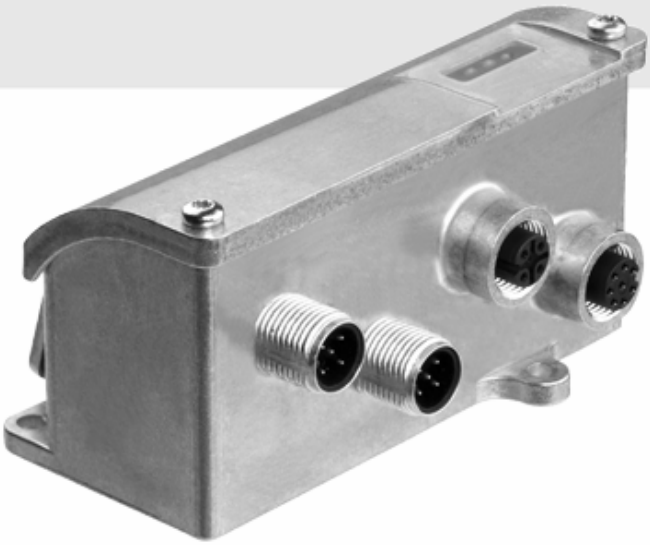
OPERATING INSTRUCTIONS

E



CANopen

SICK

SICK|STEGMANN GmbH
Dürrheimer Straße 36
D-78166 Donaueschingen
Tel: (49) 771 / 807 – 0
Fax: (49) 771 / 807 – 100
Web: http://www.sick.com
Email Info@sick.de
Version: October 2007

Schedule of revisions

The following information lists the revisions made since the first edition.

New Information (N)

New properties and additional information on existing properties

Revised information (R)

Revisions to the previous edition, which require a different procedure during commissioning.

Inf.	Revisions	Chapter	Rev	Date
	First edition of the document 01		1.00	October 2007

Contents

1 About this Document

➢ Please read this chapter carefully, before working with the documentation and the Hiperface® CANopen Adapter.

1.1 Function of this Document

This manual gives an overview of the Hiperface® CANopen Adapter with CANopen interface and describes configuration, installation, operation and maintenance of the devices in the CANopen network.

1.2 Target Group

This manual is intended for trained specialists who are responsible for installing, assembling and operating the Hiperface® CANopen Adapter and the encoder attached to it in an industrial environment (commissioning staff).

The commissioning staff must have basic knowledge of CAN and CANopen. They must understand CANopen network operations, know how slave devices function in a network and communicate with a bus master. In addition, they must have **basic understanding and experience** of electrical terminology, programming procedures, networks and software as well as sufficient knowledge of the relevant statutory and professional safety regulations of the country in which the devices will be used.

NOTE

Relevant knowledge of bus systems only is **not** sufficient, as applications with CANopen are different.

1.3 Depth of Information

These operating instructions contain information on

- assembly
- electrical installation
- commissioning and configuration
- conformity and approval

of the Hiperface CANopen Adapter.

The official and legal provisions for operating the Hiperface CANopen Adapter must always be complied with.

For commissioning, you should also have the following documents at hand:

- CiA DS 406 V3.1: Device Profile for Encoders
- DiA DS 301 V 4.02: Application Layer and Communication Profile.

NOTE

In addition, use the SICK-STEGMANN Internet homepage on www.sick-stegmann.de.

This is where you will find:

- Sample applications
- These operating instructions in different languages, for display and printing
- Data sheet in pdf format
- EDS file

1.4 Symbols Used

NOTE

Notes inform you of particular features of the device. Please observe these, as they often contain important information.

1. Act...

Instructions to act are numbered, if a certain sequence of actions must be followed.

➢ Act...

Instructions to act are not numbered if no further action follows or if the actionable step is optional. Carefully read and follow instructions to act.

CAUTION

Safety note!

A safety note makes you aware of actual or potential hazards or of incorrect handling of the application. This is to help you prevent accidents.

Carefully read and follow safety notes!

Object description

The symbol points out the objects concerned. The objects are described in chronological order, in *Chapter 9* from *page 48*.

2 About Safety

This chapter is concerned with your safety and the safety of the system operators.

➢ Please read this chapter carefully, before working with the Hiperface CANopen Adapter.

2.1 General Safety Notes

- Read the commissioning instructions prior to commissioning.
- The safety notes and instructions for installation and operation are obligatory.
- Only use the Hiperface® CANopen Adapter in industrial environments.
- SICK-STEGMANN GmbH accepts no liability and no responsibility for direct or indirect damage and consequential damage due to improper handling or incorrect selection of the products.
- Observe the relevant safety and accident prevention regulations: personal injuries or damage to the system may occur.

2.2 Authorised Persons

- All electrical installation and maintenance work shall only be performed by qualified staff trained in electrical engineering.
- The user himself is responsible for selecting the suitable product for the required application.
- Unauthorised persons must not be near the system during installation and maintenance.
- Prevent unauthorised persons from entering, e.g. by using warning signs.

2.3 Notes on Adapter / Encoder

- Observe the appropriate statutory safety guidelines.
- Observe the CANopen specifications and guidelines.
- Do not perform any mechanical or electrical changes to the devices.
- Switch off the adapter, before disconnecting the cable between adapter and encoder.
- Avoid direct UV irradiation over a longer period.
- Pay attention to the type label. It must not become damaged, as it has a sealing function.
- The adapter is operated dependent on other devices. Hence, it is not equipped with direct safety devices.
- Only operate the adapter according to the purpose for which it is designed, and only within a CANopen network.
- Avoid knocking the shaft and collet of the encoders.

2.4 Installation Notes

- During installation, comply with the accident prevention regulations of the professional and trade associations of the respective country.
- Only assemble and maintain the adapter with the electrical voltage switched off.
- Disconnect the voltage for all electrical or electronic devices, machine and system parts and check that they are potential-free.
- Observe correct cabling, earthing, screening and overcurrent protection.
- Check whether the disconnection of devices, machines or system parts causes danger.

2.5 Operational Notes

- Check for correct function of the safety devices (e.g. emergency stop).
- Under no circumstances must work be performed on a powered-up drive.
- After switching off the device: wait for five minutes so that the capacitors can discharge in the intermediate circuit, before working on the drive, motor or motor cable.
- With the mains voltage switched on, dangerously high voltages are applied to the motor cable connection terminals, even when the motor is not in operation.
- Before commencing the work, use a voltage tester to check that the drive is de-energised.
- Only operate the device within the limit values (see *Chapter 11.2* on *page 79*).

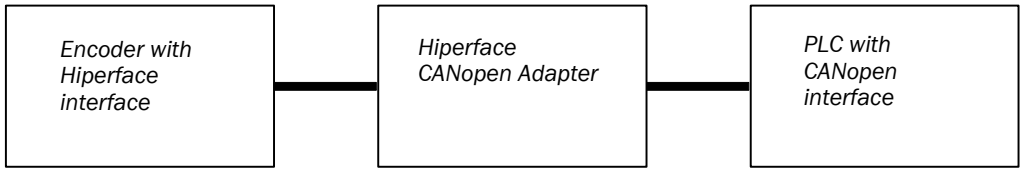
3 Product Description

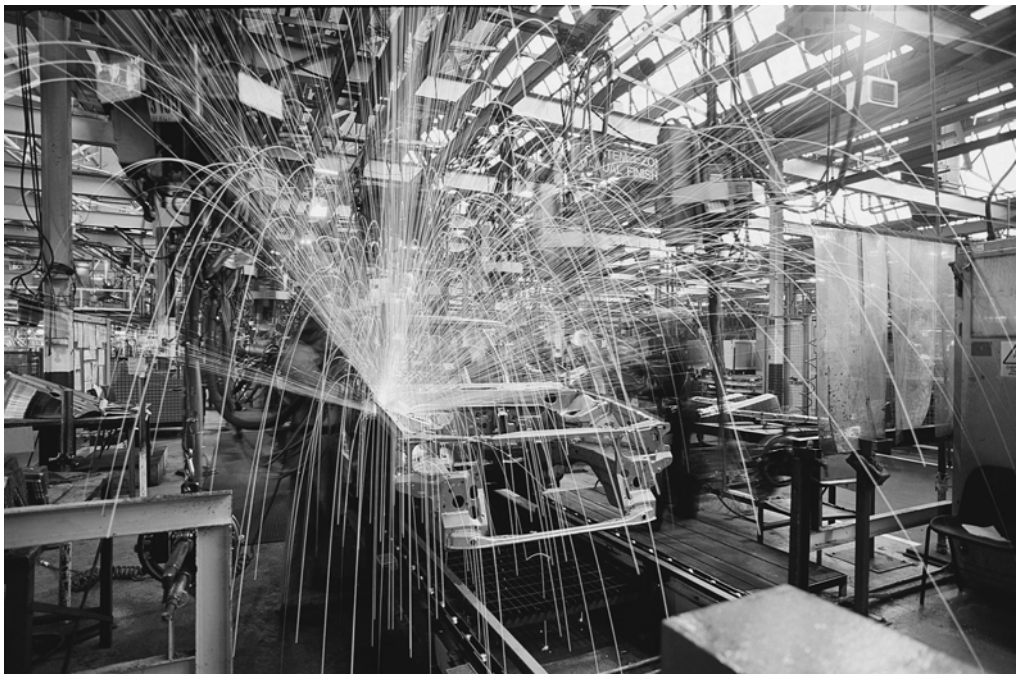
This chapter informs you the particular properties of the Hiperface CANopen Adapter. It describes the construction and the operating principle of the device.

➢ It is imperative that you read this chapter, before assembling, installing and commissioning the device.

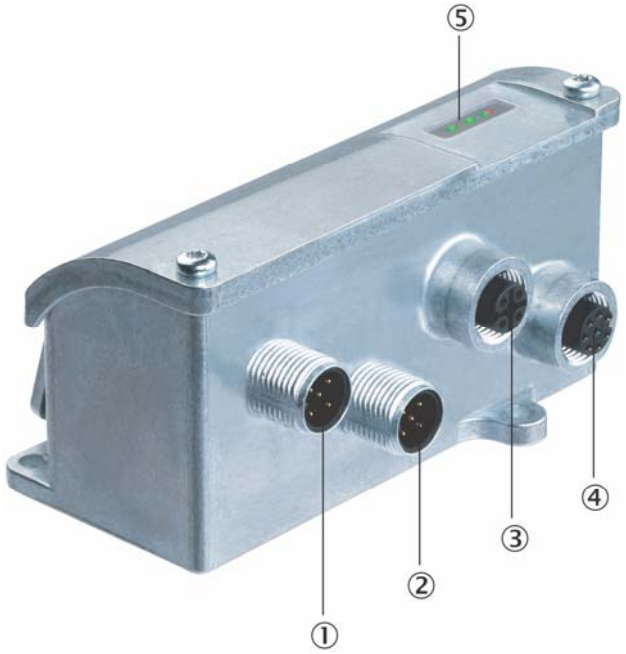
3.1 System Overview

Hiperface® stands for High Performance Interface and is a standard interface for motor feedback systems (encoders) from SICK-STEGMANN GmbH. Hiperface® CANopen Adapters enable the linking of these encoders to controllers with CANopen interface.





3.2 Properties



1 *CAN in (X4)*
2 *Operating voltage (X3)*
3 *CAN out (X2)*
4 *Hiperface in (X1)*
5 *LED displays*

Hiperface® CANopen Adapters are complex interface converters for Hiperface® encoders. They are manufactured to known industrial regulations and meet the quality requirements of ISO 9001.

The adapter communicates with the attached encoder via the digital channel (RS485) of the Hiperface® interface.
Data is transmitted from resp. to the encoder, via corresponding commands. Via the contents of the type label, each encoder defines its properties (linear or rotary, number of measuring steps, total measuring range etc.).

Thus, from the CANopen network's perspective, the adapter represents an encoder with variable basic settings.

During operation, the positions are formed by evaluating the analogue signal.

The special operating voltage of the Hiperface® encoder is also generated in the adapter.

3.2.1 Rotary Encoders

These encoders with Hiperface® interface have a resolution of up to 1024 sine/cosine periods per revolution. In multi-turn encoders, the number of revolutions is determined via a gear mechanism.

Following power-up, the adapter reads the absolute position of the encoder with an encoder-specific basic resolution via the digital channel (RS485) of the interface. The resolution is increased to higher values by the additional evaluation of the analogue signals (sine / cosine).

NOTE

The resolution of the rotary encoders is adjustable. Please also observe the particular measuring range (see *Chapter 11.2.1* on *page 79*).

3.2.2 Linear Encoders

These non-contact encoders (e.g. encoder L230) determine, via a reading head, the absolute position along a material measure (e.g. magnetic tape with sequential code). The reading head itself consists of a number of sensors which capture the absolute position.

Encoders (e.g. encoder XKS09) with wire-draw mechanism belong to the group of linear encoders although, internally, they represent a rotary system. For position determination, the length of the rope (wire) pulled-out is mapped via a drum onto the position value of the rotary system. The diameter of the drum and the number of the windings (single wrap i.e. no over wrapping) thus determine the length of the measurement path to be captured.

Following power-up, the adapter reads the absolute position of the encoder with an encoder-specific basic resolution via the digital channel (RS485) of the interface. The resolution is increased to higher values by the additional evaluation of the analogue signals (sine / cosine).

NOTE

The resolution the linear encoders is adjustable. Please also observe the particular measuring range (see *Chapter 11.2.2* on *page 79*).

3.3 Technical Data

Housing	Die-cast zinc
Weight	400 g approx.
Protection	IP 64
Operating temperature range	-20 to + 60 °C
Storage temperature	-25 to + 85 °C
Perm. rel. humidity	90%, condensation not allowed
Maximum power consumption (without load and encoder)	2.0 W
Operating voltage U_s	10-30 V DC
Encoder operating voltage via adapter	8 V DC +/- 5 % (400 mA max.)
Electrical interface	• ISO 11898, CAN specification 2.0B, galvanically separated. • Hiperface®
Position formation time	< 500 µs
Address	Adjustable via DIP switch (address 1-63)
Baud rate	Adjustable via DIP switch (10, 20, 50, 125, 250, 500, 1000) kBaud, Autobaud
Set function	Adjustable via DIP switch or protocol
Status display	• Operating condition (LED green) • Hiperface® (LED yellow) • CANopen (LED, red/green)
Bus termination	Not integrated (only external on **X2** according to *Connections: Overview* on *page 19*)

3.4 CANopen Specification

Function	Designation
Network Management (NMT)	Slave
Error control	• Node / Life Guarding • Heartbeat
Node identifier (node ID)	Hardware – switch
Process Data Objects (PDO)	2 x Tx 1 x Rx
PDO modes	• Remote • Event-triggered • Time-triggered • Synchronous, cyclic and acyclic
PDO linking	-
PDO mapping	Dynamic
SDO	• 1 x server • 0 x client
Emergency message	Supported
CANopen version	DS 301 V4.02
Device profile	DS 406 V3.1

3.5 Encoder Detection

The Hiperface® CANopen Adapter automatically detects the following encoders with Hiperface® interface:

Rotary encoders	Linear encoders
Designation / encoder series	
SRS	L230
SCK	XKS
SKS	
SEK	
SRM	
SCL	
SKM	

NOTE

Observe the respective measuring ranges of the encoders (see *Chapter 11.2* on *page 79*).

4 Introduction

4.1 Control Area Network (CAN)

CAN is a multi-master system in which all subscribers (master / slave) can independently send data. The sent message can be simultaneously received by all subscribers (broadcast). By means of acceptance filtering, the subscriber only accepts the messages intended for him. The criterion for this decision is the identifier (ID) transmitted with every message. In addition to identifying the messages, the identifier also defines the priority.

CAN possesses a line structure, with twisted pair cables used as the transmission medium. The electrical levels are internationally standardised to ISO 11898 (CAN High Speed). The most outstanding feature of the CAN protocol is its highly secure transmission (hamming distance = 6).

4.2 CAN Application Layer (CAL)

To facilitate the use of CAN in industrial applications, the CAN in Automation (CiA) user organisation has defined a universal application interface with communication and management services for CAN networks; the CAN Application Layer (CAL).

4.3 CANopen

CANopen is based on the serial CAN bus and CAL. The **Communication Profile Draft Standard 301** (DS 301) enables both cyclic and an event-triggered communication. This significantly reduces bus loading and still enables extremely short response times. The data is transmitted via message telegrams.

The telegrams can be separated into the Communication Object Identifier (COB-ID, see *Chapter 6.2 on page 31*) and a maximum of 8 successor bytes.

Further information is available from the CAN in Automation International Users and Manufacturers Group (www.CAN-CiA.de).

4.3.1 Communication Profile DS 301 V4.02

The communication profile of CANopen CiA DS 301 describes which properties of CAL are used and in which form.

CANopen enables, via the communication profile:

- Autoconfiguration of the network
- Comfortable access to all device parameters
- Device synchronisation
- Cyclic and event-triggered process data traffic
- Simultaneous reading or outputting of data

CANopen uses four classes of communication objects (COBs) with different properties:

- Process Data Objects (PDO) for real-time data
- Service Data Objects (SDO) for parameter and program transmission
- Network Management (NMT, Node / Life Guarding, Heartbeat)
- Predefined objects (for synchronisation (SYNC), time stamp, emergency (EMCY))

All parameters are deposited in an object listing (see *Chapter 9* from *page 48*). This object listing contains the description, data type and structure of the parameters as well as the address (index).

4.3.2 Device Profile DS 406 V3.1

The profile defines which CANopen functions are used and how to use them. This standard enables an open and non-manufacturer dependent bus system.

NOTE

DS 406 is the device profile for encoders. This profile is used for the Hiperface® CANopen Adapter.

5 Commissioning

From here on, the Hiperface CANopen Adapter is refered to as **adapter**.

5.1 Assembly

CAUTION

Safety note!

The safety notes and instructions for installation and operation in this document are obligatory.

The adapter must be fitted by a specialist with knowledge of electrics and precision engineering. The adapter must only be used according to the purpose it is designed for.

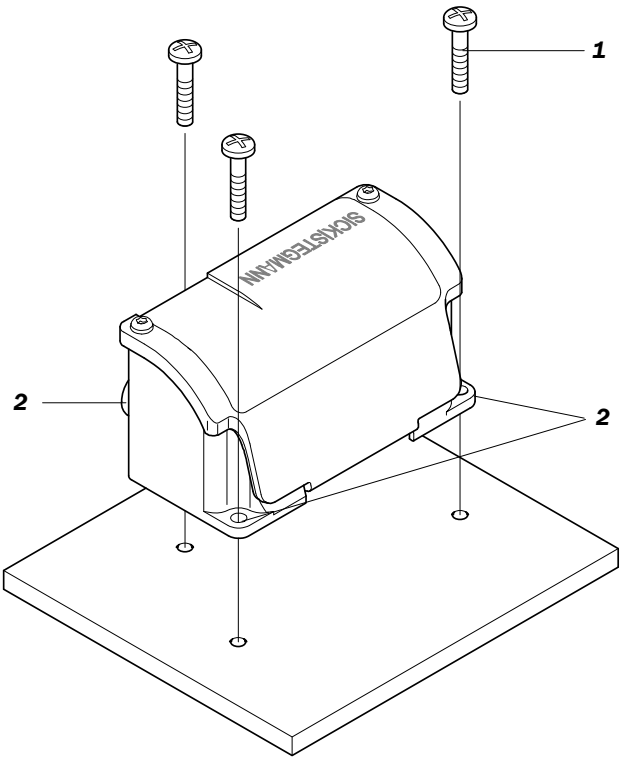
The adapter has 2 x M12 connectors for the integration into a CANopen network. An additional M12 connection is available for the voltage supply.

CAUTION

Safety note!

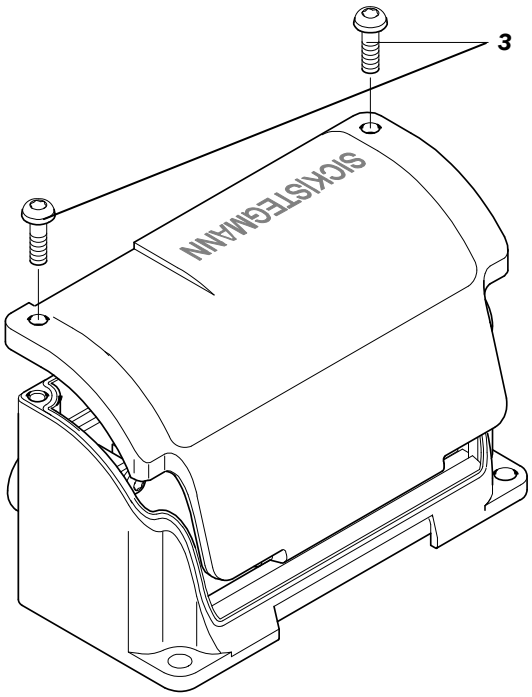
Pay attention to the sealing label in the area of the viewing window for the LEDs. The label must not be damaged!

Assemble the adapter such that it is protected from sunlight!



➢ The assembly is performed via three lugs **(2)** on the housing baseplate using screws **(1)**: 3 x M4 x 10 mm (provided by the customer).

5.1.1 Opening and Closing the Device



Opening

➢ Loosen the screws **(3)** of the housing cover using a Torx screwdriver T x 10 and open the housing cover.

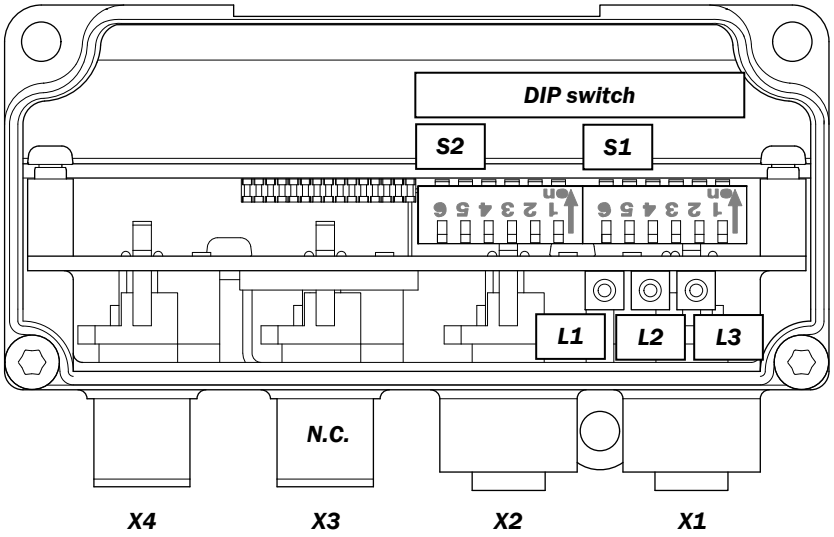
Closing

➢ Tighten the Torx screws **(3)** with a torque of 0.7 to 0.8 Nm.

The specified IP protection is now restored.

5.2 Electrical Installation

5.2.1 Connections: Overview



X1: Hiperface® input
X2: CAN out
X3: Operating voltage
X4: CAN in

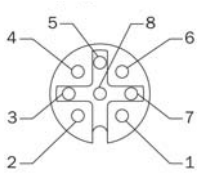
S1: Dip switch 1 (Dip 1 – 6)
S2: Dip switch 2 (Dip 1 – 6)

L1: LED status Hiperface®
L2: LED status network
L3: LED operational status

5.2.2 Pin and Cable Core Assignment

Hiperface® input (X1)

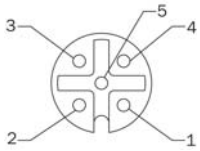
8-way A-coded



PIN	Signal	Explanation
1	REFSIN	Process data channel
2	+ SIN	Process data channel
3	REFCOS	Process data channel
4	+ COS	Process data channel
5	data +	RS485 parameter channel
6	data -	RS485 parameter channel
7	GND	Earthing
8	+ US	Encoder supply voltage

CANopen Out (X2)

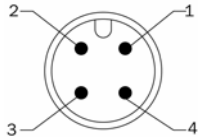
5-way A-coded



PIN	Signal	Explanation
1	Screen	Housing potential
2	V +	Supply voltage via the bus
3	V -	Earthing (GND)
4	CAN-H	H line CANopen
5	CAN-L	L line CANopen

Operating voltage (X3)

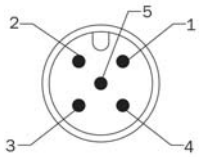
4-way A-coded



PIN	Signal	Explanation
1	Us	Operating voltage
2	n.c.	-
3	GND	Earthing
4	n.c.	-

CANopen In (X4)

5-way A-coded



PIN	Signal	Explanation
1	Screen	Housing potential
2	V +	Supply voltage via the bus
3	V -	Earthing (GND)
4	CAN-H	H line CANopen
5	CAN-L	L line CANopen

5.2.3 Electrical Installation

- Lay the bus cables at as great a distance from the motor cables as possible.
- Avoid parallel laying of motor and bus cables.
- When pulling, grip the cables by the connector. Do not pull on the cable itself.

CAUTION

Safety note!

Only cables according to CAN specification must be used for the network cabling.

Bus termination

The CAN bus loop must be terminated at both ends with a bus termination resistance of 120 Ω between CAN+ and CAN-.

NOTE

An external bus termination resistance (see **X2** according to *Connections: Overview* on *page 19*) must only be connected if the network has not already been terminated twice (at the most distant ends)!

When terminating at the adapter, this subscriber must form **one** end point of the CANopen network.

Screening

Unscreened cable is allowed in interference-free environments.

However, SICK-STEGMANN always recommends that screened cables be used for the bus cable. This recommendation also extends to any required supply cables of external voltage supplies.

Double screened cables are particularly suitable for environments subjected to heavy EMC loading. To ensure optimum protection it is necessary, in this case, to connect the outer braided screen and the inner foil screen at both cable ends by flatening them together with a grounding clamp, onto protective earth.

Adapter earthing

The housing lies, via the electrically earthed metal parts of the machine/system, on earth potential. If the housing is not connected with electrically earthed metal parts, a separate earthing conductor is recommended in order to prevent potential equalisation currents.

Voltage supply

The operating voltage is optionally fed via:

- Data cables (see **X1, X2** according to *Connections: Overview* on *page 19*) or
- Suppy cables (**X3**).

NOTE

If the current is supplied via the bus line, no further line must be connected at the plug of the supply voltage (**X3**). Cover the connection (**X3**), in order to ensure the IP protection!

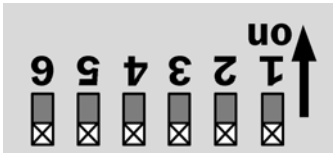
If the current is supplied via the bus line, and if several adapters are connected in series, the entire current is routed via the connectors. The maximum current flow of a connector is limited to 2 A.

5.2.4 DIP Switch Assignment

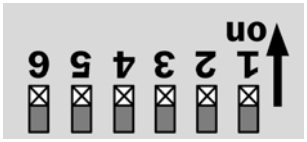
DIP switch 2						**DIP switch 1**					
6	5	4	3	2	1	6	5	4	3	2	1
					Selec-tion	2^5	2^4	2^3	2^2	2^1	2^0
Set	Baud rate				Address						

Default setting

DIP switch 2 **(S2)**



DIP switch 1 **(S1)**



5.2.5 Address (Node ID)

The addresses 1 to 63 can be set.

NOTE

The address 0 is reserved and must not be used by any node!

Address table

decimal	binary	decimal	binary	decimal	binary	decimal	binary
0	000000	16	010000	32	100000	48	110000
1	000001	17	010001	33	100001	49	110001
2	000010	18	010010	34	100010	50	110010
3	000011	19	010011	35	100011	51	110011
4	000100	20	010100	36	100100	52	110100
5	000101	21	010101	37	100101	53	110101
6	000110	22	010110	38	100110	54	110110
7	000111	23	010111	39	100111	55	110111
8	001000	24	011000	40	101000	56	111000
9	001001	25	011001	41	101001	57	111001
10	001010	26	011010	42	101010	58	111010
11	001011	27	011011	43	101011	59	111011
12	001100	28	011100	44	101100	60	111100
13	001101	29	011101	45	101101	61	111101
14	001110	30	011110	46	101110	62	111110
15	001111	31	011111	47	101111	63	111111

Set address with DIP switch

1. Set DIP-1 of DIP switch 2 (see **S2** according to *Connections: Overview* on *page 19*) to OFF.
2. Set the required address via DIP-1 to 6 of DIP switch 1 (**S1**).
3. Switch the supply voltage OFF/ON, to activate a changed in the state of the DIP switches!

Set address via the software

Currently, the address is not configurable via software. This possibility will be available as an option in the future.

5.2.6 Baud rate

Baud rates of 10, 20, 50, 125, 250, 500 kBaud or 1 MBaud can be set. The baud rate must be the same for all subscribers in the CANopen network.

Set baud rate with DIP switch

1. Set the required baud rate via DIP-2 to 5 of DIP switch 2 (see **S2** according to *Connections: Overview* on *page 19*).
2. Switch the supply voltage OFF/ON, to activate a change in the state of the DIP switches!

The following table states which baud rate will be set for the respective DIP switch positions of DIP switch 2. The value zero means OFF, the value 1 means ON.

DIP-5	DIP-4	DIP-3	DIP-2	Data transmission rate
0	0	0	0	10 kBaud
0	0	0	1	20 kBaud (default)
0	0	1	0	50 kBaud
0	0	1	1	125 kBaud
0	1	0	0	250 kBaud
0	1	0	1	500 kBaud
0	1	1	0	1 MBaud
1	1	1	1	Autobaud

Autobaud

When set to Autobaud, a one-off data communication must take place in the network for automatic detection. As a result of this communication, the adapter adjusts its baud rate to the one specified by the network. Hence, at least two further subscribers are required.

Set baud rate via the software

Currently, the baud rate is not configurable via software. This possibility will be available as an option in future.

5.2.7 Display Elements

LED status Hipferface® (L1)	Function
Off	no error, communication ok
On	no / faulty communication data interface, no basic device
Flashing	Hiperface® initialisation

LED status network (L2)	State	Description
Green on	Operational	Device works.
Flickering red/green	Auto bit rate / LSS	The automatic bit rate detection or the LSS service is in progress.
Red single flash	Warning limit reached	At least one error counter of the CAN controller has reached or exceeded the warning level. Too many error frames.
Red double flash	Error Control Event	A Guard event (NMT slave or NMT master) or a Heartbeat event (Heartbeat consumer) occurred.
Red on	Bus off	The CAN controller is in Bus OFF mode.
Green flashing	Pre-Operational	The device is in the Pre-Operational state.
Green single flash	Stopped	The device is in the Stopped state.
Green triple flash	Program / firmware download	A software download runs on the device.
Green on	Operational	The device is in the Operational state.

LED operational status (L3)	Function
On	Switched on
Off	Switched off

5.3 Adapter: Power-on

CAUTION

Safety note!

Follow the instructions below, otherwise there may operational problems.

Observe the safety notes (see C*hapter 2 on page 9*)!

The encoder must be connected to the adapter, before the operating voltage of the adapter is applied.

The following functions must be checked before powering up the encoder:

- Correct electrical installation (see *Chapter 5.2* on *page 19*)
- Correct connection of the encoder to the adapter
- Node ID (see *Chapter 5.2.5* on *page 23*)
- Baud rate (see *Chapter 5.2.6* on *page 24*).

NOTE

When connecting only one node into the CAN network, an error is always signalled due to error frames (see also *Chapter 5.2.7* on *page 25*). When commissioning a single adapter via a Master System (configuration tool, PLC), this subscriber fulfills the function of a further node.

Power-on sequence

Once the adapter has been switched on (Power-ON), the procedure for detecting the attached encoder is automatically initiated. This is signalled by LED 1 flashing (see **L1** according to *Connections: Overview* on *page 19*) (flashing phase can be reduced to briefly lighting up once). After completing this procedure (maximum of 10 sec.), the corresponding status is displayed.

- LED 1 (ON): error Hiperface® interface: (no valid detection of an encoder).
- LED 1 (OFF): no error (valid detection of an encoder).

This procedure is part of the initialisation phase in respect of the CANopen State-Machine with the further functions (see also *Chapter 7.1.1* on *page 38*):

- Reading the data from the EEPROM of the adapter (default values or previously stored values)
- Setting the objects in the object listing to the corresponding values
- Change to the Pre-Operational state. This happens after sending a so-called **Boot-UP message**.

Pre-Operational state:

If an error was detected before or during the initialisation, one or several EMCY message(s) will be sent (see *Chapter 7.4* on *page 43*). LED 2 (see **L2** according to *Connections: Overview* on *page 19*) shows the corresponding CANopen network status.

In this state, the objects in the object listing can be accessed via SDO messages. A PDO transmission is not possible. For this, the adapter must be placed in the Operational state by using a corresponding NMT control command (Start-Node).

Operational state:

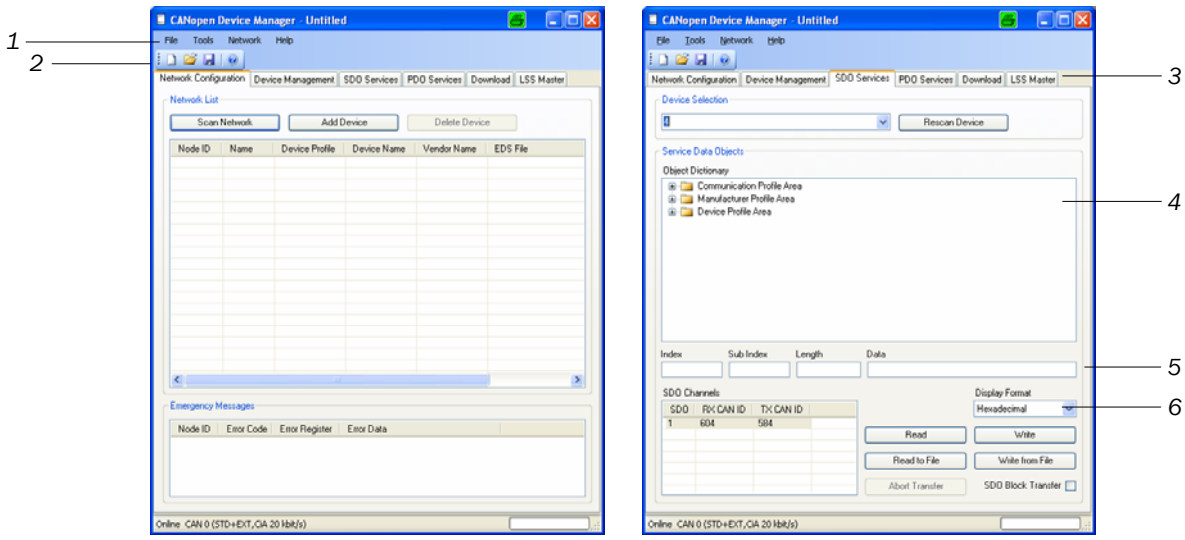
The adapter is now fully operational, and provides the data set via the PDO configuration.

5.4 Software

A PLC or a simple configuration tool can be used for commissioning. In this documentation, the **CANopen Device Manager** software from IXXAT was used. This includes a CAN-based hardware connection layout with the corresponding software for commissioning and configuration of a CANopen network.

For installation and further functions, please refer to the corresponding documentation of the software manufacturer.

5.4.1 Program Interface



1. Menu bar
2. Quick Start bar
3. Register
4. Structural tree overview
5. Data fields column
6. Setting of format

Buttons

- [Read] – The saved data of the encoder is displayed in the data fields column (Pos **(5)** in the program interface overview).
- [Write] – Amended data is transmitted and saved depending on the setting

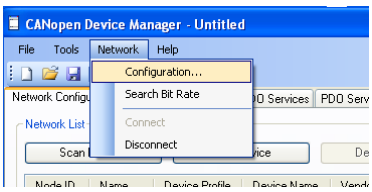
(see Object 2009_{hex} on *page 75* or Object 1010_{hex} on *page 53*).

5.4.2 Scanning the Network

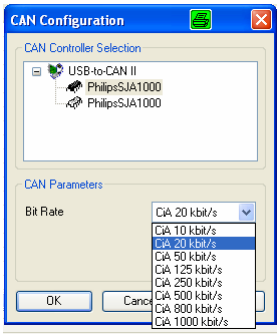
A simple scan process of the network is performed with the configuration tool. If the encoder is detected, it will be displayed as a new device in the network.

✓ The baud rate and address are set via the DIP switches (see *Chapter 5.2.6* and *Chapter 5.2.5* from *page 23*).

1. In the menu bar, select **Network > Configuration...** .

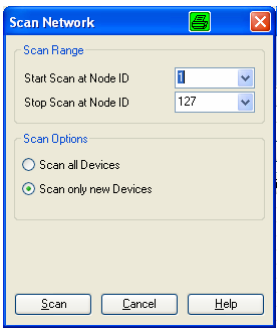


2. In the list, mark the required CAN controller and, via the pull-down menu, select the bit rate set in the network.



3. Confirm with [OK].
4. In the **Network Configuration** register, click on [Scan Network].

   The program dialogue is opened.



5. In the **Scan Range** list field, define the scanning range via the pull-down menus.
6. Activate the option field **Scan only new Devices** and click on [Scan].

   The device/devices will be displayed in the **Network Configuration** register.

   The device can now be configured.

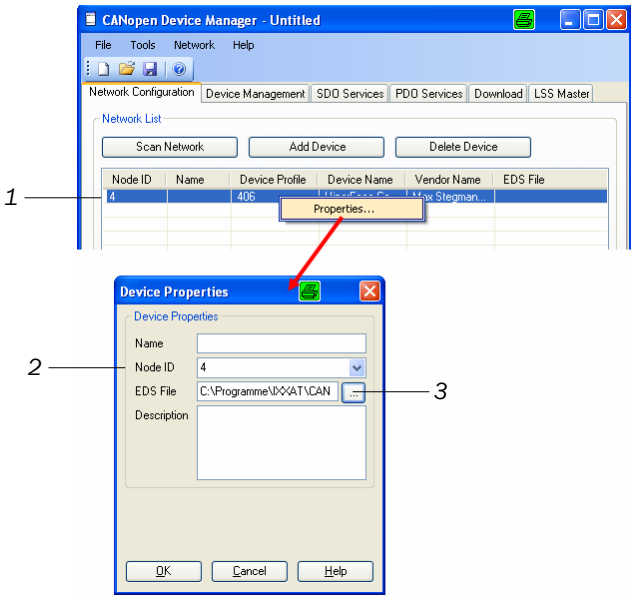
5.4.3 Integrating the EDS File

The EDS file is the electronic description of the object listing of a device, which is required for configuration resp. operation. This will be read in by a configuration tool.

Using the entries in the EDS file, the configuration tool makes all existing objects of this device available to the user. Thus, a device can be easily configured without prior knowledge of its object listing.

The EDS file can be downloaded from www.sick-stegmann.com.

✓ The CANopen Device Manager is opened.



1. Click, with the right-hand mouse button, on the selected node in the **Network List (1)** and select **Properties...** .

   The **Device Properties** dialogue is opened.

2. Select the required node, via the pull-down menu **Node-ID (2)**.
3. Click on [...] (Search File button) **(3)** and select the EDS file. Confirm with [Open].
4. Confirm the setting with [OK].

   The EDS file is loaded, and the object listing in the **SDO Services** register is displayed.

5.5 Preset Function

If the preset function is executed, the position value is updated to the new preset value. This value is always interpreted as 32-bit unsigned (bit 31 - bit 0). The actual data range is reduced to 30 bits.

The preset value and the offset value resulting from updating the position value are automatically saved in the EEPROM (adapter or encoder). You can select the memory location via Object 2006_{hex}.

If an error occurs during saving, an EMCY message is generated. The position value, however, is always updated.

NOTE

The saving takes about 100 ms and runs independently in the background. This does not affect communication to the bus.

If you activate a preset value outside the physical (PMR) or scaled (CMR) measuring range, the preset value is set to the default value (zero) and saved in the EEPROM of the adapter. **There will be no warning or error message!**

Setting the preset value:

- configuration via Object 6003_{hex} or
- evaluation of the data of the Receive PDO 1 Object 1600_{hex} or
- via DIP-6 of DIP switch 2 (see **S2** according to *Connections: Overview* on *page 19*) during transition from OFF to ON. The DIP switch must remain in the ON position for at least 1 second.

NOTE

The preset function is not intended for dynamic adjustment procedures. The function serves the electronic adjustment during commissioning, to allocate a certain position value to any mechanical position.

Due to saving the data in the EEPROM, the number of activations is limited (100,000 cycles)!

CAUTION

Personal injury, damage to the system or other items.

The use of the preset function leads to a change of the position value output by the adapter. This can cause an unexpected movement leading to damage to the system, other items or personal injury.

Object 6003_{hex}: Preset value (see *page 63*).

Object 6509_{hex}: Offset Value (see *page 72*).

Object 2006_{hex}: Memory location: preset/offset value (see *page 74*).

6 Data Transmission

6.1 File Contents and Transmission

The transmission of a message is based on the CAN protocol and is structured as follows:

- 11-bit identifier (COB)
- maximum of 8 bytes of following data.

A fragmentation is performed for Service Data Objects (SDO) for a data packet with a length of over 8 bytes.

For Process Data Objects (PDO), the data length is limited to 8 bytes. This corresponds to the format of a simple CAN message.

Data elements with a length greater than 1 byte are transmitted in the **Little Endian** format, i.e. the **Least Significant Byte** (LSB) will be transmitted first, followed by the **Most Significant Byte** (MSB).

COB-ID	**CAN data range**							
11-bit	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
	unsigned 32				unsigned 16		unsigned 8	unsigned 8
	2^7 ... 2^0	2^{15} ... 2^8	2^{23} ... 2^{16}	2^{31} ... 2^{24}	2^7 ... 2^0	2^{15} ... 2^8	2^7 ... 2^0	2^7 ... 2^0
	LSB			MSB	LSB	MSB		

6.2 Communication Object Identifier (COB-ID)

The Communication Object Identifier (COB-ID) includes the unique allocation of the communication object (message). It comprises of the function code, which considers the different message types, and the address (node ID) uniquely allocated to every encoder. The COB-ID has a length of 11 bits and is required for all types of messages (Emergency, Synchronisation, Service Data and Process Data Objects).

Structure

Bit 10	**Bit 9**	**Bit 8**	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Function code				Node ID (see *Chapter 5.2.5* on *page 23*)						

Predefined Connection Set

For easier management of the identifiers, CANopen uses the Predefined Connection Set, with all identifiers being defined with standard values in the object listing. The higher the value of the COB-ID, the lower its priority.

The identifier allocation is static and depends on the node ID. This enables fast configuration of simple networks.

Object	Function code (binary / hex)		COB-ID (dec / hex)		Reference objects
NMT network object	0000	00	0	00.00	-
SYNC	0001	80	128	00.80	1005_{hex} , 1006_{hex}
Emergency	0001	80	129 – 191	00.81 – 00.BF	1014_{hex}
PDO 1 (tx)	0011	030	385 – 447	01.81 – 01.FF	1800_{hex}, $1A00_{hex}$
PDO 1 (rx)	0100	04	513 – 575	02.00 – 02.3F	1400_{hex}, 1600_{hex}
PDO 2 (tx)	0101	05	641 – 703	02.81 – 02.BF	1801_{hex}, $1A01_{hex}$
SDO (tx)	1011	0B	1409 – 1471	05.81 – 05.BF	-
SDO (rx)	1100	0C	1537 – 1599	06.01 –06.3F	-
NMT error control	1110	0E	1793 – 1855	07.01 – 07.3F	$100C_{hex}$, $100D_{hex}$, 1017_{hex}

Example: Transmit PDO 1 of node with node ID 63

Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	1	1	0	1	1	1	1	1	1
1			B				F			

NOTE

Using a configuration tool enables dynamic ID allocation. The individual COB-IDs can be adapted independently to the node ID and the function code. If you want to allocate a dynamic ID, you must save the changed values in the EEPROM (Object 2006_{hex}).

6.3 Service Data Objects (SDOs)

The Service Data Objects (SDOs) form the communication channel for the transmission of device parameters.

SDOs provide a service for access to the object listing, an SDO client (master) having access to the object listing of the SDO server (slave). The adapter is implemented as an SDO server and has a channel with the following allocation:

- Receive SDO (rx): master → adapter (request)
- Transmit SDO (tx): adapter → master (response).

If the data value to be transmitted consists of a maximum of 4 bytes, there will be a simple communication (expedited protocol). For more than 4 bytes, first a segmented data transfer is initiated via an initialisation sequence (Initiate_seq).

The SDO's COB-ID cannot be modified.

Receive SDO

COB-ID	CAN data range							
11-bit	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
06.00_{hex} + node ID	Cmd	Index		Sub-index	Data			
	$2^7 \dots 2^0$	$2^7 \dots 2^0$	$2^{15} \dots 2^8$	$2^7 \dots 2^0$				

Transmit SDO

COB-ID	CAN data range							
11-bit	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
05.80_{hex} + node ID	Cmd	Index		Sub-index	Data			
	$2^7 \dots 2^0$	$2^7 \dots 2^0$	$2^{15} \dots 2^8$	$2^7 \dots 2^0$				

Command byte

Command	Function	Telegram type	Description
40_{hex}	Initiate Upload Request	Request	Data request to adapter
41_{hex}	Upload Response	Response	Adapter sends data length
43_{hex}	Upload Response	Response	Adapter sends data (4 bytes)
$4B_{hex}$	Upload Response	Response	Adapter sends data (2 bytes)
$4F_{hex}$	Upload Response	Response	Adapter sends data (1 byte)
60_{hex} 70_{hex}	Confirm Upload Response	Confirmation	Confirmation to adapter
21_{hex}	Initiate Download Domain	Request	Master sends data length
23_{hex}	Initiate Download Request	Request	Master sends data (4 bytes)
27_{hex}	Initiate Download Request	Request	Master sends data (3 bytes)
$2B_{hex}$	Initiate Download Request	Request	Master sends data (2 bytes)
$2F_{hex}$	Initiate Download Request	Request	Master sends data (1 byte)
60_{hex} 70_{hex}	Confirm Upload Response	Confirmation	Send confirmation to the master
80_{hex}	Abort	Reply	Adapter sends an error code

Object 1200_{hex}: Server SDO Parameter (see *page 56*).

6.4 Process Data Objects (PDOs)

The Process Data Objects (PDO) serve the highly dynamic exchange of real-time data. This data is transmitted with high priority (low COB-ID) as broadcast messages. The data length of a PDO is limited to 8 bytes.

The adapter supports the the following PDOs:

- 2 x Transmit PDO: slave (adapter) sends data to master (Master Input Data).
- 1 x Receive PDO: slave (adapter) receives data from master (Master Output Data).

Dynamic mapping is supported for the Transmit PDO. Thus, the composition of the data objects within a PDO is freely configurable using SDOs.

For Receive PDO, only a fixed configuration (preset value) is supported. The support of the Receive PDO is manufacturer-specific.

6.4.1 Configuration Transmit PDO

General structure of the communication parameters (Object 1800_{hex}- Object $19FF_{hex}$)

Sub-index	Description	Value range	Function
00_{hex}	Greatest supported sub-index	unsigned 8	Read only
01_{hex}	COB-ID	unsigned 32	Adjustable
02_{hex}	Transmission type	unsigned 8	0 – 240, 252 – 255
03_{hex}	Inhibit time	unsigned 16	100 µsec
04_{hex}	Reserved	-	-
05_{hex}	Event timer	unsigned 16	1 msec

COB-ID - sub-index 01

Bits	31	30	29	28 – 11	10 – 0
	0 / 1	0 / 1	0	00000000000000000	11-bit - identifier

Bit number	Value	Meaning
31 (MSB)	0 1	PDO activated PDO deactivated
30	0 1	RTR access to PDO RTR access blocked
29	0	11-bit ID (CAN 2.0A)
28 – 11	0	0
10 – 0 (LSB)	X	COB-ID

Transmission Type – Sub-Index 02

Value	Mode	Type	Function
0	Synchronous	Acyclic	Data capture with SYNC message. One-off sending of the PDO, in case of data change from the preceding PDO.
1 – 240	Synchronous	Cyclic	Data capture and sending of the PDO with nth SYNC message. n corresponds with the value 1 - 240.
252	Synchronous	Query (RTR)	Data capture with SYNC message. One-off sending of the PDO, in case of request via RTR telegram.
253	Asynchronous	Query (RTR)	Data capture and sending of the PDO, in case of request via RTR telegram.
254	Asynchronous	Cyclic / acyclic	Data capture and sending of the PDO are initiated by the application. Timer- (cyclically) and event- (acyclically) triggered.
255	Asynchronous	Cyclic / acyclic	Data capture and sending of the PDO are initiated by device profiles (encoder). Timer- (cyclically) and event- (acyclically) triggered.

Timer- (cyclically) triggered means that the PDO transmission is triggered by the expiry of the set time (event time). If the value of the event time is set to zero, the functionality is deactivated.

Event- (acyclically) triggered means that either value changes of the individual components within the PDO (Change of State function, see *Chapter 6.4.2* on *page 37*) or the receipt of an RTR telegram will trigger a transmission.

NOTE

In the asynchronous mode, the acyclically controlled transmission is always active, irrespective of the setting of the corresponding event time (sub-index 5).

Object 1800_{hex}: Transmit PDO-1 Communication Parameter (see *page 57*).

Object 1801_{hex}: Transmit PDO-2 Communication Parameter (see *page 58*).

Object 6200_{hex}: Cyclic Timer (see *page 65*).

General structure of the PDO mapping parameters (Object $1A00_{hex}$- Object $1BFF_{hex}$)

Sub-index	Description	Value range	Access	Default value
00_{hex}	Number of mapped application objects in the PDO	unsigned 8	ro	01_{hex}- 08_{hex}
01_{hex} - 08_{hex}	Object 1 - 8	unsigned 32	rw	Dependent on device profile

Considering the objects available for mapping, the entries of sub-index 1-8 can be freely configured. The entries comprise of the object index, the sub-index and the data length.

Via sub-index 0, the number of objects reserved for mapping and, thus, the length of the PDO are defined. If zero is entered, an empty PDO is transmitted. Hence, only the COB-ID is sent.

Object $1A00_{hex}$: Transmit PDO-1 Mapping Parameter (see *page 58*)
Object $1A01_{hex}$: Transmit PDO-2 Mapping Parameter (see *page 58*)

Mappable objects

Objects of the object listing can be allocated, as data elements, to a Transmit PDO. A total of eight objects with 1 byte data length each can be compiled via the PDO mapping table.

If the composition is activated, the total data length of all elements is checked for 8 bytes maximum.

Object | Sub-index		Data length (bytes / bits)	Designation	Entry
6004_{hex}	0	4 / 32	Position value	$60.04.00.20_{hex}$
6030_{hex}	1	2 / 16	Speed	$60.30.01.10_{hex}$
6040_{hex}	1	2 / 16	Acceleration	$60.40.01.10_{hex}$
6300_{hex}	1	1 / 08	Status register of the cams, channel 1	$63.00.01.08_{hex}$
	2	1 / 08	Status register of the cams, channel 2	$63.00.02.08_{hex}$
	3	1 / 08	Status register of the cams, channel 3	$63.00.03.08_{hex}$
	4	1 / 08	Status register of the cams, channel 4	$63.00.04.08_{hex}$
6400_{hex}	1	1 / 08	Status of the working range, channel 1	$64.00.01.08_{hex}$
6503_{hex}	0	2 / 16	Alarms (bit 13, 12, 0)	$65.03.00.10_{hex}$
6505_{hex}	0	2 / 16	Warnings (bit 13, 12, 2, 1, 0)	$65.05.00.10_{hex}$

Changing a PDO configuration

The following sequence applies to changing the parameters (communication resp. mapping) for the objects:

1. Object 1800_{hex}*, sub-index 1 (COB-ID): set bit 31 to 1.
   PDO is deactivated.
2. Object 1800_{hex}*: changing the communication parameters sub-index 1, 2, 3, 5.
3. Object $1A00_{hex}$*: changing the mapping parameters (if required).
   - Object $1A00_{hex}$, sub-index 0: set number of elements to zero.
   - Object $1A00_{hex}$, sub-index 1 - 8: enter elements / objects for mapping.
   - Object $1A00_{hex}$, sub-index 0: update elements for mapping.
4. Object 1800_{hex}, sub-index 1 (COB-ID): set bit 31 to zero.
   PDO is activated.

*Also applies to Object 1801_{hex} resp. Object 1A01.

A PDO configuration should be concluded with step 4, so that the current setting is automatically saved (value change bit 31 from 1 to zero).

Changes to the communication parameters are also implemented without step 1 resp. 4. The change, however, is not automatically saved. Without step 1, changes to the mapping parameters are always rejected.

Step 2 resp. 3 can be swapped.

6.4.2 Change of State (CoS)

Object	Designation	Definition of the change of state
6004_{hex}	Position value	Object $2004_{\mathbf{hex}}$ (Change of State) defines the necessary step width (default value: 2). When setting to values greater than the measuring range, the event-triggered position change can be blocked.
6030_{hex}	Speed	Change by the value 1 (format-dependent)
6040_{hex}	Acceleration	Change by the value 1 (format-dependent)
6300_{hex}	Status register of the cams	Change of a bit (0 / 1)
6400_{hex}	Status of the working range	Change of a bit (0 / 1)
6503_{hex}	Alarms	Change of a bit (0 / 1)
6505_{hex}	Warnings	Change of a bit (0 / 1)

For a PDO record consisting of a combination of several objects (e.g. position, warning, speed), a new message is triggered for each Change of State function of the individual objects.

If triggering is only to occur via a specific object, the Change of State functionality of the respective other objects must be switched off. For the **Position Value** object, this is effected by configuring Object 2004_{hex} to the maximum permitted value $7F.FF.FF.FF_{hex}$.

Object $2004_{\mathbf{hex}}$: Change of State (see *page 74*).

6.4.3 Receive PDO

The functionalities of a Receive PDOs are controlled via:

- Object 1400_{hex} Receive PDO-1 Communication Parameter (see *page 56*).
- Object 1600_{hex} Receive PDO-1 Mapping Parameter (see *page 57*).

Dynamic mapping is not possible.

7 Network Management (NMT)

7.1 Network Management Objects

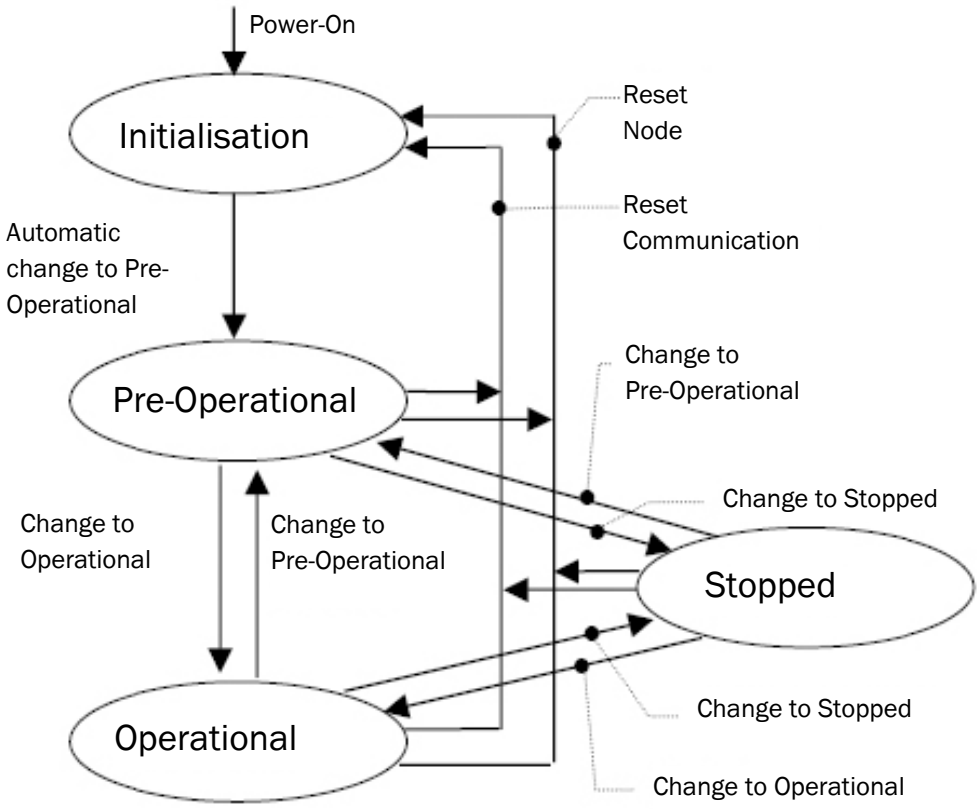
Network Management Objects serve to manage the network.

All NMT commands are transmitted as unconfirmed NMT messages. They can be sent as broadcasts or even as addressed messages.

COB-ID	CAN data range							
11-bit	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
00.00	Cmd	Node ID node						

- Node ID = 0: all subscribers are addressed (broadcast message).
- Node ID = 1 - 63: subscriber with selected node ID is addressed.

7.1.1 State Machine



Initialisation

Initial state after applying the supply voltage. After the initialisation run, the node automatically changes to the Pre-Operational state.

Pre-Operational

The SDO connections are active. The node can now be configured by accessing its object listing.

Operational

The process values are transmitted via the PDO.

Stopped:

The SDO and PDO connections are not active. Via NMT commands, the node can be placed into the Pre-Operational or Operational state.

NMT service	Action	COB-ID	Command
Start Remote Node	Change to Operational state	0	01_{hex}
Stop Remote Node	Change to Stopped state	0	02_{hex}
Enter Pre-Operational	Change to Pre-Operational state	0	80_{hex}
Reset Node	Set all parameters in the complete object listing to values in the on-state	0	81_{hex}
Reset Communication	Set all parameters of the communication profile to values in the on-state	0	82_{hex}

7.1.2 Boot-up Service

The boot-up message signals the transition of an NMT slave from the initialisation state to the Pre-Operational state. The boot-up message uses the COB-ID of the **NMT Error Control Object**.

Master					Slave
Instruction	←	COB-ID: 07.00_{hex} + node ID	Status = 0	←	Request

7.2 Node Guarding (NMT Error Control)

Node monitoring checks each node's ability to communicate. A distinction is made between the Node Guarding protocol and the Heartbeat protocol.

NOTE

Only one form of node monitoring can be used. If the Producer Heartbeat time (Object 1017_{hex}) equals zero, Node Guarding is active. If the Producer Heartbeat time does not equal zero, the Heartbeat Protocol is active.

Node monitoring should be used whenever the slave sends data on the bus at irregular intervals.

Combinations for the settings of the monitoring parameters

Mode	Guard time	Factor	Heartbeat time	Action / Reaction
(1)	x	x	> 0	Heartbeat functionality. Slave cyclically sends a Heartbeat message Master RTR-Guard telegrams are irgnored. Settings in Object $100C_{hex}$ and Object $100D_{hex}$ are not considered.
(2)	0 > 0 0	0 0 > 0	0	• No Heartbeat functionality. • No master monitoring (Life Guarding): If the master sends an RTR-Guard telegram, this will be acknowledged by the slave with the status.
(3)	> 0	> 0	0	• No Heartbeat functionality. • Node / Life Guarding: ON. If the master sends an RTR-Guard telegram, this will be acknowledged by the slave with the status. If the RTR-Guard telegram (Life Event) fails, the slave sends an EMCY message.

Object 1017_{hex}: Producer Heartbeat Time (see *page 55*).

Object $100C_{hex}$: Guard Time (see *page 52*).

Object $100D_{hex}$: Life Time Factor (see *page 52*).

7.2.1 Node / Life Guarding Protocol

For activated Node Guarding, the master monitors its slave in respect of its internal communication status. If, within the life time (Node Life Time), no corresponding status message is received from the slave, then a Node Guarding Event exists.

For activated Life Guarding, the slave simultaneously monitors its master in respect of new Guarding messages. Activation is effected with the first RTR Guard telegram. If, within the Node Life Time, no RTR arrives from the master, then a Life Guarding Event exists.

In response, the slave remains in the current state of the State-Machine (see *Chapter 7.1.1* on *page 38*) and sends an EMCY message (see *Chapter 7.4* from *page 43*).

With the receipt of a renewed Guard telegram, the error state is cancelled, and a further EMCY message is sent for signalling.

The Node / Life Guarding functionality is not activated, if the value of one of the Objects $100C_{hex}$ or $100D_{hex}$ is set to zero.

NOTE

The life time of the slave follows from the multiplication of the Guard Time (Object $100C_{hex}$) and the Life Time Factor (Object $100D_{hex}$).

Master		COB-ID	Byte 0			Slave
Request	→	(RTR) COB-ID: 07.00_{hex} + node ID			→	Instruction
Confirm	←	COB-ID: 07.00_{hex} + node ID	t*	s*	←	Response

* s: state of the NMT slave
t: toggle bit
Byte 0 for t equal 0 / 1:
04_{hex} / 84_{he} Stopped
05_{hex} / 85_{he}: Operational
$7F_{hex}$ / FF_{he}: Pre-Operational

The toggle bit in the Guarding protocol is only reset when a communication reset occurs. No other change of state resets the toggle bit. If a response with the same value of the toggle bit is received, this will be regarded as not received.

Object $100C_{hex}$: Guard Time (see *page 52*).

Object $100D_{hex}$: Life Time Factor (see *page 52*).

7.2.2 Heartbeat Protocol

As an alternative to Node Guarding, the modern Heartbeat protocol should be used today. The protocol is activated if, in Object 1017_{hex}, a value greater than zero is written in. The object defines the time (Heartbeat Producer Time) after which the adapter cyclically sends a Heartbeat message.

One or several Heartbeat consumer(s) can receive this Heartbeat message. If the cyclical sending of the Heartbeat message fails to materialise, a Heartbeat Event is triggered.

If the Heartbeat Producer Time is set in a device, the Heartbeat protocol starts immediately. If a device starts with a value unequal zero for the Heartbeat Producer Time, the Heartbeat protocol begins with the transition from the initialisation state to Pre-Operational. In this case, the boot-up message is regarded as the first Heartbeat message.

Struktur

Producer					Consumer
Request	→	COB-ID: 07.00_{hex} + node ID	Status*	→	Instruction

* Status
00_{he}: Boot-up
04_{he}: Stopped
05_{he}: Operational
$7F_{he}$: Pre-Operational

Object 1017_{hex}: Producer Heartbeat Time (see *page 55*).

7.3 Synchronisation Object (SYNC)

Many real-time applications require synchronisation (SYNC) between several bus nodes.

The CANopen Communication Profile fulfills this request by SYNC telegrams which divide the time axis into equally sized communication cycles.

Within the sending window, directly after receiving a SYNC telegram, the slave (adapter) can be configured to send the current state of its input data via a Transmit PDO. The outputs are activated using the data of the Receive PDO last received.

The COB-ID for the SYNC message is provided by Object 1005_{hex}.

NOTE

The function of the synchronisation of the outputs is not implemented for the adapter.

Object 1005_{hex}: SYNC COB-ID (see *page 51*)

7.4 Emergency Objects (EMCY)

An emergency message (EMCY) is triggered when detecting a faulty situation on the part of the adapter resp. encoder. The message is only triggered once per event. If the error is rectified, a renewed message is sent. The error code for this message either shows **Error Reset** or **No Error**.

Error Reset is displayed if an error condition has been rectified, but with other error conditions remaining.

No Error is displayed if all possible error conditions have been rectified.

The value of the error field is a copy of Object 6503_{hex}.

EMCY specific error code (LSB)		**Error register**	**Error field (copy of Object 6503_{hex}) (MSB)**				
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
EMCY error code		Error register	Error field (EF)				
LSB	MSB		LSB	MSB			

EMCY error code	**Error register**	**Error field (EF)**	**Meaning**
00.00_{hex}	00_{hex}	00.00_{hex}	Error Reset or No Error
10.00_{hex} [1)]	81_{hex}	00.01_{hex}	General position error - wrong analogue signal
10.00_{hex}	81	20.00_{hex}	Start-up position error - no RS485 communication
10.00_{hex}	81	20.01_{hex}	Position error - wrong analogue signal after start-up
10.00_{hex}	81	10.00_{hex}	General EEPROM error - saving the preset data not possible
82.10_{hex} [2)]	11	00.00_{hex}	Invalid Receive PDO - data length invalid
81.30_{hex} [3)]	11	00.00_{hex}	Node / Life Guarding Event

1) When correcting this error condition, there will be another EMCY message with the corresponding code for the errors still outstanding. For all other ones, there is no re-generation of an EMCY message, in case the corresponding error condition has been rectified.
2) Operational state is not exited. No execution of the preset function.
3) Operational state is not exited. In case of renewed receipt of a Life signal from the master, the monitoring is resumed by the slave.

Object 1001_{hex}: Error register (see *page 50*).

Object 1003_{hex}: Error field (see *page 51*).

Object 1014_{hex}: EMCY COB-ID (see *page 55*).

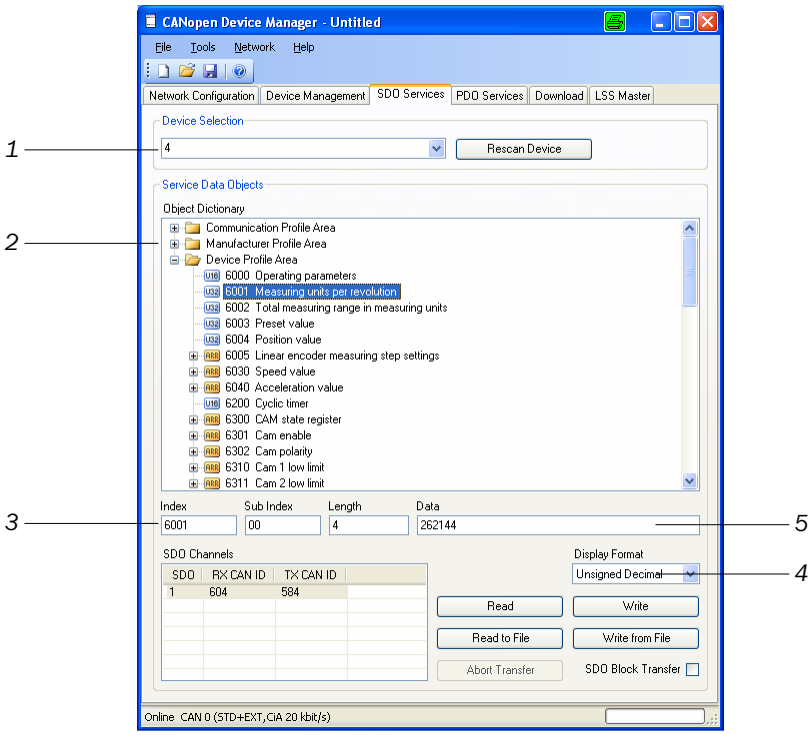
Object 6503_{hex}: Alarms (see *page 70*).

8 Sample Configurations

8.1 Configuration of the Resolution

✓ The CANopen Device Manager is opened, and the required node is scanned (see *Chapter 5.4.2* on *page 28*).

✓ The EDS file is loaded (see *Chapter 5.4.3* on *page 29*).



1. Select the **SDO Services** tab.
2. In the **Device Selection** group field, via the pull-down menu **(1)**, select the required node (node is given with the node ID).
3. In the structure tree **(2)**, open the **Device Profile Area** entry.
4. Mark the entry **6001 Measuring units per revolution** and click on [Read].

   The saved data of the node is displayed in the data fields column **(3)**.
5. In the **Data (5)** column, enter the required value and confirm with [Write].

   The data has been accepted into the node.

NOTE

Watch out for:

- **Display Format (4)**.
- Correct saving

For display in hexadecimal format, the data in the **Data (5)** field is read from right to left, i.e.:

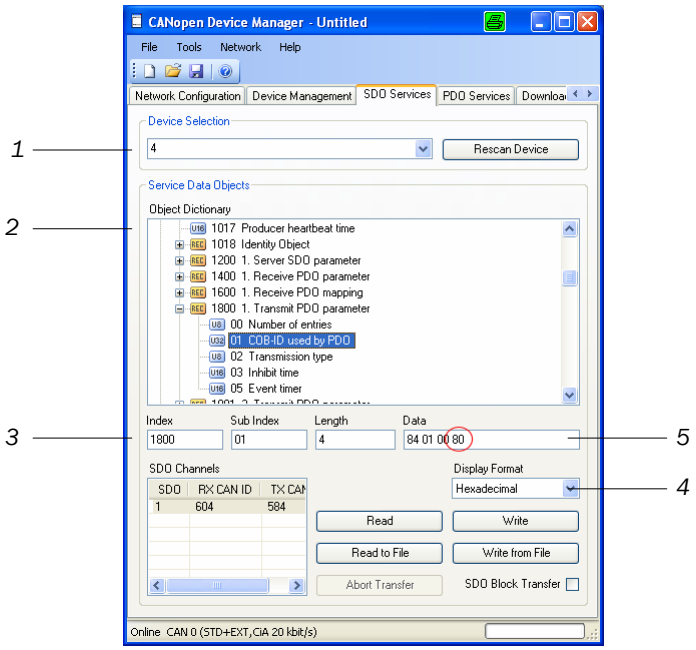
- right (MSB) values
- left (LSB) values

8.2 Changing of PDO Mapping

✓ The CANopen Device Manager is opened and the required encoder is scanned (see *Chapter 5.4.2* on *page 28*).

✓ The EDS file is loaded (see *Chapter 5.4.3* on *page 29*).

Setting bit 31 to 1 – deactivating PDO:

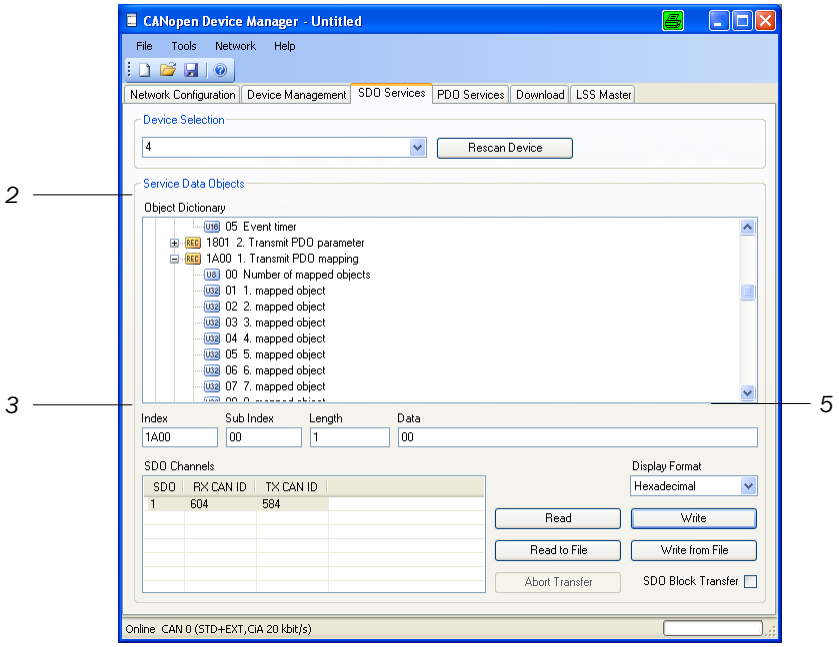


1. Select the **SDO Services** tab.
2. In the **Device Selection** group field, via the pull-down menu **(1)**, select the required encoder (encoder is given with the node ID).
3. In the structure tree **(2)**, open the **Communication Profile Area** entry and then **1800 1. Transmit PDO parameter**.
4. Mark the entry **01 COB-ID used by PDO** and click on [Read].

   The saved data of the encoder is displayed in the data fields column **(3)**.
5. In the **Display Format** pull-down menu, select the **Hexadecimal** setting.
6. In the **Data (5)** column, set the last two digits to 80_{hex} and confirm with [Write].

   The data has been accepted into the encoder. Bit 31 is set to 1.

Deactivating previous objects – setting the number of objects to zero:

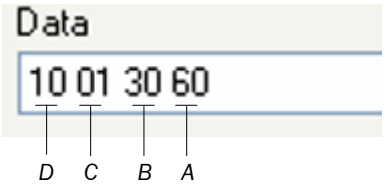


1. In the structure tree **(2)**, open the entry **1A00 1. Transmit PDO mapping**.
2. Mark the entry **00 Number of mapped objects** and click on [Read].

   The saved data of the encoder is displayed in the data fields column **(3)**.
3. In the **Data (5)** column, set the value to 00 and confirm with [Write].

Adding objects:

1. Mark the entry **01 1. mapped object**.



2. Enter the following values into the **Data (5)** column:
   - **(A)**: First two digits of the index
   - **(B)**: Second two digits of the index
   - **(C)**: Sub-index
   - **(D)**: Object length.

Here: Object 6030, sub-index 1, length 10_{hex} resp. 16 bits.

3. Conform with [Write].

   The data has been accepted into the encoder.
4. Mark the entry **0x x. mapped object** and repeat *steps 2* and *3.*

NOTE

Altogether, only 8 bytes can be occupied. An error message appears if the total number of bytes of the objects exceeds these 8 bytes.

Activating objects:

1. Mark the entry **00 Number of mapped objects**.
2. In the **Data (5)** column, set the value to 0x.
   x indicates the number of mapped objects.
3. Confirm with [Write].
   The objects are mapped in the PDO.

Setting bit 31 to 0 – activating PDO:

1. In the structure tree **(2)**, open entry **1800 1. Transmit PDO parameter**.
2. Mark the entry **01 COB-ID used by PDO** and click on [Read].
   The saved data of the encoder is displayed in the data fields column **(3)**.
3. In the **Data (5)** column, set the last digit to 00 and confirm with [Write].
   The data has been accepted into the encoder. Bit 31 is set to 0.

9 Object Listing

All properties and parameters of the encoder are deposited in the object listing.

The description of the object listing entries is structured as follows:

Index	Sub-index	Object	Name	Type	Access

- Index:
  16-bit address of the entry
- Sub-index:
  8-bit sub-entry - if no sub-entry exists, then sub-index = 0.
- Object:
  Var – individual data value (Boolean, unsigned 16 etc.),
  Array – field with same-type data,
  Record – field with any mix of data types.
- Name:
  brief description
- Type:
  Data type (Boolean, 32-bit integer – signed, etc.)
- Access:
  Access rights to the object (rw – write-/read-access, ro – read-only access, const – constant).

In the following chapters, all supported objects are listed and described in chronological order.

9.1 Communication Profile

Index (hex)	Sub-index (hex)	Object	Name / Description	Type	Access
1000	00	Var	Device type	unsigned 32	ro
1001	00	Var	Error register	unsigned 8	ro
1003	00 - 04	Array	Error field	unsigned 32	rw/ro
1005	00	Var	COB-ID for SYNC message	unsigned 32	rw
1007	00	Var	Time window for synchronous PDO	unsigned 32	rw
1008	00	Var	Manufacturer's device name	Visible String	ro
1009	00	Var	Hardware version of the manufacturer	Visible String	ro
100A	00	Var	Software version of the manufacturer	Visible String	ro
100C	00	Var	Guard time	unsigned 16	rw
100D	00	Var	Life Time Factor	unsigned 8	rw
1010	00 – 7F	Array	Save parameters	unsigned 32	rw
1011	00 – 7F	Array	Load default parameters, reset function	unsigned 32	rw
1014	00	Var	COB-ID for EMCY message	unsigned 32	rw
1015	00	Var	Inhibit Time for EMCY message	unsigned 16	rw
1017	00	Var	Producer Heartbeat Time	unsigned 16	rw
1018	00 - 04	Record	Identity	unsigned 32	ro
1200	00 - 03	Record	Server SDO Parameter	SDOParameter	ro
1400	00 - 05	Record	Receive PDO communication parameter	PDOCommPar	rw
1600	00 - 40	Record	Receive PDO mapping	PDOMapping	rw
1800	00 - 05	Record	Transmit PDO 1, communication parameter	PDOCommPar	rw
1A00	00 - 40	Record	Transmit PDO 1, mapping	PDOMapping	rw

Object 1000_{hex}: Device Type

Object code	Type
Var	unsigned 32

Encoder type		Device Profile number	
Byte 3 (bit 31...24)	**Byte 2 (bit 23...16)**	**Byte 1 (bit 15...8)**	**Byte 0 (bit 7...0)**
see comment		01.96_{hex} (406_{dez})	

Comment:

- Rotary single-turn: 00.01_{hex}
- Rotary multi-turn: 00.02_{hex}
- Linear: 00.08_{hex}

Object 1001_{hex}: Error Register

Object code	Type
Var	unsigned 8

Bit	Function
0	General error
4	Communication error (invalid Receive PDO, Guarding Event)
7	Manufacturer-specific error

Via the object, the adapter shows occurring errors. If corresponding bits have been set, then an error has occurred. For a more detailed error analysis, Object 1003_{hex} (predefined error field) or the error code of the EMCY message must be used.

A new entry is always made in connection with sending an EMCY message.

Information on:

- EMCY, see *Chapter 7.4* from *page 43*.
- LED displays, see *Chapter 5.2.7* from *page 25*.

Object 1003_{hex}: Error Field

Object code	Type
Array	unsigned 32

Sub-index	Description	Access	Value range	Default value
00_{hex}	Number of errors	rw	00_{hex} - 04_{hex}	0
01_{hex} - 04_{hex}	Standard error field	ro	unsigned 32	-

If the device finds an error, this is entered into the predefined error field. The field comprises a maximum of four error entries. Each new error is entered under sub-index 1, all other errors are moved back, by one sub-index, in the error field. The last entry under sub-index 4 is then lost.

A new entry always happens in connection with sending an EMCY message.

Delete error list:

➢ Enter the value zero under sub-index 0.

All entries of the error list are deleted.

NOTE

Values unequal zero lead to a cancellation.

Information on EMCY, see *Chapter 7.4* from *page 43*.

Object 1005_{hex}: SYNC COB-ID

Object code	Type
Var	unsigned 32

Bits	31	30	29	28 - 11	10 – 0
	0 / 1	0	0	00000000000000000	11-bit identifier

Bit number	Value	Meaning
31 (MSB)	0 / 1	No meaning
30	0	Device generates no SYNC message

Information on:

- SYNC, see *Chapter 7.3* from *page 42*.
- COB-ID, see *Chapter 6.2* on *page 31*.

Object 1007_{hex}: Synchronous window length

The object defines the length of the time window in µsec for synchronous messages. The value has no function. A PDO is sent as quickly as possible after receiving a SYNC message.

Default value: $00.00.00.00_{hex}$

Object 1008_{hex}: Manufacturer Device Name

The object defines the indication of the device designation. The representation is a String data value.

Indication in ASCII format: Hiperface Comm-Adapter CO.

Object 1009_{hex}: Hardware Device Version

The object defines the indication of the hardware version of the device. The representation is a String data value.

Indication in ASCII format: HW_V1.00.

Object $100A_{hex}$: Software Device Version

The object defines the indication of the software version of the device. The representation is a String data value.

Indication in ASCII format: SW_V1.00.

Object $100C_{hex}$: Guard Time / Object $100D_{hex}$: Life Time Factor

Object code	Type
Var	unsigned 16

Object code	Type
Var	unsigned 8

The Guard Time is output in ms.

The Node / Life Guarding functionality is deactivated when the value of one of the objects is set to zero.

The Life Time Factor multiplied with the Guard Time gives the life time for the monitoring protocol (Node / Life Guarding Protocol).

Default values:

$100C_{hex}$: 00.00_{hex}

$100D_{hex}$: 00_{hex}

Information on Node Guarding, see *Chapter 7.2* from *page 39*.

Object 1010_{hex}: Store Parameters

Object code	Type
Array	unsigned 32

Sub-index	Description	Access	Value range	Duration (msec)
00_{hex}	Greatest supported sub-index	ro	01_{hex} - 04_{hex}	-
01_{hex}	Save all parameters	rw	unsigned 32	1050
02_{hex}	Save communication profile parameters	rw	unsigned 32	350
03_{hex}	Save device profile parameters	rw	unsigned 32	650
04_{hex}	Save manufacturer-specific parameters	rw	unsigned 32	50

Write-access

If the corresponding parameters are to be stored in the EEPROM of the adapter, the **save** command must be written, as a signature, into the associated sub-index. This prevents accidental or incorrect writing to this object from causing a saving of the parameters.

MSB – LSB	Byte 3	Byte 2	Byte 1	Byte 0
ASCII	e	v	a	s
Hex	65	76	61	73

NOTE

If automatic saving is activated via Object 2009_{hex}, saving via this object can be omitted. If saving is effected via the **save** command, the preset and offset values are always saved in the EEPROM of the adapter, irrespective of Object 2006_{hex}.

Read-access

Bit number	Value	Meaning
31 - 2 (MSB)	0	Reserved
1	0	Device does not automatically save data
	1	Device automatically saves data
0	0	Device does not save data on command
	1	Device saves data on command

For the activated auto storage function (Object 2009_{hex}), one receives the value 03_{hex}, otherwise the value 01_{hex}.

Information on:

- Object 2006_{hex}, see *page 74.*
- Object 2009_{hex}, see *page 75.*

Object 1011_{hex}: Restore Default Parameters

Object code	Type
Array	unsigned 32

Sub-index	Description	Access	Value range	Duration (msec)
00_{hex}	Greatest supported sub-index (4)	ro	unsigned 8	
01_{hex}	Sets all parameters to default	rw	unsigned 32	1000
02_{hex}	Sets communication profile parameters to default	rw	unsigned 32	330
03_{hex}	Sets device profile parameters to default	rw	unsigned 32	620
04_{hex}	Sets manufacturer-specific parameters to default	rw	unsigned 32	50

Write-access

By writing the **load** command, either all or individual areas of the parameters are set to their default values. This prevents accidental accidental writing to this object from setting the parameters to default values.

The new values must be activated by the NMT commands Reset Node, Reset Communication or by switching the encoder on/off.

MSB – LSB	Byte 3	Byte 2	Byte 1	Byte 0
ASCII	d	a	o	l
Hex	64	61	6F	6C

Read-access

Bit number	Value	Meaning
31 - 1 (MSB)	0	Reserved
0	1	Restoring the default values is supported

Information on:

- Default parameters, see *page 80*.
- NMT, see *Chapter 7.1* from *page 38*.

Object 1014_{hex}: EMCY COB-ID

Object code	Type
Var	unsigned 32

Bits	31	30	29	28 – 11	10 – 0
	0 / 1	0	0	00000000000000000	COB-ID

Bit number	Value	Meaning
31 (MSB)	0	EMCY functionality activated.
	1	EMCY functionality deactivated.

Information on:

- EMCY, see *Chapter 7.4* from *page 43*.
- COB-ID, see *Chapter 6.2* on *page 31*.

Object 1015_{hex}: Inhibit Time EMCY

Object code	Type
Var	unsigned 16

The object defines the minimum time in multiples of 0.1 ms, which must elapse between two successive EMCY messages.

Default value (function switched off): 00.00_{hex}

Object 1017_{hex}: Producer Heartbeat Time

Object code	Type
Var	unsigned 16

The object defines the time interval, in ms, for sending the Heartbeat telegram. The Heartbeat telegram is a message with the COB-ID of the NMT-Error Control Object and serves node monitoring.

Default value (function switched off): 00.00_{hex}

Object 1018_{hex}: Identity

Object code	Type
Record	unsigned 32

Sub-index	Description	Access	Value (dec / hex))	
00_{hex}	Number of entries	Ro	4	04_{hex}
01_{hex}	Vendor ID	Ro	92	$00.00.00.5C_{hex}$
02_{hex}	Product code	Ro	65	$00.00.00.41_{hex}$
03_{hex}	Revision number	Ro	-	$00.01.00.00_{hex}$
04_{hex}	Serial number, also exists in $650B_{hex}$	Ro	-	$41.xx.xx.xx_{hex}$

MSB – LSB	Byte 3	Byte 2	Byte 1	Byte 0
Revision number	Major number (00.01_{hex})		Minor number (00.00_{hex})	

Object 1200_{hex}: Server SDO Parameter

Object code	Type
Record	SDOParameter

Sub-index	Description	Access	Value range	Default value
00_{hex}	Number of entries	ro	unsigned 8	02_{hex}
01_{hex}	COB-ID Client-Server (rx)	ro	unsigned 32	06.00_{hex} + node ID
02_{hex}	COB-ID Server-Client (tx)	ro	unsigned 32	05.80_{hex} + node ID

The object defines the communication behaviour of the 1st server SDO.
Structure and representation see also description in Communication Profile DS 301.

The data values of the 1st server SDO are not modifiable. This ensures that a device under the default COB-IDs receives and sends data.

Information on:

- SDO, see *Chapter 6.3* from *page 32*.
- COB-ID, see *Chapter 6.2* on *page 31.*
- Node ID, see *Chapter 5.2.5* on *page 23*.

Object 1400_{hex}: Receive PDO-1 Communication Parameter

Object code	Type
Record	PDOCommPar

Sub-index	Description	Access	Value range	Default value
00_{hex}	Greatest sub-index	ro	unsigned 8	02_{hex}
01_{hex}	COB-ID	rw	unsigned 32	02.00_{hex} + node ID
02_{hex}	Transmission type	ro	unsigned 8	255

The object defines the communication behaviour of the 1st Receive PDO.
Structure and representation see also description in Communication Profile DS 301.

Information on:

- PDO, see *Chapter 6.4* from *page 34*.
- Transmission type, see *page 34.*

Object 1600_{hex}: Receive PDO-1 Mapping Parameter

Object code	Type
Record	PDOMapping

Sub-index	Description	Access	Value range	Default value
00_{hex}	Greatest sub-index	ro	unsigned 8	01_{hex}
01_{hex}	Preset value	rw	unsigned 32	$60.03.00.20_{hex}$

The object defines the composition of the data within the 1st Receive PDO.
Structure and representation see also description in Communication Profile DS 301.

NOTE

The preset value is accepted when changing the status of bit 31 within the data value. The data range of the preset value is limited to 30 bits.

Information on the preset function see *Chapter 5.5* from *page 30*.

Object 1800_{hex}: Transmit PDO-1 Communication Parameter

Object code	Type
Record	PDOCommPar

Sub-index	Description	Access	Value range	Default value
00_{hex}	Greatest sub-index	ro	unsigned 8	05_{hex}
01_{hex}	COB-ID	rw	unsigned 32	01.80_{hex} + node ID
02_{hex}	Transmission type	rw	unsigned 8	254
03_{hex}	Inhibit Time	rw	unsigned 16	0
05_{hex}	Event Timer	rw	unsigned 16	0

The object defines the communication behaviour of the 1st Transmit PDO.
Structure and representation see also description in Communication Profile DS-301.

The default configuration means:
Asynchronous mode with acyclic transmission (Event Time = 0), and no inhibit time between the sending of two successive PDOs (Inhibit Time = 0).

Information on:

- PDO, see *Chapter 6.4* from *page 34*.
- Transmission type, see *page 34*.
- Object 1800_{hex}, see *DS 301 V4.02 page 111*.

Object 1801_{hex}: Transmit PDO-2 Communication Parameter

The object defines the communication behaviour of the 2nd Transmit PDO.

The default configuration (see *page 80*) means:
Synchronous mode with cyclic transmission after each receipt of the SYNC message.
Inhibit Time and Event Time have no meaning.

Description see Object 1800_{hex} resp. *Chapter 11.3*

Object $1A00_{hex}$: Transmit PDO-1 Mapping Parameter

Object code	Type
Record	PDOMapping

Sub-index	Description	Access	Value range	Default value
00_{hex}	Number of mapped application objects in the PDO	ro	unsigned 8	01_{hex}
01_{hex}	Object 1 (position value - default)	rw	unsigned 32	$60.04.00.20_{hex}$
02_{hex} - 08_{hex}	Object 2 - 8	rw	unsigned 32	-

The object defines the composition of the data within of the 1st Transmit PDO.
Structure and representation see also description in Communication Profile DS-301.

NOTE

PDO mapping is variable. The maximum length is 8 bytes.
Default setting: position value (4 bytes).

The values of sub-index 1-8 can be freely configured. While retaining the configuration described, 4 bytes remain for the mapping of further objects. However, a configuration without position value can also be set.

The number of objects reserved for mapping, and thus the length of the PDO, are defined via sub-index 0. If the value zero is entered, only the COB-ID will be transmitted.

Information on:

- PDO, see *Chapter 6.4* from *page 34*.
- PDO mapping, see Chapter 8.2 on *page 45.*

Object $1A01_{hex}$: Transmit PDO-2 Mapping Parameter

Object code	Type
Record	PDOMapping

The object defines the composition of the data within the 2nd Transmit PDO.
Description see Object $1A00_{hex}$.

9.2 Device Profile

Index (hex)	Sub-index (hex)	Object	Name	Type	Access
6000	00	Var	Operating parameter	unsigned 16	rw
6001	00	Var	Resolution	unsigned 32	rw
6002	00	Var	Total number of measuring steps	unsigned 32	rw
6003	00	Var	Preset value	unsigned 32	rw
6004	00	Var	Position value	unsigned 32	ro
6005	00 - 04	Array	Measuring steps for linear encoder	unsigned 32	ro
6030	00 - FE	Array	Speed	integer 16	ro
6040	00 - FE	Array	Acceleration	integer 16	ro
6200	00	Var	Cyclic Timer	unsigned 16	rw
6300	00 - FE	Array	Cams, status register	unsigned 8	ro
6301	00 - FE	Array	Cams, release register	unsigned 8	rw
6302	00 - FE	Array	Cams, polarity register	unsigned 8	rw
6310 – 6317	00 - FE	Array	Cams, under range limit	unsigned 32	rw
6320 – 6327	00 - FE	Array	Cams, over range limit	unsigned 32	rw
6330	00 - FE	Array	Cam 1, hysteresis	unsigned 16	rw
6331 – 6337	00 - FE	Array	Cams, hysteresis	unsigned 16	rw
6400	00 - FE	Array	Status: working range	unsigned 8	ro
6401	00 - FE	Array	Working range, lower limit	unsigned 32	rw
6402	00 - FE	Array	Working range, upper limit	unsigned 32	rw
6500	00	Var	Mode of operation	unsigned 16	ro
6501	00	Var	Resolution single-turn (physically)	unsigned 32	ro
6502	00	Var	Number of revolutions (physically)	unsigned 16	ro
6503	00	Var	Alarms	unsigned 16	ro
6504	00	Var	Alarms supported	unsigned 16	ro
6505	00	Var	Warnings	unsigned 16	ro
6506	00	Var	Warnings supported	unsigned 16	ro
6507	00	Var	Profile version and software version	unsigned 32	ro
6508	00	Var	Operating time counter	unsigned 32	ro
6509	00	Var	Offset value	unsigned 32	ro
650A	00 - 03	Array	Module identification	unsigned 32	ro
650B	00	Var	Serial number	unsigned 32	ro

Object 6000_{hex}: Operating Parameter

Object code	Type
Var	unsigned 16

Bit	Function	Bit = 0	Bit = 1
0	Counting direction (rotary encoders)	clockwise (CW) (default)	counter-clockwise (CCW)
2	Application scaling	deactivated	activated (default)
3	Counting direction (linear encoders)	forwards	backwards

The object defines the operating behaviour, through bit selection. The input value comprises the complete number range. However, the adapter only evaluates the bits relevant to the attached encoder. There is no error message for invalid values.

NOTE

If, after activating scaling, the value for the measuring range is smaller than the currently set preset value (Object 6003_{hex}), its value is set to zero and saved in the EEPROM of the adapter.
The parameters Resolution (CPR, Object 6001_{hex}) and Total Number of Steps (CMR, Object 6002_{hex}) are called scaling paramters.

Information on scaling see Object 6001_{hex} and Object 6002_{hex}.

Object 6001_{hex}: Measuring units per revolution (CPR)

Object code	Type
Var	unsigned 32

Encoder type	Resolution	Entry
Rotary encoders	Steps per revolution	Smaller than or equal to the measuring range physically possible.
Linear encoders	Nanometres (nm)	At least the physically representable basic resolution. Greater values mean a lower resolution.

The object defines the required resolution per measuring unit resp. span. The value range (physical measuring range) is dependent on the type of attached encoder (see *Chapter 11.2* on *page 79*). The value input is limited by the specification in the EDS file, irrespective of the physical measuring range.

To activate the object, bit 2 must be set to 1 in Object 6000_{hex} (activating the scaling function), otherwise the representation is made with the unscaled values. However, the input value is always checked against the physical limits.

NOTE

For changes of this object to values smaller than the currently set preset value (Object 6003_{hex}), its value is set to zero and saved in the EEPROM of the adapter.

Object 6002_{hex}: Total measuring range in measuring units (CMR)

The object defines the required total number of measuring steps. Thus, it defines, together with Object 6001_{hex}, the number of revolutions. The value range (physical measuring range) is dependent on the type of attached encoder (see *Chapter 11.2* on *page 79*). The value input is limited by the specification in the EDS file.

To activate the object, bit 2 must be set to 1 in Object 6000_{hex} (activating the scaling function), otherwise the representation is made with unscaled values.

NOTE

For changes of this object to values smaller than the currently set preset value (Object 6003_{hex}), its value is set to zero and saved in the EEPROM of the adapter.

Rotary encoders:
Depending on the attached encoder, the following input instructions must be observed:

- Input smaller than or equal to the physically possible measuring range (PRM)
- Input must be a 2^n multiple of CPR (Object 6001_{hex}), otherwise there will be an automatic adjustment to the next highest value.
- Input must be smaller than or equal to 2^{12} x CPR, otherwise the input value will be rejected. The max. number of revolutions is also limited to 4096.

Sample instruction 2

CPR = 300 steps, CMR = 1x300, 2x300, 4x300, 8x300.

By setting to the value 900 (=3x300), there will be an adjustment to the value 1200 (=4x300).

Sample instruction 3

In case of the following value changes

- CPR from 256 to 8192 and
- CMR from 256 to 2^{10} x 8192,

errors will occur because:

- CPR: CMR/CPR < 1
- CMR: CMR/CPR > 4096.

The values are changed as follows:

1. Change CMR to intermediate value.
2. Change CPR to end value.
3. Change CMR to end value.

Linear encoders:
Parameter has no function. (No write-access.)

Object 6003$_{hex}$: Preset Value

Object code	Type
Var	unsigned 32

Via the object, the current position value is set to the preset value.

The value, and the resulting offset value, are immediately saved in the EEPROM of the device selected via Object 2006$_{hex}$. A separate save via Object 1010$_{hex}$ is not necessary, even with the auto storage function (Object 2009$_{hex}$) switched off. The offset value can be read out via Object 6509$_{hex}$.

CAUTION

Safety note!

For values outside the physical or scaled measuring range, the value is internally set to zero and saved in the EEPROM. There will be no warning or error message!

NOTE

Note that the EEPROM has a write cycle of 100,000 cycles, i.e. more write cycles can cause damage to the memory location.

Through the following actions, the preset and offset value can be set to zero:

- Executing the Restore Default Parameter function
- Changing the resolution (Object 6001$_{hex}$)
- Changing the total number of steps (Object 6002$_{hex}$)
- Changing the setting of the scaling functions (Object 6000$_{hex}$ Bit 2)

Information on preset function see *Chapter 5.5* from *page 30*.

Object 6004$_{hex}$: Position Value

Object code	Type
Var	depending on the encoder type

Via the object, according to the set scaling, the current position is represented as a 32-bit value with steps as the unit.

Rotary encoders:

The display is an unsigned 32 bit value. The maximum position value is dependent upon scaling and equates to CRM – 1.
After a preset procedure with the value zero, the values returned for backward direction of movement are also interpreted as positive numeric values.

Linear encoders:

The display is an integer 32 value (signed) with a value range of +/- 2147483647 (corresponds to a 31-bit data value).

After a preset procedure with the value zero, values returned for backward direction of movement are displayed as negative values.
By toggling the code sequence (CW or CCW), a new display value is genereated.

Object 6005_{hex}: Measuring Steps: Linear Encoders

Object code	Type
Array	unsigned 32

Sub-index	Description	Access	Value range
00_{hex}	Number of objects	Ro	01_{hex} – 04_{hex}
01_{hex}	Position (nm) see also Object $6001_{hex.}$	Rw	unsigned 32
02_{hex}	Factor for representing the speed Format adjustment via Object $2002_{hex.}$	Rw	unsigned 32
03_{hex}	Factor for representing the acceleration Format adjustment via Object $2003_{hex.}$	Rw	unsigned 32

NOTE

Observe the measuring ranges of the attached encoders (see *Chapter 11.2.2* on *page 79).*

Examples:

The representation of the position in 100µm steps corresponds with an input value of 100,000 under sub-index 01_{hex}.

If, in Object 2002_{hex}, [m/s] is set as the speed format, one receives the format [10 m/s] for an input value of 10 under sub-index 02_{hex}.

Object 6030_{hex}: Speed Value

Object code	Type
Array	unsigned 16

Sub-index	Description	Access	Value range	Default value
00_{hex}	Number of available channels	Ro	01_{hex}	-
01_{hex}	Speed value	Rw	unsigned 16	-

Sign	Value range	Condition
+	00.00_{hex} – $7F.FF_{hex}$	Object $6000_{hex.}$= CW + forward Object $6000_{hex.}$= CCW + backward
-	$FF.FF_{hex}$ – 80.00_{hex}	Object $6000_{hex.}$= CW + backward Object $6000_{hex.}$= CCW + forward

The actual representation is a signed value with a maximum indication of +/- 32,768 (15-bit data value).

This value shows, according to the set format, the current speed. The calculation resp. updating of the value happens with a cycle of 50 ms. Several positional values are integrated over a period of 1 sec.

Definition of the format:

- Rotary encoders: Object $2002_{hex.}$
- Linear encoders: Object $2002_{hex.}$ and Object $6005_{hex.}$

Object 6040_{hex}: Acceleration

Object code	Type
Array	unsigned 16

Sub-index	Description	Access	Value range	Default value
00_{hex}	Number of available channels	ro	01_{hex}	-
01_{hex}	Acceleration value	rw	unsigned 16	-

Sign	Value range	Condition
+	00.00_{hex} – $7F.FF_{hex}$	Increasing amount of speed
+	00.00_{hex}	Constant amount of speed
-	$FF.FF_{hex}$ – 80.00_{hex}	Decreasing amount of speed

The actual representation is a signed value with a maximum indication of +/- 32,768 (15-bit data value).

This value shows, according to the set format, the current acceleration. The calculation resp. updating of the value happens with a cycle of 50 ms. Several speed values are integrated over a period of 1 sec.

- Rotary encoders: Object $2003_{hex.}$
- Linear encoders: Object $2003_{hex.\ and}$ Object $6005_{hex.}$

Object 6200_{hex}: Cyclic Timer

Object code	Type
Var	unsigned 16

The object defines the transmission time in milliseconds for a Transmit PDO set to an asynchronous mode. A timer-controlled output becomes active when a cycle time unequal zero is set.

Irrespective of the value, acyclic transmission (e.g. via position change) is always possible.

NOTE

This parameter is closely linked with the Event Timer for Transmit PDO-1. (Object 1800_{hex}). Both objects are based on only one memory cell.

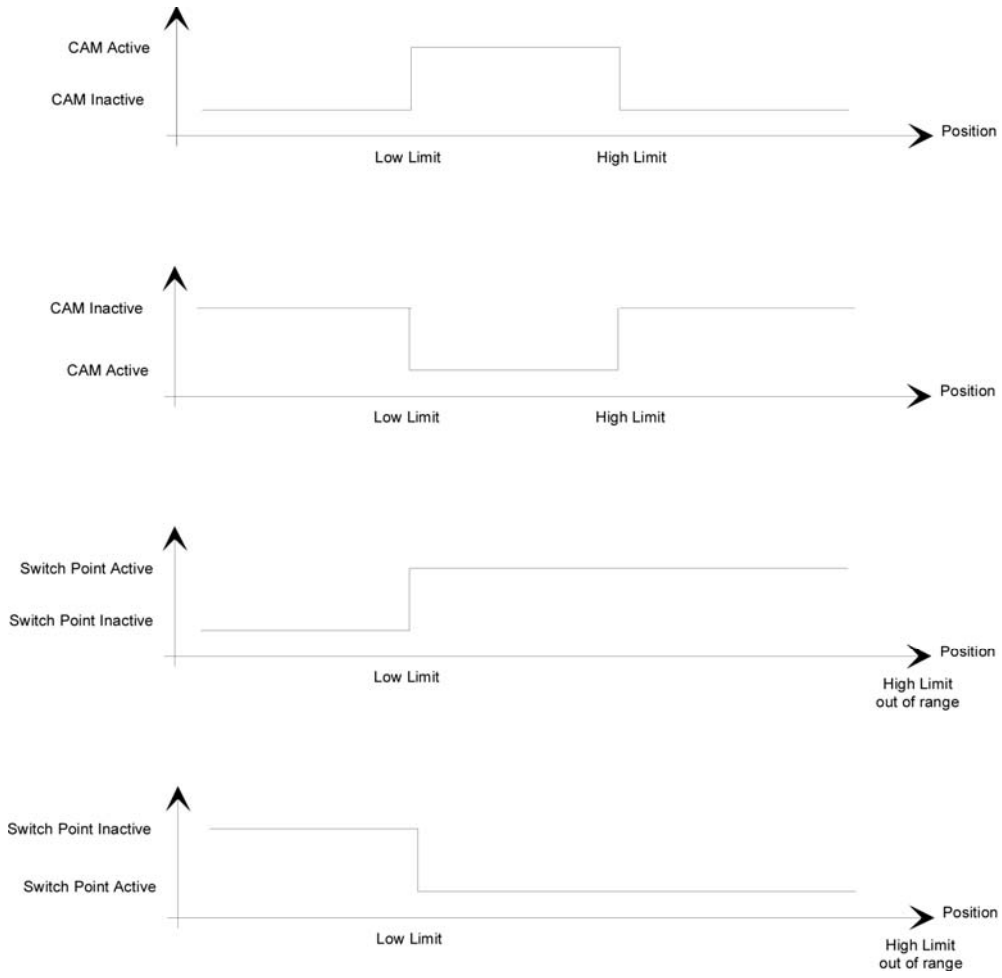
General Function of the Cams

Altogether four channels with eight cams each are supported.

The following representation results for indicating the objects 6300_{hex}, 6301_{hex}, 6302_{hex}, 6310_{hex} - 6317_{hex}, 6320_{hex} - 6327_{hex}, 6330_{hex} - 6337_{hex}:

- Sub-index (0): number of channels.
- Sub-index (1, 2, 3, 4): description of the cams of channel 1, -2, -3, -4.

Graphic representation:



Object 6300_{hex}: Cam State Register

Object code	Type
Array	unsigned 8

Sub-index	Description	Access	Value range	Default value
00_{hex}	Number of channels available	ro	01_{hex} – 04_{hex}	-
01_{hex} - 04_{hex}	Cams status channel 1-4	ro	unsigned 8	-

Example

Value	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Cam 8	Cam 7	Cam 6	Cam 5	Cam 4	Cam 3	Cam 2	Cam 1
01_{hex}	0	0	0	0	0	0	0	1
88_{hex}	1	0	0	0	1	0	0	1

The object displays the status of 8 cams each. If the corresponding polarity in Object 6302_{hex} is set to 0, a value of 1 stands for an active cam, a value of 0 for an inactive cam. Inverting the polarity introduces an inverted meaning.

Object 6301_{hex}: Cam Enable Register

The object defines released resp. inhibited cams. Description, entries and data contents according to Object 6300_{hex}: Cam Status Register. A 1 means a release resp. activation of the functionality for the corresponding cam.

Object 6302_{hex}: Cam Polarity Register

Description, entries and data contents see above: Object 6300_{hex}: Cam Status Register.

Object 6310_{hex} - 6317_{hex}: Cam low limit 1 - 8

Object code	Type
Array	unsigned 32

Sub-index	Description	Access	Value range	Default value
00_{hex}	Number of channels available	ro	01_{hex} – 04_{hex}	-
01_{hex} - 04_{hex}	Cams channel 1 – 4	rw	unsigned 32	-

Object	6317_{hex}	6316_{hex}	6315_{hex}	6314_{hex}	6313_{hex}	6312_{hex}	6311_{hex}	6310_{hex}
	Cam 8	Cam 7	Cam 6	Cam 5	Cam 4	Cam 3	Cam 2	Cam 1

The objects define the lower switching threshold of the individual cams. Under the respective sub-indices, the corresponding channels are represented as unsigned 32 values.

There will be no check of the input in respect of the measuring range supported by the encoder. Moreover, there is no check against the Upper Switching Threshold resp. Hysteresis parameters.

Object 6320_{hex} - 6327_{hex}: Cam high limit1 - 8

The object defines, in the same way as objects 6310_{hex} - 6317_{hex}, the upper monitoring limit of the individual cams.

Object 6330_{hex} - 6337_{hex}: Cam 1 - 8 Hysteresis

Object code	Type
Array	unsigned 16

Sub-index	Description	Access	Value range	Default value
00_{hex}	Number of channels available	ro	01_{hex} – 04_{hex}	-
01_{hex}	Cams channel 1	rw	unsigned 16	-
02_{hex} - 04_{hex}	Cams channel 2 – 4	rw	unsigned 16	-

Object	6037_{hex}	6036_{hex}	6035_{hex}	6034_{hex}	6033_{hex}	6032_{hex}	6031_{hex}	6030_{hex}
	Cam 8	Cam 7	Cam 6	Cam 5	Cam 4	Cam 3	Cam 2	Cam 1

The objects define the hysteresis when a switching threshold is exceeded. Description, entries and data contents see Object 6310_{hex} - 6317_{hex}.

There will be no check of the input in respect of the measuring range supported by the encoder. Moreover, there is no check against the Lower Switching Threshold resp. Upper Switching Threshold parameters.

Object 6400_{hex}: Area State Register

Object code	Type
Array	unsigned 8

Sub-index	Description	Access	Value range	Default value
00_{hex}	Number of channels available	ro	01_{hex}	-
01_{hex}	Working range channel 1	ro	unsigned 8	-

value	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
						Range underflow	Range overflow	Out of range
00_{hex}	0	0	0	0	0	0	0	0
03_{hex}	0	0	0	0	1	0	1	1
05_{hex}	0	0	0	0	1	1	0	1

- 00_{hex} : Position within the limits
- 03_{hex} : Position above the upper limit
- 05_{hex} : Position below the lower limit

Altogether one channel for defining a working range is supported.

With the definition of a working range, further (limit) switch positions can be allocated to the measuring range of the encoder. It thus represents an expansion of the number of cams.

For the definition of a working range, only a Lower (Object 6401_{hex}) and Upper Switching Threshold (Object 6402_{hex}) are defined. There will be no indication of a hysteresis. The limits for the working range are always switched active.

Object 6401_{hex}/ Object 6402_{hex}: Work Area High / LowLimit

Object code	Type
Array	unsigned 32

Sub-index	Description	Access	Value range	Default value
00_{hex}	Number of channels available	ro	01_{hex}	-
01_{hex}	Working range channel 1	rw	unsigned 32	-

There will be no check of the input in respect of the measuring range supported by the encoder. Moreover, there is no check against the respective opposite working range limits.

Object 6500_{hex}: Operatingl Status

Object code	Type
Var	unsigned 16

Bit	Function	Bit = 0	Bit = 1
0	Counting direction	CW	CCW
2	Scaling function	deactivated	activated
3	Direction of measurement (linear encoders)	forward	backward

The data value represents a copy of Object 6000_{hex}.

Object 6501_{hex}: Single Turn Resolution Step (PRS)

Object code	Type
Var	unsigned 32

- Rotary encoders: physical maximum resolution, steps per revolution
- Linear encoders: physical maximum basic resolution in nanometres (nm)

The contents of the object is dependent on the encoder type (see *Chapter 11.2.1 or 11.2.2 from page 79*). The object is read only (diagnostics).

Object 6502_{hex}: Number of distinguishable Revolutions

Object code	Type
Var	unsigned 16

For rotary encoders, the object describes the maximum number of programmable revolutions. The value is dependent on the encoder type (see *Chapter 3.2.1 or 3.2.2 from page 12*). For linear encoders, the parameter has no meaning.

Object 6503_{hex}: Alarms

Object code	Type
Var	unsigned 16

Bit	Function
0	Position error
12	EEPROM error
13	Error during encoder start-up

Via this object, critical error conditions (alarms) of the adapter are displayed in connection with the attached encoder. The possible data value is a depiction of the supported alarms in Object 6504_{hex}. If an error has occurred, the corresponding bit is set to equal 1.

The data value of this object is displayed when triggering an EMCY message (resulting from the error) in the error field area. Furthermore, there is an entry in the error register (Object 1001_{hex}) as a manufacturer-specific error code.

Information on EMCY see *Chapter 7.4* from *page 43*.

Object 6504_{hex}: Alarms Supported

Object code	Type
Var	unsigned 16

Bit	Function
0	Position error
12	EEPROM error
13	Error during encoder start-up

Object 6505_{hex}: Warnings

Object code	Type
Var	unsigned 16

Bit	Function
0	Rotational speed exceeded
1	Excessive LED current in encoder
2	CPU Watchdog Status
12	Maximum acceleration exceeded
13	Temperature exceeded

Warnings signal uncritical errors of the adapter in connection with the attached encoder. If an error has occurred, the corresponding bit is set to equal 1.

The possible data value is a depiction of the supported alarms in Object 6506_{hex}.

Object 6506_{hex}: Warnings Supported

Object code	Type
Var	unsigned 16

Bit	Function
0	Rotational speed exceeded
1	Excessive LED current in encoder
2	CPU Watchdog Status
12	Maximum acceleration exceeded
13	Temperature exceeded

Object 6507_{hex}: Software Version

Object code	Type
Var	unsigned 32

This object contains the device software version (2 bytes) and the version of the implemented encoder profile (2 bytes).

Object 6508_{hex}: Operating Time

Object code	Type
Var	unsigned 32

The indication of time is not supported. Thus, the value $FF.FF.FF.FF_{hex}$ is displayed.

Object 6509_{hex}: Offset Value

Object code	Type
Var	unsigned 32

This object contains the internal offset. The value is recalculated for each execution of a preset function and saved directly in the EEPROM of the device (adapter or encoder) set via Object 2006_{hex}. The value is calculated from the current position value and the preset value.

Object $650A_{hex}$: Module Identification

Object code	Type
Array	integer 32

Sub-index	Description	Access	Value range	Default value
00_{hex}	Number of entries	ro	01_{hex} – 03_{hex}	03_{hex}
01_{hex}	Offset	ro	integer 32	00_{hex}
02_{hex}	Minimum position value	ro	integer 32	00_{hex}
03_{hex}	Maximum position value	ro	integer 32	See Chapter 11.2

Object $650B_{hex}$: Serial Number

Object code	Type
Array	unsigned 32

32-bit number			
Byte 3	Byte 2	Byte 1	Byte 0
Bit 31 - 24	Bit 23 - 17	Bit 16 - 11	Bit 10 - 0
Device code (65)	Year (0 –99)	Week (1 – 52)	Consecutive number (0 – 2047)

This object contains the serial number of the adapter. The parameter is “hard-linked” with Object 1018_{hex} sub-index 4. Both contain the same information. The device code is an internal definition. The 32-bit number differs from the actual serial number on the type label.

Example: $41.07.18.01_{hex}$ = year 2007, calendar week 3, number 1.

9.3 Manufacturer-Specific Profile

Index (hex)	Sub-index (hex)	Object	Name	Type	Access
2002	00 - 05	Var	Format for speed	unsigned 8	rw
2003	00 - 05	Var	Format for acceleration	unsigned 8	rw
2004	00	Var	Change of State	unsigned 32	ro
2006	00	Var	Preset/Offset memory location	unsigned 8	rw
2007	00		maximum Speed limit	unsigned 16	ro
2008	00		maximum Acceleration limit	unsigned 16	ro
2009	00		Automatic saving in the EEPROM	unsigned 8	

Object 2002_{hex}: Format for Speed

Object code	Type
Var	unsigned 8

Sub-Index	Description
00_{hex}	Number/s (cps)
01_{hex}	Revolutions/s (rps)
02_{hex}	Revolutions/min (rpm)
03_{hex}	Metres/s (linear encoders)
04_{hex}	Inches/s (linear encoders)
05_{hex}	Feet/s (linear encoders)

The set default format **cps** is only suitable under certain conditions. Even with medium speed of the attached encoder, in most cases there will be an overflow onto Object 2030_{hex}, since the value range (unsigned 16) is exceeded due to the high resolution. There is a permanent or cyclic warning message (speed error).

Example – Encoder: SKM36

- For deactivated scaling, there follows a resolution of 80.00_{hex} (2^{15}) counts / revolution.
- Medium speed (3,000 rpm) gives 50 x 80.00_{hex} = $00.19.00.00_{hex}$. This value cannot be represented as unsigned 16.

Object 2003_{hex}: Format for Acceleration

Object code	Type
Var	unsigned 8

Sub-index	Description
00_{hex}	Number/s² (cps/s)
01_{hex}	Revolutions/s² (rps/s)
02_{hex}	Revolutions/min/s (rpm/s)
03_{hex}	Metres/s² (linear encoders)
04_{hex}	Inches/s² (linear encoders)
05_{hex}	Feet/s² (linear encoders)

The default setting of the format – **cps/s** – is,in the same way as for the speed format cps, only suitable under certain operating conditions.

Object 2004_{hex}: Change of State

Object code	Type
Var	unsigned 32

The object indicates the size of the position value change to trigger the sending of a PDO. For this, the transmission type (see *page 34*) of the PDO must be set to 254.

Setting to asynchronous means: the transmission type (sub-index = 2) of the corresponding PDO communication object must contain the value 254 or 255.

When setting a high value (greater than measuring range), the triggering of a PDO message due to a position change can be inhibited.
The parameter is not used for synchronous transmission.

Object 2006_{hex}: Memory Location: Preset / Offset Value

Object code	Type
Var	unsigned 8

Via the object, a definition is made as to the location where the preset and offset values are to be saved.

- Bit = 0: Adapter (default)
- Bit = 1: Encoder.

Information on preset function see *Chapter 5.5* from *page 30.*

Object 2007_{hex}: Maximum Permitted Speed

Object code	Type
Var	unsigned 16

The value range lies between 0 and 32,767. This value must always be adapted to the corresponding format of the speed from Object 2002_{hex}.

The current value of the speed of Object 6030_{hex} is also interpreted as a signed value. The value is negative if the counting direction or the movement direction changes. The check against the set limit is made using the amount of the speed.

Object 2008_{hex}: Maximum Permitted Acceleration

Object code	Type
Var	unsigned 16

The value range lies between 0 and 32,767. This value must always be adapted to the corresponding format of the acceleration from Object 2003_{hex}.

The current value of the acceleration Object 6040_{hex} is also interpreted as a signed value. The value becomes negative when the speed value is decreasing.

The check against the set limit is made using the value of the acceleration.

Object 2009_{hex}: Automatic Saving

Object code	Type
Var	unsigned 8

Via this object, it is determined whether or not the parameter is to be automatically stored in the EEPROM of the adapter.

- Bit = 0: activated (default). This means that, when changing a parameter using an SDO write procedure, the new value is immediately written to the EEPROM. Selective saving via Object 1010_{hex} is thus not necessary! The functionality applies to the objects of all data areas: Communication Profile, Encoder Profile and Manufacturer-Specific Profile.
- Bit = 1: deactivated.

10 Error Description

10.1 Hiperface® Communication

Communication by means of the Hiperface® protocol, via the RS485 interface, between adapter and encoder takes place:

- after switching the adapter on
- after reading out the relevant data from the encoder. The CANopen communication from adapter to master is started only after completing the communication sequence (successfully or with error). An EMCY message is generated in case of a prior faulty communication (adapter to encoder).
- to save the preset and offset data if the encoder setting is selected in Object 2006_{hex}. An EMCY message is generated in case of faulty communication. The preset value, however, will be accepted and displayed as a new position value.

10.2 PDO Transmission for Faulty Position (Test Scenario)

- Powering up the adapter with correct link to the encoder (no EMCY message)
- Transition to Operational (Start Node) state. PDO transmission corresponding to the configuration (4-byte position)
- Simulation of a wrong position (analogue signals are faulty). There is an EMCY message. The LED 3 (see **L3** according to *Connections: Overview page 19*) lights yellow. The PDO transmission is continued.
- Then there is a Reset Node command. The PDO transmission is stopped.
- Then there is a Start Node command. The PDO transmission is continued.
- Correction of the faulty state (analogue signals correct). A new EMCY message (Reset error) is generated. The LED extinguishes.

NOTE

The position value is no longer consistent. There is no query of the absolute position via the RS485 interface from adapter to encoder.

Solution: switch adapter, with encoder connected, off / on !!!

10.3 SDO Error Codes.

Fehler Code (hex)	Format	Beschreibung
05.04.00.00	unsigned 32	SDO protocol timed out.
06.01.00.00	unsigned 32	Unsupported access to object.
06.01.00.01	unsigned 32	Attempt to read a 'Write Only' object.
06.01.00.02	unsigned 32	Attempt to write a 'Read Only' object.
06.02.00.00	unsigned 32	Object does not exist.
06.04.00.41	unsigned 32	Object cannot be mapped to PDO.
06.04.00.42		The number and length of the objects to be mapped would exceed PDO length.
06.04.00.43	unsigned 32	General parameter oncompatibility reason.
06.07.00.10	unsigned 32	Data type does not match.
06.07.00.12	unsigned 32	Data type does not match. Length of parameter too high.
06.07.00.13	unsigned 32	Data type does not match. Length of parameter too low.
06.09.00.11	unsigned 32	Sub-Index does not exist.
06.09.00.30	unsigned 32	Invalid value for parameter.
06.09.00.31	unsigned 32	Value of parameter written too high.
06.09.00.32	unsigned 32	Value of parameter written too low.
08.00.00.20	unsigned 32	Data cannot be transferred or stored to application. Write service: Object [1010]h, [1011]h

11 Appendix

11.1 Conversion Table

dec	binary	hex	dec	Binary	hex	dec	binary	hex
0	0000 0000	00	22	0001 0110	16	44	0010 1100	2C
1	0000 0001	01	23	0001 0111	17	45	0010 1101	2D
2	0000 0010	02	24	0001 1000	18	46	0010 1110	2E
3	0000 0011	03	25	0001 1001	19	47	0010 1111	2F
4	0000 0100	04	26	0001 1010	1A	48	0100 0000	30
5	0000 0101	04	27	0001 1011	1B	49	0100 0001	31
6	0000 0110	06	28	0001 1100	1C	50	0100 0010	32
7	0000 0111	07	29	0001 1101	1D	51	0100 0011	33
8	0000 1000	08	30	0001 1110	1E	52	0100 0100	34
9	0000 1001	09	31	0001 1111	1F	53	0100 0101	34
10	0000 1010	0A	32	0010 0000	20	54	0100 0110	36
11	0000 1011	0B	33	0010 0001	21	55	0100 0111	37
12	0000 1100	0C	34	0010 0010	22	56	0100 1000	38
13	0000 1101	0D	35	0010 0011	23	57	0100 1001	39
14	0000 1110	0E	36	0010 0100	24	58	0100 1010	3A
15	0000 1111	0F	37	0010 0101	25	59	0100 1011	3B
16	0001 0000	10	38	0010 0110	26	60	0100 1100	3C
17	0001 0001	11	39	0010 0111	27	61	0100 1101	3D
18	0001 0010	12	40	0010 1000	28	62	0100 1110	3E
19	0001 0011	13	41	0010 1001	29	63	0100 1111	3F
20	0001 0100	14	42	0010 1010	2A			
21	0001 0101	15	43	0010 1011	2B			

11.2 Encoder Measuring Ranges

11.2.1 Rotary Encoders

Encoder type	Periods / revolution	PRS resolution		PMR measuring range	
SEK52	16	12-bit	1 – 4096	12-bit	1 - 4096
SKS36	128	15-bit	1 – 32,768	15-bit	1 – 32,768
SKM36	128	15-bit	1 – 32,768	27-bit	1 – 134,217,728
SCS-xx	512	17-bit	1 – 131,072	17-bit	1 – 131,072
SCM-xx	512	17-bit	1 – 131,072	29-bit	1 – 536,870,912
SRS- /SCK	1024	18-bit	1 – 262,144	18-bit	1 – 262,144
SRM- /SCL	1024	18-bit	1 – 262,144	30-bit	1 – 1,073,741,824

11.2.2 Linear Encoders

Encoder type	Period / length [mm]	Max. resolution PRS [nm]	Max. measuring length [nm]
XKS09 – 5m	1.195	4,669	$5 * 10^9$
XKS09 – 2m	1.195	4,669	$2 * 10^9$
L230 – LinCoder	5.000	19,531	$40 * 10^9$

Default, Min. / Max. Values for Shipment

Designation	Setting	Switch	Software
Baud rate	20 kBaud	Dip 5 (S2) to 1 (ON)	not supported
Address	63	Dip 1 – 6 (S1) to 1 (ON), Dip 7 (S2) to 0	not supported

Object (hex)	Designation	Minimum (hex)	Maximum (hex)	Default value
1005	COB-ID for SYNC message	-	-	80_{hex}
1007	Time window for synchronous PDO	0 - 7F.FF.FF.FF		0
100C	Monitoring Time	0	FF.FF	0
100D	Life Time Factor	0	FF	0
1010	Save parameters			0
1011	Load default parameters			0
1014	COB-ID for EMCY message			00.80_{hex} + node ID
1015	Inhibit time for EMCY message	0	FF.FF	0
1017	Producer Heartbeat Time	0	FF.FF	0
1800	Transmit PDO 1, communication parameters • COB-ID • Transmission Type • Inhibit Time • Event Time			• 01.80_{hex} + node ID • 254 • 0 • 0
1801	Transmit PDO 2, communication parameters • COB-ID • Transmission Type • Inhibit Time • Event Time			• 02.80_{hex} + node ID • 1 • 0 • 0
1A00	Transmit PDO 1, Mapping 1st object			60040020_{hex}
1A01	Transmit PDO 2, Mapping			see Object $1A00_{hex}$

Object (hex)	Designation	Minimum (hex)	Maximum (hex)	Default value
6000	Operating parameters	0	FF.FF	04_{hex}
6001	Resolution	1	≤ PRS	80.00_{hex} 32,768 (dec)
6002	Total number of measuring steps	1	≤ PMR	$40.00.00.00_{hex}$
6003	Preset value	0	≤ CMR, PMR	0
6004	Position value	$80.00.00.00_{hex}$	$7F.FF.FF.FF_{hex}$	-
6005	Position value	1	$40.00.00.00_{hex}$ $7F.FF.FF.FF_{hex}$	1
6030	Speed	80.00	7F.FF	1
6040	Acceleration	80.00	7F.FF	1
6200	Time cycle	0	FF.FF	0
6301	Cams, release register			0
6302	Cams, polarity register			0
6310 .. 6317	Cams, under range limit	$80.00.00.00_{hex}$	$7F.FF.FF.FF_{hex}$	0
6320 .. 6327	Cams, over range limit	$80.00.00.00_{hex}$	$7F.FF.FF.FF_{hex}$	$7F.FF.FF.FF_{hex}$
6330 .. 6337	Cams, hysteresis	0	FF.FF	0
6401	Working range, lower limit	$80.00.00.00_{hex}$	$7F.FF.FF.FF_{hex}$	0
6402	Working range, upper limit	$80.00.00.00_{hex}$	$7F.FF.FF.FF_{hex}$	$7F.FF.FF.FF_{hex}$

Object (hex)	Designation	Default value
2002	Format for speed	0
2003	Format for acceleration	0
2004	Change of State	2
2005	Analog monitoring	1 (activated)
2006	Location preset / offset	0 (adapter)
2007	Maximum speed	6000
2008	Maximum acceleration	$7F.FF_{hex}$
2009	Automatic saving in the EEPROM	0 (activated)

Object 1011_{hex}: Load Default Parameters (see *page 53*).

11.3 Listing of Abbreviations

Abbreviation	Description
CAN_H	CAN High
CAN_L	CAN Low
CCW	Counter clockwise. Rotation against the clock (looking at shaft).
Cmd	Command
CMR	Total number of steps over the entire measuring range (customer configurable).
CMS	CAN Message Specification. Service element of CAL
CoS	Change of State
CPR	Steps per revolution (for rotary encoders).
CPS	Encoder speed: steps/s.
CW	Clockwise. Rotation with the clock (looking at shaft)
DBT	Distributor. One of the service elements of the CAN application layer in the CAN reference model. The Distributor distributes the COB-IDs to the COB, which are used by a CMS.
DS	Draft Standard
EDS	Electronic Data Sheet. A node-specific ASCII file required for the configuration of the CAN network. The EDS file contains general information about the node and its list of objects (parameters).
EEPROM	Non-volatile memory
EMC	Electromagnetic compatibility
GND	Earthing
LMT	Layer management. One of the service elements of the CAN application layer in the CAN nominal value model. This is where parameters for the individual layers of the CAN reference model are configured.
LSS	Layer Setting Services
PMR	Physical Measuring Range: total number of steps over all revolutions resp. over the entire span (defined by the manufacturer).
RAM	Volatile memory
ro	Indicates read-only access.
RPM	Rotational speed: rev./min.
RPS	Rotational speed: rev/s.
RTR	Remote Transmission Request. data request telegram
rw	Indicates read-/write-access.

11.4 Data Type Specification

Data type	Description	Value
BOOL	Boolean	1 bit
BYTE	Bit sequence	1 byte (8 bits)
WORD	Bit sequence	2 bytes (16 bits)
Var	Individual data value	
Array	Data field with values of the same type	
Record	Field with mixed data value types	
String	Chain of characters	
integer 32	32-bit integer, signed	-2^{31} ... + (2^{31} – 1)
integer 16	16-bit integer, signed	-2^{15} ... + (2^{15} – 1)
integer 8	8-bit integer, signed	-2^{7} ... + (2^{7} – 1)
unsigned 32	32-bit integer, unsigned	0 ... (2^{32} – 1)
unsigned 16	16-bit integer, unsigned	0 ... (2^{16} – 1)
unsigned 8	8-bit integer, unsigned	0 ... (2^{8} – 1)

Australia
Phone +61 3 9497 4100
1800 334 802 – tollfree
E-Mail sales@sick.com.au

Belgium/Luxembourg
Phone +32 (0)2 466 55 66
E-Mail info@sick.be

Brasil
Phone +55 11 3215-4900
E-Mail sac@sick.com.br

Canada
Phone +1(952) 941-6780
1 800-325-7425 – tollfree
E-Mail info@sickusa.com

Ceská Republika
Phone +420 2 57 91 18 50
E-Mail sick@sick.cz

China
Phone +852-2763 6966
E-Mail ghk@sick.com.hk

Danmark
Phone +45 45 82 64 00
E-Mail sick@sick.dk

Deutschland
Phone +49 211 5301-301
E-Mail kundenservice@sick.de

España
Phone +34 93 480 31 00
E-Mail info@sick.es

France
Phone +33 1 64 62 35 00
E-Mail info@sick.fr

Great Britain
Phone +44 (0)1727 831121
E-Mail info@sick.co.uk

India
Phone +91–22–4033 8333
E-Mail info@sick-india.com

Israel
Phone +972-4-999-0590
E-Mail info@sick-sensors.com

Italia
Phone +39 02 27 43 41
E-Mail info@sick.it

Japan
Phone +81 (0)3 3358 1341
E-Mail support@sick.jp

Magyarország
Phone +36 1 371 2680
E-Mail office@sick.hu

Nederlands
Phone +31 (0)30 229 25 44
E-Mail info@sick.nl

Norge
Phone +47 67 81 50 00
E-Mail austefjord@sick.nl

Österreich
Phone +43 (0)22 36 62 28 8-0
E-Mail office@sick.at

Polska
Phone +48 22 837 40 50
E-Mail info@sick.pl

România
Phone +40 356 171 120
E-Mail office@sick.ro

Russia
Phone +7 495 775 05 30
E-Mail info@sick.ru

Schweiz
Phone +41 41 619 29 39
E-Mail contact@sick.ch

Singapore
Phone +65 6744 3732
E-Mail admin@sicksgp.com.sg

Slovenija
Phone +386 (0)1-47 69 990
E-Mail office@sick.si

South Africa
Phone +27 11 472 3733
E-Mail info@sickautomation.co.za

South Korea
Phone +82-2 786 6321/4
E-Mail info@sickkorea.net

Suomi
Phone +358-9-25 15 800
E-Mail sick@sick.fi

Sverige
Phone +46 10 110 10 00
E-Mail info@sick.se

Taiwan
Phone +886 2 2375-6288
E-Mail sales@sick.com.tw

Türkiye
Phone +90 216 528 50 00
E-Mail info@sick.com.tr

United Arab Emirates
Phone +971 4 8865 878
E-Mail info@sick.ae

USA/México
Phone +1(952) 941-6780
1 800-325-7425 – tollfree
E-Mail info@sickusa.com

More representatives and agencies
at **www.sick.com**

SICK
Sensor Intelligence.