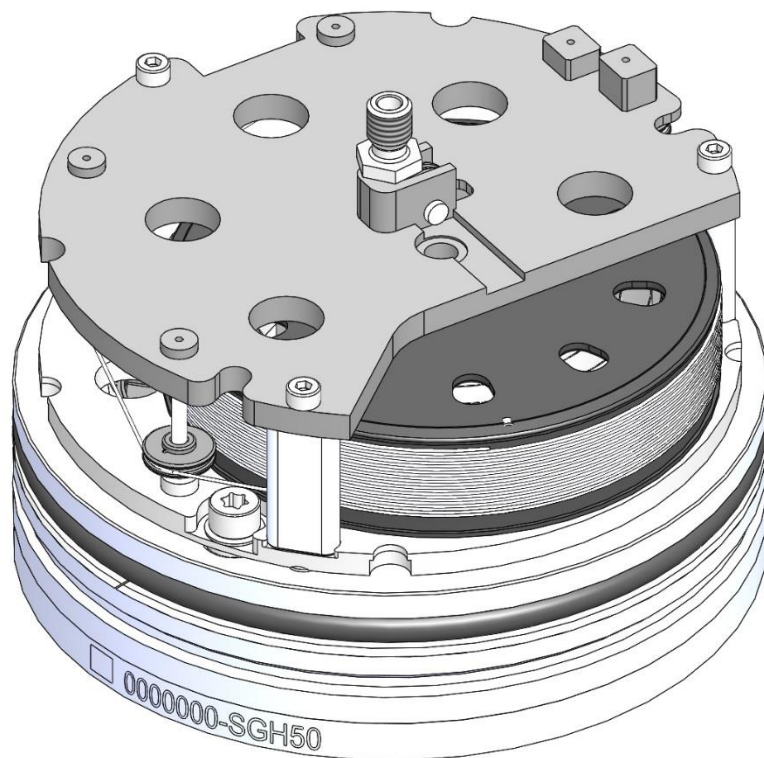


# SGH50

**(Redundant) Absolute Wire-Actuated Encoder  
with CANopen Interface**

User Manual



## Table of Contents

|          |   |          |
|----------|---|----------|
| <b>1</b> | <b>General Information</b> .....                                      | <b>5</b> |
| 1.1      | Documentation.....  | 5        |
| 1.2      | Definitions.....  | 5        |
| <b>2</b> | <b>Intended use</b> .....   | <b>5</b> |
| 2.1      | Switching on the supply voltage.....                                  | 7        |
| <b>3</b> | <b>Operation description</b> .....                                    | <b>7</b> |
| 3.1      | Counting direction.....   | 7        |
| 3.2      | Calibration.....  | 7        |
| 3.3      | Restore factory setting.....  | 7        |
| <b>4</b> | <b>Communication via CAN bus (CANopen)</b> .....                      | <b>8</b> |
| 4.1      | Telegram structure.....   | 8        |
| 4.2      | Node control.....   | 9        |
| 4.2.1    | Network Management Services (NMT).....                                | 9        |
| 4.2.1.1  | NMT Communication States.....   | 10       |
| 4.2.1.2  | Switching between the NMT communication states.....                   | 11       |
| 4.2.2    | Boot-Up.....  | 11       |
| 4.2.3    | SYNC Object.....  | 11       |
| 4.3      | Process Data Exchange.....  | 11       |
| 4.3.1    | Transmission of Process Data Objects (PDO).....                       | 11       |
| 4.3.1.1  | Transmit PDO (from the SGH50 to the master).....                      | 12       |
| 4.4      | Parameter Data Exchange.....  | 13       |
| 4.4.1    | Transmission of Service Data Objects (SDO).....                       | 13       |
| 4.4.1.1  | Accelerated request and confirmation procedure.....                   | 13       |
| 4.4.1.2  | Normal Request and Confirmation Procedure.....                        | 14       |
| 4.4.1.3  | Error response in SDO exchange.....                                   | 15       |
| 4.4.1.4  | SDO Examples.....   | 16       |
| 4.5      | Node monitoring.....  | 18       |
| 4.5.1    | Emergency service (EMCY).....   | 18       |
| 4.5.2    | Node Guarding.....  | 19       |
| 4.5.3    | Heartbeat.....  | 19       |
| 4.6      | Layer Setting Service (LSS).....                                      | 20       |
| 4.6.1    | State Change.....   | 21       |
| 4.6.1.1  | Switch states of all LSS Devices (Switch state global).....           | 21       |
| 4.6.1.2  | Switch states of individual devices (Switch state global).....        | 21       |
| 4.6.2    | Configuration.....  | 22       |
| 4.6.2.1  | Node ID Setting (Configure Node ID).....                              | 22       |
| 4.6.2.2  | Configuration of the baud rate (Configure bit timing parameters)..... | 23       |
| 4.6.2.3  | Activate Baud Rate (Activate bit timing parameters).....              | 24       |
| 4.6.2.4  | Store configuration.....  | 24       |
| 4.6.3    | Request parameters.....   | 25       |
| 4.6.3.1  | Request vendor ID.....  | 25       |

|          |   |    |
|----------|---|----|
| 4.6.3.2  | Request product code .....                    | 25 |
| 4.6.3.3  | Request revision number .....                 | 26 |
| 4.6.3.4  | Request serial number.....                    | 26 |
| 4.6.3.5  | Request Node ID.....                          | 26 |
| 4.7      | Table of Objects .....                        | 27 |
| 4.7.1    | Object Overview .....                         | 27 |
| 4.7.2    | Object Description .....                      | 29 |
| 4.7.2.1  | 1000h Device Type .....                       | 29 |
| 4.7.2.2  | 1001h: Error Register.....                    | 29 |
| 4.7.2.3  | 1002h: Manufacturer Status Register .....     | 29 |
| 4.7.2.4  | 1003h: Pre-defined Error Field.....           | 30 |
| 4.7.2.5  | 1005h: COB-ID SYNC Message.....               | 30 |
| 4.7.2.6  | 1008h: Manufacturer Device Name.....          | 31 |
| 4.7.2.7  | 1009h: Manufacturer Hardware Version .....    | 31 |
| 4.7.2.8  | 100Ah: Manufacturer Software Version .....    | 32 |
| 4.7.2.9  | 100Ch: Guard Time .....                       | 32 |
| 4.7.2.10 | 100Dh: Life Time Factor .....                 | 32 |
| 4.7.2.11 | 1010h: Store Parameter.....                   | 33 |
| 4.7.2.12 | 1011h: Restore Parameter.....                 | 34 |
| 4.7.2.13 | 1014h COB ID emergency message .....          | 36 |
| 4.7.2.14 | 1017h: Producer Heartbeat Time .....          | 37 |
| 4.7.2.15 | 1018h: Identity Object.....                   | 37 |
| 4.7.2.16 | 1200h: Server SDO Parameter .....             | 38 |
| 4.7.2.17 | 1800h 1. Transmit PDO Parameter .....         | 39 |
| 4.7.2.18 | 1801h: 2. Transmit PDO Parameter .....        | 40 |
| 4.7.2.19 | 1A00h: 1. Transmit PDO Mapping Parameter..... | 41 |
| 4.7.2.20 | 1A01h: 2. Transmit PDO Mapping Parameter..... | 42 |
| 4.7.2.21 | 5000h: Diagnose CAN bus error.....            | 43 |
| 4.7.2.22 | 5F09h: Bus terminal. ....                     | 43 |
| 4.7.2.23 | 5F0Ah: Node ID and baud rate Bus CAN.....     | 44 |
| 4.7.2.24 | 6000h: Operating Parameters .....             | 44 |
| 4.7.2.25 | 6002h: Total number of measuring steps .....  | 45 |
| 4.7.2.26 | 6003h: Preset value (calibration value) ..... | 45 |
| 4.7.2.27 | 6004h: Position value.....                    | 45 |
| 4.7.2.28 | 6005h: Resolution.....                        | 45 |
| 4.7.2.29 | 6010h: Calibration value .....                | 46 |
| 4.7.2.30 | 6020h: Position value.....                    | 47 |
| 4.7.2.31 | 6030h: Speed.....                             | 47 |
| 4.7.2.32 | 6200h: Cycle Timer.....                       | 48 |
| 4.7.2.33 | 6400h: Work area (area state register) .....  | 48 |
| 4.7.2.34 | 6401h: Work Area Low Limit.....               | 49 |
| 4.7.2.35 | 6402h: Work Area High Limit .....             | 50 |
| 4.7.2.36 | 6500h: Operating Status.....                  | 51 |
| 4.7.2.37 | 6501h: Single-turn resolution.....            | 51 |

|          |  |    |
|----------|--|----|
| 4.7.2.38 | 6502h: Number of distinguishable revolutions ..... | 51 |
| 4.7.2.39 | 6503h: Alarms .....                                | 52 |
| 4.7.2.40 | 6504h: Supported Alarms .....                      | 52 |
| 4.7.2.41 | 6505h: Warnings.....                               | 53 |
| 4.7.2.42 | 6506h: Supported Warnings .....                    | 53 |
| 4.7.2.43 | 6507h: Profile and Software Version .....          | 53 |
| 4.7.2.44 | 6508h: Operating Time.....                         | 54 |
| 4.7.2.45 | 6509h: Offset value .....                          | 54 |
| 4.7.2.46 | 650Ah: Module identification .....                 | 54 |
| 4.7.2.47 | 650Bh: Serial number .....                         | 55 |
| 4.7.2.48 | 650Ch: Offset value for multi-sensor device.....   | 55 |

## 1 General Information

### 1.1 Documentation

The following documents are available for this product:

- Data sheet; describes the technical data, the dimensions, the pin assignments, the accessories and the order code.
- Assembly Instructions; describes the mechanical and electrical assembly with all safety-relevant conditions and the associated technical specifications.
- User Manual; for commissioning and integrating the sensor in a field bus system.
- EDS file (electronic data sheet); with the help of this file, the integration and configuration in a CANopen network by means of commercial CANopen configurators is possible.

These documents are also available at <http://www.siko-global.com/p/sgh50>

### 1.2 Definitions

Decimal values are specified as numbers without addition (e.g., 1234) unless they are specified in direct conjunction with binary or hexadecimal values. Then the extension d is used (e.g., 1234d). Binary values are used with b (e.g., 1011b) and hexadecimal values with h (e.g., 280H) following the digits.

## 2 Intended use

The SGH50 is available as a redundant version with two encoders and as a non-redundant version with one encoder. The SGH50 detects the stroke of a hydraulic cylinder as absolute travel information. The cable actuator can be parameterized and read via the CAN interface using the CANopen protocol.

The redundant version of the SGH50 is designed for redundant position and speed detection. The sensor can be used for applications up to Performance Level D (PLd) in the overall system. For this purpose, a higher-level, safe encoder evaluation device is required because the wire encoder with its internal sensor diagnosis is not able to initiate actions itself such as the initiation of a safe state. There are increased demands on the electrical and mechanical coupling of the wire encoder.

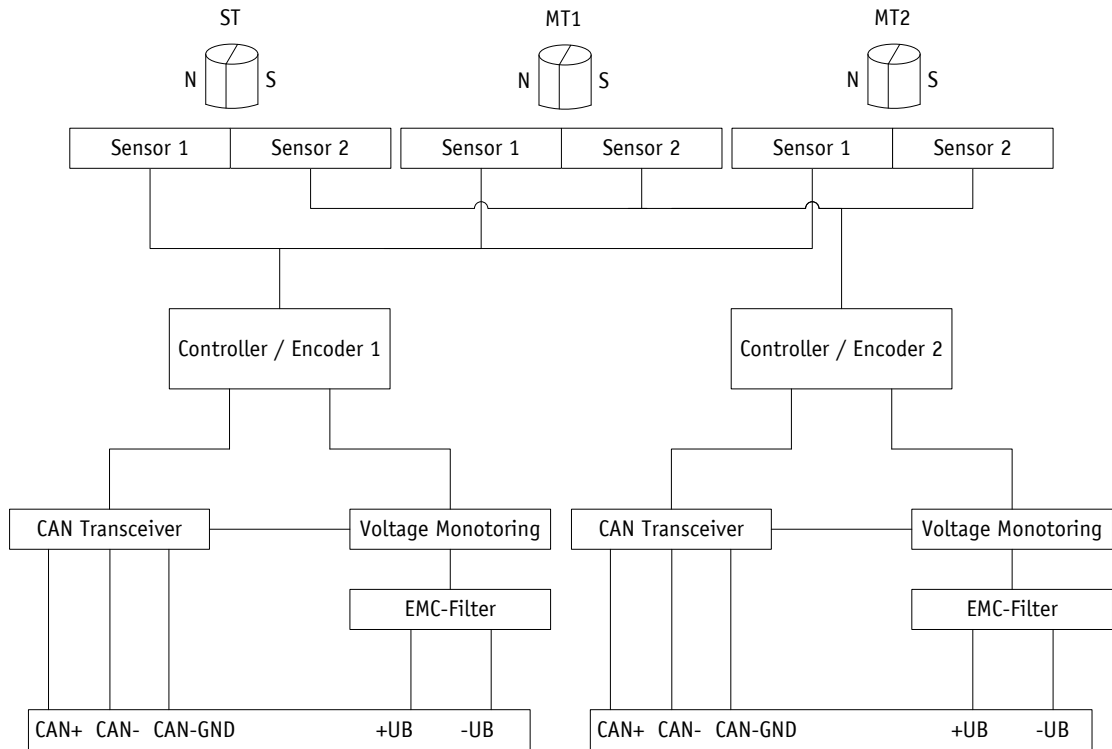


Fig. 1: Block diagram of redundant version

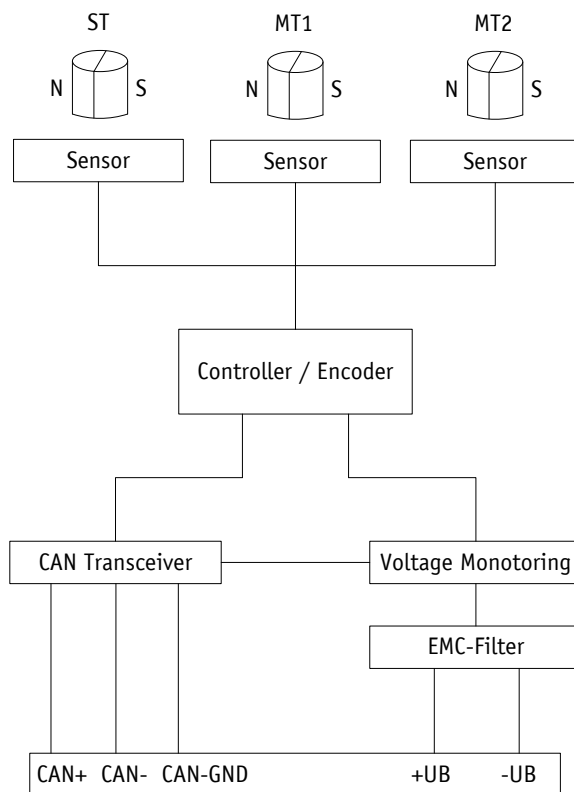


Fig. 2: Block diagram of non-redundant version

## 2.1 Switching on the supply voltage

The SGH50 initializes after it is switched on. The configuration parameters are loaded from the nonvolatile memory into the main memory of the controller.

As long as no changes have been made to the sensor, the sensor works with its default values. If parameters have been changed, the sensor works with the changed data. If these are also to be used after a power off/on, they must be stored.

Upon completion of the initialization procedure, each encoder sends a special NMT command, the boot-up message, to notify the system of its existence. The SGH50 is then in pre-operational mode. In this state, the encoder can be parameterized according to the requirements of the application via SDO commands. This concerns both the configuration parameters of the sensor system and the way in which it makes its position values available to the system (asynchronous or synchronous data transmission).

## 3 Operation description

### 3.1 Counting direction

The encoder supplies increasing position values. This property can be changed via the object [6000h: Operating Parameters](#).

### 3.2 Calibration

Due to the absolute measuring system, calibration is only required once during commissioning and can be performed at any point. As a result, the Gerber zero point can be adjusted with the mechanical zero point of the system. During calibration, the calibration value is used to calculate the position value. The resulting offset value is output in object [6509h: Offset](#) value. In the case of calibration, the following applies:

Position value = 0 + calibration value

### 3.3 Restore factory setting

To restore the delivery state of the device, the following option exists:

| Access   | Coding          | The following are set to the factory setting |                                       |
|--|-----------------|--|---------------------------------------|
| CANopen (cf. object <a href="#">1011h: Restore Parameter</a> ) | 1011h<br>"load" | Subindex 1                                   | All parameters                        |
|  |                 | Subindex 2                                   | Only bus parameters                   |
|  |                 | Subindex 3                                   | Only CiA 406 parameters               |
|  |                 | Subindex 4                                   | Only manufacturer-specific parameters |

Table 1: Access factory settings

## 4 Communication via CAN bus (CANopen)

The basis for the SGH50 is the CANopen communication profile CiA 301 V4.2, the device profile for encoders CiA 406 V3.2 as well as the layer setting service (LSS) CiA 305 V3.0.0. The SGH50 supports device class C2. The details necessary for understanding operation are provided in this documentation. If you require more detailed information, we recommend the relevant specialist literature on CAN or CANopen.

### 4.1 Telegram structure

The data telegram of a CAN message consists of the following fields:

|     |                     |               |                                   |     |         |
|-----|---------------------|---------------|-----------------------------------|-----|---------|
| SOF | Identifier (COB ID) | Control field | Data field (a maximum of 8 bytes) | CRC | ACK/EOF |
|-----|---------------------|---------------|-----------------------------------|-----|---------|

#### SOF:

(Start of Frame) Start bit of the telegram

#### Identifier (COB ID):

- All bus participants use the identifier to check whether the message is relevant to them.
- The identifier sets the priority of the message. The lower the value of the identifier, the higher the priority of the message. As a result, important messages are preferably transmitted via the bus.

The identifier field contains the identifier as well as bits for identifying the length of the identifier (11 or 29 bits). In addition, the identifier defines the device address, the channel selection and the data direction.

Consequently, the 11-bit identifier (COB identifier) is composed of a 4-bit function code and a 7-bit node number:

|            |               |   |   |   |                       |   |   |   |   |   |   |
|------------|---------------|---|---|---|-----------------------|---|---|---|---|---|---|
| Bit no.    | 10            | 9 | 8 | 7 | 6                     | 5 | 4 | 3 | 2 | 1 | 0 |
| Type       | Function code |   |   |   | Node number (Node ID) |   |   |   |   |   |   |
| Assignment | x             | x | x | x | 0                     | 0 | x | x | x | x | X |

The following function codes are defined in the "Pre-Defined Connection Set" (only those function codes that are used in this device are shown):

| Object                   | Function code | Resulting COB ID       | Object | Page |
|--------------------------|---------------|------------------------|--------|------|
| Network Management (NMT) | 0000b         | 0                      | -      | 9    |
| SYNC message             | 0001b         | 128d (80h)             | 1005h  | 30   |
| Emergency message        | 0001b         | 128d (80h) + Node ID   | 1014h  | 36   |
| TPD01                    | 0011b         | 384d (180h) + Node ID  | 1800h  | 39   |
| TPD02                    | 0101b         | 640d (280h) + Node ID  | 1801h  | 40   |
| SDO (tx)                 | 1011b         | 1408d (580h) + Node ID | 1200h  | 38   |
| SDO (rx)                 | 1100b         | 1536d (600h) + Node ID | 1200h  | 38   |
| Heartbeat message        | 1110b         | 1792d (700h) + Node ID | -      | 19   |
| Node guard message       | 1110b         | 1792d (700h) + Node ID | -      | 19   |



| Object   | Function code | Resulting COB ID | Object | Page |
|----------|---------------|------------------|--------|------|
| LSS (tx) | -             | 2021d (7E4h)     | -      | 20   |
| LSS (rx) | -             | 2020d (7E5h)     | -      | 20   |

Table 2: Overview of COB Identifiers

Changes to COB IDs are only possible in the PRE-OPERATIONAL NMT state. Via bit 31 = 1b, the COB ID must first be invalidated before it can be changed and reactivated.

An exception is the COB ID of the sync object. Bit 30 = 0b must be set there to change the COB ID. Since bit 30 cannot be set to 1b in the device, the COB ID could be changed at any time.

The node number (Node ID) (cf. Object [5F0Ah: Node ID and baud rate Bus CAN](#)) is assigned once in each bus system from the master to the SGH50 during the configuration. The node numbers are in the range from 1 to 127. Node ID = 0 is reserved and may not be used.

The transfer of a newly set node number takes place only at a renewed initialization (cf. Chapter [4.2.1](#)).

The SGH50 wire encoder is shipped from the factory in the non-redundant version with Node ID 1 (1h) and in the redundant version with Node ID 1 (1h) and 2 (2h).

#### Control field:

Contains bit-by-bit information about the number of user data and decides whether it is a data frame or remote transmission request (RTR) frame.

#### Data field:

Contains up to 8 bytes of user data. Depending on the channel selection, the user data has different meanings.

#### CRC:

Contains bits for error detection.

#### ACK/EOF:

The ACK/EOF field contains telegram confirmation bits as well as bits to identify the telegram end.

The exact description of the telegram can be found in the relevant CAN literature. In the following telegram descriptions, only the identifier (COB ID) and the data field are discussed for the sake of simplicity.

## 4.2 Node control

### 4.2.1 Network Management Services (NMT)

The master uses the NMT service to configure, manage and monitor network nodes. The device is always in one of the four communication states: "INITIALIZATION", "PRE-OPERATIONAL", "OPERATIONAL" or "STOPPED" (cf. [Fig. 3](#))

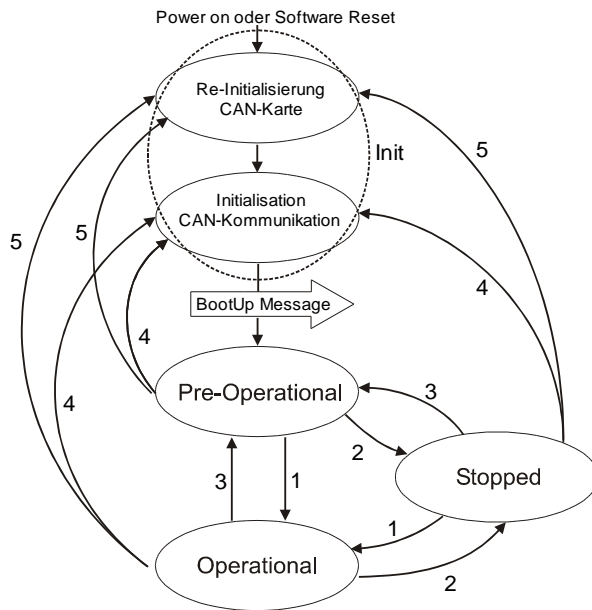


Fig. 3: NMT Status Diagram

#### 4.2.1.1 NMT Communication States

##### NMT INITIALIZATION State

In this state, the device is not involved in what is happening on the bus. All hardware and software components are initialized. This state is reached after the device is switched on or after receiving the command code 81h ("Reset Node") of its own or the global address. After receiving the command code 82h ("Reset Communication"), the display is also in the initialization state. However, only the hardware and software are reinitialized that are related to the CAN communication. The device automatically signals the completion of the initialization with a boot-up message. As soon as the boot-up message has been sent successfully, the device is in the "PRE-OPERATIONAL" state.

##### NMT PRE-OPERATIONAL State

Parameterization data (SDO) can be exchanged in Pre-Operational mode. However, no process data (PDO) are transmitted.

##### NMT OPERATIONAL State

The exchange of process data is also released. COB ID and Transmit PDO Mapping parameters cannot be changed in this state.

##### NMT STOPPED State

With the exception of Heartbeat and Node Guarding, communication is stopped. Only NMT communication is possible.

#### 4.2.1.2 Switching between the NMT communication states

To switch between the communication states, telegrams with the following structure are used:

| State Change                         |   | Transition into<br>Fig. 3 | COB ID | Command | Node ID |
|--------------------------------------|---|---------------------------|--------|---------|---------|
| From                                 | To                                      |                           |        |         |         |
| PRE-OPERATIONAL/STOPPED              | OPERATIONAL                             | 1d                        | 0h     | 01h     | x       |
| OPERATIONAL/PRE-OPERATIONAL          | STOPPED                                 | 2d                        | 0h     | 02h     | x       |
| OPERATIONAL/STOPPED                  | PRE-OPERATIONAL                         | 3d                        | 0h     | 80h     | x       |
| OPERATIONAL/ PRE-OPERATIONAL/STOPPED | INITIALIZATION<br>(Reset Node)          | 5d                        | 0h     | 81h     | x       |
| OPERATIONAL/ PRE-OPERATIONAL/STOPPED | INITIALIZATION<br>(Reset Communication) | 4d                        | 0h     | 82h     | x       |

Table 3: Switching between communication states

If x = 0h is transmitted as Node ID, then the message is intended for all bus subscribers.

#### 4.2.2 Boot-Up

The COB ID of the boot-up message consists of 700h and the Node ID. The NMT state "initialization" is output as data content.

| COB ID         | Byte 0 |
|----------------|--------|
| 700h + Node-ID | 00h    |

Table 4: Boot-up message

#### 4.2.3 SYNC Object

CANopen makes it possible to interrogate inputs and set outputs simultaneously. A synchronization message serves for this (SYNC), which is a high priority CAN message. The identifier of the sync object can be set via the object 1005h (cf. [1005h: COB-ID SYNC Message](#)).

### 4.3 Process Data Exchange

#### 4.3.1 Transmission of Process Data Objects (PDO)

Process Data Objects (PDO) are used for a fast exchange of process data. A maximum of 8 bytes of user data can be transferred in a PDO. The SGH50 supports the transmit PDO services TPDO1 and TPDO2 according to CiA 301 and CiA 406.

#### 4.3.1.1 Transmit PDO (from the SGH50 to the master)

A PDO transmission from the display to the bus master (TPDO) can be initiated by various events:

- Asynchronously controlled by internal device timer
- Synchronously in response to a SYNC message
- In response to an RTR message

TPDO1 and TPDO2 are formed from the position value and the speed value. The transmission behavior of TPDO1 is defined by the objects 1800h, 1A00h and 6200h and is assigned to asynchronous transmission. The TPDO2 is defined via the objects 1801h and 1A01h and serves for synchronous transmission. The assignment is set as a default and cannot be changed.

The messages are structured as follows:

| COB ID                     | User Data in Binary Code |        |        |            |              |            |
|----------------------------|--------------------------|--------|--------|------------|--------------|------------|
|                            | Byte 0 (LSB)             | Byte 1 | Byte 2 | Byte 3 MSB | Byte 4 (LSB) | Byte 5 MSB |
| TPDO1<br>Configure Node ID | Position value           |        |        |            | Speed value  |            |
| TPDO2<br>280h + Node ID    |                          |        |        |            |              |            |

Table 5: TPDO message

#### Asynchronous Data Transmission (TPDO1)

If a TPDO1 is to be sent cyclically, the cycle time in milliseconds must be entered in the object 1800h, subindex 05h. If the value 0 ms is written, TPDO1 is not sent. The function is switched off. The minimum value to be set is 1h (= 1 ms). Alternatively, the value can also be written in the internally linked object 6200h.

#### Synchronous data transmission (TPDO2)

Upon delivery, the device responds to each received SYNC message with the output of the TPDO2 message. In object 1801h, sub-index 02h, 1h is entered for synchronous transmission. If a value n is entered between 1d and 240d (= F0h), the device responds to every nth SYNC message.

#### RTR

Requests can be sent via RTR (cf. Chapter 4.1 Control Field) to TPDO1 and TDPO2.

## 4.4 Parameter Data Exchange

### 4.4.1 Transmission of Service Data Objects (SDO)

Service data objects are mainly used for device configuration via the object directory. SDOs are supported in the expedited request/response ("accelerated request and confirmation procedure") and in the normal request/response.

The identifier is set to 11 bits and cannot be changed.

Two different SDO services are available:

- SDO (rx) (Master → SGH50): 600h + Node ID
- SDO (tx) (SGH50 → Master): 580h + Node ID

These SDO identifiers cannot be changed!

#### 4.4.1.1 Accelerated request and confirmation procedure

Except for the reading of the object [1008h: Manufacturer Device Name](#), all SDOs are exchanged between two parties in the "expedited request/response" procedure. The user data is already delivered with the initialization message.

These SDO messages have the following structure:

| COB ID                 | User data in binary code  |               |               |          |                        |        |        |               |
|------------------------|---------------------------|---------------|---------------|----------|------------------------|--------|--------|---------------|
|                        | Byte 0<br>read /<br>write | Byte 1<br>LSB | Byte 2<br>MSB | Byte 3   | Byte 4<br>LSB          | Byte 5 | Byte 6 | Byte 7<br>MSB |
| SDO rx/tx<br>+ Node ID | Command<br>byte           | Index         |               | Subindex | User data (parameters) |        |        |               |

#### Command Byte, Byte 0:

The command byte determines the type of access and the number of valid data bytes. The following command bytes are valid for the SGH50:

| Command Byte   | Type | Function  |
|----------------|------|---|
| Write Request  | 23h  | SDO (rx), Initiate Download Request, expedited<br>Send parameter to slave (all 4 data bytes valid)        |
| Write Request  | 2Bh  | SDO (rx), Initiate Download Request, expedited<br>Send parameter to slave (2 bytes of 4 data bytes valid) |
| Write Request  | 2Fh  | SDO (rx), Initiate Download Request, expedited<br>Send parameter to slave (1 byte of 4 data bytes valid)  |
| Write Response | 60h  | SDO (tx), Initiate Download Response<br>Confirmation of data transfer to the master                       |
| Read Request   | 40h  | SDO (rx), Initiate Upload Request<br>Request parameters of the slave                                      |
| Read Response  | 43h  | SDO (tx), Initiate Upload Response, expedited<br>Report parameters to Master (all 4 data bytes valid)     |

| Command Byte   |     | Type  | Function  |
|----------------|-----|---|---|
| Read Response  | 4Bh | SDO (tx), Initiate Upload Response, expedited | Report parameters to Master (2 bytes of 4 data bytes valid) |
| Read Response  | 4Fh | SDO (tx), Initiate Upload Response, expedited | Report parameters to Master (1 byte of 4 data bytes valid)  |
| Error Response | 80h | SDO (tx), Abort Domain Transfer               | Slave reports error code to master                          |

Table 6: Command encoding

**Index, Bytes 1 and 2:**

The index (object number) is entered in Intel data format in user data byte 2 (low byte) and user data byte 3 (high byte). The index of the object to be parameterized is entered there.

**Subindex, Byte 3:**

For objects that are executed as an array, the subindex indicates the number of the field.

**User Data (Parameters), Bytes 4-7:**

In the user data, the value of the parameter is entered in left-aligned Intel representation. Byte 4 = Low-Byte ... Byte 7 = High-Byte

**4.4.1.2 Normal Request and Confirmation Procedure**

If more than 4 bytes of service data have to be transmitted, the data is exchanged between two participants using the "normal request/response procedure". This procedure is also initiated by an initialization message, and the actual user data are then transmitted in the following segment messages.

With the SGH50, this is only the case when the object [1008h: Manufacturer Device Name](#) is read.

The initialization message has the following structure:

| COB ID                 | User data in binary code  |               |               |          |                                 |        |        |               |
|------------------------|---------------------------|---------------|---------------|----------|---------------------------------|--------|--------|---------------|
|                        | Byte 0<br>read /<br>write | Byte 1<br>LSB | Byte 2<br>MSB | Byte 3   | Byte 4<br>LSB                   | Byte 5 | Byte 6 | Byte 7<br>MSB |
| SDO rx/tx<br>+ Node ID | Command<br>byte           | Index         |               | Subindex | User data (number of user data) |        |        |               |

The segment message has the following structure:

| COB ID                 | User data in binary code  |               |        |        |        |        |        |               |
|------------------------|---------------------------|---------------|--------|--------|--------|--------|--------|---------------|
|                        | Byte 0<br>read /<br>write | Byte 1<br>LSB | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7<br>MSB |
| SDO rx/tx<br>+ Node ID | Command<br>byte           | User Data     |        |        |        |        |        |               |

**Initialization and segment message: Command Byte, Byte 0:**

The command byte determines the type of access and the number of valid data bytes. The following command bytes are valid for the encoder:

| Command Byte   |     | Type                                      | Function   |
|----------------|-----|---|--|
| Read Request   | 40h | SDO (rx), Normal Initiate Upload Request  | Request parameter from slave (number of bytes to be transmitted) |
| Read Request   | 60h | SDO (rx), Normal Segment Upload Request   | Request parameters from slave (user data)                        |
| Read Response  | 41h | SDO (tx), Normal Initiate Upload Response | Report parameter to master (number of bytes to be transferred)   |
| Read Response  | 03h | SDO (tx), Normal Segment Upload Response  | Report parameters to Master (user data)                          |
| Error Response | 80h | SDO (tx), Abort Domain Transfer           | Slave reports error code to master                               |

Table 7: Command encoding

**Initialization Message: Index, Bytes 1 and 2:**

The index (object number) is entered in Intel data format in user data byte 2 (low byte) and user data byte 3 (high byte). The index of the object to be parameterized is entered there.

**Initialization message: Subindex, Byte 3:**

For objects that are executed as an array, the subindex indicates the number of the field.

**Initialization message: User Data (Parameters), Bytes 4-7:**

In the service data area, the value of the parameter is entered in left-aligned Intel representation. Byte 4 = Low-Byte ... Byte 7 = High-Byte

**Segment message: User data (parameters), bytes 1-7:**

In the user data area, the value of the parameter is entered in left-aligned Intel representation. Byte 1 = Low-Byte ... Byte 7 = High-Byte

**4.4.1.3 Error response in SDO exchange**

If the access is invalid, an error message (abort) is returned to the master. The error codes are described in the CANopen profile (CiA 301) or in the encoder profile (CiA 406). The following table shows the error codes used:

| Error Code | Description  |
|------------|--|
| 05030000h  | Toggle bit in normal transfer of request/response unequal. |
| 06010000h  | Incorrect access to an object.                             |
| 06010001h  | Read access to write-only.                                 |
| 06010002h  | Write access to read-only.                                 |

| Error Code | Description  |
|------------|--|
| 06020000h  | Object does not exist in the object dictionary.  |
| 06090011h  | Subindex does not exist.   |
| 06090030h  | Value range of the selected parameter incorrect.   |
| 08000020h  | Parameters cannot be transferred or saved to the application.  |
| 08000022h  | Parameters cannot be transferred or saved to the application due to the current state of the device. |
| 08000024h  | No data available  |

Table 8: Error codes

#### 4.4.1.4 SDO Examples

##### Example of Reading SDO Parameters with Accelerated Request and Confirmation Procedure:

From the slave with device address 1h, the calibration value stored in object 6010h subindex 01h of the object directory is to be read.

Calculation of the identifier: 600h + Node ID = 600h + 1h = 601h

Command: 40h

Index: 6010h

Subindex: 01h

The current value is 510d = 01FEh

Request from master to slave with Node ID 1h:

| COB ID | User Data |         |         |          |        |        |        |        |
|--------|-----------|---------|---------|----------|--------|--------|--------|--------|
|        | Command   | Index L | Index H | Subindex | Data 0 | Data 1 | Data 2 | Data 3 |
| 601h   | 40h       | 10h     | 60h     | 01h      | x      | x      | x      | x      |

Slave response to the request:

Calculation of the identifier: 580h + Node ID = 581h

| COB ID | User Data              |          |          |          |        |        |        |        |
|--------|------------------------|----------|----------|----------|--------|--------|--------|--------|
|        | Command                | Index LB | Index HB | Subindex | Data 0 | Data 1 | Data 2 | Data 3 |
| 581h   | 43h<br>(4 bytes valid) | 10h      | 60h      | 01h      | FEh    | 01h    | 00h    | 00h    |

##### Example of Writing SDO Parameters with Accelerated Request and Confirmation Procedure:

In the slave with device address 1h, the calibration value, which is stored with 2 bytes in object 6010h of the object directory, is to be changed.

Calculation of the identifier: 600h + Node ID = 600h + 1h = 601h

Command: 2 bytes are to be written: 2Bh

Index: 6200h



Subindex: 00h

The new value should be 4500d = 1194h

Write a value from the master to the slave with Node ID 1h:

| COB ID | User Data              |         |         |          |        |        |        |        |
|--------|------------------------|---------|---------|----------|--------|--------|--------|--------|
|        | Command                | Index L | Index H | Subindex | Data 0 | Data 1 | Data 2 | Data 3 |
| 601h   | 2Bh<br>(2 bytes valid) | 00h     | 62h     | 00h      | 94h    | 11h    | 00h    | 00h    |

Response of the slave to the command:

Calculation of the identifier: 580h + Node ID = 580h + 1h = 581h

| COB ID | User Data |         |         |          |        |        |        |        |
|--------|-----------|---------|---------|----------|--------|--------|--------|--------|
|        | Command   | Index L | Index H | Subindex | Data 0 | Data 1 | Data 2 | Data 3 |
| 581h   | 60h       | 00h     | 62h     | 00h      | 00h    | 00h    | 00h    | 00h    |

**Example of Reading SDO Parameters with Normal Request and Confirmation Procedure:**

From SGH50 with device address 1h, the manufacturer device name stored in object 1008h of the object directory is to be read.

Calculation of the identifier: 600h + Node ID = 600h + 1h = 601h

Command: 40h

Index: 1008h

Subindex 00h

First request (initialization) from master to slave with Node ID 1h:

| COB ID | User Data |         |         |          |        |        |        |        |
|--------|-----------|---------|---------|----------|--------|--------|--------|--------|
|        | Command   | Index L | Index H | Subindex | Data 0 | Data 1 | Data 2 | Data 3 |
| 601h   | 40h       | 08h     | 10h     | 00h      | x      | x      | x      | x      |

Slave response to the request:

Calculation of the identifier: 580h + Node ID = 581h

| COB ID | User Data |          |          |          |        |        |        |        |
|--------|-----------|----------|----------|----------|--------|--------|--------|--------|
|        | Command   | Index LB | Index HB | Subindex | Data 0 | Data 1 | Data 2 | Data 3 |
| 581h   | 41h       | 08h      | 10h      | 00h      | 05h    | 00h    | 00h    | 00h    |

Expected number of user data bytes: 5

Second request from master to slave with Node ID 1h:

| COB ID | User Data |         |         |          |        |        |        |        |
|--------|-----------|---------|---------|----------|--------|--------|--------|--------|
|        | Command   | Index L | Index H | Subindex | Data 0 | Data 1 | Data 2 | Data 3 |
| 601h   | 60h       | 08h     | 10h     | 00h      | x      | x      | x      | x      |

Slave response to the request:

| COB ID | User Data |              |              |              |              |              |          |          |
|--------|-----------|--------------|--------------|--------------|--------------|--------------|----------|----------|
|        | Command   | Data 0       | Data 1       | Data 2       | Data 3       | Data 4       | Data 5   | Data 6   |
| 581h   | 03h       | 53h<br>("S") | 47h<br>("G") | 48h<br>("H") | 35h<br>("5") | 30h<br>("0") | 00h<br>- | 00h<br>- |

## 4.5 Node monitoring

### 4.5.1 Emergency service (EMCY)

The state of the bus participant is transmitted in the event of a fault via high-priority emergency messages. These messages have a data length of 8 bytes and contain error information.

The emergency message is transmitted as soon as a sensor or communication error has occurred or is rectified. The cause of the fault is stored in the fault buffer (cf. Object [1003h: Pre-defined Error Field](#)). An emergency object is only sent once per error event. If a cause of the fault has been eliminated, this is signaled by sending an emergency message with error code 0000h (No Error). If there are several malfunctions and one cause of the fault is eliminated, the error code 0000h is also output, but the remaining error state is specified in the error register.

| Identifier | Byte 0               | Byte 1 | Byte 2                        | Byte 3                                       | Byte 4 | Byte 5 | Byte 6 | Byte 7 |
|------------|----------------------|--------|-------------------------------|--|--------|--------|--------|--------|
| 11/ 29 Bit | Emergency Error Code |        | Error Register (Object 1001h) | Manufacturer-specific error field (not used) |        |        |        |        |

### Emergency Error Code

| Error Description                                    | Error Code |
|--|------------|
| Malfunction cause eliminated                         | 0000h      |
| Bus status switches to Error Passive Mode            | 8120h      |
| Recovered from Bus Off                               | 8140h      |
| Manufacturer specific: Position value error          | FF05h      |
| Manufacturer specific: Position error working area 1 | FF15h      |
| Manufacturer specific: Position error working area 2 | FF16h      |

Table 9: Emergency Error Code

The identifier of the emergency object is set to 80h + Node ID by default, but can be changed via object 1014h (cf. Object [1014h COB ID](#) emergency message). Sending an emergency message is only possible in the NMT state "OPERATIONAL" or "PRE-OPERATIONAL". The sending of emergency messages can be switched off by setting the COB ID valid bit to 1.

### 4.5.2 Node Guarding

Node Guarding is available for failure monitoring of the CANopen network. With Node Guarding, the master sets remote frames (RTR, remote transmission request) to the guarding identifiers of the bus nodes to be monitored. These respond with the guarding message. This contains the current NMT state of the node as well as a toggle bit whose value must change with each message. If the NMT state or toggle bit does not match the value expected by the master or there is no response occurs, the master assumes a node failure.

The time interval (life time) is set via the objects 100Ch (Guard Time) and 100Dh (Life Time Factor), within which the NMT master expects a message. The time interval "Life-Time" is calculated from the cycle time "Guard-Time", multiplied by the factor "Life-Time-Factor". If the NMT master does not receive a response to its RTR frame within the "life time", it can react with suitable measures. After powering up, Node Guarding is activated by sending the master's first RTR frame to the slave. If the value of one of the two objects (100Ch or 100Dh) is set to 0h, Node Guarding is deactivated.

The response of the node to the RTR frame of the master is structured as follows:

| Identifier     | Byte 0            |                        |
|----------------|-------------------|------------------------|
| 700h + Node-ID | Bit 7: Toggle Bit | Bit 6 ... 0: NMT state |

#### Toggle Bit:

The toggle bit must alternate between two successive responses of the device. After the guarding protocol has been activated, the toggle bit must have the value 0 for the first response.

#### NMT State:

4: STOPPED

5: OPERATIONAL

127: PRE-OPERATIONAL

The identifier of the Node Guarding protocol is permanently set to 700h + Node ID and cannot be changed. Sending a Node Guard message is possible in the NMT state "OPERATIONAL", "PREOPERATIONAL" or "STOPPED".

Note:

The literature recommends using the heartbeat for node monitoring. With the Node Guarding protocol, only the master can detect missing communication. The heartbeat, on the other hand, can be received by all participants.

### 4.5.3 Heartbeat

The master monitors the state of the slave device via the heartbeat protocol. The device automatically sends its NMT status cyclically in this case. The SGH50 is a heartbeat producer; it does not receive and process heartbeat protocols itself. The cycle time of the heartbeat message is set via object 1017h. If the cycle time is 0h, the heartbeat protocol is deactivated.

The heartbeat message consists of the COB ID and an additional byte. This byte stores the current NMT state.

| COB ID         | Byte 0    |
|----------------|-----------|
| 700h + Node-ID | NMT state |

**NMT State:**

4: STOPPED

5: OPERATIONAL

127: PRE-OPERATIONAL

The identifier of the Heartbeat protocol is permanently set to 700h + Node ID and cannot be changed. Sending a Heartbeat message is possible in the NMT state "OPERATIONAL", "PREOPERATIONAL" or "STOPPED".

#### 4.6 Layer Setting Service (LSS)

Layer Setting Service (LSS) is a special procedure described in CiA 305 for querying and configuring various parameters (Node ID, baud rate and identity object 1018h).

Each device must have a unique LSS number composed from the entries in object 1018h.

- Vendor ID                    0000 0195h
- Product Code:                FFFF FFFFh
- Revision number:            FFFF FFFFh
- Serial number:                xxxx xxxxh (respective serial number of the encoder))

To be able to use complete LSS functionality, all devices on the bus must support the LSS procedure. There must be an LSS master, and all nodes must start at the same baud rate. After booting, the device is in LSS waiting. To make a configuration, one or all devices must be switched to the LSS configuration. If the LSS master expects an answer to its command, only one LSS slave may be switched to the LSS configuration.

Two different LSS services are available:

- LSS (rx) (LSS Master → SGH50): 7E5h
- LSS (tx) (SGH50 → LSS Master): 7E4h

These LSS identifiers cannot be changed!

A message always consists of 8 bytes. Byte 0 contains the command (command specifier cs). This is followed by a maximum of 7 data bytes. Unused data bytes are reserved and must be populated with 00h.

| Services                        | LSS waiting | LSS configuration                          |
|---------------------------------|-------------|--|
| Switch state global             | Yes         | Yes  |
| Switch state selective          | Yes         | No   |
| Activate bit timing parameters  | No          | Yes, if all devices on the bus support LSS |
| Configure bit timing parameters | No          | Yes  |
| Configure Node ID               | No          | Yes  |
| Store configuration             | No          | Yes  |

|                     |    |     |
|---------------------|----|-----|
| Request LSS address | No | Yes |
| Request Node ID     | No | Yes |

Table 10: State behavior of the supported LSS Services

#### 4.6.1 State Change

##### 4.6.1.1 Switch states of all LSS Devices (Switch state global)

With this command, all devices on the bus can be put into the LSS Waiting or LSS Configuration state. The LSS slave devices do not respond to this.

Master → All LSS Slaves

| COB ID | User Data      |        |        |        |        |        |        |        |
|--------|----------------|--------|--------|--------|--------|--------|--------|--------|
|        | Byte 0 Command | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 |
| 7E5h   | 04h            | Mode   | 00h    | 00h    | 00h    | 00h    | 00h    | 00h    |

**Mode:**

00h: Switch to LSS waiting state

01h: Switch to LSS configuration state

##### 4.6.1.2 Switch states of individual devices (Switch state global)

With this command, individual LSS slave devices can be put into the LSS configuration state via the unique LSS number.

Master → SGH50

| COB ID | User Data      |           |        |        |        |        |        |        |
|--------|----------------|-----------|--------|--------|--------|--------|--------|--------|
|        | Byte 0 Command | Byte 1    | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 |
| 7E5h   | 40h            | Vendor ID |        |        |        | 00h    | 00h    | 00h    |

| COB ID | User Data      |              |        |        |        |        |        |        |
|--------|----------------|--------------|--------|--------|--------|--------|--------|--------|
|        | Byte 0 Command | Byte 1       | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 |
| 7E5h   | 41h            | Product code |        |        |        | 00h    | 00h    | 00h    |

| COB ID | User Data      |        |        |        |        |        |        |        |
|--------|----------------|--------|--------|--------|--------|--------|--------|--------|
|        | Byte 0 Command | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 |
|        |                |        |        |        |        |        |        |        |

|      |     |                 |     |     |     |
|------|-----|-----------------|-----|-----|-----|
| 7E5h | 42h | Revision number | 00h | 00h | 00h |
|------|-----|-----------------|-----|-----|-----|

| COB ID | User Data      |               |        |        |        |        |        |        |  |
|--------|----------------|---------------|--------|--------|--------|--------|--------|--------|--|
|        | Byte 0 Command | Byte 1        | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 |  |
| 7E5h   | 43h            | Serial number |        |        | 00h    | 00h    | 00h    |        |  |

SGH50 → Master

| COB ID | User Data      |        |        |        |        |        |        |        |
|--------|----------------|--------|--------|--------|--------|--------|--------|--------|
|        | Byte 0 Command | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 |
| 7E4h   | 44h            | 00h    | 00h    | 00h    | 00h    | 00h    | 00h    | 00h    |

## 4.6.2 Configuration

### 4.6.2.1 Node ID Setting (Configure Node ID)

The LSS master can configure the Node ID of individual LSS slaves in configuration mode. If the new Node ID should still be available after a power off/on, then the "Store configuration" command must be given after the change. To immediately activate the new Node ID, the LSS slave must be put into the LSS Waiting and then an NMT "Reset Communication" 82h given. Another option is to perform a power off/on after "Store configuration".

Master → SGH50

| COB ID | User Data      |        |        |        |        |        |        |        |
|--------|----------------|--------|--------|--------|--------|--------|--------|--------|
|        | Byte 0 Command | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 |
| 7E5h   | 11h            | NID    | 00h    | 00h    | 00h    | 00h    | 00h    | 00h    |

#### NID:

01h ... 7Fh: Node ID

SGH50 → Master

| COB ID | User Data      |            |            |        |        |        |        |        |
|--------|----------------|------------|------------|--------|--------|--------|--------|--------|
|        | Byte 0 Command | Byte 1     | Byte 2     | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 |
| 7E4h   | 11h            | Error code | Spec error | 00h    | 00h    | 00h    | 00h    | 00h    |

#### Error Code:

00h: Transmission successful

01h: not a valid Node ID

FFh: Implementation error, cf. Spec error

**Spec-error:**

This byte is not equal to 0 only in the case of an implementation error and error code FFh.

**4.6.2.2 Configuration of the baud rate (Configure bit timing parameters)**

This command can be used to configure the baud rate of a single or several LSS slaves. If the new baud rate should still be available after a power off/on, then the "Store configuration" command must be given after the change. To activate the new baud rate, the command [4.6.2.3 Activate Baud Rate \(Activate bit timing parameters\)](#) must be given and the LSS slave put into LSS Waiting. Another option is to perform a power off/on after "Store configuration" to activate the new baud rate.

Master → SGH50

| COB ID | User Data      |                |             |        |        |        |        |        |
|--------|----------------|----------------|-------------|--------|--------|--------|--------|--------|
|        | Byte 0 Command | Byte 1         | Byte 2      | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 |
| 7E5h   | 13h            | Table selector | Table index | 00h    | 00h    | 00h    | 00h    | 00h    |

**Table selector:**

00h: CiA 301 bit timing table

80h...FEh: Manufacturer-specific bit timing table

**Table index:**

| Table index | Baud rate     |
|-------------|---------------|
| 0           | 1000 kbit/s   |
| 1           | 800 kbit/s    |
| 2           | 500 kbit/s    |
| 3           | 250 kbit/s    |
| 4           | 125 kbit/s    |
| 5           | Reserved      |
| 6           | 50 kbit/s     |
| 7           | 20 kbit/s     |
| 8           | Not supported |
| 9           | Not supported |

The device only supports Table selector 00h and Table index 0 to 7.

SGH50 → Master

| COB ID | User Data      |            |            |        |        |        |        |        |
|--------|----------------|------------|------------|--------|--------|--------|--------|--------|
|        | Byte 0 Command | Byte 1     | Byte 2     | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 |
| 7E4h   | 13h            | Error code | Spec error | 00h    | 00h    | 00h    | 00h    | 00h    |

**Error Code:**

00h: Transmission successful

01h: not a valid baud rate

FFh: Implementation error, cf. Spec error

**Spec-error:**

This byte is not equal to 0 only in the case of an implementation error and error code FFh.

**4.6.2.3 Activate Baud Rate (Activate bit timing parameters)**

This command activates the baud rate newly set via [4.6.2.2 Configuration of the baud rate \(Configure bit timing parameters\)](#) without the necessity of a power off/on.

Master → SGH50

| COB ID | User Data      |              |            |        |        |        |        |        |
|--------|----------------|--------------|------------|--------|--------|--------|--------|--------|
|        | Byte 0 Command | Byte 1 LSB   | Byte 2 MSB | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 |
| 7E5h   | 15h            | Switch delay |            | 00h    | 00h    | 00h    | 00h    | 00h    |

**Switch Delay:**

The Switch delay parameter defines the length of two delay times (d1, d2) of the same length and must correspond to a multiple of 1 ms. After expiry of the individual processing time and the delay time d1, the new baud rate is adopted internally. After the delay time d2 has elapsed, the LSS slave reports the boot-up via the newly set baud rate. This procedure prevents devices with different baud rates from being on the bus at the same time. During the two delay times d1 and d2, the LSS slave cannot send messages.

**4.6.2.4 Store configuration**

This command may only be executed if only one LSS slave is in configuration mode. The current settings are then be stored.

Master → SGH50

| COB ID | User Data      |        |        |        |        |        |        |        |
|--------|----------------|--------|--------|--------|--------|--------|--------|--------|
|        | Byte 0 Command | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 |
| 7E5h   | 17h            | 00h    | 00h    | 00h    | 00h    | 00h    | 00h    | 00h    |

SGH50 → Master

| COB ID | User Data      |            |            |        |        |        |        |        |
|--------|----------------|------------|------------|--------|--------|--------|--------|--------|
|        | Byte 0 Command | Byte 1     | Byte 2     | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 |
| 7E4h   | 17h            | Error code | Spec error | 00h    | 00h    | 00h    | 00h    | 00h    |

**Error Code:**



- 00h: Transmission successful
- 01h: Store configuration is not supported
- 02h: Error during saving
- FFh: Implementation error, cf. Spec error

**Spec-error:**

This byte is not equal to 0 only in the case of an implementation error and error code FFh.

**4.6.3 Request parameters**

The following requests may only be executed if only one LSS slave is in configuration mode.

**4.6.3.1 Request vendor ID**

Master → SGH50

| COB ID | User Data      |        |        |        |        |        |        |        |
|--------|----------------|--------|--------|--------|--------|--------|--------|--------|
|        | Byte 0 Command | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 |
| 7E5h   | 5Ah            | 00h    | 00h    | 00h    | 00h    | 00h    | 00h    | 00h    |

SGH50 → Master

| COB ID | User Data      |                                |        |        |            |        |        |        |
|--------|----------------|--------------------------------|--------|--------|------------|--------|--------|--------|
|        | Byte 0 Command | Byte 1 LSB                     | Byte 2 | Byte 3 | Byte 4 MSB | Byte 5 | Byte 6 | Byte 7 |
| 7E4h   | 5Ah            | Vendor ID (cf. Object 1018.1h) |        |        | 00h        | 00h    | 00h    |        |

**4.6.3.2 Request product code**

Master → SGH50

| COB ID | User Data      |        |        |        |        |        |        |        |
|--------|----------------|--------|--------|--------|--------|--------|--------|--------|
|        | Byte 0 Command | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 |
| 7E5h   | 5Bh            | 00h    | 00h    | 00h    | 00h    | 00h    | 00h    | 00h    |

SGH50 → Master

| COB ID | User Data      |                                   |        |        |            |        |        |        |
|--------|----------------|-----------------------------------|--------|--------|------------|--------|--------|--------|
|        | Byte 0 Command | Byte 1 LSB                        | Byte 2 | Byte 3 | Byte 4 MSB | Byte 5 | Byte 6 | Byte 7 |
| 7E4h   | 5Bh            | Product Code (cf. Object 1018.2h) |        |        | 00h        | 00h    | 00h    |        |

#### 4.6.3.3 Request revision number

Master → SGH50

| COB ID | User Data      |        |        |        |        |        |        |        |
|--------|----------------|--------|--------|--------|--------|--------|--------|--------|
|        | Byte 0 Command | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 |
| 7E5h   | 5Ch            | 00h    | 00h    | 00h    | 00h    | 00h    | 00h    | 00h    |

SGH50 → Master

| COB ID | User Data      |                                  |        |        |            |        |        |        |
|--------|----------------|----------------------------------|--------|--------|------------|--------|--------|--------|
|        | Byte 0 Command | Byte 1 LSB                       | Byte 2 | Byte 3 | Byte 4 MSB | Byte 5 | Byte 6 | Byte 7 |
| 7E4h   | 5Ch            | Revision number (Object 1018.3h) |        |        | 00h        | 00h    | 00h    | 00h    |

#### 4.6.3.4 Request serial number

Master → SGH50

| COB ID | User Data      |        |        |        |        |        |        |        |
|--------|----------------|--------|--------|--------|--------|--------|--------|--------|
|        | Byte 0 Command | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 |
| 7E5h   | 5Dh            | 00h    | 00h    | 00h    | 00h    | 00h    | 00h    | 00h    |

SGH50 → Master

| COB ID | User Data      |                                |        |        |            |        |        |        |
|--------|----------------|--------------------------------|--------|--------|------------|--------|--------|--------|
|        | Byte 0 Command | Byte 1 LSB                     | Byte 2 | Byte 3 | Byte 4 MSB | Byte 5 | Byte 6 | Byte 7 |
| 7E4h   | 5Dh            | Serial number (object 1018.4h) |        |        | 00h        | 00h    | 00h    | 00h    |

#### 4.6.3.5 Request Node ID

Master → SGH50

| COB ID | User Data      |        |        |        |        |        |        |        |
|--------|----------------|--------|--------|--------|--------|--------|--------|--------|
|        | Byte 0 Command | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 |
| 7E5h   | 5Eh            | 00h    | 00h    | 00h    | 00h    | 00h    | 00h    | 00h    |

SGH50 → Master

| COB ID | User Data      |               |        |        |        |        |        |        |
|--------|----------------|---------------|--------|--------|--------|--------|--------|--------|
|        | Byte 0 Command | Byte 1        | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 |
| 7E4h   | 5Eh            | Node ID (NID) |        | 00h    | 00h    | 00h    | 00h    | 00h    |

## 4.7 Table of Objects

### 4.7.1 Object Overview

The following table provides an overview of the objects of the device.

| Name                                     | Description  | Cf. page |
|--|--|----------|
| 1000h Device Type                        | Device profile and encoder type                                      | 29       |
| 1001h: Error Register                    | Current error state of the device                                    | 29       |
| 1002h: Manufacturer Status Register      | Contains the Transmit Error Counter and the Receive Error Counter    | 29       |
| 1003h: Pre-defined Error Field           | The object stores the 8 most recent error states                     | 30       |
| 1005h: COB-ID SYNC Message               | Setting of the COB ID of the SYNC object                             | 30       |
| 1008h: Manufacturer Device Name          | Device name in ASCII characters                                      | 31       |
| 1009h: Manufacturer Hardware Version     | Specifies the hardware version of the device                         | 31       |
| 100Ah: Manufacturer Software Version     | Specifies the software version of the device                         | 32       |
| 100Ch: Guard Time                        | Parameters for node guarding   | 32       |
| 100Dh: Life Time Factor                  | Parameters for node guarding   | 32       |
| 1010h: Store Parameter                   | Object for non-volatile storage of the settings                      | 33       |
| 1011h: Restore Parameter                 | Object for restoring the factory settings                            | 34       |
| 1014h COB ID emergency message           | COB ID of the emergency object                                       | 36       |
| 1017h: Producer Heartbeat Time           | Setting of the cycle time of the heartbeat timer                     | 37       |
| 1018h: Identity Object                   | Contains the manufacturer number                                     | 37       |
| 1200h: Server SDO Parameter              | SDO parameter  | 38       |
| 1800h 1. Transmit PDO Parameter          | Transmit PDO for asynchronous transmission (timer-controlled)        | 39       |
| 1801h: 2. Transmit PDO Parameter         | Transmit PDO for synchronous transmission                            | 40       |
| 1A00h: 1. Transmit PDO Mapping Parameter | Describes the arrangement of the objects, which are shown in TPDO1   | 41       |
| 1A01h: 2. Transmit PDO Mapping Parameter | Describes the arrangement of the objects that are displayed in TPDO2 | 42       |
| 5000h: Diagnose CAN bus error            | Provides information about the CAN bus errors that occurred          | 43       |
| 5F09h: Bus terminal.                     | Bus terminal.  | 43       |
| 5F0Ah: Node ID and baud rate Bus CAN     | Setting of the Node ID and the baud rate                             | 44       |
| 6000h: Operating Parameters              | Setting of the scaling and the direction of rotation                 | 44       |
| 6002h: Total number of measuring steps   | Specifies the total number of measurement steps of the system        | 45       |

| Name   | Description  | Cf. page |
|--|--|----------|
| 6003h: Preset value (calibration value)      | Use object 6010h subindex 01h  | 45       |
| 6004h: Position value                        | Use object 6020h subindex 01h  | 45       |
| 6005h: Resolution                            | Setting of the resolution  | 45       |
| 6010h: Calibration value                     | Setting of the calibration value   | 46       |
| 6020h: Position value                        | Position value   | 47       |
| 6030h: Speed                                 | Speed value  | 47       |
| 6200h: Cycle Timer                           | Identical with object 1800h, subindex 5  | 45       |
| 6400h: Work area (area state register)       | Indicates whether the position value is within the set working ranges 1 and 2                                    | 48       |
| 6401h: Work Area Low Limit                   | Setting of the lower limits of the working range 1 and 2   | 49       |
| 6402h: Work Area High Limit                  | Setting of the upper limits of the working range 1 and 2   | 50       |
| 6500h: Operating Status                      | Output of the scaling and the direction of rotation  | 48       |
| 6501h: Single-turn resolution                | The physical number of measuring steps per revolution  | 51       |
| 6502h: Number of distinguishable revolutions | Number of revolutions that the encoder can detect  | 51       |
| 6503h: Alarms                                | Display of error states  | 52       |
| 6504h: Supported Alarms                      | Specifies which alarm messages are supported   | 52       |
| 6505h: Warnings                              | Display of warnings  | 53       |
| 6506h: Supported Warnings                    | Specifies which warnings are supported   | 53       |
| 6507h: Profile and Software Version          | Displays the version number of the device profile used and the version number of the device firmware             | 51       |
| 6508h: Operating Time                        | Operating hour meter (function is not supported)   | 54       |
| 6509h: Offset value                          | Encoder states at the time of calibration  | 54       |
| 650Ah: Module identification                 | Specifies the manufacturer-specific offset value as well as the smallest and largest transferable position value | 54       |
| 650Bh: Serial number                         | Specifies the serial number  | 55       |
| 650Ch: Offset value for multi-sensor device  | Encoder states at the time of calibration  | 55       |

Table 11: Object Overview

## 4.7.2 Object Description

### 4.7.2.1 1000h Device Type

The object 1000h specifies the device profile number.

|              |  |        |              |        |
|--------------|--|--------|--------------|--------|
| Subindex     | 00h  |        |              |        |
| Description  | Information about device profile and device type |        |              |        |
| Access       | ro   |        |              |        |
| PDO mapping  | No   |        |              |        |
| Data type    | UNSIGNED 32                                      |        |              |        |
| Default      | 000A0196h  |        |              |        |
| EEPROM       | No   |        |              |        |
| Data content | Device profile number                            |        | Encoder type |        |
|              | Byte 0   | Byte 1 | Byte 2       | Byte 3 |
|              | 96h  | 01h    | 0Ah          | 00h    |

0196h (= 406d): CANopen Device Profile for Encoders

000Ah: Multi-sensor encoder interface

### 4.7.2.2 1001h: Error Register

The object 1001h displays the error state of the device.

|              |                                |  |
|--------------|--------------------------------|--|
| Subindex     | 00h                            |  |
| Description  | Currently existing error state |  |
| Access       | ro                             |  |
| PDO mapping  | No                             |  |
| Data type    | UNSIGNED 8                     |  |
| Default      | 0h                             |  |
| EEPROM       | No                             |  |
| Data content | Bit                            | Significance   |
|              | 0                              | Set bit indicates the occurrence of any error state                        |
|              | 4                              | Set bit indicates communication errors on the CAN bus (passive or Bus off) |
|              | 7                              | Manufacturer-specific (sensor error)                                       |
|              | 1 -3 5-6                       | Not used   |

Faults and errors are signaled by an emergency message at the moment of their occurrence.

### 4.7.2.3 1002h: Manufacturer Status Register

Object 1002h outputs the counter readings of the tab "Receive Error Counter" and "transmit error counter". The contents of this tab provide information about transmission faults at the mounting location of the encoder.

|              |  |                        |        |        |
|--------------|--|------------------------|--------|--------|
| Subindex     | 00h  |                        |        |        |
| Description  | Transmit Error Counter and Receive Error Counter |                        |        |        |
| Access       | ro   |                        |        |        |
| PDO mapping  | No   |                        |        |        |
| Data type    | UNSIGNED 32                                      |                        |        |        |
| Default      | 0h   |                        |        |        |
| EEPROM       | No   |                        |        |        |
| Data content | Byte 0   | Byte 1                 | Byte 2 | Byte 3 |
|              | Receive Error Counter                            | Transmit Error Counter |        |        |

#### 4.7.2.4 1003h: Pre-defined Error Field

In object 1003h, the 8 most recent error states are archived (cf. Chapter 4.5.1).

- The entry under subindex 0 specifies the number of the stored errors.
- The most recent error state is always stored in subindex 01h. Previous error messages move in the position by one subindex further.
- The entire error list is deleted when you write the value 0h at subindex 00h.
- The entries in the error list have the format described in Chapter 4.5.1.

|             |                                 |  |  |  |
|-------------|---------------------------------|--|--|--|
| Subindex    | 00h                             |  |  |  |
| Description | Number of stored error messages |  |  |  |
| Access      | rw                              |  |  |  |
| PDO mapping | No                              |  |  |  |
| Data type   | UNSIGNED 8                      |  |  |  |
| Default     | 0h                              |  |  |  |
| EEPROM      | Yes                             |  |  |  |

|             |                              |  |  |  |
|-------------|------------------------------|--|--|--|
| Subindex    | 01h-08h                      |  |  |  |
| Description | Error messages that occurred |  |  |  |
| Access      | ro                           |  |  |  |
| PDO mapping | No                           |  |  |  |
| Data type   | UNSIGNED 32                  |  |  |  |
| Default     | 0h                           |  |  |  |
| EEPROM      | Yes                          |  |  |  |

#### 4.7.2.5 1005h: COB-ID SYNC Message

The COB ID of the SYNC object is set by object 1005h.

|             |   |  |  |  |
|-------------|---|--|--|--|
| Subindex    | 00h   |  |  |  |
| Description | Defines the COB ID of the synchronization object (SYNC) |  |  |  |

|              |  |  |
|--------------|--|--|
| Access       | rw (can only be described in the "Pre-Operational" state; cf. Chapter 4.1) |  |
| PDO mapping  | No   |  |
| Data type    | UNSIGNED 32  |  |
| Default      | 80h  |  |
| EEPROM       | Yes  |  |
| Data content | Bit 31   | Not specified  |
|              | Bit 30   | 0: Device does not generate any SYNC message                         |
|              | Bit 29   | 0: 11 bit identifier (CAN 2.0A)<br>1: 29 bit identifier (CAN 2.0B)   |
|              | Bit 28 ... 11  | 0: if Bit 29 = 0<br>X: Bits 28 - 11 of the SYNC COB ID if bit 29 = 1 |
|              | Bit 10 ... 0   | X: Bits 10 - 0 of the SYNC COB ID                                    |

#### 4.7.2.6 1008h: Manufacturer Device Name

Object 1008h specifies the device name. Because this contains 5 data bytes, the SDO Normal Transfer is required for reading (cf. Chapter 4.4.1.2).

|             |                                 |           |           |           |           |         |         |
|-------------|---------------------------------|-----------|-----------|-----------|-----------|---------|---------|
| Subindex    | 00h                             |           |           |           |           |         |         |
| Description | Device name in ASCII characters |           |           |           |           |         |         |
| Access      | Const                           |           |           |           |           |         |         |
| PDO mapping | No                              |           |           |           |           |         |         |
| Data type   | Visible_String                  |           |           |           |           |         |         |
| Default     | SGH50                           |           |           |           |           |         |         |
| EEPROM      | No                              |           |           |           |           |         |         |
|             | Byte 0                          | Byte 1    | Byte 2    | Byte 3    | Byte 4    | Byte 5  | Byte 6  |
|             | 53h ("S")                       | 47h ("G") | 48h ("H") | 35h ("5") | 30h ("0") | 00h (-) | 00h (-) |

#### 4.7.2.7 1009h: Manufacturer Hardware Version

Object 1009h specifies the hardware version.

|              |                                      |           |           |           |
|--------------|--------------------------------------|-----------|-----------|-----------|
| Subindex     | 00h                                  |           |           |           |
| Description  | Hardware version in ASCII characters |           |           |           |
| Access       | Const                                |           |           |           |
| PDO mapping  | No                                   |           |           |           |
| Data type    | Visible_String                       |           |           |           |
| Default      | V100.                                |           |           |           |
| EEPROM       | No                                   |           |           |           |
| Data content | Byte 0                               | Byte 1    | Byte 2    | Byte 3    |
|              | 56h ("V")                            | 30h ("1") | 30h ("0") | 31h ("0") |

#### 4.7.2.8 100Ah: Manufacturer Software Version

The object 100Ah specifies software version of the device.

|              |                                      |           |           |           |
|--------------|--------------------------------------|-----------|-----------|-----------|
| Subindex     | 00h                                  |           |           |           |
| Description  | Software version in ASCII characters |           |           |           |
| Access       | Const                                |           |           |           |
| PDO mapping  | No                                   |           |           |           |
| Data type    | Visible_String                       |           |           |           |
| Default      | V100.                                |           |           |           |
| EEPROM       | No                                   |           |           |           |
| Data content | Byte 0                               | Byte 1    | Byte 2    | Byte 3    |
|              | 56h ("V")                            | 31h ("1") | 30h ("0") | 30h ("0") |

#### 4.7.2.9 100Ch: Guard Time

Object 100Ch specifies the cycle time set in the master for the node guarding (cf. Chapter 4.5.2). The cycle time width is entered in milliseconds. The value "0h" means that Node Guarding is deactivated.

|             |             |
|-------------|-------------|
| Subindex    | 00h         |
| Description | Guard Time  |
| Access      | rw          |
| PDO mapping | No          |
| Data type   | UNSIGNED 16 |
| Default     | 0h          |
| EEPROM      | Yes         |

#### 4.7.2.10 100Dh: Life Time Factor

Object 100Dh specifies the Life Time Factor set in the master for the node guarding (cf. Chapter 4.5.2). The value "0h" means that Node Guarding is deactivated.

|             |                  |
|-------------|------------------|
| Subindex    | 00h              |
| Description | Life Time Factor |
| Access      | rw               |
| PDO mapping | No               |
| Data type   | UNSIGNED 8       |
| Default     | 0h               |
| EEPROM      | Yes              |



#### 4.7.2.11 1010h: Store Parameter

With this object, parameters are transmitted into the EEPROM, so that they are safe from power outages. Depending on the selection of which subindex is accessed, different parameter groups are saved. The string "save" must also be transmitted as data content.

|             |   |
|-------------|---|
| Subindex    | 00h                                     |
| Description | Displays the largest supported subindex |
| Access      | const                                   |
| PDO mapping | No                                      |
| Data type   | UNSIGNED 8                              |
| Default     | 4h                                      |
| EEPROM      | No                                      |

|              |                     |   |           |           |
|--------------|---------------------|---|-----------|-----------|
| Subindex     | 01h                 |   |           |           |
| Description  | Save all parameters |   |           |           |
| Access       | rw                  |   |           |           |
| PDO mapping  | No                  |   |           |           |
| Data type    | UNSIGNED 32         |   |           |           |
| Default      | 1h                  |   |           |           |
| EEPROM       | No                  |   |           |           |
| Data content | Writing:            |   |           |           |
|              | Byte 0              | Byte 1  | Byte 2    | Byte 3    |
|              | 73h ("s")           | 61h ("a")   | 76h ("v") | 65h ("e") |
|              | Read:               |   |           |           |
|              | Bit 31 ... 2        | 0, reserved                                       |           |           |
|              | Bit 1               | 0: Device does not store parameters independently |           |           |
|              | Bit 0               | 1: Device stores parameters on command            |           |           |

|              |   |   |           |           |
|--------------|---|---|-----------|-----------|
| Subindex     | 02h   |   |           |           |
| Description  | Only save communication parameters (1000h-1FFFh, CiA 301) |   |           |           |
| Access       | rw  |   |           |           |
| PDO mapping  | No  |   |           |           |
| Data type    | UNSIGNED 32   |   |           |           |
| Default      | 1h  |   |           |           |
| EEPROM       | No  |   |           |           |
| Data content | Writing:  |   |           |           |
|              | Byte 0  | Byte 1  | Byte 2    | Byte 3    |
|              | 73h ("s")   | 61h ("a")   | 76h ("v") | 65h ("e") |
|              | Read:   |   |           |           |
|              | Bit 31 ... 2  | 0, reserved                                       |           |           |
|              | Bit 1   | 0: Device does not store parameters independently |           |           |
|              | Bit 0   | 1: Device stores parameters on command            |           |           |

|              |   |   |           |           |
|--------------|---|---|-----------|-----------|
| Subindex     | 03h   |   |           |           |
| Description  | Only save application parameters (6000h-9FFFh, CiA 406) |   |           |           |
| Access       | rw  |   |           |           |
| PDO mapping  | No  |   |           |           |
| Data type    | UNSIGNED 32   |   |           |           |
| Default      | 1h  |   |           |           |
| EEPROM       | No  |   |           |           |
| Data content | Writing:  |   |           |           |
|              | Byte 0  | Byte 1  | Byte 2    | Byte 3    |
|              | 73h ("s")   | 61h ("a")   | 76h ("v") | 65h ("e") |
|              | Read:   |   |           |           |
|              | Bit 31 ... 2  | 0, reserved                                       |           |           |
|              | Bit 1   | 0: Device does not store parameters independently |           |           |
|              | Bit 0   | 1: Device stores parameters on command            |           |           |

|              |  |   |           |           |
|--------------|--|---|-----------|-----------|
| Subindex     | 04h  |   |           |           |
| Description  | Only save manufacturer-specific parameters (2000h-5FFFh) |   |           |           |
| Access       | rw   |   |           |           |
| PDO mapping  | No   |   |           |           |
| Data type    | UNSIGNED 32  |   |           |           |
| Default      | 1h   |   |           |           |
| EEPROM       | No   |   |           |           |
| Data content | Writing:   |   |           |           |
|              | Byte 0   | Byte 1  | Byte 2    | Byte 3    |
|              | 73h ("s")  | 61h ("a")   | 76h ("v") | 65h ("e") |
|              | Read:  |   |           |           |
|              | Bit 31 ... 2   | 0, reserved                                       |           |           |
|              | Bit 1  | 0: Device does not store parameters independently |           |           |
|              | Bit 0  | 1: Device stores parameters on command            |           |           |

#### 4.7.2.12 1011h: Restore Parameter

The object 1011h restores the factory settings of the device depending on the selection. The string "load" must be sent as data content and then a reset performed. If the restored parameters are to be available permanently, they must be saved via the object [1010h: Store Parameter](#).

|             |   |  |  |
|-------------|---|--|--|
| Subindex    | 00h                                     |  |  |
| Description | Displays the largest supported subindex |  |  |
| Access      | const                                   |  |  |
| PDO mapping | No                                      |  |  |
| Data type   | UNSIGNED 8                              |  |  |

|         |    |
|---------|----|
| Default | 4h |
| EEPROM  | No |

|              |  |   |           |           |
|--------------|--|---|-----------|-----------|
| Subindex     | 01h                                    |   |           |           |
| Description  | Set all parameters to factory settings |   |           |           |
| Access       | rw                                     |   |           |           |
| PDO mapping  | No                                     |   |           |           |
| Data type    | UNSIGNED 32                            |   |           |           |
| Default      | 1h                                     |   |           |           |
| EEPROM       | No                                     |   |           |           |
| Data content | Writing:                               |   |           |           |
|              | Byte 0                                 | Byte 1  | Byte 2    | Byte 3    |
|              | 6Ch ("l")                              | 6Fh ("o")   | 61h ("a") | 64h ("d") |
|              | Read:                                  |   |           |           |
|              | Bit 31 ... 1                           | 0, reserved   |           |           |
|              | Bit 0                                  | 1: Device allows the loading of default parameters. |           |           |

|              |   |   |           |           |
|--------------|---|---|-----------|-----------|
| Subindex     | 02h   |   |           |           |
| Description  | Only set communication parameters to the factory setting (1000h-1FFFh, CiA 301) |   |           |           |
| Access       | rw  |   |           |           |
| PDO mapping  | No  |   |           |           |
| Data type    | UNSIGNED 32   |   |           |           |
| Default      | 1h  |   |           |           |
| EEPROM       | No  |   |           |           |
| Data content | Writing:  |   |           |           |
|              | Byte 0  | Byte 1  | Byte 2    | Byte 3    |
|              | 6Ch ("l")   | 6Fh ("o")   | 61h ("a") | 64h ("d") |
|              | Read:   |   |           |           |
|              | Bit 31 ... 1  | 0, reserved   |           |           |
|              | Bit 0   | 1: Device allows the loading of default parameters. |           |           |

|              |   |  |  |  |
|--------------|---|--|--|--|
| Subindex     | 03h   |  |  |  |
| Description  | Only set application parameters to the factory setting (6000h-9FFFh, CiA 406) |  |  |  |
| Access       | rw  |  |  |  |
| PDO mapping  | No  |  |  |  |
| Data type    | UNSIGNED 32   |  |  |  |
| Default      | 1h  |  |  |  |
| EEPROM       | No  |  |  |  |
| Data content | Writing:  |  |  |  |

|  |              |   |           |           |
|--|--------------|---|-----------|-----------|
|  | Byte 0       | Byte 1  | Byte 2    | Byte 3    |
|  | 6Ch ("l")    | 6Fh ("o")   | 61h ("a") | 64h ("d") |
|  | Read:        |   |           |           |
|  | Bit 31 ... 1 | 0, reserved   |           |           |
|  | Bit 0        | 1: Device allows the loading of default parameters. |           |           |

|              |  |   |           |           |
|--------------|--|---|-----------|-----------|
| Subindex     | 04h  |   |           |           |
| Description  | Only set manufacturer-specific parameters to the factory setting (2000h-5FFFh) |   |           |           |
| Access       | rw   |   |           |           |
| PDO mapping  | No   |   |           |           |
| Data type    | UNSIGNED 32  |   |           |           |
| Default      | 1h   |   |           |           |
| EEPROM       | No   |   |           |           |
| Data content | Writing:   |   |           |           |
|              | Byte 0   | Byte 1  | Byte 2    | Byte 3    |
|              | 6Ch ("l")  | 6Fh ("o")   | 61h ("a") | 64h ("d") |
|              | Read:  |   |           |           |
|              | Bit 31 ... 1   | 0, reserved   |           |           |
|              | Bit 0  | 1: Device allows the loading of default parameters. |           |           |

#### 4.7.2.13 1014h COB ID emergency message

The COB ID of the emergency object is set by object 1014h (cf. Chapter 4.5.1).

|              |  |  |
|--------------|--|--|
| Subindex     | 00h  |  |
| Description  | Defines the COB ID of the emergency object (EMCY)                          |  |
| Access       | rw (can only be described in the "Pre-Operational" state; cf. Chapter 4.1) |  |
| PDO mapping  | No   |  |
| Data type    | UNSIGNED 32  |  |
| Default      | 80h + Node ID  |  |
| EEPROM       | Yes  |  |
| Data content | Bit 31   | 0: EMCY object exists/is valid<br>1: EMCY object does not exist/is invalid |
|              | Bit 30   | Always 0b  |
|              | Bit 29   | 0: 11 bit identifier (CAN 2.0A)<br>1: 29 bit identifier (CAN 2.0B)         |
|              | Bit 28 ... 11  | 0: if Bit 29 = 0b<br>X: Bits 28 - 11 of the EMCYCOB ID if bit 29 = 1b      |
|              | Bit 10 ... 0   | X: Bits 10 - 0 of the EMCY COB ID  |

**4.7.2.14 1017h: Producer Heartbeat Time**

The cycle time "Heartbeat Time" is set for the Heartbeat protocol by object 1017h. The cycle time width is entered in milliseconds.

|              |   |
|--------------|---|
| Subindex     | 00h   |
| Description  | Defines the cycle time of the Heartbeat monitoring service  |
| Access       | rw  |
| PDO mapping  | No  |
| Data type    | UNSIGNED 16   |
| Default      | 0   |
| EEPROM       | Yes   |
| Data content | 0d, 10d ... 65535d (0h, Ah ... FFFFh); the numerical value corresponds to a multiple of 1 ms. The value 0h deactivates the service. |

**4.7.2.15 1018h: Identity Object**

The manufacturer identification number (Vendor ID) is indicated by object 1018h.

|             |   |
|-------------|---|
| Subindex    | 00h                                     |
| Description | Displays the largest supported subindex |
| Access      | const                                   |
| PDO mapping | No                                      |
| Data type   | UNSIGNED 8                              |
| Default     | 4h                                      |
| EEPROM      | No                                      |

|             |   |
|-------------|---|
| Subindex    | 01h   |
| Description | Manufacturer identification number (Vendor ID) assigned to SIKO GmbH by CiA |
| Access      | ro  |
| PDO mapping | No  |
| Data type   | UNSIGNED 32   |
| Default     | 195h  |
| EEPROM      | No  |

|             |  |
|-------------|--|
| Subindex    | 02h  |
| Description | Product Code (function is not supported; only compatibility entry for various configurators) |
| Access      | ro   |
| PDO mapping | No   |
| Data type   | UNSIGNED 32  |
| Default     | FFFFFFFFh  |
| EEPROM      | No   |

|             |   |
|-------------|---|
| Subindex    | 03h   |
| Description | Revision Number (function is not supported; only compatibility entry for various configurators) |
| Access      | ro  |
| PDO mapping | No  |
| Data type   | UNSIGNED 32   |
| Default     | FFFFFFFFh   |
| EEPROM      | No  |

|             |               |
|-------------|---------------|
| Subindex    | 04h           |
| Description | Serial Number |
| Access      | ro            |
| PDO mapping | No            |
| Data type   | UNSIGNED 32   |
| EEPROM      | Yes           |

#### 4.7.2.16 1200h: Server SDO Parameter

The COB IDs are specified for the server SDOs by object 1200h. The COB IDs cannot be changed.

|             |   |
|-------------|---|
| Subindex    | 00h                                     |
| Description | Displays the largest supported subindex |
| Access      | const                                   |
| PDO mapping | No                                      |
| Data type   | UNSIGNED 8                              |
| Default     | 2h                                      |
| EEPROM      | No                                      |

|             |                              |
|-------------|------------------------------|
| Subindex    | 01h                          |
| Description | COB ID Client -> Server (rx) |
| Access      | ro                           |
| PDO mapping | No                           |
| Data type   | UNSIGNED 32                  |
| Default     | 00000600h + Node ID          |
| EEPROM      | No                           |

|             |                              |
|-------------|------------------------------|
| Subindex    | 02h                          |
| Description | COB ID Server -> Client (tx) |
| Access      | ro                           |
| PDO mapping | No                           |

|           |                     |
|-----------|---------------------|
| Data type | UNSIGNED 32         |
| Default   | 00000580h + Node-ID |
| EEPROM    | No                  |

#### 4.7.2.17 1800h 1. Transmit PDO Parameter

According to CiA 406, TPD01 is used for asynchronous PDO transmission. The communication parameters for TPD01 are set by object 1800h.

|             |   |
|-------------|---|
| Subindex    | 00h                                     |
| Description | Displays the largest supported subindex |
| Access      | const                                   |
| PDO mapping | No                                      |
| Data type   | UNSIGNED 8                              |
| Default     | 5h                                      |
| EEPROM      | No                                      |

|             |  |
|-------------|--|
| Subindex    | 01h  |
| Description | COB ID of PDO1   |
| Access      | rw (can only be described in the "Pre-Operational" state; cf. Chapter 4.1) |
| PDO mapping | No   |
| Data type   | UNSIGNED 32  |
| Default     | Configure Node ID  |
| EEPROM      | Yes  |

|              |                          |   |
|--------------|--------------------------|---|
| Subindex     | 02h                      |   |
| Description  | Transmission Type        |   |
| Access       | rw                       |   |
| PDO mapping  | No                       |   |
| Data type    | UNSIGNED 8               |   |
| Default      | FEh (254d)               |   |
| EEPROM       | Yes                      |   |
| Data content | FEh (254d)<br>FFh (255d) | PDO has asynchronous characteristics (PDO is sent depending on the "Event Timer").  |
|              | FDh (253d)               | The device only responds to an RTR request if RTR Bit 30 is released in the COB ID. |

|             |  |
|-------------|--|
| Subindex    | 03h  |
| Description | Inhibit time (function is not supported; only compatibility entry for various configurators) |
| Access      | ro   |
| PDO mapping | No   |

|           |             |
|-----------|-------------|
| Data type | UNSIGNED 16 |
| Default   | 0h          |
| EEPROM    | No          |

|          |  |
|----------|--|
| Subindex | 04h (is not used; access generates an error message) |
|----------|--|

|              |  |
|--------------|--|
| Subindex     | 05h  |
| Description  | Event timer for TPD01 hard-wired (CiA 406) with cyclic timer 6200h   |
| Access       | rw   |
| PDO mapping  | No   |
| Data type    | UNSIGNED 16  |
| Default      | 0h   |
| EEPROM       | Yes  |
| Data content | Writing the value 0h switches off the service. The content of this object is identical with object 6200h. If the value is changed when the timer is running, the change only takes effect after the next run of the timer. |

|          |  |
|----------|--|
| Subindex | 06h (is not used; access generates an error message) |
|----------|--|

#### 4.7.2.18 1801h: 2. Transmit PDO Parameter

According to CiA 406, TPD02 is used for synchronous PDO transmission. The communication parameters for TPD02 are set by object 1801h.

|             |   |
|-------------|---|
| Subindex    | 00h                                     |
| Description | Displays the largest supported subindex |
| Access      | const                                   |
| PDO mapping | No                                      |
| Data type   | UNSIGNED 8                              |
| Default     | 5h                                      |
| EEPROM      | No                                      |

|             |  |
|-------------|--|
| Subindex    | 01h  |
| Description | COB ID of PDO2   |
| Access      | rw (can only be described in the "Pre-Operational" state; cf. Chapter 4.1) |
| PDO mapping | No   |
| Data type   | UNSIGNED 32  |
| Default     | 280h + Node ID   |
| EEPROM      | Yes  |

|          |     |
|----------|-----|
| Subindex | 02h |
|----------|-----|



|              |                     |   |
|--------------|---------------------|---|
| Description  | Transmission Type   |   |
| Access       | rw                  |   |
| PDO mapping  | No                  |   |
| Data type    | UNSIGNED 8          |   |
| Default      | 1h                  |   |
| EEPROM       | Yes                 |   |
| Data content | 1h (1)<br>F0h (240) | PDO is sent after received 1d ... 240d SYNC messages.                               |
|              | FCh (252)           | The device only responds to an RTR request if RTR Bit 30 is released in the COB ID. |

|             |  |  |
|-------------|--|--|
| Subindex    | 03h  |  |
| Description | Inhibit time (function is not supported; only compatibility entry for various configurators) |  |
| Access      | ro   |  |
| PDO mapping | No   |  |
| Data type   | UNSIGNED 16  |  |
| Default     | 0h   |  |
| EEPROM      | No   |  |

|          |  |  |
|----------|--|--|
| Subindex | 04h (is not used; access generates an error message) |  |
|----------|--|--|

|             |   |  |
|-------------|---|--|
| Subindex    | 05h   |  |
| Description | Event timer (function is not supported; only compatibility entry for various configurators) |  |
| Access      | ro  |  |
| PDO mapping | No  |  |
| Data type   | UNSIGNED 16   |  |
| Default     | 0h  |  |
| EEPROM      | No  |  |

|          |  |  |
|----------|--|--|
| Subindex | 06h (is not used; access generates an error message) |  |
|----------|--|--|

#### 4.7.2.19 1A00h: 1. Transmit PDO Mapping Parameter

The objects are specified by object 1A00h that are depicted in the first Transmit PDO (TPDO1).

|             |                          |  |
|-------------|--------------------------|--|
| Subindex    | 00h                      |  |
| Description | Number of mapped objects |  |
| Access      | const                    |  |
| PDO mapping | No                       |  |
| Data type   | UNSIGNED 8               |  |

|         |    |
|---------|----|
| Default | 2h |
| EEPROM  | No |

|             |  |
|-------------|--|
| Subindex    | 01h  |
| Description | 1. Object of the PDO1 message (data bytes 0 to 3)                |
| Access      | ro   |
| PDO mapping | No   |
| Data type   | UNSIGNED 32  |
| Default     | 60200120h (position value of object 6020h, subindex 01h, 32 bit) |
| EEPROM      | No   |

|             |   |
|-------------|---|
| Subindex    | 02h   |
| Description | 2. Object of the PDO1 message (data bytes 4 to 5) |
| Access      | ro  |
| PDO mapping | No  |
| Data type   | UNSIGNED 32                                       |
| Default     | 60300110h (speed 6030h, subindex 01h, 16 bit)     |
| EEPROM      | No  |

#### 4.7.2.20 1A01h: 2. Transmit PDO Mapping Parameter

The objects are specified by object1A01h that are depicted in the second Transmit PDO (TPDO2).

|             |                          |
|-------------|--------------------------|
| Subindex    | 00h                      |
| Description | Number of mapped objects |
| Access      | const                    |
| PDO mapping | No                       |
| Data type   | UNSIGNED 8               |
| Default     | 2h                       |
| EEPROM      | No                       |

|             |  |
|-------------|--|
| Subindex    | 01h  |
| Description | 1. Object of the PDO2 message (data bytes 0 to 3)                |
| Access      | ro   |
| PDO mapping | No   |
| Data type   | UNSIGNED 32  |
| Default     | 60200120h (position value of object 6020h, subindex 01h, 32 bit) |
| EEPROM      | No   |

|          |     |
|----------|-----|
| Subindex | 02h |
|----------|-----|

|             |   |
|-------------|---|
| Description | 2. Object of the PDO2 message (data bytes 4 to 5) |
| Access      | ro  |
| PDO mapping | No  |
| Data type   | UNSIGNED 32                                       |
| Default     | 60300110h (speed 6030h, subindex 01h, 16 bit)     |
| EEPROM      | No  |

#### 4.7.2.21 5000h: Diagnose CAN bus error

A prioritized list of the CAN bus errors that occurred can be read via object 5000h.

|              |   |               |               |               |
|--------------|---|---------------|---------------|---------------|
| Subindex     | 00h   |               |               |               |
| Description  | Provides the CAN bus errors Acknowledge, Form, CRC and Stuff Error sorted according to frequency. |               |               |               |
| Access       | ro  |               |               |               |
| PDO mapping  | No  |               |               |               |
| Data type    | UNSIGNED 32   |               |               |               |
| Default      | 0h  |               |               |               |
| EEPROM       | No  |               |               |               |
| Data content | Byte 0  | Byte 1        | Byte 2        | Byte 3        |
|              | General Acknowledgment Error  | Form error    | CRC error     | Stuff error   |
|              | 0, 1, 2, 3, 4   | 0, 1, 2, 3, 4 | 0, 1, 2, 3, 4 | 0, 1, 2, 3, 4 |

Explanation of the data content:

0: Error does not occur at all

4: Most frequently occurring error

#### 4.7.2.22 5F09h: Bus terminal.

Internal CAN bus termination resistance can be activated by object 5F09h.

|              |  |
|--------------|--|
| Subindex     | 00h  |
| Description  | Switch CAN bus termination on or off   |
| Access       | rw   |
| PDO mapping  | No   |
| Data type    | UNSIGNED 8   |
| Default      | 0h   |
| EEPROM       | Yes  |
| Data content | 0: internal bus termination not activated<br>1: internal bus termination activated |

**4.7.2.23 5FOAh: Node ID and baud rate Bus CAN**

Node ID and the baud rate bus can be set by object 5FOAh.

|             |   |
|-------------|---|
| Subindex    | 00h                                     |
| Description | Displays the largest supported subindex |
| Access      | const                                   |
| PDO mapping | No                                      |
| Data type   | UNSIGNED 8                              |
| Default     | 2h                                      |
| EEPROM      | No                                      |

|              |             |
|--------------|-------------|
| Subindex     | 01h         |
| Description  | Node ID     |
| Access       | rw          |
| PDO mapping  | No          |
| Data type    | UNSIGNED 8  |
| Default      | 1h          |
| EEPROM       | Yes         |
| Data content | 01h ... 7Fh |

|              |  |
|--------------|--|
| Subindex     | 02h  |
| Description  | Baud rate of the CAN bus   |
| Access       | rw   |
| PDO mapping  | No   |
| Data type    | UNSIGNED 8   |
| Default      | 5h (500 kBaud)   |
| EEPROM       | Yes  |
| Data content | 1: 20 kbit/s<br>2: 50 kbit/s<br>3: 125 kbit/s<br>4: 250 kbit/s<br>5: 500 kbit/s<br>6: 800 kbit/s<br>7: 1000 kbit/s |

**4.7.2.24 6000h: Operating Parameters**

Operating parameters can be set via object 6000h.

|             |                      |
|-------------|----------------------|
| Subindex    | 00h                  |
| Description | Operating Parameters |
| Access      | rw                   |
| PDO mapping | No                   |

|              |              |  |
|--------------|--------------|--|
| Data type    | UNSIGNED 16  |  |
| Default      | 4h           |  |
| EEPROM       | Yes          |  |
| Data content | Bit 15 ... 4 | Not used   |
|              | Bit 3        | 0: Counting direction of increasing values<br>1: Counting direction of decreasing values |
|              | Bit 2        | 1: Release scaling   |
|              | Bit 1        | Not used   |
|              | Bit 0        | Not used   |

**Scaling:** The encoder works with its set resolution, which can be parameterized via object 6005h. It is not possible to switch off the scaling function.

**Count direction is positive:** increasing position values when you pull out the wire.

**Counting direction is negative:** falling position values when you pull out of the wire.

Note:

After you switch the count direction from negative to positive, a desired calibration value must be sent again.

#### 4.7.2.25 6002h: Total number of measuring steps

Object 6002h specifies the total number of measurement steps of the system

|             |                                 |
|-------------|---------------------------------|
| Subindex    | 00h                             |
| Description | Total number of measuring steps |
| Access      | rw                              |
| PDO mapping | No                              |
| Data type   | UNSIGNED 32                     |
| Default     | 73840d (00012070h)              |
| EEPROM      | Yes                             |

#### 4.7.2.26 6003h: Preset value (calibration value)

This object is not used. Cf. Object [6010h: Calibration value](#)

#### 4.7.2.27 6004h: Position value

This object is not used. Cf. Object [6020h: Position value](#)

#### 4.7.2.28 6005h: Resolution

The resolution is specified by object 6005h.

|             |   |
|-------------|---|
| Subindex    | 00h                                     |
| Description | Displays the largest supported subindex |

|             |            |
|-------------|------------|
| Access      | ro         |
| PDO mapping | No         |
| Data type   | UNSIGNED 8 |
| Default     | 2h         |
| EEPROM      | No         |

|              |  |
|--------------|--|
| Subindex     | 01h  |
| Description  | Resolution of the linear sensor. According to CiA 406, the parameter must be specified in multiples of nm. |
| Access       | rw   |
| PDO mapping  | No   |
| Data type    | UNSIGNED 32  |
| Default      | 100000d (000186A0h)  |
| EEPROM       | Yes  |
| Data content | 100000d (000186A0h)  |

|              |   |
|--------------|---|
| Subindex     | 02h   |
| Description  | Increment of the speed of the linear sensor. According to CiA 406, the parameter must be specified in multiples of 0.01 mm/s. |
| Access       | rw  |
| PDO mapping  | No  |
| Data type    | UNSIGNED 32   |
| Default      | 100d (64h)  |
| EEPROM       | Yes   |
| Data content | 100d (64h)  |

#### 4.7.2.29 6010h: Calibration value

The position value of the encoder for calibration to a calibration value can be set via object 6010h. Position value = measurement value + calibration value

|             |   |
|-------------|---|
| Subindex    | 00h                                     |
| Description | Displays the largest supported subindex |
| Access      | ro                                      |
| PDO mapping | No                                      |
| Data type   | UNSIGNED 8                              |
| Default     | 1h                                      |
| EEPROM      | No                                      |

|              |  |
|--------------|--|
| Subindex     | 01h                                    |
| Description  | Calibration value                      |
| Access       | ro                                     |
| PDO mapping  | Yes                                    |
| Data type    | SIGNED 32                              |
| Default      | 0h                                     |
| EEPROM       | No                                     |
| Data content | -73839d...73839d (FFFE91h...0001206Fh) |

#### 4.7.2.30 6020h: Position value

Object 6020h indicates the current position value of the encoder.

|             |   |
|-------------|---|
| Subindex    | 00h                                     |
| Description | Displays the largest supported subindex |
| Access      | ro                                      |
| PDO mapping | No                                      |
| Data type   | UNSIGNED 8                              |
| Default     | 1h                                      |
| EEPROM      | No                                      |

|             |                |
|-------------|----------------|
| Subindex    | 01h            |
| Description | Position value |
| Access      | ro             |
| PDO mapping | No             |
| Data type   | SIGNED 32      |
| Default     | 0h             |
| EEPROM      | No             |

Position value = measurement value + calibration value

#### 4.7.2.31 6030h: Speed

The speed can be read via object 6030h.

|          |     |
|----------|-----|
| Subindex | 00h |
|----------|-----|

|             |   |
|-------------|---|
| Description | Displays the largest supported subindex |
| Access      | ro                                      |
| PDO mapping | No                                      |
| Data type   | UNSIGNED 8                              |
| Default     | 1h                                      |
| EEPROM      | No                                      |

|             |                     |
|-------------|---------------------|
| Subindex    | 01h                 |
| Description | Speed value in mm/s |
| Access      | ro                  |
| PDO mapping | No                  |
| Data type   | SIGNED 16           |
| Default     | 0h                  |
| EEPROM      | No                  |

#### 4.7.2.32 6200h: Cycle Timer

Object 6200h sets a cycle time, with which the PDO1 is to be output. This value is firmly linked to the object [1800h 1. Transmit PDO Parameter](#) subindex 05h. The timer-controlled output is active as soon as a valid cycle time is entered and the device is operated in operational mode. The value 0h deactivates the function.

|              |                            |
|--------------|----------------------------|
| Subindex     | 00h                        |
| Description  | Cycle Timer                |
| Access       | rw                         |
| PDO mapping  | No                         |
| Data type    | UNSIGNED 16                |
| Default      | 0h                         |
| EEPROM       | Yes                        |
| Data content | 0d ... 65535d (0h...FFFFh) |

#### 4.7.2.33 6400h: Work area (area state register)

Object 6400h indicates whether the position value is within the set working ranges 1 and 2

|             |   |
|-------------|---|
| Subindex    | 00h                                     |
| Description | Displays the largest supported subindex |
| Access      | ro                                      |
| PDO mapping | No                                      |
| Data type   | UNSIGNED 8                              |
| Default     | 2h                                      |
| EEPROM      | No                                      |



|              |                      |  |
|--------------|----------------------|--|
| Subindex     | 01h                  |  |
| Description  | State of work area 1 |  |
| Access       | ro                   |  |
| PDO mapping  | No                   |  |
| Data type    | UNSIGNED 8           |  |
| Default      | 0h                   |  |
| EEPROM       | No                   |  |
| Data content | Bit 7 ... 3          | Not used   |
|              | Bit 2                | 0: Position value is within the work area<br>1: Position value is smaller than the set limit value in object 6401.1h |
|              | Bit 1                | 0: Position value is within the work area<br>1: Position value is greater than the set limit value in object 6402.1h |
|              | Bit 0                | 0: Position value is within the set work area<br>1: Position value is outside of the set work area                   |

|              |                      |  |
|--------------|----------------------|--|
| Subindex     | 02h                  |  |
| Description  | State of work area 2 |  |
| Access       | ro                   |  |
| PDO mapping  | No                   |  |
| Data type    | UNSIGNED 8           |  |
| Default      | 0h                   |  |
| EEPROM       | No                   |  |
| Data content | Bit 7 ... 3          | Not used   |
|              | Bit 2                | 0: Position value is within the work area<br>1: Position value is smaller than the set limit value in object 6401.2h |
|              | Bit 1                | 0: Position value is within the work area<br>1: Position value is greater than the set limit value in object 6402.2h |
|              | Bit 0                | 0: Position value is within the set work area<br>1: Position value is outside of the set work area                   |

#### 4.7.2.34 6401h: Work Area Low Limit

A lower limit value for one of the two working areas can be set via object 6401h.

|             |   |
|-------------|---|
| Subindex    | 00h                                     |
| Description | Displays the largest supported subindex |
| Access      | ro                                      |
| PDO mapping | No                                      |
| Data type   | UNSIGNED 8                              |
| Default     | 2h                                      |
| EEPROM      | No                                      |

|              |   |
|--------------|---|
| Subindex     | 01h                                       |
| Description  | Lower limit value of work area 1          |
| Access       | rw  |
| PDO mapping  | No  |
| Data type    | SIGNED 32                                 |
| Default      | -147678d(FFFDBF22h)                       |
| EEPROM       | Yes                                       |
| Data content | -147678d(FFFDBF22h)... 221517d(0003614Dh) |

|              |   |
|--------------|---|
| Subindex     | 02h                                       |
| Description  | Lower limit value of work area 2          |
| Access       | rw  |
| PDO mapping  | No  |
| Data type    | SIGNED 32                                 |
| Default      | -147678d(FFFDBF22h)                       |
| EEPROM       | Yes                                       |
| Data content | -147678d(FFFDBF22h)... 221517d(0003614Dh) |

#### 4.7.2.35 6402h: Work Area High Limit

A higher limit value for one of the two working areas can be set via object 6402h.

|             |   |
|-------------|---|
| Subindex    | 00h                                     |
| Description | Displays the largest supported subindex |
| Access      | ro                                      |
| PDO mapping | No                                      |
| Data type   | UNSIGNED 8                              |
| Default     | 2h                                      |
| EEPROM      | No                                      |

|              |   |
|--------------|---|
| Subindex     | 01h                                       |
| Description  | Higher limit value of work area 1         |
| Access       | rw  |
| PDO mapping  | No  |
| Data type    | SIGNED 32                                 |
| Default      | 221517d(0003614Dh)                        |
| EEPROM       | Yes                                       |
| Data content | -147678d(FFFDBF22h)... 221517d(0003614Dh) |

|          |     |
|----------|-----|
| Subindex | 02h |
|----------|-----|

|              |   |
|--------------|---|
| Description  | Higher limit value of work area 2         |
| Access       | rw  |
| PDO mapping  | No  |
| Data type    | SIGNED 32                                 |
| Default      | 221517d(0003614Dh)                        |
| EEPROM       | Yes                                       |
| Data content | -147678d(FFFDBF22h)... 221517d(0003614Dh) |

#### 4.7.2.36 6500h: Operating Status

The object 6500h indicates the settings programmed with object 6000h.

|              |                  |  |
|--------------|------------------|--|
| Subindex     | 00h              |  |
| Description  | Operating Status |  |
| Access       | ro               |  |
| PDO mapping  | No               |  |
| Data type    | UNSIGNED 16      |  |
| Default      | 4h               |  |
| EEPROM       | No               |  |
| Data content | Bit 15 ... 4     | Not used   |
|              | Bit 3            | 0: Counting direction of increasing values<br>1: Counting direction of decreasing values |
|              | Bit 2            | 1: Release scaling   |
|              | Bit 1            | Not used   |
|              | Bit 0            | Not used   |

#### 4.7.2.37 6501h: Single-turn resolution

Object 6501h specifies the physical number of measuring steps per revolution.

|             |                     |
|-------------|---------------------|
| Subindex    | 00h                 |
| Description | Physical resolution |
| Access      | ro                  |
| PDO mapping | No                  |
| Data type   | UNSIGNED 32         |
| Default     | 100000d (000186A0h) |
| EEPROM      | No                  |

#### 4.7.2.38 6502h: Number of distinguishable revolutions

Object 6502h specifies the number of revolutions that the encoder can record.

|             |                     |
|-------------|---------------------|
| Subindex    | 00h                 |
| Description | Physical resolution |

|             |             |
|-------------|-------------|
| Access      | ro          |
| PDO mapping | No          |
| Data type   | UNSIGNED 16 |
| Default     | 1           |
| EEPROM      | No          |

#### 4.7.2.39 6503h: Alarms

In addition to the errors that are reported via the emergency message, object 6503h provides additional device-specific alarm messages. In the event of an error, the corresponding bit is set to 1.

|              |                |  |
|--------------|----------------|--|
| Subindex     | 00h            |  |
| Description  | Alarm messages |  |
| Access       | ro             |  |
| PDO mapping  | No             |  |
| Data type    | UNSIGNED 16    |  |
| Default      | 0h             |  |
| EEPROM       | No             |  |
| Data content | Bit 15 ... 14  | Not used   |
|              | Bit 13         | 0: Position value within the working range 2<br>1: Position limit value 2 exceeded or fallen below (work area 2) |
|              | Bit 12         | 0: Position value within the working range 1<br>1: Position limit value 1 exceeded or fallen below (work area 1) |
|              | Bit 11 ... 1   | Not used   |
|              | Bit 0          | 0: Position value valid<br>1: Position value invalid   |

#### 4.7.2.40 6504h: Supported Alarms

This object 6504h indicates which alarm messages are supported. The corresponding bits are set.

|              |                          |                               |
|--------------|--------------------------|-------------------------------|
| Subindex     | 00h                      |                               |
| Description  | Supported alarm messages |                               |
| Access       | ro                       |                               |
| PDO mapping  | No                       |                               |
| Data type    | UNSIGNED 16              |                               |
| Default      | 3001h                    |                               |
| EEPROM       | No                       |                               |
| Data content | Bit 15 ... 14            | Not used                      |
|              | Bit 13                   | Position limit value 2 errors |
|              | Bit 12                   | Position limit value 1 error  |

|  |              |                |
|--|--------------|----------------|
|  | Bit 11 ... 1 | Not used       |
|  | Bit 0        | Position error |

#### 4.7.2.41 6505h: Warnings

Warning messages can be output via object 6505h. In the case of a warning message, the position value can still be valid unlike in the case of an alarm message.

|              |              |          |
|--------------|--------------|----------|
| Subindex     | 00h          |          |
| Description  | Warnings     |          |
| Access       | ro           |          |
| PDO mapping  | No           |          |
| Data type    | UNSIGNED 16  |          |
| Default      | 0h           |          |
| EEPROM       | No           |          |
| Data content | Bit 0 ... 15 | Not used |

#### 4.7.2.42 6506h: Supported Warnings

Object 6506h indicates which warning messages are supported.

|              |                    |          |
|--------------|--------------------|----------|
| Subindex     | 00h                |          |
| Description  | Supported warnings |          |
| Access       | ro                 |          |
| PDO mapping  | No                 |          |
| Data type    | UNSIGNED 16        |          |
| Default      | 0000h              |          |
| EEPROM       | No                 |          |
| Data content | Bit 0 ... 15       | Not used |

#### 4.7.2.43 6507h: Profile and Software Version

The object 6507h displays the used encoder profile (CANopen device profiles for encoders) and the firmware version number.

|             |                              |        |                  |            |
|-------------|------------------------------|--------|------------------|------------|
| Subindex    | 00h                          |        |                  |            |
| Description | Profile and Software Version |        |                  |            |
| Access      | ro                           |        |                  |            |
| PDO mapping | No                           |        |                  |            |
| Data type   | UNSIGNED 32                  |        |                  |            |
| Default     | 01000302h                    |        |                  |            |
| EEPROM      | No                           |        |                  |            |
|             | Profile version              |        | Software version |            |
|             | Byte 0 (LSB)                 | Byte 1 | Byte 2           | Byte 3 MSB |

|  |     |     |     |     |
|--|-----|-----|-----|-----|
|  | 02h | 03h | 01h | 00h |
|--|-----|-----|-----|-----|

#### 4.7.2.44 6508h: Operating Time

The operating hours can be displayed via object 6508h. This function is not supported.

|             |                         |
|-------------|-------------------------|
| Subindex    | 00h                     |
| Description | Operation hours counter |
| Access      | ro                      |
| PDO mapping | No                      |
| Data type   | UNSIGNED 32             |
| Default     | FFFFFFFFh               |
| EEPROM      | No                      |

#### 4.7.2.45 6509h: Offset value

Object 6509h specifies the difference between the encoder value and the scaled position value calculated with the calibration value.

|             |   |
|-------------|---|
| Subindex    | 00h                                       |
| Description | Encoder states at the time of calibration |
| Access      | ro  |
| PDO mapping | No  |
| Data type   | SIGNED 32                                 |
| Default     | 0h  |
| EEPROM      | Yes                                       |

#### 4.7.2.46 650Ah: Module identification

Object 650Ah specifies the manufacturer-specific offset value as well as the smallest and largest transferable position value

|             |   |
|-------------|---|
| Subindex    | 00h                                     |
| Description | Displays the largest supported subindex |
| Access      | ro                                      |
| PDO mapping | No                                      |
| Data type   | UNSIGNED 8                              |
| Default     | 3h                                      |
| EEPROM      | No                                      |

|             |                                    |
|-------------|------------------------------------|
| Subindex    | 01h                                |
| Description | Manufacturer-specific offset value |
| Access      | ro                                 |
| PDO mapping | No                                 |

|           |           |
|-----------|-----------|
| Data type | SIGNED 32 |
| Default   | 0h        |
| EEPROM    | No        |

|             |                                      |
|-------------|--------------------------------------|
| Subindex    | 02h                                  |
| Description | Smallest transferable position value |
| Access      | ro                                   |
| PDO mapping | No                                   |
| Data type   | SIGNED 32                            |
| Default     | -147678d(FFFDBF22h)                  |
| EEPROM      | No                                   |

|             |                                     |
|-------------|-------------------------------------|
| Subindex    | 03h                                 |
| Description | Largest transferable position value |
| Access      | ro                                  |
| PDO mapping | No                                  |
| Data type   | SIGNED 32                           |
| Default     | 221517d(0003614Dh)                  |
| EEPROM      | No                                  |

#### 4.7.2.47 650Bh: Serial number

Object 650Bh supplies the serial number of the encoder.

|             |               |
|-------------|---------------|
| Subindex    | 00h           |
| Description | Serial number |
| Access      | ro            |
| PDO mapping | No            |
| Data type   | UNSIGNED 32   |
| Default     | 0h            |
| EEPROM      | Yes           |

#### 4.7.2.48 650Ch: Offset value for multi-sensor device

Object 650C9h specifies the difference between the encoder value and the scaled position value calculated with the calibration value (equivalent to object [6509h: Offset](#) value).

|             |   |
|-------------|---|
| Subindex    | 00h                                     |
| Description | Displays the largest supported subindex |
| Access      | ro                                      |
| PDO mapping | No                                      |
| Data type   | UNSIGNED 8                              |
| Default     | 1h                                      |

|        |    |
|--------|----|
| EEPROM | No |
|--------|----|

|             |              |
|-------------|--------------|
| Subindex    | 01h          |
| Description | Offset value |
| Access      | ro           |
| PDO mapping | No           |
| Data type   | SIGNED 32    |
| Default     | 0h           |
| EEPROM      | Yes          |