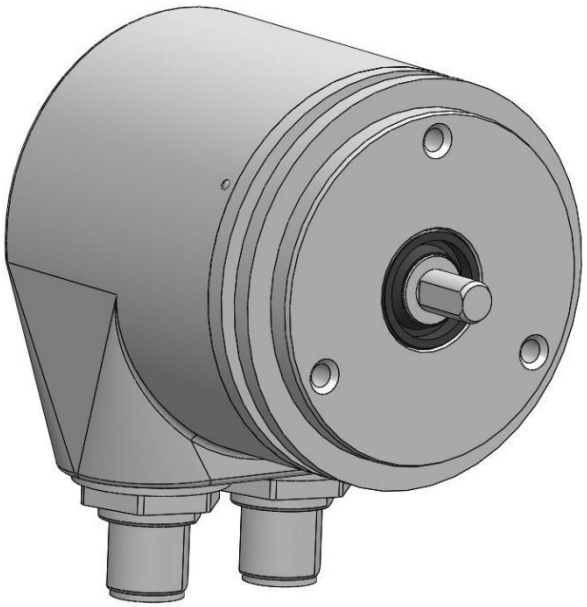
# WV58MR/WH58MR

## Redundant rotary encoder with CANopen Safety interface extension

User manual



066/19

## Inhaltsverzeichnis

# 1 General Information

## 1.1 Documentation

The following documents are associated with this product:

- Data sheet; it describes the technical data, the dimensions, the pin assignments, the accessories and the order key.
- Mounting instructions; they describe the mechanical and electrical installation with all safety-relevant conditions and the associated technical specifications.
- User manual; for commissioning the sensor and integrating it into a fieldbus system.
- EDS file (electronic data sheet); this file enables integration and configuration in a CANopen network by means of commercial CANopen configurators.

These documents can also be found at ***http://www.siko-global.com/p/wv58mr***

and ***http://www.siko-global.com/p/wh58MR*** .

## 1.2 Definitions

Decimal values are given as numbers without addition (e.g. 1234), except when indicated in direct connection with binary or hexadecimal values, in which case the extension d will be used (e.g. 1234d). Binary values are identified by adding b (e.g. 1011b) to the figures whereas hexadecimal values are extended by h (e.g. 280h).

# 2 Intended use

The WV58MR / WH58MR consists of two encoders, which collect redundantly absolute travel information. By means of the CANopen protocol and CANopen Safety protocol, both encoders can be configured and read out via the CAN interface. The evaluation and assessment of the data must occur in the superordinate control unit.

Both encoders are equipped with each 3 LEDs (yellow, red, green), which indicate error or status information for diagnostic purposes.

The rotary encoder WV58MR / WH58MR is designed for redundant position and speed detection. It can be used for applications up to Performance Level D (PLd) in the overall system. For this purpose an overriding safe encoder evaluation device is required. Since the encoder is incapable with its encoder-internal diagnostic function to initiate actions such as obtaining a safe state on its own. Increased demands are placed on the electrical and mechanical connection of the rotary encoder.

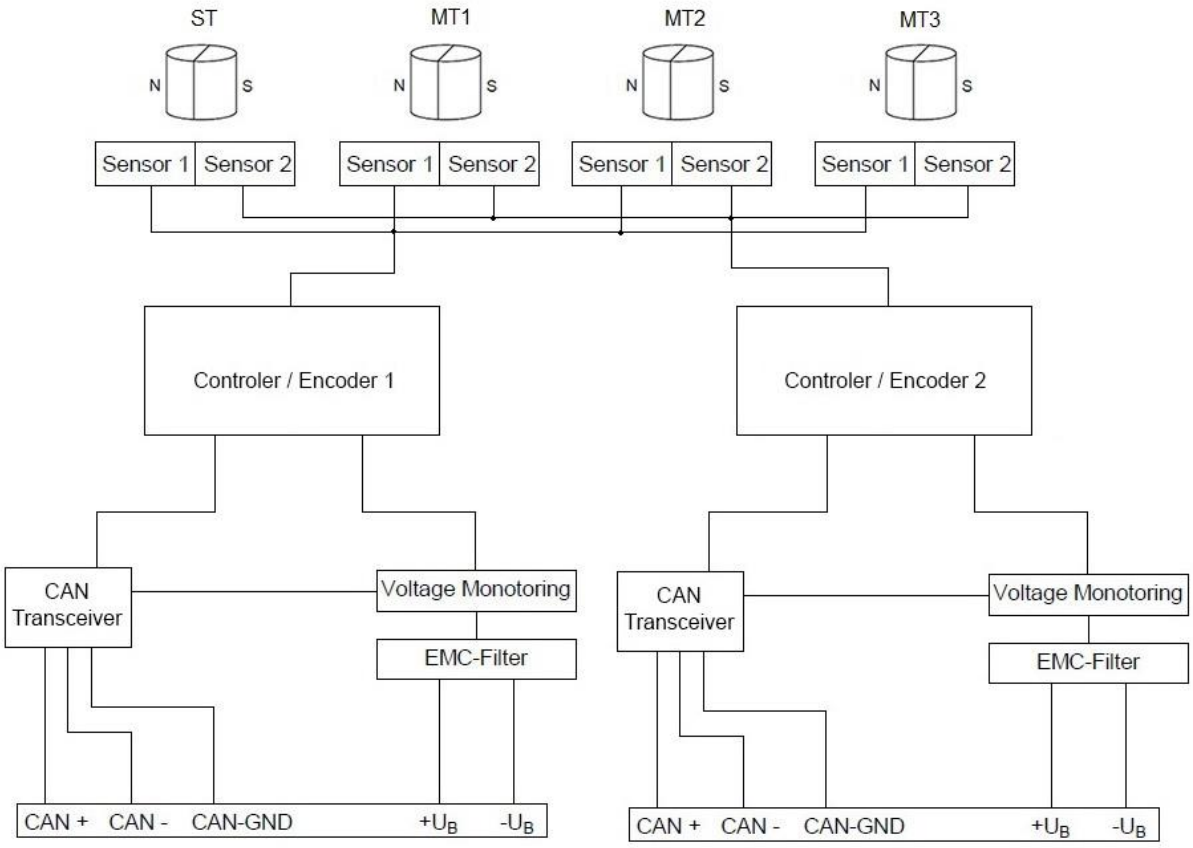



*Fig. 1: Block diagram*

## 2.1 Switching on the supply voltage

WV58MR / WH58MR initializes after being switched on. During initialization, the LEDs light up one after the other and the configuration parameters are loaded from the non-volatile memory to the random memory of the controller.

Each sensor will work with its default values as long as no changes have been made to it. With parameters changed, the sensor will work with the changed data, which must be stored if they are intended to be used after power off/on.

After completing the initialization procedure, each sensor sends a specific NMT command, the boot-up message, which informs the system about their availability. The WV58MR / WH58MR is now in the pre-operational mode. In this state, the encoder can be parameterized via SDO commands in accordance with the requirements of the application. This applies to configuration parameters of the sensor unit as well as to the way it makes available to the system its position values (asynchronous or synchronous data transmission.

# 3 LED-signal

Each transmitter has 3 LEDs in the colors yellow, green and red for diagnosis and status purposes.

- A yellow LED for device-specific states
- A green LED for indicating the NMT status or the LSS configuration status (CAN Run LED)
- A red LED for CAN error states or for indicating the LSS configuration status (CAN Err LED)

The LSS waiting status is not indicated via the LEDs.

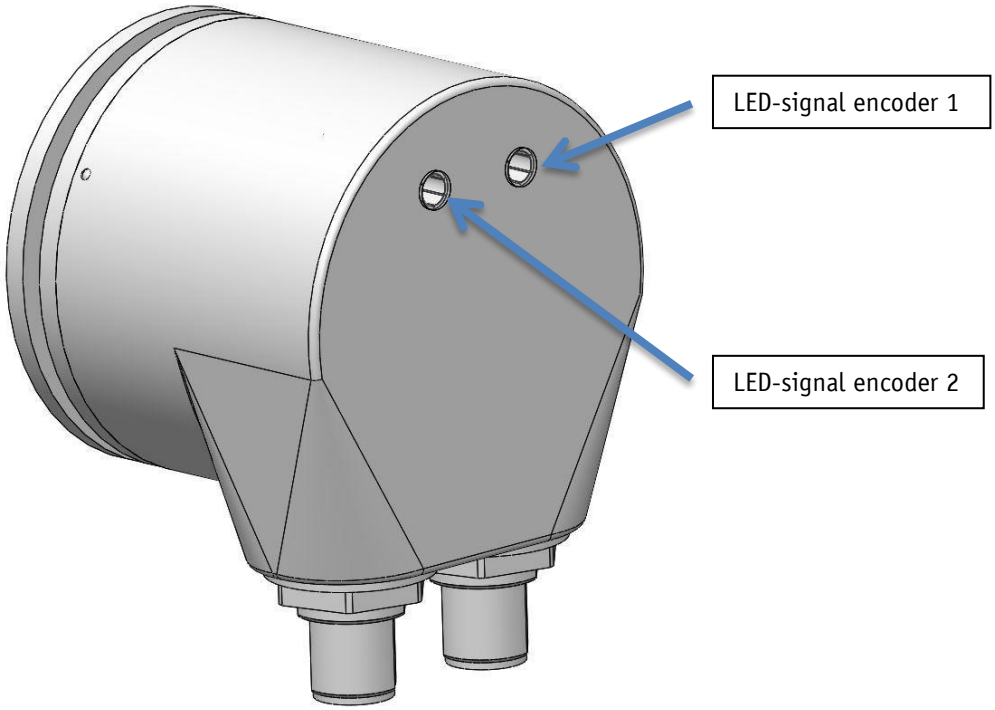



*Fig. 2: LED-signal*

### Device-specific diagnosis

| Error status | LED status |
|---|---|
| Maximum speed exceeded | On |
| encoder is in the valid speed range | Off |

*Table 1: Device-specific status LED*

### CAN diagnosis:

The CiA DS-303 Part 3 V1.4.0 indicator specification is the basis of the CAN diagnosis

| LED status | Description |
|---|---|
| On | LED is permanently on |
| Off | LED is permanently off |
| Flickering | Both LEDs alternately with the frequency of 10 Hz |

| LED status | Description |
|---|---|
| | (50 ms on/off) |
| Flashing | LED flashes at a frequency of 2.5 Hz (200 ms on/off) |
| Single Flash | LED is 200 ms on, 1000 ms off |
| Double Flash | LED is 200 ms on, 200 ms off, 200 ms on, 1000 ms off |

*Table 2: CAN LED statuses acc. to CiA DS-303*

**CAN Run LED:**

| NMT state | LED status |
|---|---|
| Pre-Operational | Flashing |
| Operational | On |
| Stopped | Single Flash |

*Table 3: CAN Run LED*

**CAN Err LED:**

| Error states | LED status |
|---|---|
| No error | Off |
| Warning limit reached (at least one error counter (Transmit Error Counter CANTEC or Receive Error Counter CANREC) of the CAN controller has reached or exceeded the warning limit (too many error frames). | **Single Flash** |
| Error control event<br>=> A Guard Event (if no RTR Node guard received from master within the lifetime set) | **Double Flash** |
| Bus off | On |

*Table 4: CAN Err LED*

**CAN Run LED and CAN Err LED alternately:**

| LSS state | LED status |
|---|---|
| configuration | Flickering |

*Table 5: LSS configuration*

# 4 Functional description

## 4.1 Measuring range

The measuring range depends on the chosen device design and the APU set.

| Design | Default measuring range | With changed APU (Object 6001h) |
|---|---|---|
| Singleturn | 0…16383 | 0…((APU*1) – 1) |

| Design | Default measuring range | With changed APU (Object 6001h) |
|---|---|---|
| 4 Bit Multiturn | 0...262143 | 0...((APU*16) – 1) |
| 8 Bit Multiturn | 0...4194303 | 0...((APU*256) – 1) |
| 12 Bit Multiturn | 0...67108863 | 0...((APU*4096) – 1) |

**Counting direction:**

The encoder delivers ascending position values when the shaft is rotated clockwise (CW, view on the shaft). This property can be changed via Object 6000h: Operating Parameters

## 4.2 Calibration

Owing to the absolute system, calibration is required only once when the system is taken into operation and can be performed at any position. This enables alignment of the encoder zero point with the system's mechanical zero point. With calibration, the calibration value is adopted for calculation of the position value. The resulting offset value is output in Object 6509h: Offset value. The following equation is applied in case of calibration:

Position value = 0 + calibration value + application offset

## 4.3 Reset to factory settings

To return to the original condition of the device as delivered, there exist the following options:

| Access | Coding | | Settings are restored |
|---|---|---|---|
| CANopen (see Object 1011h: Restore Parameter) | 1011h "load" | Sub-index 1 | All parameters |
| | | Sub-index 2 | Only bus parameters |
| | | Sub-index 3 | Only CiA DS-406 parameters |
| | | Sub-index 4 | Only manufacturer-specific parameters |

*Table 6: Access to factory settings*

# 5 Communication via CAN-Bus (CANopen and CANopen Safety)

The CANopen communication profile CiA DS-301 V4.2, the Device profile for Encoders CiA DS-406 V4.02 as well as the indicator specification CiA DS-303 Part 3 V1.4.0 for CAN diagnosis form the basis for the WV58MR / WH58MR. The CANopen Safety protocol EN50325-5 is additionally implemented for safety-relevant applications.

Safety-relevant applications as well as safety-non-relevant applications can be operated on a CAN bus.

The WV58MR / WH58MR supports device class C3. The details required for a better understanding of safety-relevant operation are included in this documentation. If more in-depth information is required, we recommend reading the applicable technical literature on CANopen Safety. For fundamental information on CANopen communication refer to the WV58MR / WH58MR CANopen manual.

## 5.1 Telegram structure

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

| SOF: | Identifier (COB-ID) | Control field: | Data field (max. 8 byte) | CRC | ACK/EOF: |
|---|---|---|---|---|---|

**SOF:**

Start of Frame start bit of the telegram

**Identifier (COB-ID):**

- By means of the identifier, all bus subscribers check whether the message is relevant for each of them.
- The identifier determines the priority of the message. The lower the value of the identifier, the higher is the priority of the message This enables preferential transmission of important messages via the bus.

The Identifier field contains the identifier as well as bits for the recognition of the length of the identifiers (11 or 29 bits). The device address, channel selection as well as data direction are determined via the identifier as well.

Thus, the 11bits identifier (COB identifier) consists of a 4bit function code and a 7bit node number.

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

The following functional codes have been defined in the “Pre-defined Connection Set” (only the functional codes used in the present device are shown):

| Object | Functional code | Resulting COB-ID | Object | Page |
|---|---|---|---|---|
| Network management (NMT) | 0000b | 0 | - | 11 |
| SYNC message | 0001b | 128d (80h) | 1005h | 56 |
| Emergency message | 0001b | 128d (80h) + Node-ID | 1014h | 61 |
| TPD01 | 0011b | 384d (180h) + Node-ID | 1800h | 64 |
| TPD02 | 0101b | 640d (280h) + Node-ID | 1801h | 74 |
| SDO (tx) | 1011b | 1408d (580h) + Node-ID | 1200h | 63 |
| SDO (rx) | 1100b | 1536d (600h) + Node-ID | 1200h | 63 |
| Heartbeat message | 1110b | 1792d (700h) + Node-ID | - | 24 |
| Node Guard message | 1110b | 1792d (700h) + Node-ID | - | 23 |
| LSS (tx) | - | 2021d (7E4h) | - | 24 |
| LSS (rx) | - | 2020d (7E5h) | - | 24 |

*Table 7: Overview of COB identifiers*

Changes to COB IDs are only possible in the PRE-OPERATIONAL NMT status. First, the COB ID must be switched invalid via bit 31 = 1b before it can be changed and reactivated.
The COB ID of the Sync object is an exception, where bit 30 must be = 0b to enable the COB ID to be changed. As bit 30 cannot be set to 1b in the WV58MR / WH58MR, the COB ID could be changed at any time.

The node number (Node ID) (see also object 5F0Ah: Node-ID and baud rate Bus CAN) is assigned once in every bus system with configuration of the master on WV58MR / WH58MR. The node numbers range from 1 to 127. Node ID = 0 is reserved and must not be used.

The adoption of a node ID or baud rate which was reset occurs only after re-initialization (see chapter 5.2.1: Network management (NMT) services).

With the WV58MR / WH58MR rotary encoder, an encoder with the node ID 1 (1h) and the redundant encoder with the node ID 2 (2h) is delivered ex works.

**Control field:**

contains bit-by-bit information concerning the number of user data and determines whether a data frame or RTR frame (Remote Transmission Request frame) is concerned.

**Data field:**

contains up to 8 bytes of user data. The user data has a different meaning depending on the channel selection.

**CRC:**

contains bits for error detection.

**ACK/EOF:**

The ACK/EOF field contains telegram acknowledgment bits as well as bits for determining the end of telegram.

For a detailed description of the telegram please refer to the applicable technical CAN literature. For simplification, only identifier (COB ID) and data field will be dealt with in the subsequent telegram descriptions.

## 5.2 Node control

### 5.2.1 Network management (NMT) services

The master configures, manages and monitors network nodes via the NMT service. The device is always in one of the four communication states "INITIALIZATION", "PRE-OPERATIONAL", "OPERATIONAL" or "STOPPED" (see Fig. 3)

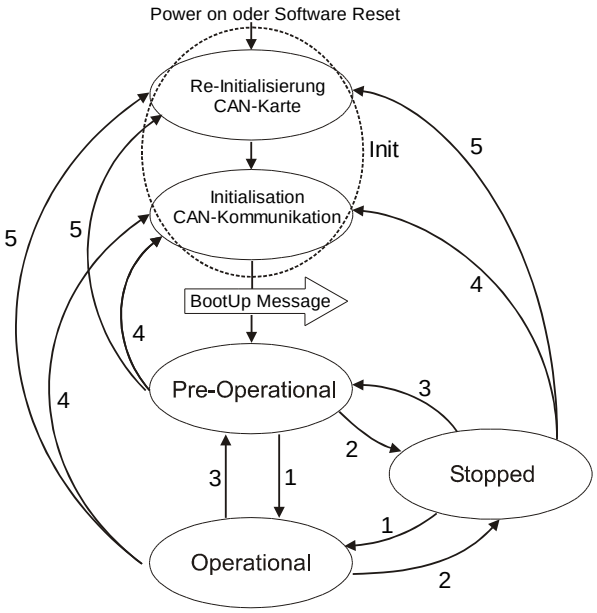



*Fig. 3: NMT Status diagram*

### 5.2.1.1 NMT communication states

**NMT Status 'INITIALIZATION'**

The device is not involved in the bus actions in this state. All hardware and software components are initialized. This state is attained after switching on the device or after receipt of the command code 81h ("Reset node") of the own or global addresses. Following receipt of the command code 82h ("Reset Communication"), the display will enter the initialization stage as well. But only hardware and software associated with CAN communication will be reinitialized. The device signals automatically the completion of initialization by sending a boot-up message. As soon as the boot-up message was sent successfully, the device will enter the "PRE-OPERATIONAL" status.

**NMT Status PRE-OPERATIONAL**

Parameterization data (SDO) can be exchanged in the pre-operational mode. However, no process data (PDO's) is transferred.

**NMT Status OPERATIONAL**

The exchange of process data is enabled as well. However, COB-ID and Transmit PDO Mapping parameters can no longer be changed in this status.

**NMT Status STOPPED**

Communication is stopped except for heartbeat and node guarding Only NMT communication is enabled.

#### 5.2.1.2 Toggling between the NMT communication states

For toggling between the communication states, telegrams with the following structures are used

| Change of state | | Transition in Fig. 3 | COB-ID | Com-mand | 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 (Reset Node) | 5d | 0h | 81h | x |
| OPERATIONAL / PRE-OPERATIONAL / STOPPED | INITIALIZATION (Reset Communication) | 4d | 0h | 82h | x |

*Table 8: Toggling between communication states*

If x = 0h is transferred as node ID, then the message is intended for all bus subscribers

### 5.2.2 Boot-Up

The COB ID of the boot-up message is made up of 700h and the node ID. The “Initialization” NMT status is output as data content.

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

*Table 9: Boot-Up message*

### 5.2.3 SYNC object

CANopen enables the simultaneous query of all inputs and the simultaneous setting of all outputs. The synchronization message (SYNC), a CAN message with high priority serves this purpose. The identifier of the Sync object can be set via object 1005h (see 1005h: COB-ID SYNC-message).

## 5.3 Process data exchange

### 5.3.1 Transfer of process data objects (PDO)

Process data objects (PDO) serve for fast exchange of process data. A maximum of 8 bytes of user data can be transferred in a PDO. The WV58MR / WH58MR supports the Transmit PDO services TPDO1 and TPDO2 according to CiA DS-301 and CiA DS-406.

#### 5.3.1.1 Transmit PDO (from the WV58MR / WH58MR to the master)

PDO transfer from the display to the bus master (TPDO) can be initiated as a result of various events:

- asynchronous, controlled by an internal device timer
- synchronous as a response to a SYNC telegram
- as a response to an RTR message

TPDO1 and TPDO2 are generated from the position value and the speed value. The transfer behavior of TPDO1 is determined via the objects 1800h, 1A00h and 6200h and is assigned to asynchronous transmission. TPDO2 is defined via the objects 1801h and 1A01h and serves synchronous transmission. Assignment is static and cannot be changed.

Messages are structured as shown in Table 10.

| COB-ID | Process data in binary code | | | | | |
|---|---|---|---|---|---|---|
| | Byte 0 (LSB) | Byte 1 | Byte 2 | Byte 3 (MSB) | Byte 4 (LSB) | Byte 5 (MSB) |
| TPDO1<br>180h + Node-ID | Position value | | | | Speed value | |
| TPDO2<br>280h + Node-ID | | | | | | |

*Table 10: TPDO message*

**Asynchronous data transmission (TPDO1)**

If a TPDO1 is to be sent cyclically, then the cycle time must be entered in milliseconds into object 1800h, sub-index 05h. The TPDO1 will not be sent if the value 0 ms is written. The function is disabled. The minimum value to be set is 1 (= 1 ms). Alternately, the value can also be written into the permanently internally linked object 6200h.

**Synchronous data transfer (TPDO2)**

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

**RTR**

Queries can be sent via RTR (see chapter 5.1: Telegram structure, control field) toTPDO1 and TDPO2.

## 5.4 Parameter data exchange

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

Service data objects serve mainly device configuration via the directory of objects. SDOs in the expedited Request/Response and in the normal normal Request/Response are supported.

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

Two SDO services are available:

- SDO (rx) (Master → WV58MR / WH58MR): 600h + Node-ID
- SDO (tx) (WV58MR / WH58MR → Master): 580h + Node-ID

These SDO identifiers cannot be changed!

#### 5.4.1.1 Expedited Request/Response

Except for reading the object 1008h: Manufacturer Device Name, all SDOs are exchanged between two subscribers in the expedited Request/Response method. The user data is provided already with the initialization message.

SDO messages are set up as follows:

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

**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 WV58MR / WH58MR:

| Command byte | | Type | Function |
|---|---|---|---|
| Write Request | 23h | SDO (rx), Initiate Download Request, expedited | Send parameter to slave (All 4 data bytes valid) |
| Write Request | 2Bh | SDO (rx), Initiate Download Request, expedited | Send parameter to slave (2Bytes of 4 data bytes valid) |
| Write Request | 2Fh | SDO (rx), Initiate Download Request, expedited | Send parameter to slave (1Byte of 4 data bytes valid) |
| Write Response | 60h | SDO (tx), Initiate Download Response | Acknowledgment of data acquisition to master |
| Read Request | 40h | SDO (rx), Initiate Upload Request | Request parameter from slave |
| Read Response | 43h | SDO (tx), Initiate Upload Response, expedited | Report parameter to master (All 4 data bytes valid) |
| Read Response | 4Bh | SDO (tx), Initiate Upload Response, expedited | Report parameter to master (2Bytes of 4 data bytes valid) |

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

*Table 11: Command coding*

**Index, bytes 1 and 2:**

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

**Sub-index, Byte 3:**

The sub-index indicates the number of the fields for objects realized as an array.

**User data (parameters), byte 4-7:**

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

#### 5.4.1.2 Normal Request/Response

If more than 4 bytes of service data are to be transferred, the data is exchanged between two subscribers via the normal Request/Response. This procedure is also initiated by an initialization message, and the actual user data will be transferred in the subsequent segment messages.

For the WV58MR / WH58MR this is only the case with reading of the object 1008h: Manufacturer Device Name.

The initialization message has the following structure:

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

The segment message has the following structure:

| COB-ID | User data in binary code | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Byte 0 (read / write) | Byte 1 LSB | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 (MSB) |
| SDO rx/tx + Node-ID | Command 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 WV58MR / WH58MR:

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

*Table 12: Command coding*

**Initialization message : Index, bytes 1 and 2:**

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

**Initialization message : Sub-index, Byte 3:**

The subindex indicates the number of the fields for objects realized as an array.

**Initialization message : User data (parameters), byte 4-7:**

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

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

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

#### 5.4.1.3 Error Response in SDO exchange

With invalid access, an error message (Abort) is returned to the master.
The error codes are described in the CANopen profile (CiA DS-301) or in the encoder profile (CiA DS- 406), respectively. The table below shows the error codes used:

| Error code | Description |
|---|---|
| 05030000h | Toggle bit in Normal Transfer of Request/Response unequal. |
| 06010000h | Wrong access to an object. |
| 06010001h | Read access to Write-Only |
| 06010002h | Write access to Read-Only. |

| Error code | Description |
|---|---|
| 06020000h | Object doesn't exist in the object directory. |
| 06090011h | Sub-index does not exist |
| 06090030h | Wrong value range of selected parameter. |
| 08000020h | Parameters cannot be transferred to application or stored. |
| 08000022h | Parameters cannot be transferred to application or stored due to the current device status. |
| 08000024h | No data available |

*Table 13: Error codes*

#### 5.4.1.4 SDO examples

**Example of reading SDO parameters with the expedited Request/Response:**

The calibration value stored in object 6003 of the directory of objects is to be read from the slave with device address 1h.

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

Command: 40h
index: 6003h
sub-index: 00h
The current value is 510d = 01FEh

Request of master from slave with node ID 1h:

| COB-ID | User data | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Command | Index L | Index H | Sub-index | Data 0 | Data 1 | Data 2 | Data 3 |
| 601h | 40h | 03h | 60h | 00h | x | x | x | x |

Response to the request by the slave

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

| COB-ID | User data | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Command | Index LB | Index HB | Sub-index | Data 0 | Data 1 | Data 2 | Data 3 |
| 581h | 43h (4 bytes valid) | 03h | 60h | 00h | FEh | 01h | 00h | 00h |

**Example of writing SDO parameters with the expedited Request/Response:**

The calibration value stored with 2 bytes in object 6002 of the directory of objects is to be changed in the slave with device address 1h.

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

Command: 2 bytes are to be written 2Bh
Index: 6200h
Sub-index: 00h
The new value shall be 4500d = 1194h

Writing of a value from master to slave with node ID 1h:

| COB-ID | User data | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Command | Index L | Index H | Sub-index | Data 0 | Data 1 | Data 2 | Data 3 |
| 601h | 2Bh (2 bytes valid) | 00h | 62h | 00h | 94h | 11h | 00h | 00h |

Response to the command by the slave:

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

| COB-ID | User data | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Command | Index L | Index H | Sub-index | 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/Response:**

The manufacturer device name stored in object 1008h of the directory of objects is to be read from the WV58MR / WH58MR with device address 1h.

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

Command: 40h

Index: 1008h

Sub-index: 00h

First request (= initialization) of master from slave with node ID 1h:

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

Response to the request by the slave

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

| COB-ID | User data | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Command | Index LB | Index HB | Sub-index | Data 0 | Data 1 | Data 2 | Data 3 |
| 581h | 41h | 08h | 10h | 00h | 06h | 00h | 00h | 00h |

Number of expected user data bytes: 6

Second request of master from slave with node ID 1h:

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

Response to the request by the slave

| COB-ID | User data | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Command | Data 0 | Data 1 | Data 2 | Data 3 | Data 4 | Data 5 | Data 6 |
| 581h | 03h | 57h ("W") | 56h ("V") | 35h ("5") | 38h ("8") | 4Dh ("M") | 52h ("R") | 00h - |

## 5.5 Safety data exchange

### 5.5.1 Transfer of safety-relevant data objects (SRDO)

• Safety-relevant data objects (SRDO) serve the cyclic exchange of safe data in the Operational NMT state. An SRDO consists always of 2 messages with different COB-IDs and bit-by-bit inverted data. A maximum of 8 bytes of user data can be transferred in an SRDO. The WV58MR / WH58MR supports the Safety Transmit services SRDO1for the position value (4 bytes) and SRDO2 for speed (2 bytes) according to EN50325-5 and CiA DS-406. The transmit behavior of SRDO is determined via the objects 1301h, 1381h and 6100h. SRDO is determined via the objects 1302h, 1382h and 6101h. Mapping is static and cannot be changed.

•

| COB-ID | Data in binary code | | | |
|---|---|---|---|---|
| | Byte 0 (LSB) | Byte 1 | Byte 2 | Byte 3 (MSB) |
| SRDO1 COB-ID1<br>000000FFh + 2*Node-ID | Position value | | | |
| SRDO1 COB-ID2<br>00000100h + 2*Node-ID | Bit-inverted position value | | | |

*Table 14: SRDO1 message*

•

| COB-ID | Data in binary code | |
|---|---|---|
| | Byte 0 (LSB) | Byte 1 |
| SRDO2 COB-ID1<br>0000010Fh + 2*Node-ID | speed value | |
| SRDO2 COB-ID1<br>00000110h + 2*Node-ID | Bit-inverted speed value | |

*Table 15: SRDO2 message*

•

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

• All safety concerning service data objects for device configuration are addressed in expedited Request/Response

• All changes to safety concerning service data objects are monitored via a checksum CRC-16-CCITT and can only be executed in the Pre-Operational NMT state. The polynomial is g(x) = $x^{16}+x^{12}+x^5+1$.

• Procedure of changing the configuration data in the objects 1301h, 1302h, 6100h and 6101h:

- Change values
- Enter the new valid checksum (sub-index of the relevant object 13FFh or object 61FFh).
- Switch the relevant configuration to valid via object 13FEh or 61FEh.
-

### 5.5.3 Example for calculating a checksum

• In the following example the checksum CRC-16-CCITT (polynomial 1021h) for SRDO2 should be calculated. The checksum is calculated byte by byte and the low byte by the contents of the two objects 1302h and 1382h.

| Object | Sub-index | Name | Größe | Data |
|---|---|---|---|---|
| 1302h | SRDO2 Communication parameter | | | |
| | 01h | Direction of information | 1 Byte | 01h |
| | 02h | Refresh time | 2 Byte | 0019h |
| | 03h | tx: reserved<br>rx: SRVT | 1 Byte | 14h |
| | 05h | COB-ID 1 | 4 Byte | 00000111h |
| | 06h | COB-ID 2 | 4 Byte | 00000112h |
| 1382h | SRDO2 Mapping Parameter | | | |
| | 00h | Highest sub-index | 1 Byte | 04h |
| | 01h | Sub-index | 1 Byte | 01h |
| | 01h | Speed value 1.Byte | 4 Byte | 61240108h |
| | 02h | Sub-index | 1 Byte | 02h |
| | 02h | Inverted speed value 1.Byte | 4 Byte | 61250108h |
| | 03h | Sub-index | 1 Byte | 03h |
| | 03h | Speed value 2.Byte | 4 Byte | 61240208h |
| | 04h | Sub-index | 1 Byte | 04h |
| | 04h | Inverted speed value 2.Byte | 4 Byte | 61250208h |

• The following data bytes are used to calculate the checksum:

• 0x01 0x19 0x00 0x14 0x11 0x01 0x00 0x00 0x12 0x01 0x00 0x00 0x04 0x01 0x08 0x01 0x24 0x61 0x02 0x08 0x01 0x25 0x61 0x03 0x08 0x02 0x24 0x61 0x04 0x08 0x02 0x25 0x61

• Checksum = 1C7Fh

### 5.5.4 Example of change of a configuration

• In the following example, the direction of information is to be changed from valid to invalid for SRDO2 while maintaining he configuration of SRDO1.

•

• Changing the direction of information:

| COB-ID | User data | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Command | Index L | Index H | Sub-index | Data 0 | Data 1 | Data 2 | Data 3 |
| 600h + Node-ID | 2Fh | 02h | 13h | 01h | 00h | - | - | - |

•

• For the checksum, the default value 0000h is entered. Therefore, both checksums must be indicated for the first change even if only one configuration is changed.

| COB-ID | User data | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Command | Index L | Index H | Sub-index | Data 0 | Data 1 | Data 2 | Data 3 |
| 600h + Node-ID | 2Bh | FFh | 13h | 01h | 0Dh | 25h | - | - |
| 600h + Node-ID | 2Bh | FFh | 13h | 02h | 7Fh | 1Ch | - | - |

•

• Enable the configuration for SRDO1 and SRDO2. During activation, the configuration is monitored by means of the checksum in object 13FFh sub-indexes 01h and 02h. The configurations can only be enabled with the correct checksums.

| COB-ID | User data | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Command | Index L | Index H | Sub-index | Data 0 | Data 1 | Data 2 | Data 3 |
| 600h + Node-ID | 2Fh | FEh | 13h | 00h | A5h | - | - | - |

## 5.6 Node monitoring

### 5.6.1 Emergency Service (EMCY)

In the case of an error, the status of the bus subscriber is transferred via high-priority emergency messages (emergency telegrams). These messages have a data length of 8 bytes and contain error information.

The emergency message is transferred as soon as a sensor or communication error has occurred or when such errors have been corrected. The cause of the error is deposited in the error buffer (see object 1003h: Pre-defined Error Field). An emergency object is sent only once per error event. Removal of the cause of the error is signaled by sending an emergency message with the error code 0000h (no error). If multiple errors have occurred and one cause of error is removed, the error code 0000h is output as well; the persisting error status is indicated in the error register, however.

| 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 |
|---|---|
| Cause of the error removed | 0000h |
| Bus status changed over to the error passive mode | 8120h |
| recovered from Bus Off | 8140h |
| Manufacturer-specific Position value error | FF05h |
| Manufacturer-specific Velocity error | FF12h |
| Manufacturer-specific Error limit speed low | FF13h |
| Manufacturer-specific Error limit speed high | FF14h |
| Manufacturer-specific Position error work area 1 | FF15h |
| Manufacturer-specific Position error work area 2 | FF16h |

*Table 16: Emergency Error Code*

The identifier of the emergency object is set to 80h + node ID by default; however, it can be changed via object 1014h (see 1014h: COB-ID Emergency message). Transmission of an emergency message is enabled in the NMT statuses “OPERATIONAL” or “PRE-OPERATIONAL” only! Transmission of the emergency messages can be disabled by setting the COB-ID Valid bit to 1.

### 5.6.2 Node Guarding

Node guarding is available for failure monitoring of the CANopen network. During node guarding, the master transmits remote frames (RTR, remote transmit request, message request telegrams) on the guarding identifiers of the nodes to be monitored. The latter respond with the guarding message. This message contains the current NMT status of the node as well as a toggle bit whose value must change after each message. The master assumes that a node error has occurred if status or toggle bits do not correspond with those expected by the master or if there is no response.

Via objects 100Ch (Guard Time) and 100Dh (Life Time Factor) the time interval (Life-Time) is set within which the NMT master expects to receive a response. The time interval “Life Time” is calculated from the cycle time “Guard Time”, multiplied with the factor “Life Time Factor”. If the NMT master does not receive a response to its RTR frame within the “Life Time”, it may react with suitable measures. Upon switching on, node guarding will be enabled by sending the first RTR frame of the master to the slave. Node Guarding is deactivated if the value of either object (100Ch or 100Dh) is set to 0h.

The answer of the node to the RTR frame of the master is formed 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 subsequent responses of the device. After the guarding protocol has been enabled, the toggle bit must have the value 0 with 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. A node guard message can be sent in the NMT statuses “OPERATIONAL”, “PRE-OPERATIONAL” or “STOPPED”.

Note:

Literature recommends heartbeat to be used for node monitoring. Only the master can detect missing communication via the node guarding protocol as opposed to the heartbeat that can be received by all subscribers.

### 5.6.3 Heartbeat

The master monitors the state of the slave device via Heartbeat protocol. While doing this, the device sends independently its NMT status cyclically. The WV58MR / WH58MR is a heartbeat producer, it does not receive nor process heartbeat protocols itself. The cycle time of the heartbeat message is set via object 1017h. The heartbeat protocol is deactivated if the cycle time is 0h.

The heartbeat message consists of the COB ID and an additional byte. In this byte, the current NMT state is deposited.

| 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. Heartbeat messages are sent in the NMT statuses “OPERATIONAL”, “PRE-OPERATIONAL” or “STOPPED”.

## 5.7 Layer Setting Service (LSS)

Layer Setting Service (LSS) is a special method described in CiA DS-305it serves for retrieving and configuring various parameters (node ID, baud rate, and Identity Object 1018h).

Every device must have a unique LSS number composed of the entries in Object 1018h.

- Vendor ID: 0000 0195h
- Product Code: FFFF FFFFh

- Revision number: FFFF FFFFh
- Serial number: xxxx xxxxh (serial number of the encoder concerned)

In order to enable the use of full LSS functionality, all devices on the bus must support the LSS method. An LSS master must exist and all nodes must start with the same baud rate. After starting, the device will be in the LSS waiting state. To enable configuration, one or all devices must be switched to the LSS configuration state. If the LSS master expects to receive an answer to its command, only one LSS slave must be switched to the LSS configuration mode. If an encoder is in the LSS configuration mode, this will be indicated via the green and red LEDs (see chapter 3 LED-signal).

Two LSS services are available:

- LSS (rx) (LSS Master → WV58MR / WH58MR): 7E5h
- LSS (tx) (WV58MR / WH58MR → LSS Master): 7E4h

These LSS identifiers cannot be changed!

A message consists always of 8 bytes. Byte 0 contains the command (Command – Specifier cs), followed by max. 7 data bytes Unused data bytes are reserved and must be filled 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 17: State behavior of the supported LSS services*

### 5.7.1 State change

#### 5.7.1.1 Switch states of all LSS devices (Switch state global)

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

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

### 5.7.1.2 Switch states of individual LSS devices (Switch state selective)

With this command, Individual LSS slave devices can be set to the LSS Configuration state via the unique LSS number.

Master → WV58MR / WH58MR

| 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 |

WV58MR / WH58MR → 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 |

## 5.7.2 Configuration

### 5.7.2.1 Setting the node ID (Configure Node-ID)

The LSS master can configure the node ID of single LSS slaves switched to the configuration mode. If the new node ID is intended to still be available after Power off/on, the “Save configuration” command must be output after the change. For immediate activation of the new node ID, the LSS slave must be set to the LSS Waiting mode, followed by an NMT "Reset Communication" 82h. Another possibility would be to execute power off/on after “Save configuration”.

Master → WV58MR / WH58MR

| 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

WV58MR / WH58MR → 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:no valid node ID

FFh: Implementation error see Spec error

**Spec error:**

This byte is nonzero only in case of an implementation error and Error Code FFh

### 5.7.2.2 Configuration of the baud rate (Configure bit timing parameters)

The baud rate of a single or of multiple LSS slaves can be configured via this command. If the new baud rate is intended to still be available after Power off/on, the “Save configuration” command must be output after the change. To activate the new baud rate the 5.7.2.3 Activate baud rate (Activate bit timing parameters) command must be output and the LSS slave set to the LSS Waiting state. Another possibility of activating the new baud rate would be to execute power off/on after “Save configuration”.

Master → WV58MR / WH58MR

| | User data |
|---|---|

| COB-ID | 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 DS-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 supports only Table selector 00h and Table index 0 until 7.

WV58MR / WH58MR → 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: no valid baud rate

FFh: Implementation error see Spec error

**Spec error:**

This byte is nonzero only in case of an implementation error and Error Code FFh

### 5.7.2.3 Activate baud rate (Activate bit timing parameters)

This command activates the new baud rate set via 5.7.2.2 Configuration of the baud rate (Configure bit timing parameters) without requiring Power off/on.

Master → WV58MR / WH58MR

| 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 periods (d1, d2) of the same length and must correspond with a multiple of 1 ms. After expiry of the individual processing time and delay time d1, the new baud rate will be adopted internally. After expiry of the delay time d2, the LSS slave will report with the boot up via the newly set baud rate. This procedure prevents the synchronous presence on the bus of devices with different baud rates. The LSS slave cannot send messages during the two delay periods d1 and d2.

#### 5.7.2.4 Store configuration

This command must only be executed if only one LSS slave is in the configuration mode. The current settings will be stored subsequently.

Master → WV58MR / WH58MR

| 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 |

WV58MR / WH58MR → 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 occurred during storing

FFh: Implementation error see Spec error

**Spec error:**

This byte is nonzero only in case of an implementation error and Error Code FFh

### 5.7.3 Requesting parameters

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

#### 5.7.3.1 Request Vendor ID

Master → WV58MR / WH58MR

| 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 |

WV58MR / WH58MR → 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 (see Object 1018.1h) | | | | 00h | 00h | 00h |

#### 5.7.3.2 Request Product Code

Master → WV58MR / WH58MR

| 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 |

WV58MR / WH58MR → 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 (see Object 1018.2h) | | | | 00h | 00h | 00h |

#### 5.7.3.3 Request revision number

Master → WV58MR / WH58MR

| 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 |

WV58MR / WH58MR → 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 |

#### 5.7.3.4 Request serial number

Master → WV58MR / WH58MR

| 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 |

WV58MR / WH58MR → 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 |

#### 5.7.3.5 Request Node ID

Master → WV58MR / WH58MR

| 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 |

WV58MR / WH58MR → 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 | 00h |

## 5.8 Directory of objects

### 5.8.1 Overview of objects

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

| Name | Description | See page |
|---|---|---|
| 1000h: Device Type | Device profile and encoder type | 54 |
| 1001h: Error Register | Current error state of the device | 54 |
| 1002h: Manufacturer Status Register | Contains the Transmit Error Counter and the Receive Error Counter | 55 |
| 1003h: Pre-defined Error Field | the object stores the 8 error states that have occurred last | 55 |
| 1005h: COB-ID SYNC-message | Setting of the COB ID of the SYNC object | 56 |
| 1008h: Manufacturer Device Name | Device name in ASCII notation | 56 |
| 1009h: Manufacturer Hardware Version | Indicates the hardware version of the device | 56 |
| 100Ah: Manufacturer Software Version | Indicates the software version of the device | 57 |
| 100Ch: Guard Time | Parameter for Node Guarding | 57 |
| 100Dh: Life Time Factor | Parameter for Node Guarding | 57 |
| 1010h: Store Parameter | Object for non-volatile storage of the settings | 58 |
| 1011h: Restore Parameter | Object for restoring the factory settings | 59 |
| 1014h: COB-ID Emergency message | COB ID of the Emergency object | 61 |
| 1017h: Producer Heartbeat Time | Setting of the cycle time of the heartbeat timer | 62 |
| 1018h: Identity Object | Contains the manufacturer number | 62 |
| 1200h: Server SDO Parameter | SDO parameter | 63 |

| | | | |
|---|---|---|---|
| | | Transmit PDO for asynchronous transfer (timercontrolled) | 64 |

### 1.1 1301h: SRDO1 communication parameters

The communication behavior of SRDO1 can be determined via object 1301h.

| Sub-index | 00h |
|---|---|
| Description | indicat |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGN |
| Default | 6h |
| EEPROM | no |

| Sub-index | 01h |
|---|---|
| Description | Directi |
| Access | ro, if N<br>rw, if N |
| PDO mapping | no |
| Data type | UNSIGN |
| Default | 1 |
| EEPROM | yes |
| Data content | 0: SRD<br>1: SRD |

| Sub-index | 02h |
|---|---|
| Description | refresh |
| Access | ro, if N<br>rw, if N |
| PDO mapping | no |
| Data type | UNSIGN |
| Default | 25 |
| EEPROM | yes |
| Data content | 10...65 |

| Sub-index | 03h |
|---|---|
| Description | tx: rese<br>rx: SRV |
| Access | ro, if N |

| | | | | |
|---|---|---|---|---|
| | rw, if NMT state is pre-operational | | | |
| PDO mapping | no | | | |
| Data typ | UNSIGNED 8 | | | |
| Default | 20 | | | |
| EEPROM | no | | | |
| Data content | 20 | | | |

| | |
|---|---|
| Sub-index | 04h |
| Description | transm |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGN |
| Default | 254 |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 05h |
| Description | COB-ID |
| Access | ro, if N<br>rw, if N |
| PDO mapping | no |
| Data type | UNSIGN |
| Default | 000000 |
| EEPROM | yes |
| Data content | 257...3 |

| | |
|---|---|
| Sub-index | 06h |
| Description | COB-ID |
| Access | ro, if N<br>rw, if N |
| PDO mapping | no |
| Data type | UNSIGN |
| Default | 000001 |
| EEPROM | yes |
| Data content | 258...3 |

### 1.2 1302h: SRDO2 communication parameters

The communication behavior of SRDO2 can be determined via 1302h.

| | |
|---|---|
| Sub-index | 00h |
| Description | indicat |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGN |
| Default | 6h |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 01h |
| Description | Directi |
| Access | ro, if N<br>rw, if N |
| PDO mapping | no |
| Data type | UNSIGN |
| Default | 1 |
| EEPROM | yes |
| Data content | 0: SRDO<br>1: SRDO |

| | |
|---|---|
| Sub-index | 02h |
| Description | refresh |
| Access | ro, if N<br>rw, if N |
| PDO mapping | no |
| Data type | UNSIGN |
| Default | 25 |
| EEPROM | yes |
| Data content | 10...65 |

| | |
|---|---|
| Sub-index | 03h |
| Description | tx: rese<br>rx: SRV |
| Access | ro, if N<br>rw, if N |
| PDO mapping | no |
| Data type | UNSIGN |
| Default | 20 |
| EEPROM | nein |
| Dateninhalt | 20 |

| | |
|---|---|
| Sub-index | 04h |
| Description | transm |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGN |
| Default | 254 |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 05h |
| Description | COB-ID |
| Access | ro, if N<br>rw, if N |
| PDO mapping | no |
| Data type | UNSIGN |
| Default | 000001 |
| EEPROM | yes |
| Data content | 257...3 |

| | |
|---|---|
| Sub-index | 06h |
| Description | COB-ID |
| Access | ro, if N<br>rw, if N |
| PDO mapping | no |
| Data type | UNSIGN |
| Default | 000001 |
| EEPROM | yes |
| Data content | 258...3 |

### 1.3 1381h: SRDO1 mapping parameters

Object 1381h determines the objects that are mapped in the first Transmit SRDO (SRDO1).

| | |
|---|---|
| Sub-index | 00h |
| Description | indicat |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGN |

| | |
|---|---|
| Default | 8h |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 01h |
| Description | 1st obje |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGN |
| Default | 612001 |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 02h |
| Description | 1st obje |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGN |
| Default | 612101 |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 03h |
| Description | 2nd obje |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGN |
| Default | 612002 |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 04h |
| Description | 2nd obje |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGN |
| Default | 612102 |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 05h |
| Description | 3rd obje |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGN |

| | |
|---|---|
| Default | 61200308h (position value object 6120h, sub-index 03h, 8bit) |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 06h |
| Description | 3rd obje |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGN |
| Default | 612003 |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 07h |
| Description | 4th obje |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGN |
| Default | 612004 |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 08h |
| Description | 4th obje |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGN |
| Default | 612104 |
| EEPROM | no |

## 1.4 1382h: SRDO2 mapping parameters

Object 1382h determines the objects that are mapped in the second Safety Transmit SRDO (SRDO2).

| | |
|---|---|
| Sub-index | 00h |
| Description | indicat |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGN |
| Default | 4h |

| | |
|---|---|
| EEPROM | no |

| | |
|---|---|
| Sub-index | 01h |
| Description | 1st obje |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGN |
| Default | 612401 |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 02h |
| Description | 1st obje |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGN |
| Default | 612501 |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 03h |
| Description | 2nd obje |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGN |
| Default | 612402 |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 04h |
| Description | 2nd obje |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGN |
| Default | 612502 |
| EEPROM | no |

## 1.5 13FEh: Safety configuration

The SRDO configuration can be switched to valid

by means of object 13FEh.

| | |
|---|---|
| Sub-index | 00h |
| Description | This pa<br>parame<br>this ob<br>entered |
| Access | rw |
| PDO mapping | no |
| Data type | UNSIGN |
| Default | 0h |
| EEPROM | yes |
| Data content | A5h: SI<br>00h...A |

### 1.6 13FFh: Safety configuration signature (checksum)

This object 13FFh receives the signatures (checksums) via the CANopen Safety parameters of SRDO1 and SRDO2. Only a checksum which is valid at that time can be transmitted. The checksum is checked anew before switching the configuration to valid. Any change to the configuration will be valid only after passing.

| | |
|---|---|
| Sub-index | 00h |
| Description | indicat |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGN |
| Default | 2h |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 01h |
| Description | SRDO1 |
| Access | rw |

| | | | | | |
|---|---|---|---|---|---|
| PDO mapping | no | | | | |
| Data type | UNSIGNED 16 | | | | |
| Default | 0000h | | | | |
| EEPROM | yes | | | | |
| Data content | The checksum CRC-16-CCITT is calculated via the content of the two objects 1301h and 1381h. | | | | |
| | Object | Sub-index | Name | Extent | Value |
| | 1301h | SRDO1 communication parameter | | | |
| | | 01h | Direction of information | 1 byte | Object 1301h sub-index 01h |
| | | 02h | Refresh time | 2 bytes | Object 1301h sub-index 02h |
| | | 03h | tx: reserved<br>rx: SRVT | 1 Byte | Object 1301h sub-index 03h |
| | | 05h | COB ID 1 | 4 bytes | Object 1301h sub-index 05h |
| | | 06h | COB ID 2 | 4 bytes | Object 1301h sub-index 06h |
| | 1381h | SRDO1 mapping parameters | | | |
| | | 00h | Highest sub-index | 1 byte | 08h |
| | | 01h | Sub-index | 1 byte | 01h |
| | | 01h | Position value 1st byte | 4 bytes | 61200108h |
| | | 02h | Sub-index | 1 byte | 02h |
| | | 02h | Inverted position value 1st byte | 4 bytes | 61210108h |
| | | 03h | Sub-index | 1 byte | 03h |
| | | 03h | Position value 2nd byte | 4 bytes | 61200208h |
| | | 04h | Sub-index | 1 byte | 04h |
| | | 04h | Inverted position value 2nd byte | 4 bytes | 61210208h |
| | | 05h | Sub-index | 1 byte | 05h |
| | | 05h | Position value 3rd byte | 4 bytes | 61200308h |
| | | 06h | Sub-index | 1 byte | 06h |
| | | 06h | Inverted position value 3rd byte | 4 bytes | 61210308h |
| | | 07h | Sub-index | 1 byte | 07h |
| | | 07h | Position value 4th byte | 4 bytes | 61200408h |
| | | 08h | Sub-index | 1 byte | 08h |
| | | 08h | Inverted position value 4th byte | 4 bytes | 61210408h |

| | |
|---|---|
| Sub-index | 02h |
| Description | SRDO2 |

| Name | | Description | | | | See page |
|---|---|---|---|---|---|---|
| Access | rw | | | | | |
| PDO mapping | no | | | | | |
| Data type | UNSIGNED 16 | | | | | |
| Default | 0000h | | | | | |
| EEPROM | yes | | | | | |
| Data content | The checksum CRC-16-CCITT is calculated via the content of the two objects 1302h and 1382h. | | | | | |
| | Object | Sub-index | Name | Extent | Value | |
| | 1302h | SRDO2 communication parameter | | | | |
| | | 01h | Direction of information | 1 byte | Object 1302h sub-index 01h | |
| | | 02h | Refresh time | 2 bytes | Object 1302h sub-index 02h | |
| | | 03h | tx: reserved<br>rx: SRVT | 1 Byte | Object 1302h sub-index 03h | |
| | | 05h | COB ID 1 | 4 bytes | Object 1302h sub-index 05h | |
| | | 06h | COB ID 2 | 4 bytes | Object 1302h sub-index 06h | |
| | 1382h | SRDO2 mapping parameters | | | | |
| | | 00h | Highest sub-index | 1 byte | 04h | |
| | | 01h | Sub-index | 1 byte | 01h | |
| | | 01h | speed value 1st byte | 4 bytes | 61240108h | |
| | | 02h | Sub-index | 1 byte | 02h | |
| | | 02h | Inverted speed value 1st byte | 4 bytes | 61250108h | |
| | | 03h | Sub-index | 1 byte | 03h | |
| | | 03h | speed value 2nd byte | 4 bytes | 61240208h | |
| | | 04h | Sub-index | 1 byte | 04h | |
| | | 04h | Inverted speed value 2nd byte | 4 bytes | 61250208h | |
| 1800h: 1. Transmit PDO Parameter | | | | | | |
| 1801h: 2. Transmit PDO Parameter | | Transmit PDO for synchronous transfer | | | | 74 |
| 1A00h: 1. Transmit PDO Mapping Parameter | | Describes the arrangement of the objects,which are mapped in TPDO1 | | | | 75 |
| 1A01h: 2. Transmit PDO Mapping Parameter | | Describes the arrangement of the objects,which are mapped in TPDO2 | | | | 76 |
| 2001h: Application offset | | manufacturer-specific offset value (is added to the position value encoder-internally) | | | | 77 |
| 2002h: Calibrate encoder value | | Set the position value to the calibration value | | | | 77 |
| 2003h: Limit speed low | | Setting of the lower limit speed | | | | 78 |

| Name | Description | See page |
|---|---|---|
| 2004h: Limit speed High | Setting of the upper limit speed | 78 |
| 5000h: Diagnosis CAN Bus error | Informs on the CAN bus errors that occurred | 78 |
| 5F0Ah: Node-ID and baud rate Bus CAN | Setting of Node ID and baud rate | 78 |
| 6000h: Operating Parameters | Setting of scaling and sense of rotation | 79 |
| 6001h: Measurement steps per revolution (Display per revolution = APU) | Setting of the displayed measurement steps per revolution (Display per revolution = APU) | 80 |
| 6002h: Overall number of measurement steps | Indicates the overall number of the system's measuring steps | 80 |
| 6003h: Preset value (calibration value) | Setting the calibration value | 80 |
| 6004h: Position value | position value (offset with calibration and offset value) | 81 |
| 600Ch: Absolute accuracy | Raw value of position | 82 |
| 6030h: Velocity value | Velocity value | 82 |
| 6031h: Speed parameters | adjustment the speed parameter | 82 |

| | | | |
|---|---|---|---|
| | | Identical with object 1800h, sub-index 5 | 82 |

### 1.7 6100h: Safety configuration parameters of position

Via object 6100h, settings for the position and its transmission can be made.

| Sub-index | 00h |
|---|---|
| Description | indicat |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGN |
| Default | 2h |
| EEPROM | no |

| Sub-index | 01h |
|---|---|
| Description | Safety |
| Access | rw (wri |
| PDO mapping | no |
| Data type | UNSIGN |
| Default | 0000h |
| EEPROM | yes |
| Data content | 0: CW<br>1: CCW |

| Sub-index | 02h |
|---|---|
| Description | Safety |
| Access | rw (wri |
| PDO mapping | no |
| Data type | SIGNED |
| Default | 0h |
| EEPROM | yes |
| Data content | Encode |
| | Single- |
| | 4 bit m |
| | 8 bit M |
| | 12 bit turn |

## 1.8 6101h: Safety configuration parameters of speed

Via object 6101h, settings for speed and its transmission can be made.

| Sub-index | 00h |
|---|---|
| Description | indicat |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGN |
| Default | 7h |
| EEPROM | no |

| Sub-index | 01h |
|---|---|
| Description | Safety |
| Access | rw (wri |
| PDO mapping | no |
| Data type | UNSIGN |
| Default | 0000h |
| EEPROM | yes |
| Data content | 0: CW<br>1: CCW |

| Sub-index | 02h |
|---|---|
| Description | Safety |
| Access | rw (wri |
| PDO mapping | no |
| Data type | SIGNED |
| Default | 0h |
| EEPROM | yes |
| Data content | Encode |
| | Single- |
| | 4 bit m |
| | 8 bit m |
| | 12 bit turn |

| | | | | |
|---|---|---|---|---|
| Sub-index | 04h | | | |
| Description | Safety speed source selector | | | |
| Access | rw (write in the Pre-operational NMT state only) | | | |
| PDO mapping | no | | | |
| Data type | UNSIGNED 8 | | | |
| Default | 02h | | | |
| EEPROM | yes | | | |
| Data content | 02h: Object 600C raw value of position is used | | | |

| | |
|---|---|
| Sub-index | 05h |
| Description | Safety |
| Access | rw (wri |
| PDO mapping | no |
| Data type | UNSIGN |
| Default | 64h |
| EEPROM | yes |
| Data content | 64h: 1C<br>C8h: 2C |

| | |
|---|---|
| Sub-index | 06h |
| Description | Safety |
| Access | rw (wri |
| PDO mapping | no |
| Data type | UNSIGN |
| Default | 01h |
| EEPROM | yes |
| Data content | 01h |

| | |
|---|---|
| Sub-index | 07h |
| Description | Safety |
| Access | rw (wri |
| PDO mapping | no |
| Data type | UNSIGN |
| Default | 01h |
| EEPROM | yes |
| Data content | 01h |

Speed value [Inc/ms] = (new raw position value object 600Ch - old raw position value object 600Ch) / (integration time [ms] object 6101h

sub-index 05h * 10-3) * multiplier object 6101h sub-index 06h / divisor object 6101h sub-index 07h

## 1.9 6120h: Safety position value

Via object 6120h, the position value can be read byte by byte.

| | |
|---|---|
| Sub-index | 00h |
| Description | indicat |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGN |
| Default | 4h |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 01h |
| Description | Safety |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGN |
| Default | 0h |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 02h |
| Description | Safety |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGN |
| Default | 0h |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 03h |
| Description | Safety |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGN |
| Default | 0h |

| | |
|---|---|
| EEPROM | no |

| | |
|---|---|
| Sub-index | 04h |
| Description | Safety |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGN |
| Default | 0h |
| EEPROM | no |

## 1.10 6121h: Safety inverted position value

Via object 6121h, the inverted position value can be read byte by byte.

| | |
|---|---|
| Sub-index | 00h |
| Description | indicat |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGN |
| Default | 4h |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 01h |
| Description | Safety |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGN |
| Default | FFh |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 02h |
| Description | Safety |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGN |
| Default | FFh |
| EEPROM | no |

| | | | | |
|---|---|---|---|---|
| Sub-index | 03h | | | |
| Description | Safety inverted position value byte 3 | | | |
| Access | ro | | | |
| PDO mapping | no | | | |
| Data type | UNSIGNED 8 | | | |
| Default | FFh | | | |
| EEPROM | no | | | |

| | |
|---|---|
| Sub-index | 04h |
| Description | Safety |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGN |
| Default | FFh |
| EEPROM | no |

## 1.11 6124h: Safety speed value

Via object 6124h, the speed value can be read byte by byte.

| | |
|---|---|
| Sub-index | 00h |
| Description | indicat |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGN |
| Default | 2h |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 01h |
| Description | Safety |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGN |
| Default | 0h |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 02h |
| Description | Safety |
| Access | ro |

| | |
|---|---|
| PDO mapping | no |
| Data type | UNSIGNED 8 |
| Default | 0h |
| EEPROM | no |

Conversion of increments per ms in rpm

Speed [rpm] = speed value [Inc/ms] * 6000 [rpm] / 1638.4 [Inc/ms]

## 1.12 6125h: Safety inverted speed value

Via object 6125h, the inverted speed value can be read byte by byte.

| | |
|---|---|
| Sub-index | 00h |
| Description | indicat |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGN |
| Default | 2h |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 01h |
| Description | Safety |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGN |
| Default | FFh |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 02h |
| Description | Safety |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGN |
| Default | FFh |
| EEPROM | no |

### 1.13 61FEh: Safety application configuration

Via object 61FEh, the configuration of position and speed can be switched to valid.

| | |
|---|---|
| Sub-index | 00h |
| Description | This pa<br>parame<br>this ob<br>entered |
| Access | rw |
| PDO mapping | no |
| Data type | UNSIGN |
| Default | 0h |
| EEPROM | yes |
| | A5h: Co<br>00h...A |

### 1.14 61FFh: Safety configuration signature (checksum)

This object 61FFh receives the signatures (checksums) via the CANopen Safety parameters for position and speed. Only a checksum which is valid at that time can be transmitted. The checksum is checked anew before switching the configuration to valid. Any change to the configuration will be valid only after passing.

| | |
|---|---|
| Sub-index | 00h |
| Description | indicat |
| Access | ro |
| PDO mapping | no |

| | |
|---|---|
| Data type | UNSIGNED 8 |
| Default | 2h |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 01h |
| Description | Signatu |
| Access | rw |
| PDO mapping | no |
| Data type | UNSIGN |
| Default | 0000h |
| EEPROM | yes |
| Data content | The che<br>6100h.<br>Object<br>6100h |

| | |
|---|---|
| Sub-index | 02h |
| Description | Signatu |
| Access | rw |
| PDO mapping | no |
| Data type | UNSIGN |
| Default | 0000h |
| EEPROM | yes |
| Data content | The che<br>6101h.<br>Object<br>6101h |

| Name | | | Description | | | See page |
|---|---|---|---|---|---|---|
| | | | 03h | Safety speed source selector | 1 byte | Object 6101h sub-index 04h |
| | | | 04h | Sub-index | 1 byte | 05h |
| | | | 04h | Safety integration time of speed | 2 bytes | Object 6101h sub-index 05h |
| | | | 05h | Sub-index | 1 byte | 06h |
| | | | 05h | Safety multiplier | 2 bytes | 0001h |
| | | | 06h | Sub-index | 1 byte | 07h |
| | | | 06h | Safety divisor | 2 bytes | 0001h |
| 6200h: Cycle timer | | | | | | |
| 6400h: Operating range (Area state register) | | | Indicates whether the position value is within the sewt work areas 1 and 2. | | | 101 |
| 6401h: Work Area Low Limit | | | Setting of the lower limits of the work areas 1 and 2 | | | 102 |
| 6402h: Work Area High Limit | | | Setting of the lower limits of the work areas 1 and 2 | | | 103 |
| 6500h: Operating Status | | | Output of scaling and sense of rotation | | | 101 |
| 6501h: Single-turn resolution | | | The physical number of measurement steps per revolution | | | 104 |
| 6502h: Number of distinguishable revolutions | | | Number of revolution the encoder is able to sense | | | 104 |
| 6503h: Alarms | | | indication of error states | | | 105 |
| 6504h: Supported Alarms | | | indicates the alarm messages that are supported | | | 105 |
| 6505h: Warnings | | | indication of warnings | | | 106 |
| 6506h: Supported Warnings | | | indicates the warnings that are supported | | | 106 |
| 6507h: Profile and Software Version | | | Indicates the version number of the device profile used and the version number of the encoder's firmware | | | 104 |
| 6508h: Operating Time | | | Hourmeter (function is not supported) | | | 107 |
| 6509h: Offset value | | | Encoder state at the time of calibration | | | 107 |
| 650Ah: Module Identification | | | Indicates the manufacturer-specific offset value as well as the smallest and largest transferable position value | | | 107 |
| 650Bh: Serial number | | | Indicates the serial number | | | 108 |
| 650Dh: Absolute accuracy | | | provides the absolute accuracy of the encoder in bits | | | 109 |
| 650Eh: Device functionality | | | provides information on device functionality | | | 109 |

*Table 18: Overview of objects*

### 5.8.2 Object Description

#### 5.8.2.1 1000h: Device Type

Object 1000h indicates the device profile number.

| | | | | |
|---|---|---|---|---|
| Subindex | 00h | | | |
| Description | Information about the device profile and encoder type | | | |
| Access | ro | | | |
| PDO-Mapping | no | | | |
| Data type | UNSIGNED 32 | | | |
| Default | CANopen:<br>Singleturn: 00010196h<br>Multiturn: 00020196h | | | |
| EEPROM | no | | | |
| Data content | Device profile -number | | Gebertyp | |
| | Byte 0 | Byte 1 | Byte 2 | Byte 3 |
| | 96h | 01h | 01h / 02h | 00h |

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

CANopen:

0001h: Absolute single-turn encoder

0002h: Absolute multi-turn encoder

#### 5.8.2.2 1001h: Error Register

Object 1001h indicates the error state of the device.

| | | |
|---|---|---|
| Sub-index | 00h | |
| Description | pending error status | |
| Access | ro | |
| PDO mapping | no | |
| Data type | UNSIGNED 8 | |
| Default | 0h | |
| EEPROM | no | |
| Data content | Bit | Meaning |
| | 0 | set bit indicates the occurrence of any error condition |
| | 4 | set bit indicates communication error on the CAN bus (passive or bus-off) |
| | 7 | manufacturer-specific (sensor error) |
| | 1-3, 5-6 | Not used |

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

#### 5.8.2.3 1002h: Manufacturer Status Register

Object 1002h outputs the counter readings of the “Receive Error Counter” and “Transmit Error Counter” registers. The contents of these registers provide information on the transmit faults present at the mounting site of the encoder.

| | | | | |
|---|---|---|---|---|
| Sub-index | 00h | | | |
| Description | Transmit Error Counter and the 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 | | |

#### 5.8.2.4 1003h: Pre-defined Error Field

In object 1003h, the 8 latest error states are archived (see chapter 5.6.1: Emergency Service (EMCY)).

- the entry under sub-index 0 indicates the number of errors saved.
- The latest error status is always stored in sub-index 01h. Previous error messages “slip onwards” in their position by one sub-index.
- The whole error list is deleted by writing the value 0 in sub-index 00h.
- The entries in the error list have the format described in chapter 5.6.1: Emergency Service (EMCY) .

| | |
|---|---|
| Sub-index | 00h |
| Description | number of the error messages stored |
| Access | rw |
| PDO mapping | no |
| Data type | UNSIGNED 8 |
| Default | 0h |
| EEPROM | yes |

| | |
|---|---|
| Sub-index | 01h-08h |
| Description | error messages that occurred |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 32 |
| Default | 0h |
| EEPROM | yes |

#### 5.8.2.5 1005h: COB-ID SYNC-message

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

| | | |
|---|---|---|
| Sub-index | 00h | |
| Description | Defines the COB ID of the synchronization object (SYNC) | |
| Access | rw (writable in the "Pre-Operational" state only see chapter 5.1) | |
| PDO mapping | no | |
| Data type | UNSIGNED 32 | |
| Default | 80h | |
| EEPROM | yes | |
| Data content | Bit 31 | not defined |
| | Bit 30 | 0: The device generates no SYNC message |
| | Bit 29 | 0: 11bits identifier (CAN 2.0A)<br>1: 29bits 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 |

#### 5.8.2.6 1008h: Manufacturer Device Name

Object 1008h indicates the device name. Since the latter comprises 6 or 7 data bytes, normal transfer is required for reading the SDO (see chapter 5.4.1.2: Normal Request/Response).

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Sub-index | 00h | | | | | | |
| Description | Device name in ASCII notation | | | | | | |
| Access | Const | | | | | | |
| PDO mapping | no | | | | | | |
| Data type | Visible_String | | | | | | |
| Default | WV58MR / WH58MR | | | | | | |
| EEPROM | no | | | | | | |
| | Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 |
| | 57h ("W") | 56h ("V") | 35h ("5") | 38h ("8") | 4Dh ("M") | 52h ("R") | 00h - |

#### 5.8.2.7 1009h: Manufacturer Hardware Version

Object 1009h indicates the hardware version.

| | |
|---|---|
| Sub-index | 00h |
| Description | Hardware version in ASCII notation |
| Access | Const |
| PDO mapping | no |
| Data type | Visible_String |
| Default | V001 |
| EEPROM | no |

| | | | | |
|---|---|---|---|---|
| Data content | Byte 0 | Byte 1 | Byte 2 | Byte 3 |
| | 56h ("V") | 30h ("1") | 30h ("0") | 31h ("0") |

#### 5.8.2.8 100Ah: Manufacturer Software Version

Object 100Ah indicates the software version of the device.

| | | | | |
|---|---|---|---|---|
| Sub-index | 00h | | | |
| Description | Software version in ASCII notation | | | |
| Access | Const | | | |
| PDO mapping | no | | | |
| Data type | Visible_String | | | |
| Default | V001 | | | |
| EEPROM | no | | | |
| Data content | Byte 0 | Byte 1 | Byte 2 | Byte 3 |
| | 56h ("V") | 31h ("1") | 30h ("0") | 30h ("0") |

#### 5.8.2.9 100Ch: Guard Time

Object 100Ch indicates the cycle time set in the master for node guarding (see chapter 5.6.2: Node Guarding). The cycle time is indicated in milliseconds Value "'0h'" means that Node Guarding is deactivated.

| | |
|---|---|
| Sub-index | 00h |
| Description | Guard Time |
| Access | rw |
| PDO mapping | no |
| Data type | UNSIGNED 16 |
| Default | 0h |
| EEPROM | yes |

#### 5.8.2.10 100Dh: Life Time Factor

Object 100Dh indicates the life time factor set in the master for node guarding (see chapter 5.6.2: Node Guarding). Value "'0h'" means that Node Guarding is deactivated.

| | |
|---|---|
| Sub-index | 00h |
| Description | Life Time Factor |
| Access | rw |
| PDO mapping | no |
| Data type | UNSIGNED 8 |
| Default | 0h |
| EEPROM | yes |

### 5.8.2.11 1010h: Store Parameter

Parameters are transferred into the EEPROM with this object in order to ensure that they are protected from loss of voltage. Different parameter groups are stored depending on the selection of the sub-index to be accessed. The string “Save” must be sent as data content.

| | |
|---|---|
| Sub-index | 00h |
| Description | indicates the largest supported sub-index |
| Access | const |
| PDO mapping | no |
| Data type | UNSIGNED 8 |
| Default | 4h |
| EEPROM | no |

| | | | | |
|---|---|---|---|---|
| Sub-index | 01h | | | |
| Description | Save all parameters | | | |
| Access | rw | | | |
| PDO mapping | no | | | |
| Data type | UNSIGNED 32 | | | |
| Default | 1h | | | |
| EEPROM | no | | | |
| Data content | Write: | | | |
| | 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 independently store parameters | | |
| | Bit 0 | 1: Device stores parameters after command | | |

| | | | | |
|---|---|---|---|---|
| Sub-index | 02h | | | |
| Description | Save only communication parameters (1000h-1FFFh, CiA DS-301) | | | |
| Access | rw | | | |
| PDO mapping | no | | | |
| Data type | UNSIGNED 32 | | | |
| Default | 1h | | | |
| EEPROM | no | | | |
| Data content | Write: | | | |
| | 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 independently store parameters | | |
| | Bit 0 | 1: Device stores parameters after command | | |

| | | | | |
|---|---|---|---|---|
| Sub-index | 03h | | | |
| Description | Save only application parameters (1000h-1FFFh, CiA DS-406) | | | |
| Access | rw | | | |
| PDO mapping | no | | | |
| Data type | UNSIGNED 32 | | | |
| Default | 1h | | | |
| EEPROM | no | | | |
| Data content | Write: | | | |
| | 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 independently store parameters | | |
| | Bit 0 | 1: Device stores parameters after command | | |

| | | | | |
|---|---|---|---|---|
| Sub-index | 04h | | | |
| Description | Save only manufacturer-specific parameters (2000h-5FFFh) | | | |
| Access | rw | | | |
| PDO mapping | no | | | |
| Data type | UNSIGNED 32 | | | |
| Default | 1h | | | |
| EEPROM | no | | | |
| Data content | Write: | | | |
| | 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 independently store parameters | | |
| | Bit 0 | 1: Device stores parameters after command | | |

### 5.8.2.12 1011h: Restore Parameter

Object 1011h restores the factory settings of the device depending on the selection . The string “Load” must be sent as data content and the device reset thereafter. If the restored parameters are intended to be permanently available, they must be stored via object 1010h: Store Parameter.

| | |
|---|---|
| Sub-index | 00h |
| Description | indicates the largest supported sub-index |
| Access | const |
| PDO mapping | no |
| Data type | UNSIGNED 8 |

| | |
|---|---|
| Default | 4h |
| EEPROM | no |

| | | | | |
|---|---|---|---|---|
| Sub-index | 01h | | | |
| Description | Reset all parameters to factory settings | | | |
| Access | rw | | | |
| PDO mapping | no | | | |
| Data type | UNSIGNED 32 | | | |
| Default | 1h | | | |
| EEPROM | no | | | |
| Data content | Write: | | | |
| | 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 permits loading of default parameters. | | |

| | | | | |
|---|---|---|---|---|
| Sub-index | 02h | | | |
| Description | Set only communication parameters to factory settings (1000h-1FFFh, CiA DS-301) | | | |
| Access | rw | | | |
| PDO mapping | no | | | |
| Data type | UNSIGNED 32 | | | |
| Default | 1h | | | |
| EEPROM | no | | | |
| Data content | Write: | | | |
| | 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 permits loading of default parameters. | | |

| | |
|---|---|
| Sub-index | 03h |
| Description | Set only application parameters to factory settings (6000h-9FFFh, CiA DS-406) |
| Access | rw |
| PDO mapping | no |
| Data type | UNSIGNED 32 |
| Default | 1h |
| EEPROM | no |
| Data content | Write: |

| | | | | |
|---|---|---|---|---|
| | 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 permits loading of default parameters. | | |

| | | | | |
|---|---|---|---|---|
| Sub-index | 04h | | | |
| Description | Set only manufacturer-specific parameters to factory settings (2000h-5FFFh) | | | |
| Access | rw | | | |
| PDO mapping | no | | | |
| Data type | UNSIGNED 32 | | | |
| Default | 1h | | | |
| EEPROM | no | | | |
| Data content | Write: | | | |
| | 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 permits loading of default parameters. | | |

### 5.8.2.13 1014h: COB-ID Emergency message

The COB ID of the Emergency object is set via object 1014h (see chapter 5.6.1: Emergency Service (EMCY)).

| | | |
|---|---|---|
| Sub-index | 00h | |
| Description | Defines the COB ID of the Emergency object (EMCY) | |
| Access | rw (writable in the "Pre-Operational" state only see chapter 5.1: Telegram structure) | |
| 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 exists / is invalid |
| | Bit 30 | Always 0b |
| | Bit 29 | 0: 11bits identifier (CAN 2.0A)<br>1: 29bits identifier (CAN 2.0B) |
| | Bit 28 ... 11 | 0: if Bit 29 = 0b<br>X: Bits 28 – 11 of the EMCY-COB-ID, if Bit 29 = 1b |
| | Bit 10 ... 0 | X: bits 10 – 0 of the EMCY-COB-ID |

#### 5.8.2.14 1017h: Producer Heartbeat Time

The cycle time “Heartbeat Time” for the heartbeat protocol is set via object 1017h. The cycle time is indicated in milliseconds

| | |
|---|---|
| Sub-index | 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. Value 0h disables the service. |

#### 5.8.2.15 1018h: Identity Object

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

| | |
|---|---|
| Sub-index | 00h |
| Description | indicates the largest supported sub-index |
| Access | const |
| PDO mapping | no |
| Data type | UNSIGNED 8 |
| Default | 4h |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 01h |
| Description | The manufacturer identification number (vendor ID) for the company SIKO GmbH allocated by the CiA |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 32 |
| Default | 195h |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 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 |

| | |
|---|---|
| Sub-index | 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 |

| | |
|---|---|
| Sub-index | 04h |
| Description | Serial Number |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 32 |
| Default | 1h |
| EEPROM | yes |

#### 5.8.2.16 1200h: Server SDO Parameter

The COB IDs for the server SDOs are indicated via object 1200h. The COB IDs cannot be changed.

| | |
|---|---|
| Sub-index | 00h |
| Description | indicates the largest supported sub-index |
| Access | const |
| PDO mapping | no |
| Data type | UNSIGNED 8 |
| Default | 2h |
| EEPROM | no |

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

| | |
|---|---|
| Sub-index | 02h |
| Description | COB-ID Server -> Client (tx) |
| Access | ro |

| | |
|---|---|
| PDO mapping | no |
| Data type | UNSIGNED 32 |
| Default | 00000580h + Node-ID |
| EEPROM | no |

### 5.8.2.17 1301h: SRDO1 communication parameters

The communication behavior of SRDO1 can be determined via object 1301h.

| | |
|---|---|
| Sub-index | 00h |
| Description | indicates the largest supported sub-index |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 8 |
| Default | 6h |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 01h |
| Description | Direction of information |
| Access | ro, if NMT state is operational<br>rw, if NMT state is pre-operational |
| PDO mapping | no |
| Data type | UNSIGNED 8 |
| Default | 1 |
| EEPROM | yes |
| Data content | 0: SRDO1 is not valid<br>1: SRDO1 is TX and valid |

| | |
|---|---|
| Sub-index | 02h |
| Description | refresh time |
| Access | ro, if NMT state is operational<br>rw, if NMT state is pre-operational |
| PDO mapping | no |
| Data type | UNSIGNED 16 |
| Default | 25 |
| EEPROM | yes |
| Data content | 10...65535 |

| | |
|---|---|
| Sub-index | 03h |
| Description | tx: reserved<br>rx: SRVT |

| | |
|---|---|
| Access | ro, if NMT state is operational<br>rw, if NMT state is pre-operational |
| PDO mapping | no |
| Data typ | UNSIGNED 8 |
| Default | 20 |
| EEPROM | no |
| Data content | 20 |

| | |
|---|---|
| Sub-index | 04h |
| Description | transmission type |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 8 |
| Default | 254 |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 05h |
| Description | COB-ID1 |
| Access | ro, if NMT state is operational<br>rw, if NMT state is pre-operational |
| PDO mapping | no |
| Data type | UNSIGNED 32 |
| Default | 000000FFh + 2*Node ID |
| EEPROM | yes |
| Data content | 257...383. Only odd numbers are allowed. |

| | |
|---|---|
| Sub-index | 06h |
| Description | COB-ID2 |
| Access | ro, if NMT state is operational<br>rw, if NMT state is pre-operational |
| PDO mapping | no |
| Data type | UNSIGNED 32 |
| Default | 00000100h + 2*Node ID |
| EEPROM | yes |
| Data content | 258...384. Only even numbers are allowed. |

#### 5.8.2.18 1302h: SRDO2 communication parameters

The communication behavior of SRDO2 can be determined via 1302h.

| | |
|---|---|
| Sub-index | 00h |

| | |
|---|---|
| Description | indicates the largest supported sub-index |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 8 |
| Default | 6h |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 01h |
| Description | Direction of information |
| Access | ro, if NMT state is operational<br>rw, if NMT state is pre-operational |
| PDO mapping | no |
| Data type | UNSIGNED 8 |
| Default | 1 |
| EEPROM | yes |
| Data content | 0: SRDO2 is not valid<br>1: SRDO2 is TX and valid |

| | |
|---|---|
| Sub-index | 02h |
| Description | refresh time |
| Access | ro, if NMT state is operational<br>rw, if NMT state is pre-operational |
| PDO mapping | no |
| Data type | UNSIGNED 16 |
| Default | 25 |
| EEPROM | yes |
| Data content | 10...65535 |

| | |
|---|---|
| Sub-index | 03h |
| Description | tx: reserved<br>rx: SRVT |
| Access | ro, if NMT state is operational<br>rw, if NMT state is pre-operational |
| PDO mapping | no |
| Data type | UNSIGNED 8 |
| Default | 20 |
| EEPROM | nein |
| Dateninhalt | 20 |

| | |
|---|---|
| Sub-index | 04h |
| Description | transmission type |

| | |
|---|---|
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 8 |
| Default | 254 |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 05h |
| Description | COB-ID1 |
| Access | ro, if NMT state is operational<br>rw, if NMT state is pre-operational |
| PDO mapping | no |
| Data type | UNSIGNED 32 |
| Default | 0000010Fh + 2*Node ID |
| EEPROM | yes |
| Data content | 257...383. Only odd numbers are allowed. |

| | |
|---|---|
| Sub-index | 06h |
| Description | COB-ID2 |
| Access | ro, if NMT state is operational<br>rw, if NMT state is pre-operational |
| PDO mapping | no |
| Data type | UNSIGNED 32 |
| Default | 00000110h + 2*Node ID |
| EEPROM | yes |
| Data content | 258...384. Only even numbers are allowed. |

#### 5.8.2.19 1381h: SRDO1 mapping parameters

Object 1381h determines the objects that are mapped in the first Transmit SRDO (SRDO1).

| | |
|---|---|
| Sub-index | 00h |
| Description | indicates the largest supported sub-index |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 8 |
| Default | 8h |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 01h |
| Description | 1st object of the SRDO1 message with the COB ID1 (data byte 0) |
| Access | ro |
| PDO mapping | no |

| | |
|---|---|
| Data type | UNSIGNED 32 |
| Default | 61200108h (position value object 6120h, sub-index 01h, 8bit) |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 02h |
| Description | 1st object of the SRDO1 message with the COB ID2 (data byte 0) |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 32 |
| Default | 61210108h (inverted position value object 6121h, sub-index 01h, 8bit) |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 03h |
| Description | 2nd object of the SRDO1 message with the COB ID1 (data byte 1) |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 32 |
| Default | 61200208h (position value object 6120h, sub-index 02h, 8bit) |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 04h |
| Description | 2nd object of the SRDO1 message with the COB ID2 (data byte 1) |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 32 |
| Default | 61210208h (inverted position value object 6121h, sub-index 02h, 8bit) |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 05h |
| Description | 3rd object of the SRDO1 message with the COB ID1 (data byte 2) |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 32 |
| Default | 61200308h (position value object 6120h, sub-index 03h, 8bit) |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 06h |
| Description | 3rd object of the SRDO1 message with the COB-ID2 (data byte 2) |
| Access | ro |
| PDO mapping | no |

| | |
|---|---|
| Data type | UNSIGNED 32 |
| Default | 61200308h (inverted position value object 6121h, sub-index 03h, 8bit) |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 07h |
| Description | 4th object of the SRDO1 message with the COB ID1 (data byte 3) |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 32 |
| Default | 61200408h (position value object 6120h, sub-index 04h, 8bit) |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 08h |
| Description | 4th object of the SRDO1 message with the COB ID2 (data byte 3) |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 32 |
| Default | 61210408h (inverted position value object 6121h, sub-index 04h, 8bit) |
| EEPROM | no |

#### 5.8.2.20 1382h: SRDO2 mapping parameters

Object 1382h determines the objects that are mapped in the second Safety Transmit SRDO (SRDO2).

| | |
|---|---|
| Sub-index | 00h |
| Description | indicates the largest supported sub-index |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 8 |
| Default | 4h |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 01h |
| Description | 1st object of the SRDO2 message with the COB ID1 (data byte 0) |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 32 |
| Default | 61240108h (speed value object 6124h, sub-index 01h, 8bit) |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 02h |
| Description | 1st object of the SRDO2 message with the COB ID2 (data byte 0) |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 32 |
| Default | 61250108h (inv. speed value object 6125h, sub-index 01h, 8bit) |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 03h |
| Description | 2nd object of the SRDO2 message with the COB ID1 (data byte 1) |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 32 |
| Default | 61240208h (speed value object 6124h, sub-index 02h, 8bit) |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 04h |
| Description | 2nd object of the SRDO2 message with the COB ID2 (data byte 1) |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 32 |
| Default | 61250208h (inv. speed value object 6125h, sub-index 02h, 8bit) |
| EEPROM | no |

#### 5.8.2.21 13FEh: Safety configuration

The SRDO configuration can be switched to valid by means of object 13FEh.

| | |
|---|---|
| Sub-index | 00h |
| Description | This parameter will be switched to invalid automatically after changing a parameter in object 1301h or 1302h and must be switched valid again via this object. Switching to valid is only enabled if the correct signatures are entered in object 13FFh. |
| Access | rw |
| PDO mapping | no |
| Data type | UNSIGNED 8 |
| Default | 0h |
| EEPROM | yes |
| Data content | A5h: SRDO1 and SRDO2 configuration valid<br>00h...A4h and A6h...FFh: SRDO1 and SRDO2 configuration invalid |

### 5.8.2.22 13FFh: Safety configuration signature (checksum)

This object 13FFh receives the signatures (checksums) via the CANopen Safety parameters of SRDO1 and SRDO2. Only a checksum which is valid at that time can be transmitted. The checksum is checked anew before switching the configuration to valid. Any change to the configuration will be valid only after passing.

| | |
|---|---|
| Sub-index | 00h |
| Description | indicates the largest supported sub-index |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 8 |
| Default | 2h |
| EEPROM | no |

| | | | | | |
|---|---|---|---|---|---|
| Sub-index | 01h | | | | |
| Description | SRDO1 signature (checksum) | | | | |
| Access | rw | | | | |
| PDO mapping | no | | | | |
| Data type | UNSIGNED 16 | | | | |
| Default | 0000h | | | | |
| EEPROM | yes | | | | |
| Data content | The checksum CRC-16-CCITT is calculated via the content of the two objects 1301h and 1381h. | | | | |
| | Object | Sub-index | Name | Extent | Value |
| | 1301h | SRDO1 communication parameter | | | |
| | | 01h | Direction of information | 1 byte | Object 1301h sub-index 01h |
| | | 02h | Refresh time | 2 bytes | Object 1301h sub-index 02h |
| | | 03h | tx: reserved<br>rx: SRVT | 1 Byte | Object 1301h sub-index 03h |
| | | 05h | COB ID 1 | 4 bytes | Object 1301h sub-index 05h |
| | | 06h | COB ID 2 | 4 bytes | Object 1301h sub-index 06h |
| | 1381h | SRDO1 mapping parameters | | | |
| | | 00h | Highest sub-index | 1 byte | 08h |
| | | 01h | Sub-index | 1 byte | 01h |
| | | 01h | Position value 1st byte | 4 bytes | 61200108h |
| | | 02h | Sub-index | 1 byte | 02h |
| | | 02h | Inverted position value 1st byte | 4 bytes | 61210108h |
| | | 03h | Sub-index | 1 byte | 03h |
| | | 03h | Position value 2nd byte | 4 bytes | 61200208h |

| | | | | | |
|---|---|---|---|---|---|
| | | 04h | Sub-index | 1 byte | 04h |
| | | 04h | Inverted position value 2nd byte | 4 bytes | 61210208h |
| | | 05h | Sub-index | 1 byte | 05h |
| | | 05h | Position value 3rd byte | 4 bytes | 61200308h |
| | | 06h | Sub-index | 1 byte | 06h |
| | | 06h | Inverted position value 3rd byte | 4 bytes | 61210308h |
| | | 07h | Sub-index | 1 byte | 07h |
| | | 07h | Position value 4th byte | 4 bytes | 61200408h |
| | | 08h | Sub-index | 1 byte | 08h |
| | | 08h | Inverted position value 4th byte | 4 bytes | 61210408h |

| | | | | | |
|---|---|---|---|---|---|
| Sub-index | 02h | | | | |
| Description | SRDO2 signature (checksum) | | | | |
| Access | rw | | | | |
| PDO mapping | no | | | | |
| Data type | UNSIGNED 16 | | | | |
| Default | 0000h | | | | |
| EEPROM | yes | | | | |
| Data content | The checksum CRC-16-CCITT is calculated via the content of the two objects 1302h and 1382h. | | | | |
| | Object | Sub-index | Name | Extent | Value |
| | 1302h | SRDO2 communication parameter | | | |
| | | 01h | Direction of information | 1 byte | Object 1302h sub-index 01h |
| | | 02h | Refresh time | 2 bytes | Object 1302h sub-index 02h |
| | | 03h | tx: reserved<br>rx: SRVT | 1 Byte | Object 1302h sub-index 03h |
| | | 05h | COB ID 1 | 4 bytes | Object 1302h sub-index 05h |
| | | 06h | COB ID 2 | 4 bytes | Object 1302h sub-index 06h |
| | 1382h | SRDO2 mapping parameters | | | |
| | | 00h | Highest sub-index | 1 byte | 04h |
| | | 01h | Sub-index | 1 byte | 01h |
| | | 01h | speed value 1st byte | 4 bytes | 61240108h |
| | | 02h | Sub-index | 1 byte | 02h |
| | | 02h | Inverted speed value 1st byte | 4 bytes | 61250108h |
| | | 03h | Sub-index | 1 byte | 03h |

| | | | | | |
|---|---|---|---|---|---|
| | | 03h | speed value 2nd byte | 4 bytes | 61240208h |
| | | 04h | Sub-index | 1 byte | 04h |
| | | 04h | Inverted speed value 2nd byte | 4 bytes | 61250208h |

### 5.8.2.23 1800h: 1. Transmit PDO Parameter

TPDO1 is used for asynchronous PDO transfer according to CiA DS-406.
The communication parameters for TPDO1 are set via object 1800h.

| | |
|---|---|
| Sub-index | 00h |
| Description | indicates the largest supported sub-index |
| Access | const |
| PDO mapping | no |
| Data type | UNSIGNED 8 |
| Default | 5h |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 01h |
| Description | COB ID of the PDO1 |
| Access | rw (writable in the “Pre-Operational” state only see chapter 5.1) |
| PDO mapping | no |
| Data type | UNSIGNED 32 |
| Default | 180h + Node-ID |
| EEPROM | yes |

| | | |
|---|---|---|
| Sub-index | 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) | Device responds only to RTR request if RTR Bit 30 is enabled in the COB-ID. |

| | |
|---|---|
| Sub-index | 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 |

| | |
|---|---|
| Sub-index | 04h (is not used, access attempt generates error message) |

| | |
|---|---|
| Sub-index | 05h |
| Description | Event timer for TPDO1 hard-wired (CiA DS-406) with cyclic timer 6200h |
| Access | rw |
| PDO mapping | no |
| Data type | UNSIGNED 16 |
| Default | 0h |
| EEPROM | yes |
| Data content | The service is disabled by writing the value 0h. the content of this object is identical with object 6200h. If the value is changed with the timer running, the change will be applied only with the next timer operation. |

| | |
|---|---|
| Sub-index | 06h (is not used, access attempt generates error message) |

#### 5.8.2.24 1801h: 2. Transmit PDO Parameter

TPDO2 is used for synchronous PDO transfer according to CiA DS-406.
The communication parameters for TPDO2 are set via object 1801h.

| | |
|---|---|
| Sub-index | 00h |
| Description | indicates the largest supported sub-index |
| Access | const |
| PDO mapping | no |
| Data type | UNSIGNED 8 |
| Default | 5h |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 01h |
| Description | COB ID of the PDO2 |
| Access | rw (writable in the “Pre-Operational” state only see chapter 5.1) |
| PDO mapping | no |
| Data type | UNSIGNED 32 |
| Default | 280h + Node-ID |
| EEPROM | yes |

| | | |
|---|---|---|
| Sub-index | 02h | |
| Description | Transmission Type | |
| Access | rw | |
| PDO mapping | no | |
| Data type | UNSIGNED 8 | |
| Default | 1h | |
| EEPROM | yes | |
| Data content | FEh (254d)<br>FFh (255d) | PDO is sent after 1d ... 240d received SYNC messages. |
| | FDh (253d) | Device responds only to RTR request if RTR Bit 30 is enabled in the COB-ID. |

| | |
|---|---|
| Sub-index | 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 |

| | |
|---|---|
| Sub-index | 04h (is not used, access attempt generates error message) |

| | |
|---|---|
| Sub-index | 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 |

| | |
|---|---|
| Sub-index | 06h (is not used, access attempt generates error message) |

### 5.8.2.25 1A00h: 1. Transmit PDO Mapping Parameter

Object 1A00h determines the objects that are mapped on the first Transmit PDO (TPDO1).

| | |
|---|---|
| Sub-index | 00h |
| Description | number of objects mapped |
| Access | const |
| PDO mapping | no |

| | |
|---|---|
| Data type | UNSIGNED 8 |
| Default | 2h |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 01h |
| Description | 1. Object of the PDO1 message (data byte 0 to 3) |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 32 |
| Default | 60040020h (position value object 6004h, sub-index 00h, 32bit) |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 02h |
| Description | 2. Object of the PDO1 message (data byte 4 + 5) |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 16 |
| Default | 60300110h (speed object 6030h, sub-index 01h, 16bit) |
| EEPROM | no |

#### 5.8.2.26 1A01h: 2. Transmit PDO Mapping Parameter

Object 1A01h determines the objects that are mapped in the second Transmit PDO (TPDO2).

| | |
|---|---|
| Sub-index | 00h |
| Description | number of objects mapped |
| Access | const |
| PDO mapping | no |
| Data type | UNSIGNED 8 |
| Default | 2h |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 01h |
| Description | 1. Object of the PDO2 message (data byte 0 to 3) |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 32 |
| Default | 60040020h (position value object 6004h, sub-index 00h, 32bit) |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 02h |
| Description | 2. Object of the PDO2 message (data byte 4 + 5) |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 32 |
| Default | 60300110h (speed object 6030h, sub-index 01h, 16bit) |
| EEPROM | no |

#### 5.8.2.27 2001h: Application offset

The offset value is determined via object 2001h.

| | | |
|---|---|---|
| Sub-index | 00h | |
| Description | The offset enables the shifting of a scaled value range. The offset value is added to the position value in the encoder. Positive as well as negative values are permitted.<br>Position value = measured value + calibration value + application offset | |
| Access | rw | |
| PDO mapping | no | |
| Data type | SIGNED 32 | |
| Default | 0h | |
| EEPROM | yes | |
| Data content | Single-turn | -16384...16383 |
| | 4 Bit Multi-turn | -262144...262143 |
| | 8 Bit Multi-turn | -4194304...4194303 |
| | 12 Bit Multi-turn | -67108864...67108863 |

#### 5.8.2.28 2002h: Calibrate encoder value

Calibration can be executed via Object 2002h.

| | | |
|---|---|---|
| Sub-index | 00h | |
| Description | This object enables “zeroing” of the measured value. This serves for setting the position value to the calibration value.<br>Position value = measured value + calibration value + application offset | |
| Access | wo | |
| PDO mapping | no | |
| Data type | UNSIGNED 8 | |
| Default | 0h | |
| EEPROM | yes | |
| Data content | 1 | Writing of the value 1h sets the position value to the calibration value. |

#### 5.8.2.29 2003h: Limit speed low

The lower value for the limit speed can be set via Object 2003h.

| | |
|---|---|
| Sub-index | 00h |
| Description | Lower limit speed |
| Access | rw |
| PDO mapping | no |
| Data type | SIGNED 16 |
| Default | 0h |
| EEPROM | yes |
| Data content | -32768...32767 |

#### 5.8.2.30 2004h: Limit speed High

The upper value for the limit speed can be set via Object 2003h.

| | |
|---|---|
| Sub-index | 00h |
| Description | Upper limit speed |
| Access | rw |
| PDO mapping | no |
| Data type | SIGNED 16 |
| Default | 0h |
| EEPROM | yes |
| Data content | -32768...32767 |

#### 5.8.2.31 5000h: Diagnosis CAN Bus error

A prioritized list of CAN bus errors occurring can be read via Object 5000h.

| | | | | |
|---|---|---|---|---|
| Sub-index | 00h | | | |
| Description | Indicates the CAN Bus errors Acknowledge, Form, CRC and Stuff Error sorted by 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: No error occurring at all

4: Error occurring most frequently

### 5.8.2.32 5F0Ah: Node-ID and baud rate Bus CAN

Node ID and baud rate of the bus can be set via Object 5F0Ah.

| Sub-index | 00h |
|---|---|
| Description | indicates the largest supported sub-index |
| Access | const |
| PDO mapping | no |
| Data type | UNSIGNED 8 |
| Default | 2h |
| EEPROM | no |

| Sub-index | 01h |
|---|---|
| Description | Node ID |
| Access | rw |
| PDO mapping | no |
| Data type | UNSIGNED 8 |
| Default | 1h (redundant encoder 2h) |
| EEPROM | yes |
| Data content | 01h ... 7Fh |

| Sub-index | 02h |
|---|---|
| Description | Baud rate of the CAN bus |
| Access | rw |
| PDO mapping | no |
| Data type | UNSIGNED 8 |
| Default | 5h (500kBaud) |
| 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 |

### 5.8.2.33 6000h: Operating Parameters

Settings of the operating parameters can be made through object 6000h.

| Sub-index | 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 | not used |
| | Bit 2 | 1: Scaling enabled |
| | Bit 1 | not used |
| | Bit 0 | 0: Clockwise (CW) sense of rotation I<br>1: Counter-clockwise (CCW) sense of rotation E |

**Scaling:** The encoder works with the preset resolution indicator (measuring step per resolution) which can be configured by object 6001h. The scaling function cannot be disabled.

**Sense of rotation I:** ascending position values with clockwise (CW) encoder rotation (view on the encoder shaft)

**Sense of rotation E:** ascending position values with counter-clockwise (CCW) encoder rotation (view on the encoder shaft)

#### 5.8.2.34 6001h: Measurement steps per revolution (Display per revolution = APU)

The number of measurement steps is determined via Object 6001h.

| | |
|---|---|
| Sub-index | 00h |
| Description | Number of measurement steps per revolution |
| Access | rw |
| PDO mapping | no |
| Data type | UNSIGNED 32 |
| Default | 16384d (4000h) |
| EEPROM | yes |
| Data content | 1d...65535d (1h...FFFFh) |

#### 5.8.2.35 6002h: Overall number of measurement steps

Object 6002h indicates the overall number of the system's measuring steps.

| | | |
|---|---|---|
| Sub-index | 00h | |
| Description | Overall number of measuring units | |
| Access | rw | |
| PDO mapping | no | |
| Data type | UNSIGNED 32 | |
| Default | Single-turn | 16384 |
| | 4 Bit Multi-turn | 262144 |
| | 8 Bit Multi-turn | 4194304 |
| | 12 Bit Multi-turn | 67108864 |

| | | |
|---|---|---|
| EEPROM | Yes | |
| Data content | Encoder type | With changed APU (Object 6001h) |
| | Single-turn | (APU*1) – 1 |
| | 4 Bit Multi-turn | (APU*16) – 1 |
| | 8 Bit Multi-turn | (APU*256) – 1 |
| | 12 Bit Multi-turn | (APU*4096) – 1 |

### 5.8.2.36 6003h: Preset value (calibration value)

Via object 6003h, the position value of the encoder can be set to a

calibration value with calibration. Position value = measured value + calibration value + application offset (see chapter 4.2 Calibration)

| | | | |
|---|---|---|---|
| Sub-index | 00h | | |
| Description | Calibration value | | |
| Access | rw | | |
| PDO mapping | no | | |
| Data type | SIGNED 32 | | |
| Default | 0h | | |
| EEPROM | yes | | |
| Data content | Encoder type | Default | With changed APU (Object 6001h) |
| | Single-turn | 0...16383 | 0...((APU*1) – 1) |
| | 4 Bit Multi-turn | 0...262143 | 0...((APU*16) – 1) |
| | 8 Bit Multi-turn | 0...4194303 | 0...((APU*256) – 1) |
| | 12 Bit Multi-turn | 0...67108863 | 0...((APU*4096) – 1) |

### 5.8.2.37 6004h: Position value

Object 6004h indicates the actual position value of the device.

| | |
|---|---|
| Sub-index | 00h |
| Description | Position value |
| Access | ro |
| PDO mapping | yes |
| Data type | UNSIGNED 32 |
| Default | 0h |
| EEPROM | no |

Position value = measured value + calibration value + application offset

#### 5.8.2.38 600Ch: Absolute accuracy

The object 600Ch provides the absolute accuracy of the encoder in bits.

| | |
|---|---|
| Subindex | 00h |
| Description | Raw Value (Absolute accuracy) |
| Access | ro |
| PDO-Mapping | yes |
| Data type | UNSIGNED 32 |
| Default | 0h |
| EEPROM | no |

#### 5.8.2.39 6030h: Velocity value

Velocity can be read via object 6030h.

| | |
|---|---|
| Sub-index | 00h |
| Description | indicates the largest supported sub-index |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 8 |
| Default | 1h |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 01h |
| Description | Speed value in increments per ms (Inc/ms) |
| Access | ro |
| PDO mapping | no |
| Data type | SIGNED 16 |
| Default | 0h |
| EEPROM | no |

Conversion of increments per ms in rpm

Speed [rpm] = speed value [Inc/ms] * 6000 [rpm] / 1638.4 [Inc/ms]

#### 5.8.2.40 6031h: Speed parameters

Speed settings can be made via object 6031h.

| | |
|---|---|
| Subindex | 00h |
| Description | Indicates the largest supported sub-index |
| Access | ro |
| PDO-Mapping | no |
| Data type | UNSIGNED 8 |
| Default | 4h |

| | |
|---|---|
| EEPROM | no |

| | |
|---|---|
| Subindex | 01h |
| Description | Speed source selector |
| Access | rw |
| PDO-Mapping | no |
| Data type | UNSIGNED 8 |
| Default | 02h |
| EEPROM | yes |
| Data content | 02h: Object 600C raw value of position is used |

| | |
|---|---|
| Subindex | 02h |
| Description | Integration time of speed |
| Access | ro |
| PDO-Mapping | no |
| Data type | UNSIGNED 16 |
| Default | 64h |
| EEPROM | yes |
| Data content | 64h: 100 ms<br>C8h: 200 ms |

| | |
|---|---|
| Subindex | 03h |
| Description | Multiplikator |
| Access | rw |
| PDO-Mapping | no |
| Data type | UNSIGNED 16 |
| Default | 01h |
| EEPROM | yes |
| Data content | 01h |

| | |
|---|---|
| Subindex | 04h |
| Description | Divisor |
| Access | rw |
| PDO-Mapping | no |
| Data type | UNSIGNED 16 |
| Default | 01h |
| EEPROM | yes |
| Data content | 01h |

Velocity value [Inc/ms] = (new raw position value object 600Ch- old raw position value object 600Ch) / (Integration time [ms] object 6031h sub-index 02h * 10-3) * multiplier object 6031h sub-index 03h / divisor object 6031h sub-index 04h

### 5.8.2.41 6100h: Safety configuration parameters of position

Via object 6100h, settings for the position and its transmission can be made.

| | |
|---|---|
| Sub-index | 00h |
| Description | indicates the largest supported sub-index |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 8 |
| Default | 2h |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 01h |
| Description | Safety sense of direction |
| Access | rw (write in the Pre-operational NMT state only) |
| PDO mapping | no |
| Data type | UNSIGNED 16 |
| Default | 0000h |
| EEPROM | yes |
| Data content | 0: CW<br>1: CCW |

| | | | |
|---|---|---|---|
| Sub-index | 02h | | |
| Description | Safety preset value (calibration value) | | |
| Access | rw (write in the Pre-operational NMT state only) | | |
| PDO mapping | no | | |
| Data type | SIGNED 32 | | |
| Default | 0h | | |
| EEPROM | yes | | |
| Data content | Encoder type | Default | With changed APU (object 6001h) |
| | Single-turn | 0…16383 | 0…((APU*1)-1) |
| | 4 bit multi-turn | 0…262143 | 0…((APU*16)-1) |
| | 8 bit Multi-turn | 0…4194303 | 0…((APU*256)-1) |
| | 12 bit multi-turn | 0…67108863 | 0…((APU*4096)-1) |

### 5.8.2.42 6101h: Safety configuration parameters of speed

Via object 6101h, settings for speed and its transmission can be made.

| | |
|---|---|
| Sub-index | 00h |
| Description | indicates the largest supported sub-index |
| Access | ro |
| PDO mapping | no |

| | |
|---|---|
| Data type | UNSIGNED 8 |
| Default | 7h |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 01h |
| Description | Safety sense of direction |
| Access | rw (write in the Pre-operational NMT state only) |
| PDO mapping | no |
| Data type | UNSIGNED 16 |
| Default | 0000h |
| EEPROM | yes |
| Data content | 0: CW<br>1: CCW |

| | | | |
|---|---|---|---|
| Sub-index | 02h | | |
| Description | Safety preset value (calibration value) | | |
| Access | rw (write in the Pre-operational NMT state only) | | |
| PDO mapping | no | | |
| Data type | SIGNED 32 | | |
| Default | 0h | | |
| EEPROM | yes | | |
| Data content | Encoder type | Default | With changed APU (Object 6001h) |
| | Single-turn | 0...16383 | 0...((APU*1)-1) |
| | 4 bit multi-turn | 0...262143 | 0...((APU*16)-1) |
| | 8 bit multi-turn | 0...4194303 | 0...((APU*256)-1) |
| | 12 bit multi-turn | 0...67108863 | 0...((APU*4096)-1) |

| | |
|---|---|
| Sub-index | 04h |
| Description | Safety speed source selector |
| Access | rw (write in the Pre-operational NMT state only) |
| PDO mapping | no |
| Data type | UNSIGNED 8 |
| Default | 02h |
| EEPROM | yes |
| Data content | 02h: Object 600C raw value of position is used |

| | |
|---|---|
| Sub-index | 05h |
| Description | Safety integration time of speed |
| Access | rw (write in the Pre-operational NMT state only) |
| PDO mapping | no |

| | |
|---|---|
| Data type | UNSIGNED 16 |
| Default | 64h |
| EEPROM | yes |
| Data content | 64h: 100 ms<br>C8h: 200 ms |

| | |
|---|---|
| Sub-index | 06h |
| Description | Safety multiplier |
| Access | rw (write in the Pre-operational NMT state only) |
| PDO mapping | no |
| Data type | UNSIGNED 16 |
| Default | 01h |
| EEPROM | yes |
| Data content | 01h |

| | |
|---|---|
| Sub-index | 07h |
| Description | Safety divisor |
| Access | rw (write in the Pre-operational NMT state only) |
| PDO mapping | no |
| Data type | UNSIGNED 16 |
| Default | 01h |
| EEPROM | yes |
| Data content | 01h |

Speed value [Inc/ms] = (new raw position value object 600Ch - old raw position value object 600Ch) / (integration time [ms] object 6101h sub-index 05h * 10-3) * multiplier object 6101h sub-index 06h / divisor object 6101h sub-index 07h

#### 5.8.2.43 6120h: Safety position value

Via object 6120h, the position value can be read byte by byte.

| | |
|---|---|
| Sub-index | 00h |
| Description | indicates the largest supported sub-index |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 8 |
| Default | 4h |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 01h |
| Description | Safety position byte 1 |
| Access | ro |

| | |
|---|---|
| PDO mapping | no |
| Data type | UNSIGNED 8 |
| Default | 0h |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 02h |
| Description | Safety position byte 2 |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 8 |
| Default | 0h |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 03h |
| Description | Safety position byte 3 |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 8 |
| Default | 0h |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 04h |
| Description | Safety position byte 4 |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 8 |
| Default | 0h |
| EEPROM | no |

#### 5.8.2.44 6121h: Safety inverted position value

Via object 6121h, the inverted position value can be read byte by byte.

| | |
|---|---|
| Sub-index | 00h |
| Description | indicates the largest supported sub-index |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 8 |
| Default | 4h |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 01h |
| Description | Safety inverted position value byte 1 |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 8 |
| Default | FFh |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 02h |
| Description | Safety inverted position value byte 2 |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 8 |
| Default | FFh |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 03h |
| Description | Safety inverted position value byte 3 |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 8 |
| Default | FFh |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 04h |
| Description | Safety inverted position value byte 4 |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 8 |
| Default | FFh |
| EEPROM | no |

#### 5.8.2.45 6124h: Safety speed value

Via object 6124h, the speed value can be read byte by byte.

| | |
|---|---|
| Sub-index | 00h |
| Description | indicates the largest supported sub-index |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 8 |

| | |
|---|---|
| Default | 2h |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 01h |
| Description | Safety speed value byte 1 |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 8 |
| Default | 0h |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 02h |
| Description | Safety speed value byte 2 |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 8 |
| Default | 0h |
| EEPROM | no |

Conversion of increments per ms in rpm

Speed [rpm] = speed value [Inc/ms] * 6000 [rpm] / 1638.4 [Inc/ms]

#### 5.8.2.46 6125h: Safety inverted speed value

Via object 6125h, the inverted speed value can be read byte by byte.

| | |
|---|---|
| Sub-index | 00h |
| Description | indicates the largest supported sub-index |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 8 |
| Default | 2h |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 01h |
| Description | Safety inverted speed value byte 1 |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 8 |
| Default | FFh |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 02h |
| Description | Safety inverted speed value byte 2 |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 8 |
| Default | FFh |
| EEPROM | no |

#### 5.8.2.47 61FEh: Safety application configuration

Via object 61FEh, the configuration of position and speed can be switched to valid.

| | |
|---|---|
| Sub-index | 00h |
| Description | This parameter will be switched to invalid automatically after changing a parameter in object 6100h or 6101h and must be switched valid again via this object. Switching to valid is only enabled if the correct signatures are entered in object 61FFh. |
| Access | rw |
| PDO mapping | no |
| Data type | UNSIGNED 8 |
| Default | 0h |
| EEPROM | yes |
| | A5h: Configuration for position and speed valid<br>00h...A4h und A6h...FFh: Configuration invalid |

#### 5.8.2.48 61FFh: Safety configuration signature (checksum)

This object 61FFh receives the signatures (checksums) via the CANopen Safety parameters for position and speed. Only a checksum which is valid at that time can be transmitted. The checksum is checked anew before switching the configuration to valid. Any change to the configuration will be valid only after passing.

| | |
|---|---|
| Sub-index | 00h |
| Description | indicates the largest supported sub-index |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 8 |
| Default | 2h |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 01h |
| Description | Signature of position (checksum) |
| Access | rw |
| PDO mapping | no |

| | | | | | |
|---|---|---|---|---|---|
| Data type | UNSIGNED 16 | | | | |
| Default | 0000h | | | | |
| EEPROM | yes | | | | |
| Data content | The checksum CRC-16-CCITT is calculated via the content of the object 6100h. | | | | |
| | Object | Sub-index | Name | Extent | Value |
| | 6100h | Safety configuration of the position parameters | | | |
| | | 00h | Highest sub-index | 1 byte | 02h |
| | | 01h | Sub-index | 1 byte | 01h |
| | | 01h | Safety sense of direction | 2 bytes | Object 6100h sub-index 01h |
| | | 02h | Sub-index | 1 byte | 02h |
| | | 02h | Safety preset value (calibration value) | 4 bytes | Object 6100h sub-index 02h |

| | | | | | |
|---|---|---|---|---|---|
| Sub-index | 02h | | | | |
| Description | Signature of speed (checksum) | | | | |
| Access | rw | | | | |
| PDO mapping | no | | | | |
| Data type | UNSIGNED 16 | | | | |
| Default | 0000h | | | | |
| EEPROM | yes | | | | |
| Data content | The checksum CRC-16-CCITT is calculated via the content of the object 6101h. | | | | |
| | Object | Sub-index | Name | Extent | Value |
| | 6101h | Safety configuration of the speed parameters | | | |
| | | 00h | Highest sub-index | 1 byte | 07h |
| | | 01h | Sub-index | 1 byte | 01h |
| | | 01h | Safety sense of direction | 2 bytes | Object 6101h sub-index 01h |
| | | 02h | Sub-index | 1 byte | 02h |
| | | 02h | Safety Preset value (calibration value) | 4 bytes | Object 6101h sub-index 02h |
| | | 03h | Sub-index | 1 byte | 04h |
| | | 03h | Safety speed source selector | 1 byte | Object 6101h sub-index 04h |
| | | 04h | Sub-index | 1 byte | 05h |
| | | 04h | Safety integration time of speed | 2 bytes | Object 6101h sub-index 05h |
| | | 05h | Sub-index | 1 byte | 06h |
| | | 05h | Safety multiplier | 2 bytes | 0001h |
| | | 06h | Sub-index | 1 byte | 07h |
| | | 06h | Safety divisor | 2 bytes | 0001h |

### 5.8.2.49 6200h: Cycle timer

Object 6200h sets a cycle time for the output of PDO1. This value is permanently linked to object 1301h: SRDO1 communication parameters

The communication behavior of SRDO1 can be determined via object 1301h.

| | |
|---|---|
| Sub-index | 00h |
| Description | indicates the largest supported sub-index |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 8 |
| Default | 6h |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 01h |
| Description | Direction of information |
| Access | ro, if NMT state is operational<br>rw, if NMT state is pre-operational |
| PDO mapping | no |
| Data type | UNSIGNED 8 |
| Default | 1 |
| EEPROM | yes |
| Data content | 0: SRDO1 is not valid<br>1: SRDO1 is TX and valid |

| | |
|---|---|
| Sub-index | 02h |
| Description | refresh time |
| Access | ro, if NMT state is operational<br>rw, if NMT state is pre-operational |
| PDO mapping | no |
| Data type | UNSIGNED 16 |
| Default | 25 |
| EEPROM | yes |
| Data content | 10...65535 |

| | |
|---|---|
| Sub-index | 03h |
| Description | tx: reserved<br>rx: SRVT |
| Access | ro, if NMT state is operational<br>rw, if NMT state is pre-operational |
| PDO mapping | no |
| Data typ | UNSIGNED 8 |
| Default | 20 |

| | |
|---|---|
| EEPROM | no |
| Data content | 20 |

| | |
|---|---|
| Sub-index | 04h |
| Description | transmission type |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 8 |
| Default | 254 |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 05h |
| Description | COB-ID1 |
| Access | ro, if NMT state is operational<br>rw, if NMT state is pre-operational |
| PDO mapping | no |
| Data type | UNSIGNED 32 |
| Default | 000000FFh + 2*Node ID |
| EEPROM | yes |
| Data content | 257...383. Only odd numbers are allowed. |

| | |
|---|---|
| Sub-index | 06h |
| Description | COB-ID2 |
| Access | ro, if NMT state is operational<br>rw, if NMT state is pre-operational |
| PDO mapping | no |
| Data type | UNSIGNED 32 |
| Default | 00000100h + 2*Node ID |
| EEPROM | yes |
| Data content | 258...384. Only even numbers are allowed. |

#### 5.8.2.50 1302h: SRDO2 communication parameters

The communication behavior of SRDO2 can be determined via 1302h.

| | |
|---|---|
| Sub-index | 00h |
| Description | indicates the largest supported sub-index |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 8 |
| Default | 6h |

| | |
|---|---|
| EEPROM | no |

| | |
|---|---|
| Sub-index | 01h |
| Description | Direction of information |
| Access | ro, if NMT state is operational<br>rw, if NMT state is pre-operational |
| PDO mapping | no |
| Data type | UNSIGNED 8 |
| Default | 1 |
| EEPROM | yes |
| Data content | 0: SRDO2 is not valid<br>1: SRDO2 is TX and valid |

| | |
|---|---|
| Sub-index | 02h |
| Description | refresh time |
| Access | ro, if NMT state is operational<br>rw, if NMT state is pre-operational |
| PDO mapping | no |
| Data type | UNSIGNED 16 |
| Default | 25 |
| EEPROM | yes |
| Data content | 10...65535 |

| | |
|---|---|
| Sub-index | 03h |
| Description | tx: reserved<br>rx: SRVT |
| Access | ro, if NMT state is operational<br>rw, if NMT state is pre-operational |
| PDO mapping | no |
| Data type | UNSIGNED 8 |
| Default | 20 |
| EEPROM | nein |
| Dateninhalt | 20 |

| | |
|---|---|
| Sub-index | 04h |
| Description | transmission type |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 8 |
| Default | 254 |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 05h |
| Description | COB-ID1 |
| Access | ro, if NMT state is operational<br>rw, if NMT state is pre-operational |
| PDO mapping | no |
| Data type | UNSIGNED 32 |
| Default | 0000010Fh + 2*Node ID |
| EEPROM | yes |
| Data content | 257...383. Only odd numbers are allowed. |

| | |
|---|---|
| Sub-index | 06h |
| Description | COB-ID2 |
| Access | ro, if NMT state is operational<br>rw, if NMT state is pre-operational |
| PDO mapping | no |
| Data type | UNSIGNED 32 |
| Default | 00000110h + 2*Node ID |
| EEPROM | yes |
| Data content | 258...384. Only even numbers are allowed. |

#### 5.8.2.51 1381h: SRDO1 mapping parameters

Object 1381h determines the objects that are mapped in the first Transmit SRDO (SRDO1).

| | |
|---|---|
| Sub-index | 00h |
| Description | indicates the largest supported sub-index |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 8 |
| Default | 8h |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 01h |
| Description | 1st object of the SRDO1 message with the COB ID1 (data byte 0) |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 32 |
| Default | 61200108h (position value object 6120h, sub-index 01h, 8bit) |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 02h |

| | |
|---|---|
| Description | 1st object of the SRDO1 message with the COB ID2 (data byte 0) |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 32 |
| Default | 61210108h (inverted position value object 6121h, sub-index 01h, 8bit) |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 03h |
| Description | 2nd object of the SRDO1 message with the COB ID1 (data byte 1) |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 32 |
| Default | 61200208h (position value object 6120h, sub-index 02h, 8bit) |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 04h |
| Description | 2nd object of the SRDO1 message with the COB ID2 (data byte 1) |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 32 |
| Default | 61210208h (inverted position value object 6121h, sub-index 02h, 8bit) |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 05h |
| Description | 3rd object of the SRDO1 message with the COB ID1 (data byte 2) |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 32 |
| Default | 61200308h (position value object 6120h, sub-index 03h, 8bit) |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 06h |
| Description | 3rd object of the SRDO1 message with the COB-ID2 (data byte 2) |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 32 |
| Default | 61200308h (inverted position value object 6121h, sub-index 03h, 8bit) |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 07h |

| | |
|---|---|
| Description | 4th object of the SRDO1 message with the COB ID1 (data byte 3) |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 32 |
| Default | 61200408h (position value object 6120h, sub-index 04h, 8bit) |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 08h |
| Description | 4th object of the SRDO1 message with the COB ID2 (data byte 3) |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 32 |
| Default | 61210408h (inverted position value object 6121h, sub-index 04h, 8bit) |
| EEPROM | no |

#### 5.8.2.52 1382h: SRDO2 mapping parameters

Object 1382h determines the objects that are mapped in the second Safety Transmit SRDO (SRDO2).

| | |
|---|---|
| Sub-index | 00h |
| Description | indicates the largest supported sub-index |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 8 |
| Default | 4h |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 01h |
| Description | 1st object of the SRDO2 message with the COB ID1 (data byte 0) |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 32 |
| Default | 61240108h (speed value object 6124h, sub-index 01h, 8bit) |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 02h |
| Description | 1st object of the SRDO2 message with the COB ID2 (data byte 0) |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 32 |
| Default | 61250108h (inv. speed value object 6125h, sub-index 01h, 8bit) |

| | |
|---|---|
| EEPROM | no |

| | |
|---|---|
| Sub-index | 03h |
| Description | 2nd object of the SRDO2 message with the COB ID1 (data byte 1) |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 32 |
| Default | 61240208h (speed value object 6124h, sub-index 02h, 8bit) |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 04h |
| Description | 2nd object of the SRDO2 message with the COB ID2 (data byte 1) |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 32 |
| Default | 61250208h (inv. speed value object 6125h, sub-index 02h, 8bit) |
| EEPROM | no |

#### 5.8.2.53 13FEh: Safety configuration

The SRDO configuration can be switched to valid by means of object 13FEh.

| | |
|---|---|
| Sub-index | 00h |
| Description | This parameter will be switched to invalid automatically after changing a parameter in object 1301h or 1302h and must be switched valid again via this object. Switching to valid is only enabled if the correct signatures are entered in object 13FFh. |
| Access | rw |
| PDO mapping | no |
| Data type | UNSIGNED 8 |
| Default | 0h |
| EEPROM | yes |
| Data content | A5h: SRDO1 and SRDO2 configuration valid<br>00h...A4h and A6h...FFh: SRDO1 and SRDO2 configuration invalid |

#### 5.8.2.54 13FFh: Safety configuration signature (checksum)

This object 13FFh receives the signatures (checksums) via the CANopen Safety parameters of SRDO1 and SRDO2. Only a checksum which is valid at that time can be transmitted. The checksum is checked anew before switching the configuration to valid. Any change to the configuration will be valid only after passing.

| | |
|---|---|
| Sub-index | 00h |
| Description | indicates the largest supported sub-index |

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

| | | | | | |
|---|---|---|---|---|---|
| Sub-index | 01h | | | | |
| Description | SRDO1 signature (checksum) | | | | |
| Access | rw | | | | |
| PDO mapping | no | | | | |
| Data type | UNSIGNED 16 | | | | |
| Default | 0000h | | | | |
| EEPROM | yes | | | | |
| Data content | The checksum CRC-16-CCITT is calculated via the content of the two objects 1301h and 1381h. | | | | |
| | Object | Sub-index | Name | Extent | Value |
| | 1301h | SRDO1 communication parameter | | | |
| | | 01h | Direction of information | 1 byte | Object 1301h sub-index 01h |
| | | 02h | Refresh time | 2 bytes | Object 1301h sub-index 02h |
| | | 03h | tx: reserved<br>rx: SRVT | 1 Byte | Object 1301h sub-index 03h |
| | | 05h | COB ID 1 | 4 bytes | Object 1301h sub-index 05h |
| | | 06h | COB ID 2 | 4 bytes | Object 1301h sub-index 06h |
| | 1381h | SRDO1 mapping parameters | | | |
| | | 00h | Highest sub-index | 1 byte | 08h |
| | | 01h | Sub-index | 1 byte | 01h |
| | | 01h | Position value 1st byte | 4 bytes | 61200108h |
| | | 02h | Sub-index | 1 byte | 02h |
| | | 02h | Inverted position value 1st byte | 4 bytes | 61210108h |
| | | 03h | Sub-index | 1 byte | 03h |
| | | 03h | Position value 2nd byte | 4 bytes | 61200208h |
| | | 04h | Sub-index | 1 byte | 04h |
| | | 04h | Inverted position value 2nd byte | 4 bytes | 61210208h |
| | | 05h | Sub-index | 1 byte | 05h |
| | | 05h | Position value 3rd byte | 4 bytes | 61200308h |
| | | 06h | Sub-index | 1 byte | 06h |

| | | | | | |
|---|---|---|---|---|---|
| | | 06h | Inverted position value 3rd byte | 4 bytes | 61210308h |
| | | 07h | Sub-index | 1 byte | 07h |
| | | 07h | Position value 4th byte | 4 bytes | 61200408h |
| | | 08h | Sub-index | 1 byte | 08h |
| | | 08h | Inverted position value 4th byte | 4 bytes | 61210408h |

| | | | | | |
|---|---|---|---|---|---|
| Sub-index | 02h | | | | |
| Description | SRDO2 signature (checksum) | | | | |
| Access | rw | | | | |
| PDO mapping | no | | | | |
| Data type | UNSIGNED 16 | | | | |
| Default | 0000h | | | | |
| EEPROM | yes | | | | |
| Data content | The checksum CRC-16-CCITT is calculated via the content of the two objects 1302h and 1382h. | | | | |
| | Object | Sub-index | Name | Extent | Value |
| | 1302h | SRDO2 communication parameter | | | |
| | | 01h | Direction of information | 1 byte | Object 1302h sub-index 01h |
| | | 02h | Refresh time | 2 bytes | Object 1302h sub-index 02h |
| | | 03h | tx: reserved<br>rx: SRVT | 1 Byte | Object 1302h sub-index 03h |
| | | 05h | COB ID 1 | 4 bytes | Object 1302h sub-index 05h |
| | | 06h | COB ID 2 | 4 bytes | Object 1302h sub-index 06h |
| | 1382h | SRDO2 mapping parameters | | | |
| | | 00h | Highest sub-index | 1 byte | 04h |
| | | 01h | Sub-index | 1 byte | 01h |
| | | 01h | speed value 1st byte | 4 bytes | 61240108h |
| | | 02h | Sub-index | 1 byte | 02h |
| | | 02h | Inverted speed value 1st byte | 4 bytes | 61250108h |
| | | 03h | Sub-index | 1 byte | 03h |
| | | 03h | speed value 2nd byte | 4 bytes | 61240208h |
| | | 04h | Sub-index | 1 byte | 04h |
| | | 04h | Inverted speed value 2nd byte | 4 bytes | 61250208h |

1800h: 1. Transmit PDO Parameter sub-index 05h. Timer-controlled output is active as soon as a valid cycle time has been entered and the device is operated in the operational mode. Value 0h disables the function.

| | |
|---|---|
| Sub-index | 00h |
| Description | Cycle timer |
| Access | rw |
| PDO mapping | no |
| Data type | UNSIGNED 16 |
| Default | 0h |
| EEPROM | yes |
| Data content | 0d ... 65535d (0h...FFFFh) |

### 5.8.2.55 6400h: Operating range (Area state register)

Object 6400h outputs whether the position value is within the set work areas 1 and 2.

| | |
|---|---|
| Sub-index | 00h |
| Description | indicates the largest supported sub-index |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 8 |
| Default | 2h |
| EEPROM | no |

| | | |
|---|---|---|
| Sub-index | 01h | |
| Description | Status of operating range 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 limit set in Object 6401.1h |
| | Bit 1 | 0: Position value is within the work area<br>1: Position value is larger than the limit set in Object 6402.1h |
| | Bit 0 | 0: Position value is within the work area set<br>1: Position value is beyond the work area set |

| | |
|---|---|
| Sub-index | 02h |
| Description | Status 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 limit set in Object 6401.2h |
| | Bit 1 | 0: Position value is within the work area<br>1: Position value is larger than the limit set in Object 6402.2h |
| | Bit 0 | 0: Position value is within the work area set<br>1: Position value is beyond the work area set |

### 5.8.2.56 6401h: Work Area Low Limit

Each a lower limit can be set for one of the two work areas via Object 6401h.

| | |
|---|---|
| Sub-index | 00h |
| Description | indicates the largest supported sub-index |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 8 |
| Default | 2h |
| EEPROM | no |

| | | | |
|---|---|---|---|
| Sub-index | 01h | | |
| Description | Lower limit of work area 1 | | |
| Access | rw | | |
| PDO mapping | no | | |
| Data type | SIGNED 32 | | |
| Default | 0h | | |
| EEPROM | yes | | |
| Data content | Encoder type | Default | With changed APU (Object 6001h) |
| | Single-turn | 0...16383 | 0...((APU*1) – 1) |
| | 4 Bit Multi-turn | 0...262143 | 0...((APU*16) – 1) |
| | 8 Bit Multi-turn | 0...4194303 | 0...((APU*256) – 1) |
| | 12 Bit Multi-turn | 0...67108863 | 0...((APU*4096) – 1) |

| | |
|---|---|
| Sub-index | 02h |
| Description | Lower limit of work area 2 |
| Access | rw |
| PDO mapping | no |
| Data type | SIGNED 32 |

| | | | |
|---|---|---|---|
| Default | 0h | | |
| EEPROM | Yes | | |
| Data content | Encoder type | Default | With maximum APU (Object 6001h) |
| | Single-turn | 0...16383 | 0...((APU*1) – 1) |
| | 4 Bit Multi-turn | 0...262143 | 0...((APU*16) – 1) |
| | 8 Bit Multi-turn | 0...4194303 | 0...((APU*256) – 1) |
| | 12 Bit Multi-turn | 0...67108863 | 0...((APU*4096) – 1) |

### 5.8.2.57 6402h: Work Area High Limit

Each an upper limit can be set for one of the two work areas via Object 6402h.

| | |
|---|---|
| Sub-index | 00h |
| Description | indicates the largest supported sub-index |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 8 |
| Default | 2h |
| EEPROM | no |

| | | | |
|---|---|---|---|
| Sub-index | 01h | | |
| Description | Upper limit of work area 1 | | |
| Access | rw | | |
| PDO mapping | no | | |
| Data type | SIGNED 32 | | |
| Default | 0h | | |
| EEPROM | yes | | |
| Data content | Encoder type | Default | With changed APU (Object 6001h) |
| | Single-turn | 0...16383 | 0...((APU*1) – 1) |
| | 4 Bit Multi-turn | 0...262143 | 0...((APU*16) – 1) |
| | 8 Bit Multi-turn | 0...4194303 | 0...((APU*256) – 1) |
| | 12 Bit Multi-turn | 0...67108863 | 0...((APU*4096) – 1) |

| | |
|---|---|
| Sub-index | 02h |
| Description | Upper limit of work area 2 |
| Access | rw |
| PDO mapping | no |
| Data type | SIGNED 32 |
| Default | 0h |
| EEPROM | yes |

| | | | |
|---|---|---|---|
| Data content | Encoder type | Default | With changed APU (Object 6001h) |
| | Single-turn | 0...16383 | 0...((APU*1) – 1) |
| | 4 Bit Multi-turn | 0...262143 | 0...((APU*16) – 1) |
| | 8 Bit Multi-turn | 0...4194303 | 0...((APU*256) – 1) |
| | 12 Bit Multi-turn | 0...67108863 | 0...((APU*4096) – 1) |

#### 5.8.2.58 6500h: Operating Status

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

| | | |
|---|---|---|
| Sub-index | 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 | not used |
| | Bit 2 | 0: Scaling disabled<br>1: Scaling enabled |
| | Bit 1 | not used |
| | Bit 0 | 0: Clockwise (CW) sense of rotation I<br>1: Counter-clockwise (CCW) sense of rotation E |

#### 5.8.2.59 6501h: Single-turn resolution

Object 6501h indicates the physical number of measurement steps per revolution.

| | |
|---|---|
| Sub-index | 00h |
| Description | Physical resolution |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 32 |
| Default | 16384d (4000h) |
| EEPROM | no |

#### 5.8.2.60 6502h: Number of distinguishable revolutions

Object 6502h indicates the number of resolutions the encoder is able to sense.

| | |
|---|---|
| Sub-index | 00h |
| Description | Physical resolution |

| | | |
|---|---|---|
| Access | ro | |
| PDO mapping | no | |
| Data type | UNSIGNED 16 | |
| Default | Single-turn | 1 |
| | 4 Bit Multi-turn | 16 |
| | 8 Bit Multi-turn | 256 |
| | 12 Bit Multi-turn | 4096 |
| EEPROM | no | |

#### 5.8.2.61 6503h: Alarms

Object 6503h indicates other device-specific alarm messages in addition to the errors reported via the Emergency message. In the case of an error, the associated bit is set to 1.

| | | |
|---|---|---|
| Sub-index | 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 work area 2<br>1: Position limit 2 exceeded or undershot (Work area 2) |
| | Bit 12 | 0: Position value within work area 1<br>1: Position limit 1 exceeded or undershot (Work area 1) |
| | Bit 11 ... 1 | Not used |
| | Bit 0 | 0: Position value valid<br>1: position value invalid |

#### 5.8.2.62 6504h: Supported Alarms

This Object 6504h indicates the alarm messages that are supported. The relevant bits are set.

| | | |
|---|---|---|
| Sub-index | 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 2 error |
| | Bit 12 | Position limit 1 error |
| | Bit 11 ... 1 | Not used |
| | Bit 0 | position error |

#### 5.8.2.63 6505h: Warnings

The warnings displayed via Object 6505h indicate that tolerances of internal encoder parameters have been exceeded. However, unlike with alarm messages, the position value can be valid in case of a warning.

| | | |
|---|---|---|
| Sub-index | 00h | |
| Description | Warnings | |
| Access | ro | |
| PDO mapping | no | |
| Data type | UNSIGNED 16 | |
| Default | 0h | |
| EEPROM | no | |
| Data content | Bit 1 ... 15 | Not used |
| | Bit 0 | 0: Speed ok<br>1: maximum speed exceeded or limit speed set exceeded or undershot (2003h: Limit speed low) |

#### 5.8.2.64 6506h: Supported Warnings

The Object 6506h indicates the warnings that are supported.

| | | |
|---|---|---|
| Sub-index | 00h | |
| Description | Supported warnings | |
| Access | ro | |
| PDO mapping | no | |
| Data type | UNSIGNED 16 | |
| Default | 0001h | |
| EEPROM | no | |
| Data content | Bit 1 ... 15 | Not used |
| | Bit 0 | Speed warning |

#### 5.8.2.65 6507h: Profile and Software Version

The object 6507h indicates the encoder profile used (CANopen Device profile for encoders) and the version number of the firmware state.

| | |
|---|---|
| Sub-index | 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 |

#### 5.8.2.66 6508h: Operating Time

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

| | |
|---|---|
| Sub-index | 00h |
| Description | Hourmeter |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 32 |
| Default | FFFFFFFFh |
| EEPROM | no |

#### 5.8.2.67 6509h: Offset value

The Object 6509h indicates the difference between encoder value and the scaled and offset against

preset and application offset position value.

| | |
|---|---|
| Sub-index | 00h |
| Description | Encoder state at the time of calibration |
| Access | ro |
| PDO mapping | no |
| Data type | SIGNED 32 |
| Default | 0h |
| EEPROM | yes |

#### 5.8.2.68 650Ah: Module Identification

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

| | |
|---|---|
| Sub-index | 00h |
| Description | indicates the largest supported sub-index |
| Access | ro |
| PDO mapping | no |
| Data type | UNSIGNED 8 |

| | |
|---|---|
| Default | 3h |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 01h |
| Description | Manufacturer-specific offset value |
| Access | ro |
| PDO mapping | no |
| Data type | SIGNED 32 |
| Default | 0h |
| EEPROM | no |

| | |
|---|---|
| Sub-index | 02h |
| Description | Smallest transferable position value |
| Access | ro |
| PDO mapping | no |
| Data type | SIGNED 32 |
| Default | 0h |
| EEPROM | no |

| | | | |
|---|---|---|---|
| Sub-index | 03h | | |
| Description | Largest transferable position value | | |
| Access | ro | | |
| PDO mapping | no | | |
| Data type | SIGNED 32 | | |
| EEPROM | yes | | |
| Data content | Encoder type | Default | With maximum APU (Object 6001h) |
| | Single-turn | 16383 | 65534 ((65535 * 1) - 1) |
| | 4 Bit Multi-turn | 262143 | 1048559 ((65535 * 16) - 1) |
| | 8 Bit Multi-turn | 4194303 | 16776959 ((65535 * 256) - 1) |
| | 12 Bit Multi-turn | 67108863 | 268431359 ((65535 * 4096) - 1) |

#### 5.8.2.69 650Bh: Serial number

Object 650Bh provides the serial number of the encoder. The serial number is composed of the imprinted serial number and a suffixed 1 or 2 to enable unambiguous identification of each encoder.

| | |
|---|---|
| Sub-index | 00h |
| Description | Serial number |
| Access | ro |
| PDO mapping | no |

| | |
|---|---|
| Data type | UNSIGNED 32 |
| Default | 0h |
| EEPROM | yes |

### 5.8.2.70 650Dh: Absolute accuracy

The object 650Dh provides the absolute accuracy of the encoder in bits.

| | |
|---|---|
| Subindex | 00h |
| Description | Absolute accuracy |
| Access | ro |
| PDO-Mapping | no |
| Data type | UNSIGNED 8 |
| Default | 08h |
| EEPROM | no |

### 5.8.2.71 650Eh: Device functionality

Speed settings can be made via object 6031h.

| | | |
|---|---|---|
| Subindex | 00h | |
| Description | Device functionality | |
| Access | ro | |
| PDO-Mapping | no | |
| Data type | UNSIGNED 32 | |
| Default | CANopen: 0002h (class 2)<br>CANopen Safety: 0013h (class 3 + CANopen Safety) | |
| EEPROM | no | |
| Data content | Bit 15 ... 8 | not used |
| | Bit 7 ... 6 | 00: CANopen Safety<br>01 ... 11: not used |
| | Bit 5 | 0: Safety is not supported<br>1: Safety is supported |
| | Bit 4 | not used |
| | Bit 3 | 0: normal resolution<br>1: high resolution |
| | Bit 2 ... 0 | 000: reserved<br>001: class C1<br>010: class C2<br>011: class C3 |