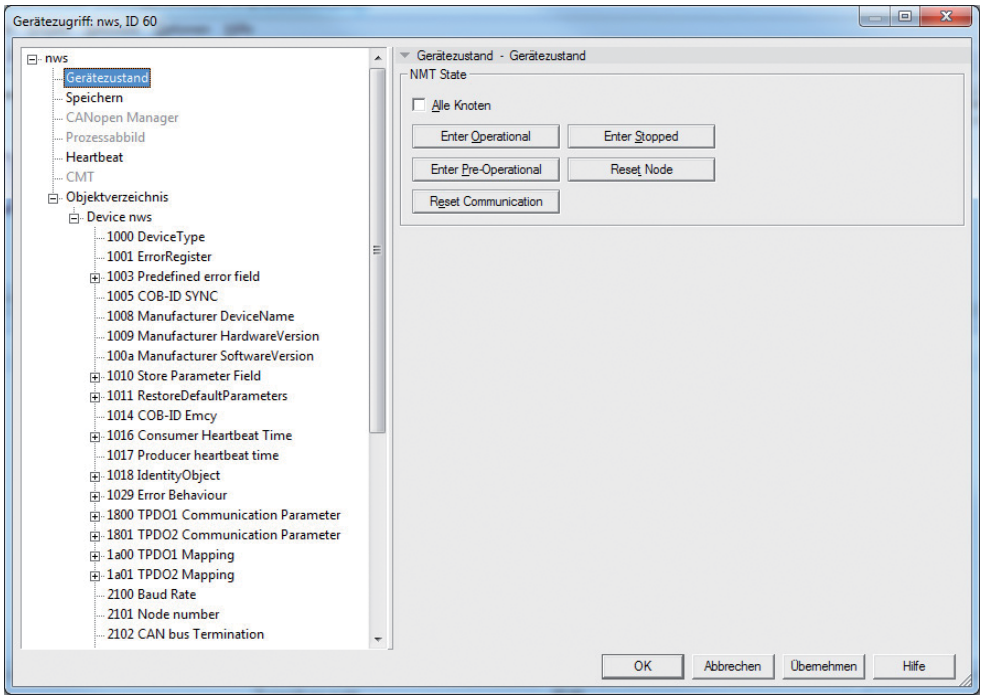
Example: Cyclic sending of TPDO1 with event timer
The event timer of TPDO1 is for example set to 100ms:



The settings are then stored in the non-volatile memory via object 1010 01h:



Switch then over to Operational Mode. The mapped objects of TPDO1 are sent in a 100ms cycle:



4. LSS services DS 305

LSS hardware requirements (LSS address)

All LSS slaves must have a valid object entry in the objects dictionary for the Identity object [1018h] in order to be able to carry out a selective configuration of the node. This object consists of the following sub-indices:

- Vendor ID (numerical number)
- Product Code (numerical number)
- Revision Number (major an minor revision as numerical number)
- Serial Number (numerical number)

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

LSS operative restructions
In order to ensure trouble-free LSS functionality, all devices in the network must support the LSS services. There can be only one LSS master. All nodes must start with the same baud rate.
LSS communication can only take place in “Stop" mode or in “Pre-operational” mode.

Exactly two conditions must be met for devices connected to a CANopen network - all devices must have the same baud rate and their node addresses must be unique within the network. The condition for operation under LSS is that a 1:1 CAN connection is established with the device. A special dialog mode then allows modifying the baud rate and node address. COB-ID **0x7E5** is sent by the master to the slave, the slave answers with COB-ID **0x7E4**.
LSS messages always have a length of 8 bytes. Unused bytes are reserved and should be filled with 0.
The LSS service can also modify the node address of a LSS slave. To do so, the LSS master sets the LSS slave back to the configuration mode. Then it communicates the new node address to the LSS slave. The LSS slave then answers to inform the LSS master whether this node address is within the supported range. After switching back to Operation mode, the LSS slave carries out a software reset in order to be able to reconfigure the communication objects with its new node number. Further LSS services are described in standard CiA DSP-305.

CANopen error display after powering

Identifier	DLC	Data							
		0	1	2	3	4	5	6	7
0x7E5	8	0x04	mod	reserved					

mod: new LSS mode
0 = switch on Operation mode
1 = switch on Configuration mode

Configure bit timing

Identifier	DLC	Data							
		0	1	2	3	4	5	6	7
0x7E5	8	0x13	tab	ind	reserved				

tab: indicates which baud rate table is to be used
0 = baud rate table defined according to CiA DSP-305 1 … 127 = reserved
128 … 255 = can be defined by the user himself

Ind: index within the baud rate table in which the new baud rate for the CANopen device is stored.

Standardized baud rates according to CiA DS305

Baudrate table 0x00	
Table index	Baud rate
0	1000 kBit/s
1	800 kBit/s
2	500 kBit/s
3	250 kBit/s
4	125 kBit/s
5	---
6	50 kBit/s
7	20 kBit/s
8	10 kBit/s

Configure Node ID
Allocation of a new node address

Identifier	DLC	Data							
		0	1	2	3	4	5	6	7
0x7E5	8	0x11	nid	reserved					

nid: new node address for the LSS slave (values from 1 to 127 are allowed)

Answer to Configure Node ID

Identifier	DLC	Data							
		0	1	2	3	4	5	6	7
0x7E4	8	0x11	err	spec	reserved				

err: error code
0 = successfully performed
1 = node address invalid (only values from 1 to 127 allowed)
2 … 254 = reserved
255 = special error code in spec
spec: manufacturer-specific error code (when err =255)

Switch Mode Global
To conclude the LSS service, the "Switch Mode Global" command switches the device back from LS configuration mode to Preoperational mode.

Identifier	DLC	Data							
		0	1	2	3	4	5	6	7
0x7E4	8	0x04	mod	reserved					

mod: new LSS mode
0 = switch on Operation mode
1 = switch on Configuration mode

All new settings become valid after a **Boot-up (Reset node) sequence**.

5. CANopen objects

Index (hex)	Sub Index	Object symbol	Attribute	Type	M / O	Name	Standard value	Mappable
1000h	00	CONST	RO	U32	M	Device Type		N
1001h	00	VAR	RO	U8	M	Error Register		N
1003h	xx	RECORD	RO	U32	O	Predefined Error Field		N
1005h	00	VAR	RW	U32	O	COB-ID Sync	80h	N
1008h	00	VAR	RO	STRING	O	Device Name		N
1009h	00	VAR	RO	STRING	O	Hardware Version		N
100Ah	00	VAR	RO	STRING	O	Software Version		N
1010h						Store Parameter		N
	01	VAR	RW	U32	O	Save All Parameters		N
1011h						Restore Parameter		N
	01	VAR	RW	U32	O	Restore All Default Parameters		N
1014h	00	VAR	RW	U32	O	COB-ID Emcy	BEh	N
1017h	00	VAR	RW	U32	O	Producer heartbeat time	0	N
1018h		RECORD				Identity Object		N
	01	VAR	RO	U32	M	Vendor ID		N
	02	VAR	RO	U32	M	Product Code		N
	03	VAR	RO	U32	M	Revision Number		N
	04	VAR	RO	U32	M	Serial Number		N
1029h		ARRAY				Error Behavior		N
	01	VAR	RW	U8	O	Communication Error	0	N
	02	VAR	RW	U8	O	Sync Error	0	N
	03	VAR	RW	U8	O	Internal Device Error	0	N

1800h					0	**TPDO1** Communication Parameter		N
	01	VAR	RW	U32	>	COB-ID	1BEh	N
	02	VAR	RW	U8		Transmission Type	255	N
	05	VAR	RW	U16		Event timer	0 [step 1 ms]	N
1801h					0	**TPDO2** Communication Parameter		N
	01	VAR	RW	U32	M	COB-ID	2BEh	N
	02	VAR	RW	U8		Transmission Type	1	N
	05	VAR	RW	U16		Event timer	0 [step 1 ms]	N
Mapping 2-dimensional								
1A00h					M	TPDO1 Mapping		N
	00	VAR	RW	U8		Number of Entries	3	N
	01	VAR	RW	U32		1st Mapped Object	0x60100010	N
	02	VAR	RW	U32		2nd Mapped Object	0x60200010	N
	03	VAR	RW	U32		3rd Mapped Object	0x50000010	N
	04	VAR	RW	U32		4th Mapped Object	0	N
1A01h					0	TPDO2 Mapping		N
	00	VAR	RW	U8		Number of Entries	3	N
	01	VAR	RW	U32		1st Mapped Object	0x60100010	N
	02	VAR	RW	U32		2nd Mapped Object	0x60200010	N
	03	VAR	RW	U32		3rd Mapped Object	0x50000010	N
	04	VAR	RW	U32		4th Mapped Object	0	N
Mapping 1-dimensional								
1A00h					M	TPDO1 Mapping		N
	00	VAR	RW	U8		Number of Entries	2	N
	01	VAR	RW	U32		1st Mapped Object	0x60100010	N
	02	VAR	RW	U32		2nd Mapped Object	0x50000010	N
	03	VAR	RW	U32		3rd Mapped Object	0	N
	04	VAR	RW	U32		4th Mapped Object	0	N
1A01h					0	TPDO2 Mapping		N
	00	VAR	RW	U8		Number of Entries	2	N
	01	VAR	RW	U32		1st Mapped Object	0x60100010	N
	02	VAR	RW	U32		2nd Mapped Object	0x50000010	N
	03	VAR	RW	U32		3rd Mapped Object	0	N
	04	VAR	RW	U32		4th Mapped Object	0	N
Profile DS410 Inclinometer								
6000h	00	VAR		U16	M	Resolution	0	N

6010h	00	VAR	RO	I16	M	Slope long 16		J
6011h	00	VAR	RW	U8	M	Slope long16 operating parameter	0	N
6012h	00	VAR	RW	I16	O	Slope long16 preset value	0	N
6013h	00	VAR	RW	I16	O	Slope long16 offset	0	N
6014h	00	VAR	RW	I16	O	Differential Slope long16 offset	0	N
6021h .. 6024h only 2-dimensional								
6020h	00	VAR	RO	I16	M	Slope lateral 16		J
6021h	00	VAR	RW	U8	M	Slope lateral16 operating parameter	0	N
6022h	00	VAR	RW	I16	O	Slope lateral16 preset value	0	N
6023h	00	VAR	RW	I16	O	Slope lateral16 offset	0	N
6024h	00	VAR	RW	I16	O	Differential Slope lateral16 offset	0	N
Manufacturer-specific objects								
2100h	00	VAR	RW	U8	O	Baud rate	5 (250 kBit/s)	N
2101h	00	VAR	RW	U8	O	Node Number	0x3E (62d)	N
2102h	00	VAR	RW	U8	O	Termination	1 = ON	N
2105h	00	VAR	RW	U32	O	Save All Bus Parameters	0x65766173	N
3000h	00	VAR	RW	U16	O	Digital Filter Active	1 = ON	N
3001h	00	VAR	RW	F32	O	Digital Filter Coefficient	10.0	N
5000h	00	VAR	RO	I16	O	Internal Temperature		J
5001h	00	VAR	RO	U16	O	Sensor Information		J
5002h only 1-dimensional								
5002h	00	VAR	RO	U16	O	*Raw Slope long16 High Resolution		J

Uxx = UNSIGNED
Ixx = SIGNED
Fxx = FLOAT
VAR = Variable
ARRAY = Array of variables
RW = Read/Write
RO = Read only
const = Constant
Name = Object name
M/O = Mandatory or Optional.

*Raw Slope long16 High Resolution with a resolution of 0.01°.

6. The communication profile DS 301

All communication objects and all user objects are summarized in the **Object Dictionary** (OD). In the CANopen device model, the OD is the **link** between the application and the CANopen communication unit.
Every entry in the object dictionary represents an object and is identified by a **16-bit index**. An index can include up to 256 sub-indexes. This allows differentiating up to 65536 × 254 elements independently of the "11-bit identifiers". (Subindexes 0 and 255 cannot be freely used.)
The allocation of communication and device profile objects to a individual index is precisely defined in profiles; therefore, the object dictionary defines a unique interface between the application and the external communication.
So, for example, every CANopen node in the network knows that the heartbeat interval can be found in index 1017h, and every node or every configuration program has a read or write access to it.

Index range	Use
0000	Unused
0001-009F	Data types (special case)
00A0-0FFF	Reserved
1000-1FFF	Communication profile
2000-5FFF	Manufacturer-specific area
6000-9FFF	Up to 8 standardized device profiles
A000-AFFF	Process image of IEC61131 devices
B000-BFFF	Process image of CANopen gateways according to CiA 302-7
C000-FFFF	Reserved

Service data objects (SDO) provide an access service to the object dictionary. Every CANopen device needs at least one SDO server that receives and processes SDO requests from other devices. As the default setting, messages to the SDO server of a device use the node number of the addressee + 1536 as COB-ID or as "Identifier" for the CAN message. The answer of the SDO server uses the node number of the sender + 1408 as "identifier". These relatively high and therefore low-priority IDs are used for transferring entries in the OD. There is a protocol for this SDO transfer that requires 4 bytes for coding the send direction, the index and the subindex. So only 4 bytes of the 8 bytes of a CAN data field remain for the data content. For objects with a data content exceeding 4 bytes, there are two other protocols for **fragmented SDO transfer.**

In contrast to the low-priority and protocol data-overloaded SDO transfer, the **process data objects (PDO)** offer a faster possibility for process data transfer.

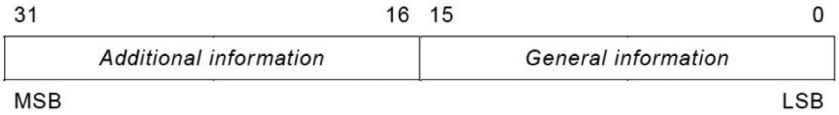
The "identifiers" used for PDO transfer have **default settings in the COB-ID range from 385 to 1407** and have therefore a higher priority than the SDO messages. Moreover, they only contain payload, making 8 bytes available. The content of the payload is defined by PDO mapping entries. These are objects in the OD, which define, like an allocation table, which data is to be transferred via a PDO. This data is in turn contained in other objects of the OD.

A PDO can also transfer the values of several objects, and the addressees of the PDO can use only parts of the data, according to their PDO mapping entries. When receiving a PDO, the data is in turn written in other objects of the OD according to the mapping entries, e.g. in a digital output object. PDO transfer can take place cyclically, event-driven, upon request or synchronized.

Network management objects (NMT) are used to manage the network. So there are, among others, messages that lead to a status change in a device or that diffuse global error messages.

The **Sync object** transmits or receives for example the high-priority SYNC message used for the synchronization of the nodes in the network and for ensuring, with the Timestamp object, a unified time all over the network. Besides this, a communication profile and, in particular in the device profiles, many other objects are available.

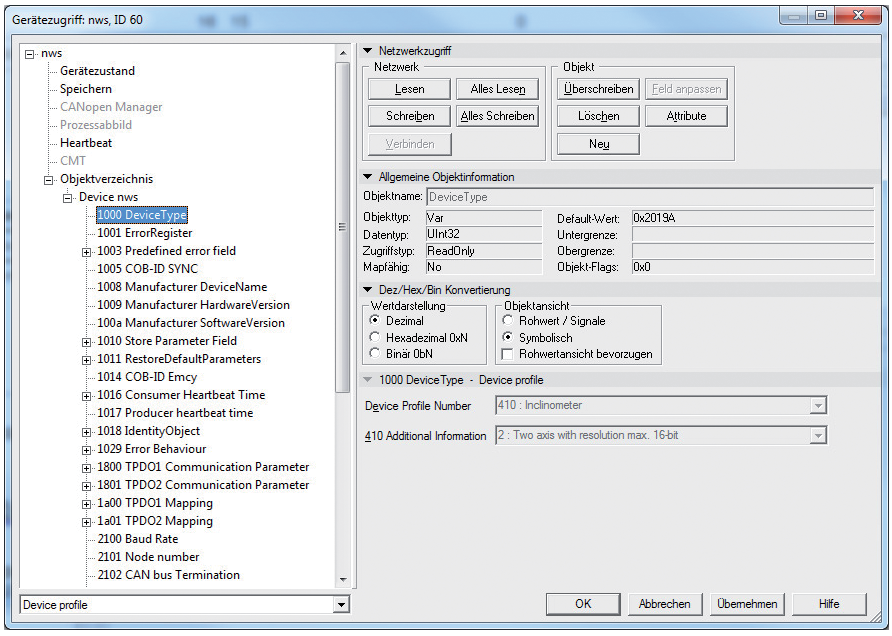
Object 1000h Device Type
This object shows information about the device and the device profile.



Bits 0-15 indicate the device profile version
Bits 16-23 specify the inclinometer type

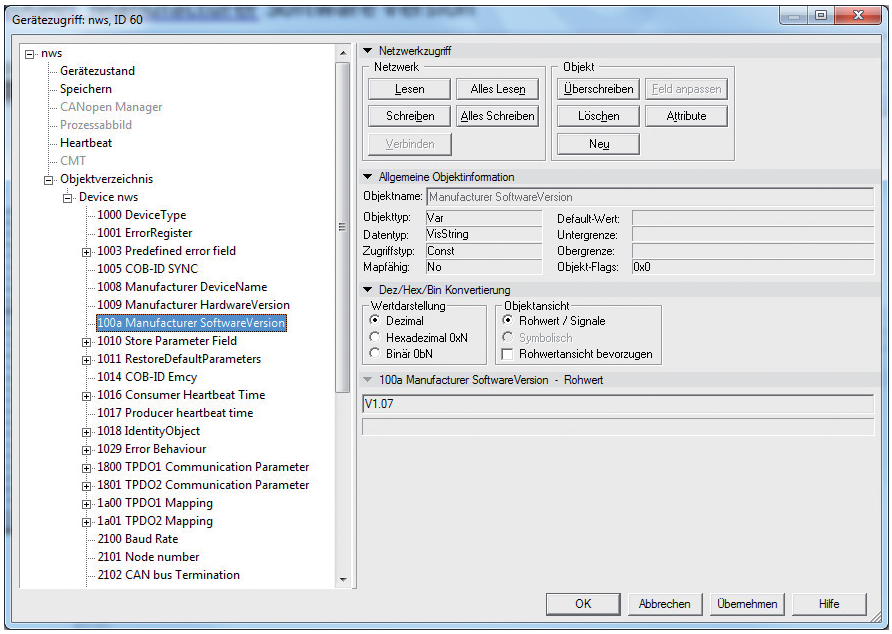
Field	Value	Definition
General Information	410_d	Device profile number
Additional information	0000_h 0001_h 0002_h 0003_h 0004_h 0005_h to $FFFE_h$ $FFFF_h$	see / CiA 301 / One axis with resolution max. 16 bits Two axes with resolution max. 16 bits One axis with resolution max. 32 bits Two axes with resolution max. 32 bits reserved see / CiA 301 /

The Kübler encoder M3668 uses type **2019Ah**



Object 100Ah Manufacturer Software Version

Information about the current implemented software



Objekt 1010h Save CANopen Parameters

Command "save“ under subindex 1h (save all Parameters) saves the parameters in the non-volatile memory (FLASH MEMORY). All communication objects, application objects and manufacturer-specific objects are saved under this subitem. **This operation requires about 20 ms.**

In order to prevent accidental saving, the command is only carried out when the code word string "save" is entered in this subindex.
Parameter "save“ (hexadecimal 0x65766173)

Object 1011h Load CANopen Factory Default Values

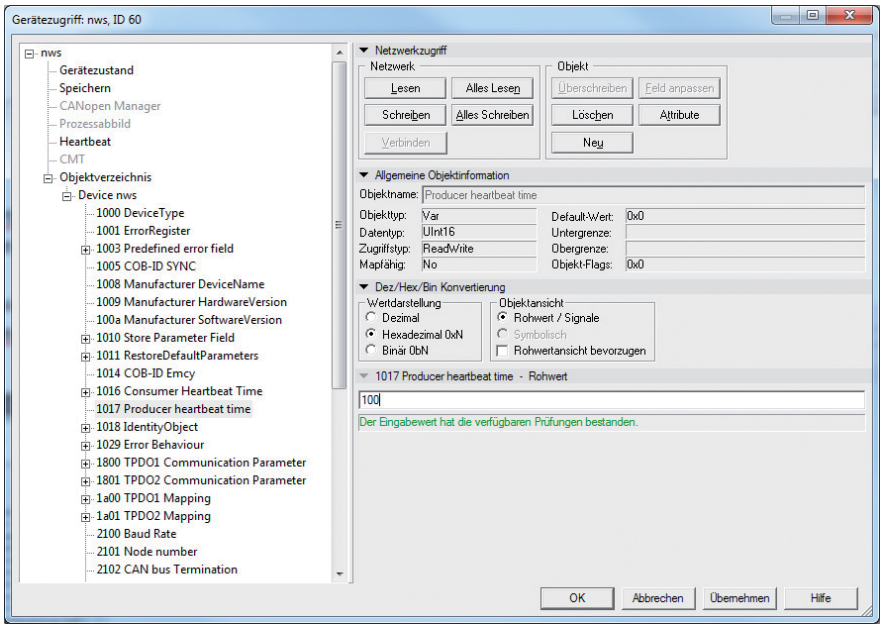
Command 'load" under subindex 1h resets all parameters to their standard values. In order to prevent accidental loading of the standard values, the command is only carried out when the code word string "load" is entered in this subindex.
Parameter "load“ (hexadecimal 0x64616F6C)

Object 1017h Producer Heartbeat Object
Heartbeat Producer Protocol
The producer heartbeat time defines the heartbeat cycle. If this function is not required, time must be set to 0. This function is activated with a time between **1 ms** and max. **65535ms**

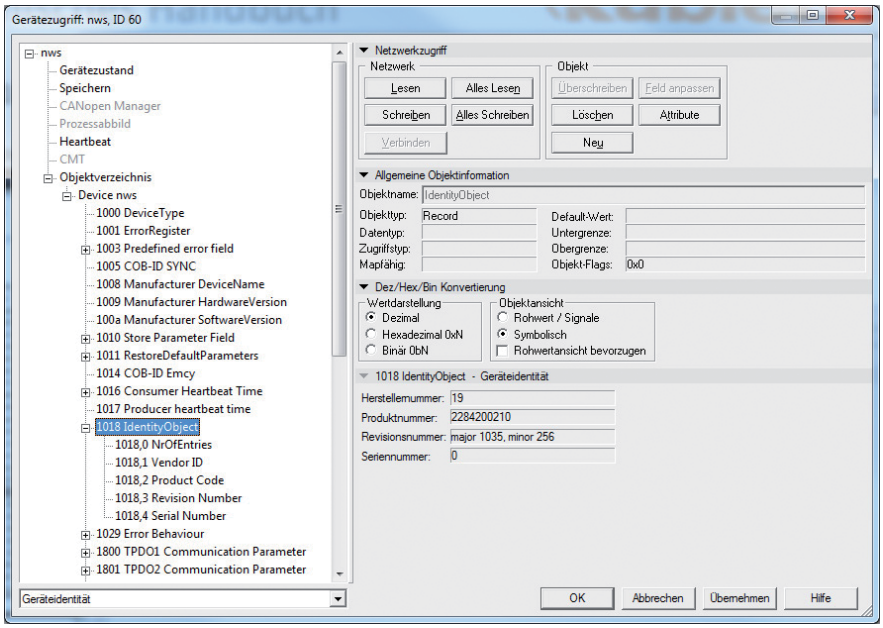
Object Description	
INDEX	1017h
Name	Producer Heartbeat Time
Object Code	VAR
Data Type	UNSIGNED16
Category	Conditional; Mandatory if guarding not supported
Entry Description	
Access	rw
PDO Mapping	No
Value Range	UNSIGNED16
Dafault Value	0

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

Heartbeat is used for monitoring the node. Here, for example, a value of **100ms** has been set for the heartbeat.



Object 1018h Identity Object
Information about the manufacturer and the device:

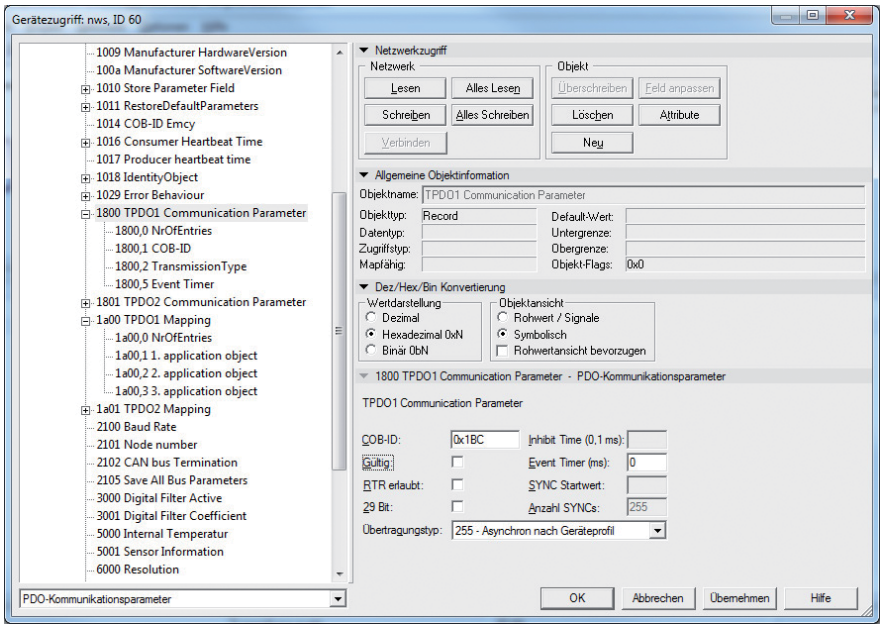


1018 RECORD Device – Identification read only
Sub-Index 0h : "Number of Subindexes" returns the value 4
Sub-Index 1h: "read" only returns the **Vendor ID (000000013h) Fritz Kübler GmbH**
Sub-Index 2h: returns the **Product Code** (e.g. 0x08082721 CANopen Inclinometer 2-axes)
Sub-Index 3h: "read" only, returns the **software revision number** (e.g. 108)
Sub-Index 4h: "read" only, returns the 10-digit **serial number** of the encoder

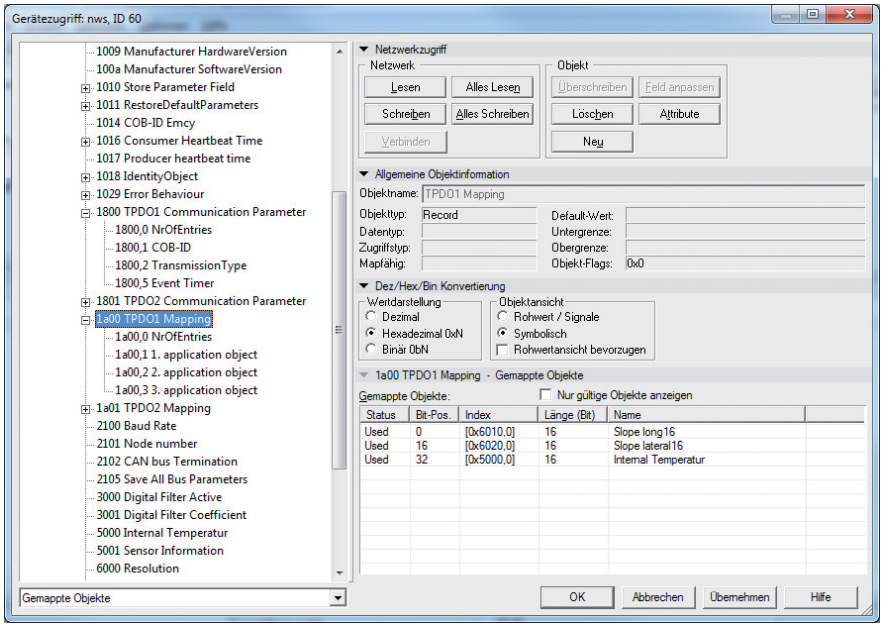
Dynamic mapping

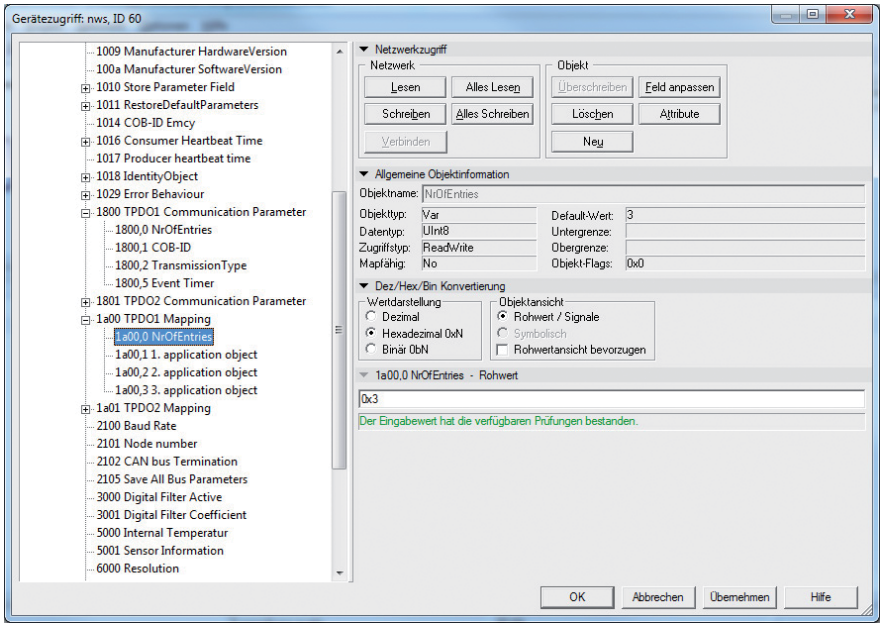
The mapping entries of TPDO 1&2 can be modified according to the needs. Four 16-bit mappable objects can be transferred per TPDO.

Example: Changing the mapping entries of TPDO 1:
1. TPDO1 is set to "invalid" in object 1800h:

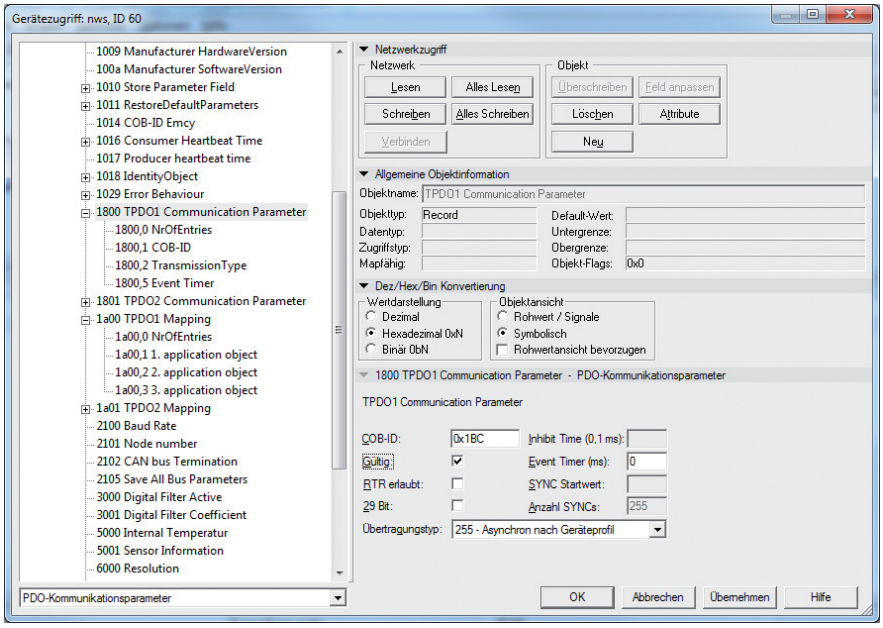


2. The mapping entries of TPDO 1 are modified in object 1A00A, and then the number of the actual entries is updated in 1A00_00h:





3. TPDO1 is then set back to "valid":



Emergency message

Emergency objects appear in case of error situations within a CAN network and are triggered according to the event and transferred on the bus with **high priority**.
Important: An Emergency object is only triggered **once per "event"**. No new object is generated until the error is corrected. Once the error is corrected, a new Emergency object with content 0 (Error Reset or No Error) is generated and sent on the bus.

"Emergency"-type messages are used to signal the errors of a device. The Emergency telegram transfers a code that identifies clearly the error (defined in the communication profile CiA 301and in the respective device profiles CiA 410).

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

Example of a message in case of 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**	Error register
[ManufacturerSpecific2]	**56**	current temperature
[ManufacturerSpecific3]	**20**	current threshold value lower range
[ManufacturerSpecific4]	**50**	current threshold value upper range
[ManufacturerSpecific5]	**2E**	Version register

The behavior in case of an error is described in **Object 1029h Error Behavior**.

Object 1029h **Error Behavior**
In case of a serious error, the device should switch automatically to Pre-Operational mode. This object allows setting how the device has to behave in case of an error. The following error classes are covered:

1029h, Subindex 1 Communication error
- Bus off status of the CAN interface
- Life guarding event occurred
- Heartbeat monitoring 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 set up as follows:

Byte 0
$2^7 \ldots 2^0$

Value range 8 bits
- 0 Pre-Operational mode (only if the Operational mode was active previously)
- 1 no mode change
- 2 Stopped mode
- 3 … 127 reserved

7. Objects of Encoder Profile DS 410

Object 6000h **Resolution**
Default setting 2-axes sensor:10d = Resolution 0.01°
Default setting 1-axis sensor:100d = Resolution 0.1° (0.01 ° Resolution with Object 5002h without scaling)

Value	Definition
1d (01h)	0.001° not supported
10d (0Ah)	**0.01° only for 2-dimensional**
100d (64h)	0.1°
1000d (3E8h)	1.0°
other	not supported

Parameter "6000 Resolution" influences measuring axes long16 and lateral16 !

Angle calculations
2-axis inclinometer

Orientation angles
Indicating the two orientation angles describes the inclination of the coordinates system of the sensor with respect to the gravitational direction. The first value output corresponds to a rotation around the y-axis of the sensor and is called "Orientation angle X". This value corresponds to the angle [°] formed by the gravity vector with the yz plane of the sensor. The second value output corresponds to a rotation around the x-axis of the sensor and is called "Orientation angle Y". This value corresponds to the angle [°] formed by the gravity vector with the zx plane of the sensor.

Orientation angle X $= \sin^{-1}\left(\frac{x}{\sqrt{x^2+y^2+z^2}}\right)$

Orientation angle Y $= \sin^{-1}\left(\frac{y}{\sqrt{x^2+y^2+z^2}}\right)$

1-axis inclinometer

Euler angles
In this setting, both angle values output are to be interpreted as Euler angles. The current sensor position is given by two revolutions performed consecutively, based on its horizontally aligned position. The "Euler angle Z" indicates the angle [°] by which the z axis of the sensor is deflected. The "Euler angle XY" corresponds to the angle [°] by which the sensor has then been rotated around the (deflected) z axis.

Euler Angle Z $= \cos^{-1}\left(\frac{z}{\sqrt{x^2+y^2+z^2}}\right)$

Euler Angle Z $= \tan 2^{-1}(x, y)$

Object 6010h Slope long16
The inclinometer supplies the value measured by the measuring axis long16 as a **signed** 16-bit value in degrees [°]. The measured value depends on the **settings of objects 6011h - 6014h**. These values influence the calculation and the result.

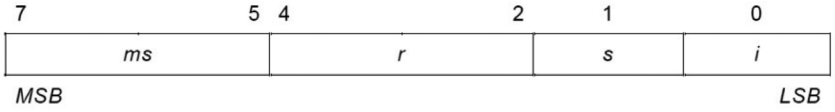
Data content:

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

2-dimensional:
Values range: 0....+/- 85.00 ° (signed value)
Update rate of the measured value: 20ms

1-dimensional:
Values range: 0.. 360.0 °
Update rate of the measured value: 20ms

Object 6011h Slope long16 Operating parameter
This object allows switching on and off the scaling with Offset/Preset of objects 6012h-6014h and the measurement value inversion of Slope long16 in object 6010h.



Field	Value	Definition
ms		Manufacturer-specific
r	0_b	reserved
s (scaling)	0_b 1_b	Scaling not enabled Scaling enabled
i (inversion)	0_b 1_b	Inversion not enabled Inversion enabled

Scaling:
If scaling is switched on, the measured value of 6010h long16 is calculated as follows:
Slope long16 = physically measured angle + Differential slope long16 offset + Slope long16 offset
If scaling is switched off, the measured value of 6010h corresponds to the physically measured value.

Inversion:
If inversion is switched on, the measured value of 6010h is output inverted.

Object 6012 Slope long16 preset value
Object 6012 allows setting the measured value of 6010h long16 to a desired angle value (PRESET). The desired angle value is transmitted to object 6000 as a signed 16-bit value, taking into consideration the resolution set previously.
The differential offset of object 6014h is included in the Preset calculation.
The angle offset calculated by the Preset value in 6012h can be read or modified via object 6013h.

Angle offset calculation:
Slope long16 offset = Slope long16 preset value at t_{acc} – slope physical measured at t_{acc} – Differential slope long16 offset
t_{acc} = time when accessing object 6012_h

Calculation of the measured value object 6010h long16:
Slope long16 = physically measured angle + Differential slope long16 offset + Slope long16 offset

Example:
The measured value of object 6010h is to be set to +45.00 °. The resolution in object 6000h is set to 0.01° = 10d:
Object 6012h = 4500 (SIGNED16)

Data content:

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

2-dimensional:
Values range: 0 ... +/-85.00°. Example: +45.00° = 4500 (SIGNED16)

1-dimensional:
Values range: 0 … 360.0°. Example: 45.0° = 450 (SIGNED16)

CAUTION! The entry must be adapted to the selected resolution of object 6000h!

Object 6013h Slope long16 offset
Object 6013h allows transferring directly an angle offset that will be used with the measured value of 6010h long16 in the calculation. The angle offset is transferred with a signed 16-bit value, depending on the resolution set in object 6000h.
Values range:

2-dimensional:
+/-180.00°. Example: +45.00° = 4500 (SIGNED16)

1-dimensional:
+/-360.0°. Example: +45.0° = 450 (SIGNED16)

CAUTION! The input must be adapted to the selected resolution of object 6000h!

Slope long16 = physically measured angle + Differential slope long16 offset + Slope long16 offset

Data content:

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

Object 6014h Differential Slope long16 offset
Object 6014h allows shifting the measuring range with an offset independently of objects 6012h Preset and 6013h Offset.
To that purpose, a signed 16-bit angular value, depending on the resolution set in object 6000h, can be transferred in object 6014.
Values range:

2-dimensional:
+/-85.00°. Example: +45.00° = 4500 (SIGNED16)

1-dimensional:
+/-360.0°. Example: +45.0° = 450 (SIGNED16)

CAUTION! The input must be adapted to the selected resolution of object 6000h!

Slope long16 = physically measured angle + Differential slope long16 offset + Slope long16 offset

Data content:

Byte 0	Byte 1
00	10h

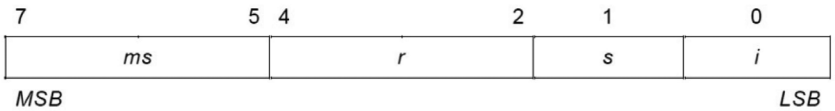
Object 6020h Slope lateral16 (ONLY FOR 2-dimensional)
The inclinometer supplies the value measured by the measuring axis lateral16 as a **signed** 16-bit value in degrees [°]. The measured value depends on the **settings of objects 6012h - 6024h**. These values influence the calculation and the result.

Data content:

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

Values range: +/- 85,00 ° (signed value)
Update rate of the measured value: 20ms

Object 6021h Slope lateral16 operating parameter (ONLY FOR 2-dimensional)
This object allows switching on and off the scaling with Offset/Preset of objects 6022h-6024h and the measurement value inversion of Slope lateral16 in object 6020h.



Field	Value	Definition
ms		Manufacturer-specific
r	0_b	reserved
s (scaling)	0_b 1_b	Scaling not enabled Scaling enabled
i (inversion)	0_b 1_b	Inversion not enabled Inversion enabled

Scaling:
If scaling is switched on, the measured value of 6020h lateral16 is calculated as follows:
Slope long16 = physically measured angle + Differential slope lateral16 offset + Slope lateral16 offset
If scaling is switched off, the measured value of 6020h corresponds to the physically measured value.

Inversion:
If inversion is switched on, the measured value of 6020h is output inverted.

Object 6022 Slope lateral16 preset value (ONLY FOR 2-dimensional)
Object 6022 allows setting the measured value of 6020h lateral16 to a desired angle value (PRESET). The desired angle value is transmitted to object 6000 as a signed 16-bit value, taking into consideration the resolution set previously.
The differential offset of object 6024h is included in the Preset calculation.
The angle offset calculated by the Preset value in 6022h can be read or modified via object 6023h.

Angle offset calculation:
Slope lateral16 offset = Slope lateral16 preset value at t_{acc} – slope physical measured at t_{acc} – Differential slope lateral16 offset
t_{acc} = time when accessing object 6022_h

Calculation of the measured value object 6020h lateral16:
Slope lateral16 = physically measured angle + Differential slope lateral16 offset + Slope lateral16 offset

Example:

The measured value of object 6020h is to be set to +45.00 °. The resolution in object 6000h is set to 0.01° = 10d:
Object 6022h = 4500 (SIGNED16)

Data content:

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

Values range: +/-85.00°. Example: +45.00° = 4500 (SIGNED16)

Object 6023h Slope lateral16 offset (ONLY FOR 2-dimensional)

Object 6023h allows transferring directly an angle offset that will be used with the measured value of 6020h lateral16 in the calculation. The angle offset is transferred with a signed 16-bit value, depending on the resolution set in object 6000h.
Values range: +/-180.00°. Example: +45.00° = 4500 (SIGNED16)

Slope lateral16 = physically measured angle + Differential slope lateral16 offset + Slope lateral16 offset

Data content:

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

Object 6024h Differential Slope lateral16 offset
Object 6024h allows shifting the measuring range with an offset independently of objects 6022h Preset and 6023h Offset.
To that purpose, a signed 16-bit angular value, depending on the resolution set in object 6000h, can be transferred in object 6024.
Values range: +/-85.00°. Example: +45.00° = 4500 (SIGNED16)

Slope lateral16 = physically measured angle + Differential slope lateral16 offset + Slope lateral16 offset

Data content:

Byte 0	Byte 1
00	10h

8. Manufacturer-specific objects

Object 2100h Baud rate
This object allows modifying the baud rate by software. As a standard, the value is set to 05h, i.e. 250kBit/s. If the value is set between 0..8 and the parameter is saved using **object 2105h Save All Bus Parameters**, the device will boot with the modified baud rate at the following powering or **Reset Node.**

Data content:

Byte 0
$2^7 \ldots 2^0$

Values range **0 ...8 (see baud rate table)**

A new baud rate is only taken over at the following booting (reset/power-on) of the encoder or via a **NMT-Reset Node** command. All other settings in the objects table remain retained.

Object 2101h Node address
This object allows modifying the node address by software. As a standard, the value is set to 0x3Eh, i.e. Node-ID= 0x3E. If the value is set between 1..127 and the parameter is saved using **object 2105h Save All Bus Parameters**, the device will boot with the modified node address at the following powering or **Reset Node.**

Data content:

Byte 0
$2^7 \ldots 2^0$

Values range **1 ...127 or 1..7Fh**

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

A new node number is only taken over at the following booting (reset/power-on) of the encoder or via a **NMT-Reset Node** command. All other settings in the objects table remain retained.

Object 2102h CAN bus termination off/on
This object allows switching the bus termination on by software. As a standard, this value is set to 0, i.e. the termination is switched off.

Data content:

Byte 0
$2^7 \ldots 2^0$

Values range **0..1**

***for devices with cable exit and a CAN connection = 1**

Object 2105h Save All Bus Parameters
This parameter saves the desired bus parameters (objects 2100h, 2101h, 2102h) permanently in the Flash memory. This object serves as an additional protection against accidental changes of the baudrate and node address.
Only targeted saving with parameter **"save" (hexadecimal 0x65766173)** will save permanently the bus parameters **baudrate, node address and termination**.

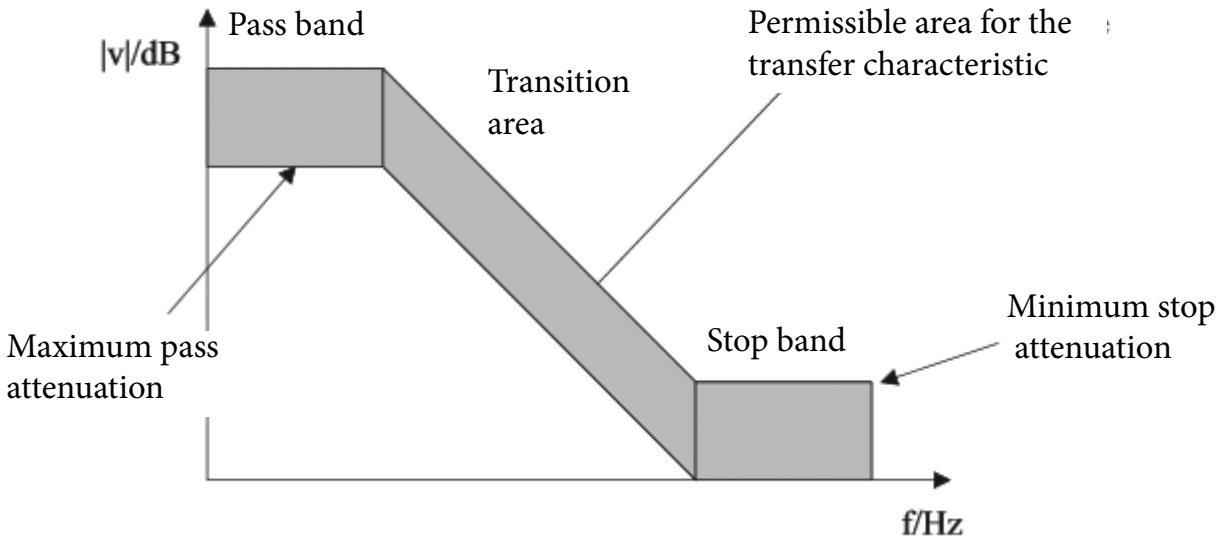
Data content:

Byte 0	Byte 1	Byte 2	Byte 3
$2^7 \ldots 2^0$	$2^{15} \ldots 2^6$	$2^{23} \ldots 2^{16}$	$2^{31} \ldots 2^{24}$

Values range: **"save" in hexadecimal 0x65766173**

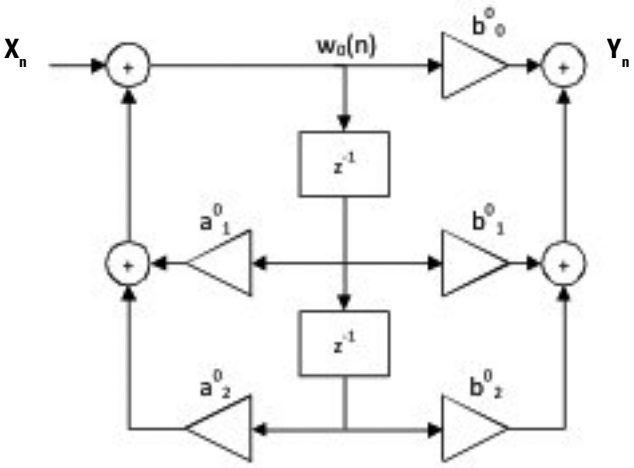
Low-pass filter

In electronics, low-pass filters are filters that let pass signal portions with frequencies lower than their limit frequencies almost without attenuation and attenuate signal portions with higher frequencies.



Setting possibilities: Filter on/off
Filter operating frequency b :defines the starting point of the stop band (area 0.1 … 10.0 Hz)

Filter description 2nd order:
An IIR filter is generally realized with the help of **2nd order subsystems** in direct form.
The following picture shows the corresponding block diagram. A subsystem consists of 2 delay elements or memory elements that contain the intermediate values w0(n), as well as of the two coefficients a01, a02 in the recursive portion and the three coefficients b00, b01 and b02.
The second index (j) is used for differentiation in case of several subsystems. A subsystem is described by equations, see below. The device uses 4 2nd order subsystems, resulting in
an **8th order Butterworth filter.**



$$w_0(n) = x(n) + a^0_1 * w_0(n-1) + a^0_2 * w_0(n-2)$$

$$y_0(n) = b^0_0 * w_0(n) + b^0_1 * w_0(n-1) + b^0_2 * w_0(n-2)$$

X_n is here the input signal, Y_n is the **filter output and simultaneously the input** of another subsystem.

Object 3000h Low-pass filter on/off
Data content:

Byte 0	Byte 1
$2^7 \ldots 2^0$	$2^{15} \ldots 2^6$

Low-pass filter on 0x1
Low-pass filter off 0x0

Object 3001h Filter coefficient
Standard setting: Filter operating frequency b Value 10.0

Data content:

Byte 0	Byte 1
$2^7 \ldots 2^0$	$2^{15} \ldots 2^6$

Possible settings: 0.1, 0.3, 0.5, 1.0, 2.0, 5.0, 10.0 Hz
Other values are set by default to 10.0 Hz.
Values range: Real32 0.1 ... 10.0 Hz

Object 5000h Actual Temperature Sensor *
This object indicates the current temperature inside of the sensor as a signed 16-bit hexadecimal value. This value allows checking the current temperature of the device.

Data content:

Byte 0	Byte 1
$2^7 \ldots 2^0$	$2^{15} \ldots 2^6$

Values range up to 00…FFFFh
Example: 0x103 corresponds to + 25.9°C

* The temperature value can be mapped as a 16-bit value with the process data. It is updated there all 2s. Accuracy is ± 0.2°C, measurement takes place inside of the encoder electronics.

Object 5001h Sensor Information
This object allows displaying sensor information during operation.

Data content:

Byte 0	Byte 1
$2^7 \ldots 2^0$	$2^{15} \ldots 2^6$

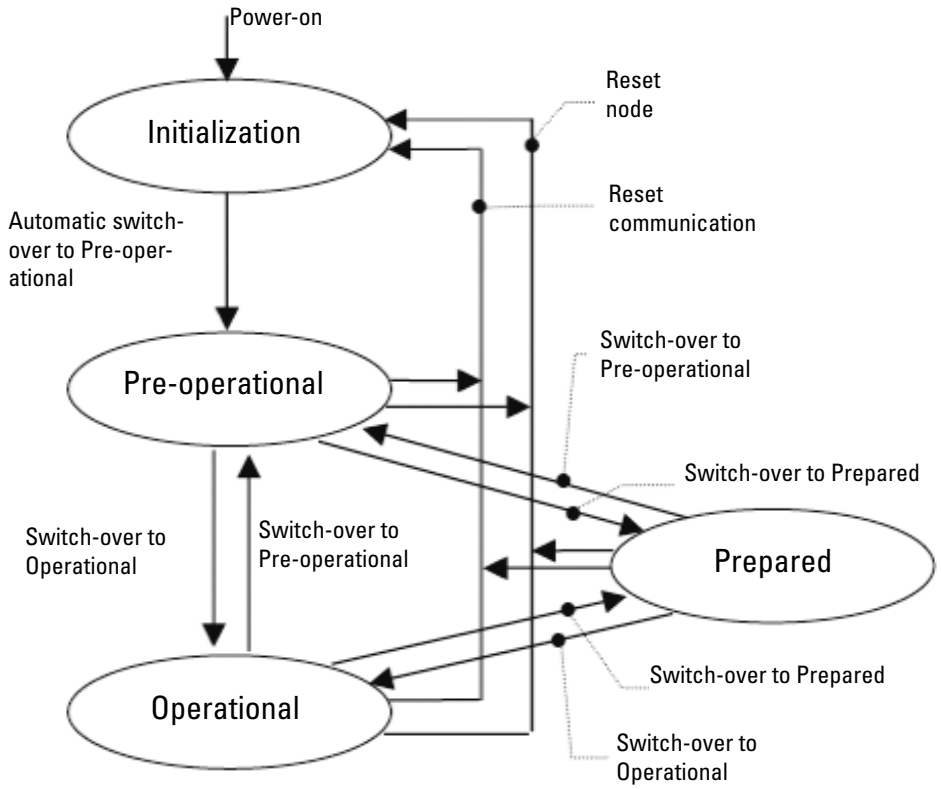
Bit 0..1 = Overflow: Measuring range exceeded
Value = 0: Sensor positioned within the valid measuring range > -85,00° & <+85,00°
Value = 1: Sensor positioned outside of the positive measuring range > +85,00°
Value = 2: Sensor positioned outside of the negative measuring range < -85,00°

Bit 2 = Back / Front Indication:
Value = 0 Sensor upside down
Value = 1 Sensor in normal position/installation position

Bit 8..15 = Orientation:
Value = 0 Upside down X , normal position Y
Value = 2 X rotated, Y rotated
Value = 6 Normal position X , normal position Y

9. Network management

This encoder supports the simplified network management (minimum boot up) defined in the profile for "minimum capability devices". The following status diagram according to DS 301 shows the various node statuses and the corresponding network commands (controlled by the network master via NMT services):



Initialization: this is the initial state after the power supply is applied, following a device Reset or Power ON. After performing the reset/initialization routines, the node switches automatically to the Pre-operational status. The LEDs display the momentary status.

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

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

Prepared or **Stopped:** In this status, the node is no longer active, i.e. neither SDO nor PDO communication is possible. The node can be set to Operational or Pre-operational status by means of NMT commands.

10. 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 participant.
An NMT object is structured as follows:

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

COB-ID = 0
Byte 0 = Command byte
Byte 1 = Node number (e.g. 3F or 00 for all participants)

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.

Command byte (hex)	Description
01h	Start_Remote_Node: Switch to Operational
02h	Stop_Remote_Node: Switch to Prepared
80h	Enter_Pre-Operational_State: Switch to Pre-operational
81h	Reset_Node: Reset the node[1]
82h	Reset_Communication: Reset the communication[2]

[1]All parameters of the whole object dictionary are set to Power-on values.
[2]only the parameter in section Communication profile of the object dictionary are set to Power-on values.

11. Abbreviations used

CAL	CAN Application Layer. Application layer (Layer 7) in the CAN communication model
CAN	Controller Area Network
CiA	CAN in Automation. International association of CAN products users and manufacturers
CMS	CAN Message Specification. Service element of CAL
COB	Communication Object. Transport unit in the CAN network (CAN message). Data is sent through the network in a COB.
COB-ID	COB-Identifier. Univocal designation of a CAN message. The identifier determines the priority of the COBs in the network.
DBT	Distributor. Service element of CAL, responsible for the dynamic allocation 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 in the various 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 initialization, configuration and errors handling within the network.
MT	Multiturn encoder
OSI	Open Systems Interconnection. Layers model for describing the functional areas in a data communication system.
PDO	Process Data Object. Object for the exchange of process data.
RTR	Remote Transmission Request; data request telegram
SDO	Service Data Object; communication object used by the master to access the object dictionary of a node.
SYNC	Synchronization telegram. Bus participants answer the SYNC command with their process value.

12. Glossary

Baud rate
The baud rate is the transmission rate. It is related with the nominal bit timing. The maximum posible baud rate depends on many factors that influence the signal propagation time on the bus. There is a substantial link between the maximum baud rate and the bus length and cable type. Various baud rates are defined between 10 kbit/s and 1 Mbit/s in CANopen.

CANopen
CANopen is a CAN-based protocol developed originally for industrial control systems. The specifications include various device profiles as well 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 many optional features are ideal for customized solutions. A wide range of configuration tools is moreover available. The user can define on this basis application-specific device profiles. Further information about CANopen can be found in the Internet at the address www.can-cia.org.

EDS file
The EDS (Electronic Data Sheet) file is provided by the manufacturer of a CANopen device. It has a standardized format for the description of devices. The EDS file contains information about:
- File description (name, version, date of creation etc.)
- 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 through their attributes.

Node number
Within a CanOpen network, every device is defined by its node number (node ID). The permissible node numbers are in the range of 1-127 and can only be used once within a network.

Network management
Various tasks are to be performed in a distributed system in connection with the configuration, initialization and monitoring of the network participants. The service element "Network Management (NMT)" defined in CANopen provides this functionality..

PDO
The process data objects (PDO) are the actual means of transport 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 can reach 8 bytes. It can be used to transport several application objects. The PDO mapping describes the arrangement of the application objects within the data field of the PDO.

SDO
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.

www.kuebler.com

Kübler

wir geben Impulse

Kübler Group
Fritz Kübler GmbH
Schubertstraße 47
D-78054 Villingen-Schwenningen
Germany
Phone: +49 7720 3903-0
Fax: +49 7720 21564
info@kuebler.com
www.kuebler.com

R67028.0002 - Index 2