

# LXM32M

## Module CANopen Fieldbus manual

V1.02, 12.2009



**CAN**open

## **Important information**

This manual is part of the product.

Carefully read this manual and observe all instructions.

Keep this manual for future reference.

Hand this manual and all other pertinent product documentation over to all users of the product.

Carefully read and observe all safety instructions and the chapter "Before you begin - safety information".

Some products are not available in all countries.

For information on the availability of products, please consult the catalog.

Subject to technical modifications without notice.

All details provided are technical data which do not constitute warranted qualities.

Most of the product designations are registered trademarks of their respective owners, even if this is not explicitly indicated.

# Table of contents



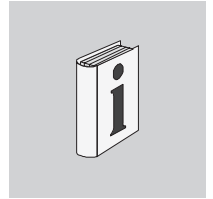
<b>Important information</b>	<b>2</b>
<b>Table of contents</b>	<b>3</b>
<b>About this manual</b>	<b>7</b>
Further reading	8
<b>1 Introduction</b>	<b>9</b>
1.1 CAN bus	9
1.2 CANopen technology	10
1.2.1 CANopen description language	10
1.2.2 Communication layers	10
1.2.3 Objects	11
1.2.4 CANopen profiles	12
1.3 Certifications	12
<b>2 Before you begin - safety information</b>	<b>13</b>
2.1 Qualification of personnel	13
2.2 Intended use	13
2.3 Hazard categories	14
2.4 Basic information	15
2.5 Standards and terminology	16
<b>3 Basics</b>	<b>17</b>
3.1 Communication profile	17
3.1.1 Object dictionary	17
3.1.2 Communication objects	18
3.1.3 Communication relationships	21
3.2 Service data communication	23
3.2.1 Overview	23
3.2.2 SDO data exchange	23
3.2.3 SDO message	24
3.2.4 Reading and writing data	25
3.2.5 Reading data longer than 4 bytes	27
3.3 Process data communication	28
3.3.1 Overview	28
3.3.2 PDO data exchange	28
3.3.3 PDO message	29
3.3.4 PDO mapping	33
3.4 Synchronization	36


3.5	Emergency service . . . . .	38
3.5.1	Error evaluation and handling . . . . .	38
3.6	Network management services . . . . .	40
3.6.1	NMT services for device control . . . . .	41
3.6.2	NMT services for connection monitoring . . . . .	43
<b>4</b>	<b>Installation. . . . .</b>	<b>47</b>
4.1	Installation of the module . . . . .	47
4.2	Electrical installation . . . . .	48
<b>5</b>	<b>Commissioning. . . . .</b>	<b>51</b>
5.1	Commissioning the device . . . . .	51
5.2	Address and baud rate . . . . .	52
<b>6</b>	<b>Operation. . . . .</b>	<b>53</b>
6.1	Operating states . . . . .	54
6.1.1	Indication of the operating state . . . . .	54
6.1.2	Changing the operating state . . . . .	56
6.2	Operating modes . . . . .	57
6.2.1	Starting and changing an operating mode . . . . .	57
6.2.2	Operating mode Jog . . . . .	58
6.2.3	Operating mode Electronic Gear . . . . .	60
6.2.4	Operating mode Profile Torque . . . . .	61
6.2.5	Operating mode Profile Velocity . . . . .	63
6.2.6	Operating mode Profile Position . . . . .	65
6.2.7	Operating mode Homing . . . . .	67
6.2.8	Operating mode Motion Sequence . . . . .	69
<b>7</b>	<b>Diagnostics and troubleshooting . . . . .</b>	<b>71</b>
7.1	Fieldbus communication error diagnostics . . . . .	71
7.2	Fieldbus status LEDs . . . . .	72
7.3	Error diagnostics via fieldbus . . . . .	74
7.3.1	Message objects . . . . .	74
7.3.2	Messages on the device status . . . . .	74
7.4	CANopen error messages . . . . .	75
7.4.1	Error register . . . . .	75
7.4.2	Error code table . . . . .	75
7.4.3	SDO error message ABORT . . . . .	76

<b>8</b>	<b>Object dictionary</b>	<b>79</b>
8.1	Specifications for the objects	79
8.2	Overview of object group 1000h	80
8.3	Assignment object group 3000h	84
8.4	Assignment object group 6000h	93
8.5	Details of object group 1000h	94
8.5.1	1000h Device type	94
8.5.2	1001h Error register	95
8.5.3	1003h Predefined error field	96
8.5.4	1005h COB ID SYNC message	97
8.5.5	1008h Manufacturer device name	98
8.5.6	1009h Manufacturer hardware version	98
8.5.7	100Ah Manufacturer software version	99
8.5.8	100Ch Guard time	99
8.5.9	100Dh Life time factor	100
8.5.10	1010h Save Parameters	101
8.5.11	1011h Restore default parameters	102
8.5.12	1014h COB ID-Emergency message	103
8.5.13	1015h Inhibit time emergency message	104
8.5.14	1016h Consumer Heartbeat Time	104
8.5.15	1017h Producer Heartbeat Time	105
8.5.16	1018h Identity Object	106
8.5.17	1020h data on configuration	107
8.5.18	1029h error behavior	108
8.5.19	1200h 1st server SDO parameter	109
8.5.20	1201h 2nd server SDO parameter	110
8.5.21	1400h 1st receive PDO parameter	111
8.5.22	1401h 2nd receive PDO parameter	113
8.5.23	1402h 3rd receive PDO-Parameter	115
8.5.24	1403h 4th receive PDO parameter	116
8.5.25	1600h 1st receive PDO mapping	117
8.5.26	1601h 2nd receive PDO mapping	118
8.5.27	1602h 3rd receive PDO mapping	119
8.5.28	1603h 4th receive PDO mapping	120
8.5.29	1800h 1st transmit PDO parameter	121
8.5.30	1801h 2nd transmit PDO parameter	123
8.5.31	1802h 3rd transmit PDO parameter	125
8.5.32	1803h 4th transmit PDO parameter	127
8.5.33	1A00h 1st transmit PDO mapping	129
8.5.34	1A01h 2nd transmit PDO mapping	130
8.5.35	1A02h 3rd transmit PDO mapping	131
8.5.36	1A03h 4th transmit PDO mapping	132

<b>9</b>	<b>Accessories and spare parts</b>	<b>133</b>
9.1	CANopen cable with connectors	133
9.2	CANopen connectors, distributors, terminating resistors	134
9.3	CANopen cables	134
9.4	Spare parts connectors, fans, cover plates	135
<b>10</b>	<b>Glossary</b>	<b>137</b>
10.1	Units and conversion tables	137
10.1.1	Length	137
10.1.2	Mass	137
10.1.3	Force	137
10.1.4	Power	137
10.1.5	Rotation	138
10.1.6	Torque	138
10.1.7	Moment of inertia	138
10.1.8	Temperature	138
10.1.9	Conductor cross section	138
10.2	Terms and Abbreviations	139
<b>11</b>	<b>Index</b>	<b>141</b>

## About this manual



	<p>This manual applies to the module CANopen for the product LXM32M, module identification CAN.</p> <p>The information provided in this manual supplements the product manual.</p>
<i>Source manuals</i>	<p>The latest versions of the manuals can be downloaded from the Internet at:</p> <p><a href="http://www.schneider-electric.com">http://www.schneider-electric.com</a></p>
<i>Source EPLAN Macros</i>	<p>For easier engineering, macro files and product master data are available for download from the Internet at:</p> <p><a href="http://www.schneider-electric.com">http://www.schneider-electric.com</a></p>
<i>Corrections and suggestions</i>	<p>We always try to further optimize our manuals. We welcome your suggestions and corrections.</p> <p>Please get in touch with us by e-mail:  <a href="mailto:techcomm@schneider-electric.com">techcomm@schneider-electric.com</a>.</p>
<i>Work steps</i>	<p>If work steps must be performed consecutively, this sequence of steps is represented as follows:</p> <ul style="list-style-type: none"> <li>■ Special prerequisites for the following work steps</li> <li>▶ Step 1</li> <li>◁ Specific response to this work step</li> <li>▶ Step 2</li> </ul> <p>If a response to a work step is indicated, this allows you to verify that the work step has been performed correctly.</p> <p>Unless otherwise stated, the individual steps must be performed in the specified sequence.</p>
<i>Making work easier</i>	<p>Information on making work easier is highlighted by this symbol:</p> <div style="text-align: center;">  </div> <p><i>Sections highlighted this way provide supplementary information on making work easier.</i></p>
<i>Parameters</i>	<p>In text sections, parameters are shown with the parameter name, for example <code>_IO_act</code>. A list of the parameters can be found in the product manual in the chapter Parameters.</p>
<i>SI units</i>	<p>SI units are the original values. Converted units are shown in brackets behind the original value; they may be rounded.</p> <p>Example:          Minimum conductor cross section: 1.5 mm<sup>2</sup> (AWG 14)</p>

<i>Inverted signals</i>	Inverted signals are represented by an overline, for example $\overline{\text{STO\_A}}$ or $\overline{\text{STO\_B}}$ .
<i>Glossary</i>	Explanations of special technical terms and abbreviations.
<i>Index</i>	List of keywords with references to the corresponding page numbers.

## Further reading

	Recommended literature for further reading
<i>CAN users and manufacturers organization</i>	CiA - CAN in Automation Am Weichselgarten 26 D-91058 Erlangen <a href="http://www.can-cia.org/">http://www.can-cia.org/</a>
<i>CANopen standards</i>	<ul style="list-style-type: none"><li>• CiA Standard 301 (DS301) CANopen application layer and communication profile</li><li>• CiA Standard 402 (DSP402) Device profile for drives and motion control</li><li>• ISO 11898: Controller Area Network (CAN) for high speed communication</li><li>• EN 50325-4: Industrial communications subsystem based on ISO 11898 for controller device interfaces (CANopen)</li></ul>
<i>Literature</i>	Controller Area Network Konrad Etschberger, Carl Hanser Verlag ISBN 3-446-19431-2



# 1 Introduction

# 1

## 1.1 CAN bus

The CAN bus (**C**ontroller **A**rea **N**etwork) was originally developed for fast, economical data transmission in the automotive industry. Today, the CAN bus is also used in industrial automation technology and has been further developed for communication at fieldbus level.

### *Features of the CAN bus*

The CAN bus is a standardized, open bus enabling communication between devices, sensors and actuators from different manufacturers. The features of the CAN bus comprise

- Multimaster capability  
Each device in the fieldbus can transmit and receive data independently without depending on an "ordering" master functionality.
- Message-oriented communication  
Devices can be integrated into a running network without reconfiguration of the entire system. The address of a new device does not need to be specified on the network.
- Prioritization of messages  
Messages with higher priority are sent first for time-critical applications.
- Residual error probability  
Various security features in the network reduce the probability of undetected incorrect data transmission to less than  $10^{-11}$ .

### *Transmission technology*

In the CAN bus, multiple devices are connected via a bus cable. Each network device can transmit and receive messages. Data between network devices are transmitted serially.

### *Network devices*

Examples of CAN bus devices are

- Automation devices, e.g. PLCs
- PCs
- Input/output modules
- Drives
- Analysis devices
- Sensors and actuators

## 1.2 CANopen technology

### 1.2.1 CANopen description language

CANopen is a device- and manufacturer-independent description language for communication via the CAN bus. CANopen provides a common basis for interchanging commands and data between CAN bus devices.

### 1.2.2 Communication layers

CANopen uses the CAN bus technology for data communication.

CANopen is based on the basic network services for data communication as per the ISO-OSI model model. 3 layers enable data communication via the CAN bus.

- Physical Layer
- Data Link Layer
- Application Layer

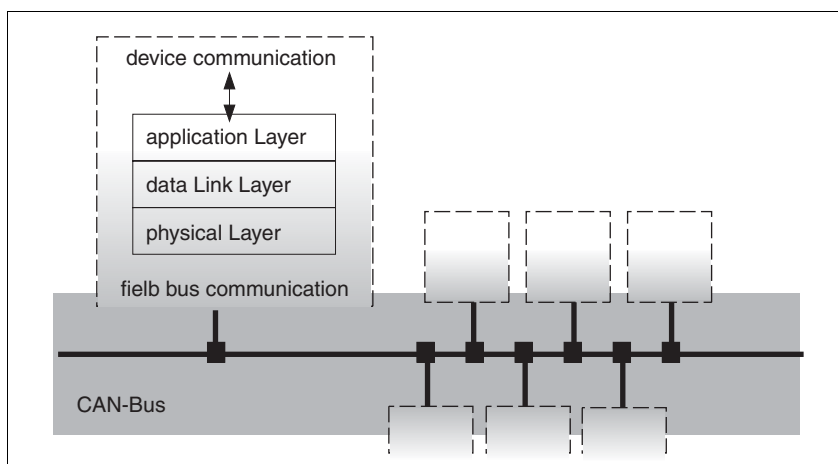


Figure 1.1 CANopen layer model

*Physical Layer* The physical layer defines the electrical properties of the CAN bus such as connectors, cable length and cable properties such as bit coding and bit timing.

*Data Link Layer* The data link layer connects the network devices. It assigns priorities to individual data packets and monitors and corrects errors.

*Application Layer* The application layer uses communication objects (COB) to exchange data between the various devices. Communication objects are elementary components for creating a CANopen application.

### 1.2.3 Objects

All processes under CANopen are executed via objects. Objects carry out different tasks; they act as communication objects for data transport to the fieldbus, control the process of establishing a connection or monitor the network devices. If objects are directly linked to the device (device-specific objects), the device functions can be used and changed via these objects.



*The product provides corresponding parameters for CANopen object groups 3000<sub>h</sub> and 6000<sub>h</sub>. The names of the parameters and the data type of the parameters may be different from the DSP402 definition for object group 6000<sub>h</sub>. In this case, enter the data type according to the DSP402. A detailed description of all parameters can be found in the product manual in the Parameters chapter.*

**Object dictionary** The object dictionary of each network device allows for communication between the devices. Other devices find all objects with which they can communicate in this dictionary.

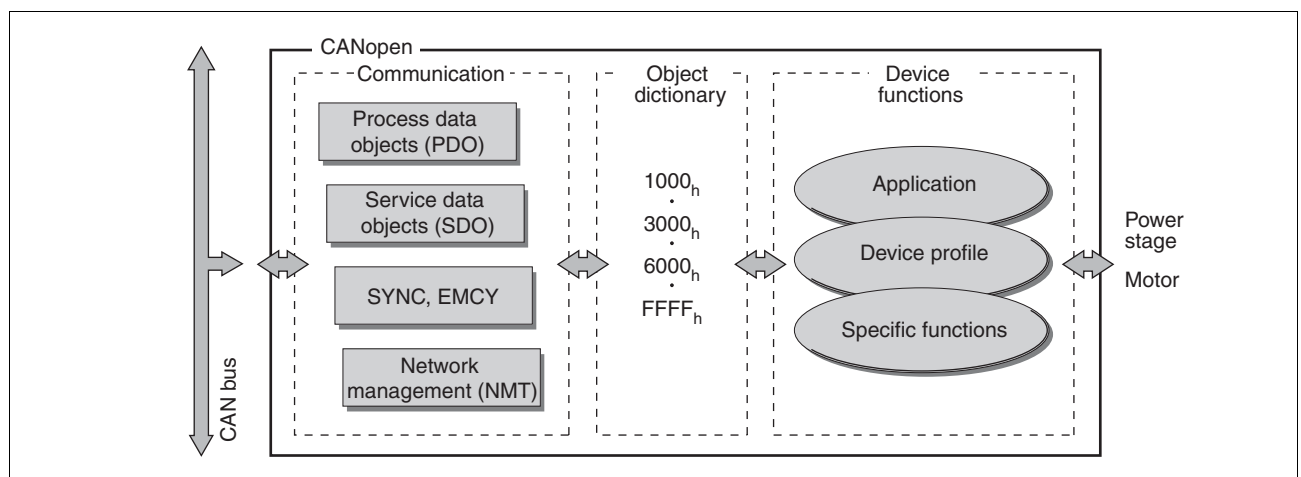


Figure 1.2 Device model with object dictionary

Objects for describing the data types and executing the communication tasks and device functions under CANopen are included in the object dictionary.

**Object index** Each object is addressed by means of a 16 bit index, which is represented as a four-digit hexadecimal number. The objects are arranged in groups in the object dictionary. The following table shows an overview of the object dictionary as per the CANopen.

Index range (hex)	Object groups
1000 <sub>h</sub> -2FFF <sub>h</sub>	Communication profile
3000 <sub>h</sub> -5FFF <sub>h</sub>	Vendor-specific objects
6000 <sub>h</sub> -9FFF <sub>h</sub>	Standardized device profiles
A000 <sub>h</sub> -FFFF <sub>h</sub>	Reserved

For a list of all CANopen objects see chapter 8 "Object dictionary".

## 1.2.4 CANopen profiles

### *Standardized profiles*

Standardized profiles describe objects that are used with different devices without additional configuration. The users and manufacturers organization CAN in Automation has standardized various profiles. These include:

- DS301 communication profile
- DSP402 device profile

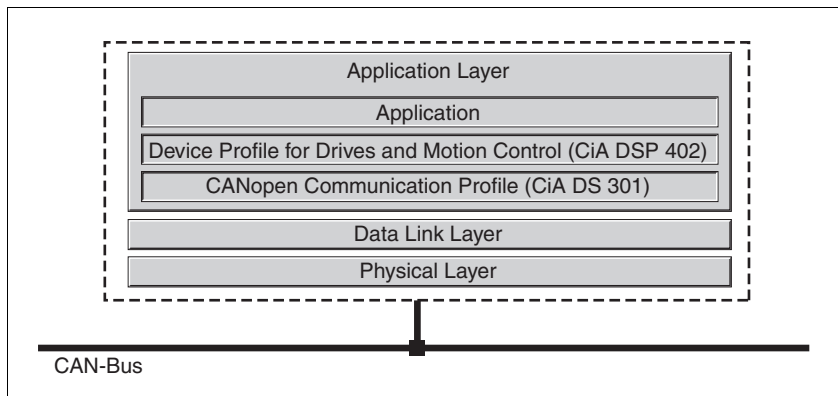


Figure 1.3 CANopen reference model

### *DS301 communication profile*

The DS301 communication profile is the interface between device profiles and CAN bus. It was specified in 1995 under the name DS301 and defines uniform standards for common data exchange between different device types under CANopen.

The objects of the communication profile in the device carry out the tasks of data exchange and parameter exchange with other network devices and initialize, control and monitor the device in the network.

### *DSP402 device profile*

The DSP402 device profile describes standardized objects for positioning, monitoring and settings of drives. The tasks of the objects include:

- Device monitoring and status monitoring (Device Control)
- Standardized parameterization
- Changing, monitoring and execution of operating modes

### *Vendor-specific profiles*

The basic functions of a device can be used with objects of standardized device profiles. Only vendor-specific device profiles offer the full range of functions. The objects with which the special functions of a device can be used under CANopen are defined in these vendor-specific device profiles.

## 1.3 Certifications

Product certifications:

Certified by	Assigned number	Validity
CiA (Can in Automation)	CiA200906-301V402/20-0104	

## 2 Before you begin - safety information

# 2

The information provided in this manual supplements the product manual. Carefully read the product manual before using the product.

### 2.1 Qualification of personnel

Only appropriately trained persons who are familiar with and understand the contents of this manual and all other pertinent product documentation are authorized to work on and with this product. In addition, these persons must have received safety training to recognize and avoid hazards involved. These persons must have sufficient technical training, knowledge and experience and be able to foresee and detect potential hazards that may be caused by using the product, by changing the settings and by the mechanical, electrical and electronic equipment of the entire system in which the product is used.

All persons working on and with the product must be fully familiar with all applicable standards, directives, and accident prevention regulations when performing such work.

### 2.2 Intended use

The functions described in this manual are only intended for use with the basic product; you must read and understand the appropriate product manual.

The product may only be used in compliance with all applicable safety regulations and directives, the specified requirements and the technical data.

Prior to using the product, you must perform a risk assessment in view of the planned application. Based on the results, the appropriate safety measures must be implemented.

Since the product is used as a component in an entire system, you must ensure the safety of persons by means of the design of this entire system (for example, machine design).

Operate the product only with the specified cables and accessories. Use only genuine accessories and spare parts.

Any use other than the use explicitly permitted is prohibited and can result in hazards.

Electrical equipment should be installed, operated, serviced, and maintained only by qualified personnel.

The product must NEVER be operated in explosive atmospheres (hazardous locations, Ex areas).

## 2.3 Hazard categories

Safety instructions to the user are highlighted by safety alert symbols in the manual. In addition, labels with symbols and/or instructions are attached to the product that alert you to potential hazards.

Depending on the seriousness of the hazard, the safety instructions are divided into 4 hazard categories.

### DANGER

DANGER indicates an imminently hazardous situation, which, if not avoided, **will result** in death or serious injury.

### WARNING

WARNING indicates a potentially hazardous situation, which, if not avoided, **can result** in death, serious injury, or equipment damage.

### CAUTION

CAUTION indicates a potentially hazardous situation, which, if not avoided, **can result** in injury or equipment damage.

### CAUTION

CAUTION used without the safety alert symbol, is used to address practices not related to personal injury (e.g. **can result** in equipment damage).

## 2.4 Basic information

### DANGER

#### UNINTENDED CONSEQUENCES OF EQUIPMENT OPERATION

When the system is started, the drives are usually out of the operator's view and cannot be visually monitored.

- Only start the system if there are no persons in the hazardous area.

**Failure to follow these instructions will result in death or serious injury.**

### WARNING

#### LOSS OF CONTROL

- The designer of any control scheme must consider the potential failure modes of control paths and, for certain critical functions, provide a means to achieve a safe state during and after a path failure. Examples of critical control functions are EMERGENCY STOP, overtravel stop, power outage and restart.
- Separate or redundant control paths must be provided for critical functions.
- System control paths may include communication links. Consideration must be given to the implication of unanticipated transmission delays or failures of the link.
- Observe the accident prevention regulations and local safety guidelines.<sup>1)</sup>
- Each implementation of the product must be individually and thoroughly tested for proper operation before being placed into service.

**Failure to follow these instructions can result in death or serious injury.**

<sup>1)</sup> For USA: Additional information, refer to NEMA ICS 1.1 (latest edition), Safety Guidelines for the Application, Installation, and Maintenance of Solid State Control and to NEMA ICS 7.1 (latest edition), Safety Standards for Construction and Guide for Selection, Installation for Construction and Operation of Adjustable-Speed Drive Systems.

### 2.5 Standards and terminology

Technical terms, terminology and the corresponding descriptions in this manual are intended to use the terms or definitions of the pertinent standards.

In the area of drive systems, this includes, but is not limited to, terms such as "safety function", "safe state", "fault", "fault reset", "failure", "error", "error message", "warning", "warning message", etc.

Among others, these standards include:

- IEC 61800 series: "Adjustable speed electrical power drive systems"
- IEC 61158 series: "Industrial communication networks - Fieldbus specifications"
- IEC 61784 series: "Industrial communication networks - Profiles"
- IEC 61508 series: "Functional safety of electrical/electronic/programmable electronic safety-related systems"

Also see the glossary at the end of this manual.



## 3 Basics

# 3

### 3.1 Communication profile

CANopen manages communication between the network devices with object dictionaries and objects. A network device can use process data objects (PDO) and service data objects (SDO) to request the object data from the object dictionary of another device and, if permissible, write back modified values.

The following can be done by accessing the objects of the network devices

- Exchange parameter values
- Start motion functions of individual CAN bus devices
- Request status information

#### 3.1.1 Object dictionary

Each CANopen device manages an object dictionary which contains all objects for communication.

*Index, subindex*

The objects are addressed in the object dictionary via a 16 bit index. One or more 8 bit subindex entries for each object specify individual data fields in the object. Index and subindex are shown in hexadecimal notation with a subscript "h".

*Example*

The following table shows index and subindex entries using the example of the object `software position limit (607Dh)` for specifying the positions of software limit switches.

Index	Subindex	Name	Meaning
607D <sub>h</sub>	00 <sub>h</sub>	-	Number of data fields
607D <sub>h</sub>	01 <sub>h</sub>	min. position limit	Lower limit switch
607D <sub>h</sub>	02 <sub>h</sub>	max. position limit	Upper limit switch

Table 3.1 Example of index and subindex entries

*Object descriptions in the manual*

For CAN programming of a device, the objects of the following object groups are described in detail:

- 1xxx<sub>h</sub> objects: Communication objects in this chapter
- 3xxx<sub>h</sub> objects: Vendor-specific objects required to control the device in chapter 6 "Operation".
- 6xxx<sub>h</sub> objects: Standardized objects of the device profile in chapter 6 "Operation"

*Standardized objects*

Standardized objects allow you to use the same application program for different network devices of the same device type. This requires these objects to be contained in the object dictionary of the network devices. Standardized objects are defined in the DS301 communication profile and the DSP402 device profile.

### 3.1.2 Communication objects

*Overview* The communication objects are standardized with the DS301 CANopen communication profile. The objects can be classified into 4 groups according to their tasks.

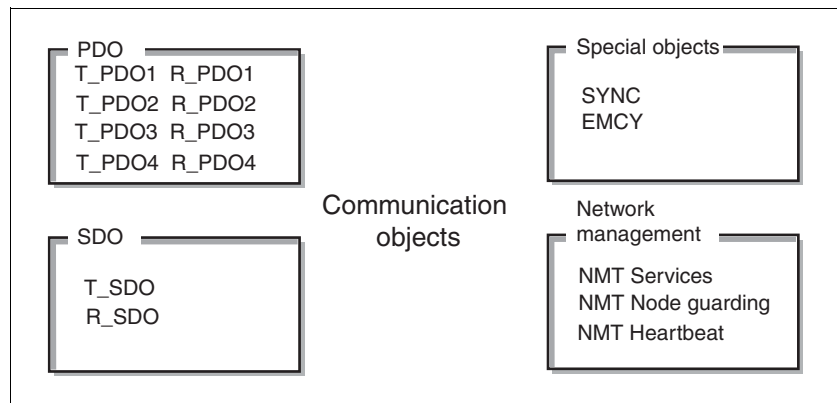


Figure 3.1 Communication objects; the following applies to the perspective of the network device: T\_...: "Transmit", R\_...: "Receive"

- PDOs (process data objects) for real-time transmission of process data
- SDOs (service data object) for read and write access to the object dictionary
- Objects for controlling CAN messages:
  - SYNC object (synchronization object) for synchronization of network devices
  - EMCY object (emergency object), for signaling errors of a device or its peripherals.
- Network management services:
  - NMT services for initialization and network control (NMT: network management)
  - NMT Node Guarding for monitoring the network devices
  - NMT Heartbeat for monitoring the network devices

**CAN message** Data is exchanged via the CAN bus in the form of CAN messages. A CAN message transmits the communication object as well as numerous administration and control data.

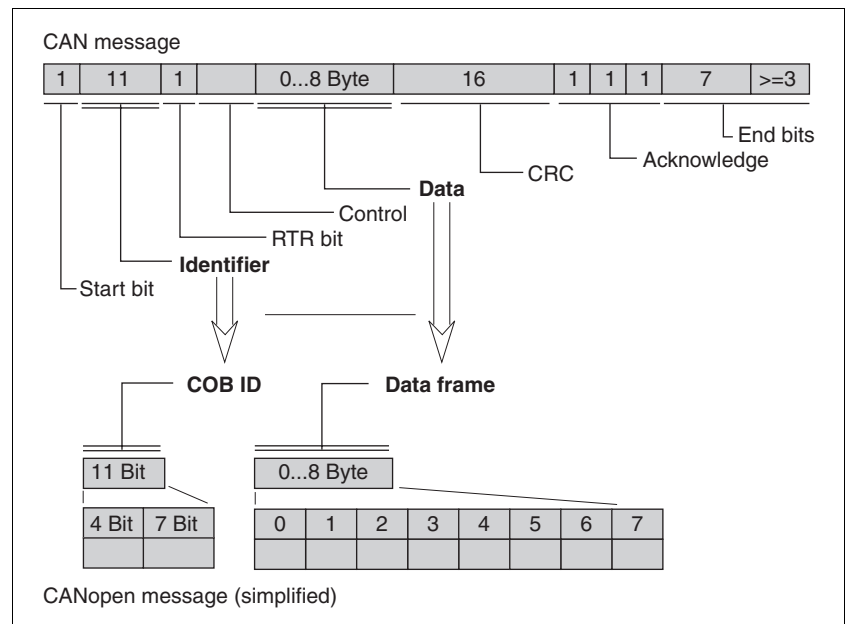


Figure 3.2 CAN message and simplified representation of CANopen message

**CANopen message** For work with CANopen objects and for data exchange, the CAN message can be represented in simplified form because most of the bits are used for error correction. These bits are automatically removed from the receive message by the data link layer of the OSI model, and added to a message before it is transmitted.

The two bit fields "Identifier" and "Data" form the simplified CANopen message. The "Identifier" corresponds to the "COB ID" and the "Data" field to the data frame (maximum length 8 bytes) of a CANopen message.

**COB ID** The COB ID (**C**ommunication **O**bject **I**dentifier) has 2 tasks as far as controlling communication objects is concerned:

- Bus arbitration: Specification of transmission priorities
- Identification of communication objects

An 11 bit COB identifier as per the CAN 3.0A specification is defined for CAN communication; it comprises 2 parts

- Function code, 4 bits
- Node address (node ID), 7 bits.

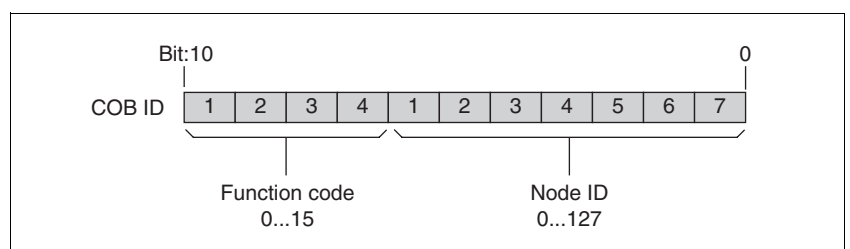


Figure 3.3 COB ID with function code and node address

### COB IDs of the communication objects

The following table shows the COB IDs of all communication objects with the factory settings. The column "Index of object parameters" shows the index of special objects with which the settings of the communication objects can be read or modified via an SDO.

Communication object	Function code	Node address, node ID [1...127]	COB ID decimal (hexadecimal)	Index of object parameters
NMT Start/Stop Service	0 0 0 0	0 0 0 0 0 0 0	0 (0 <sub>h</sub> )	-
SYNC object	0 0 0 1	0 0 0 0 0 0 0	128 (80 <sub>h</sub> )	1005 <sub>h</sub> ...1007 <sub>h</sub>
EMCY object	0 0 0 1	x x x x x x x	128 (80 <sub>h</sub> ) + node ID	1014 <sub>h</sub> , 1015 <sub>h</sub>
T_PDO1 <sup>1)</sup>	0 0 1 1	x x x x x x x	384 (180 <sub>h</sub> ) + node ID	1800 <sub>h</sub>
R_PDO1 <sup>1)</sup>	0 1 0 0	x x x x x x x	512 (200 <sub>h</sub> ) + node ID	1400 <sub>h</sub>
T_PDO2 <sup>1)</sup>	0 1 0 1	x x x x x x x	640 (280 <sub>h</sub> ) + node ID	1801 <sub>h</sub>
R_PDO2 <sup>1)</sup>	0 1 1 0	x x x x x x x	768 (300 <sub>h</sub> ) + node ID	1401 <sub>h</sub>
T_PDO3 <sup>1)</sup>	0 1 1 1	x x x x x x x	896 (380 <sub>h</sub> ) + node ID	1802 <sub>h</sub>
R_PDO3 <sup>1)</sup>	1 0 0 0	x x x x x x x	1024 (400 <sub>h</sub> ) + node ID	1402 <sub>h</sub>
T_PDO4	1 0 0 1	x x x x x x x	1152 (480 <sub>h</sub> ) + node ID	1803 <sub>h</sub>
R_PDO4	1 0 1 0	x x x x x x x	1280 (500 <sub>h</sub> ) + node ID	1403 <sub>h</sub>
T_SDO	1 0 1 1	x x x x x x x	1408 (580 <sub>h</sub> ) + node ID	-
R_SDO	1 1 0 0	x x x x x x x	1536 (600 <sub>h</sub> ) + node ID	-
NMT error control	1 1 1 0	x x x x x x x	1792 (700 <sub>h</sub> ) + node ID	
LMT Services <sup>1)</sup>	1 1 1 1	1 1 0 0 1 0 x	2020 (7E4 <sub>h</sub> ), 2021 (7E5 <sub>h</sub> )	
NMT Identify Service <sup>1)</sup>	1 1 1 1	1 1 0 0 1 1 0	2022 (7E6 <sub>h</sub> )	
DBT Services <sup>1)</sup>	1 1 1 1	1 1 0 0 x x x	2023 (7E7 <sub>h</sub> ), 2024 (7F8 <sub>h</sub> )	
NMT Services <sup>1)</sup>	1 1 1 1	1 1 0 1 0 0 x	2025 (7E9 <sub>h</sub> ), 2026 (7EA <sub>h</sub> )	

1) Not supported by the device

Table 3.2 COB IDs of all communication objects



*COB IDs of PDOs can be changed as required. The assignment pattern for COB IDs only specifies a basic setting.*

#### Function code

The function code classifies the communication objects. Since the bits of the function code in the COB ID are more significant, the function code simultaneously controls the transmission priorities: Objects with a lower function code are transmitted with higher priority. For example, an object with function code "1" is transmitted prior to an object with function code "3" in the case of simultaneous bus access.

#### Node address

Each network device is configured before it can be operated on the network. The device is assigned a unique 7 bit node address (node ID) between 1 (01<sub>h</sub>) and 127 (7F<sub>h</sub>). The device address "0" is reserved for "broadcast" transmissions which are used to send messages to all devices simultaneously.

#### Example

Selection of a COB ID

For a device with the node address 5, the COB ID of the communication object T\_PDO1 is:

$$384 + \text{node ID} = 384 (180_{\text{h}}) + 5 = 389 (185_{\text{h}}).$$

*Data frame* The data frame of the CANopen message can hold up to 8 bytes of data. In addition to the data frame for SDOs and PDOs, special frame types are specified in the CANopen profile:

- Error data frame
- Remote data frame for requesting a message

The data frames contain the respective communication objects.

### 3.1.3 Communication relationships

CANopen uses 3 relationships for communication between network devices:

- Master-slave relationship
- Client-server relationship
- Producer-consumer relationship

*Master-slave relationship* A network master controls the message traffic. A slave only responds when it is addressed by the master.

The master-slave relationship is used with network management objects for a controlled network start and to monitor the connection of devices.

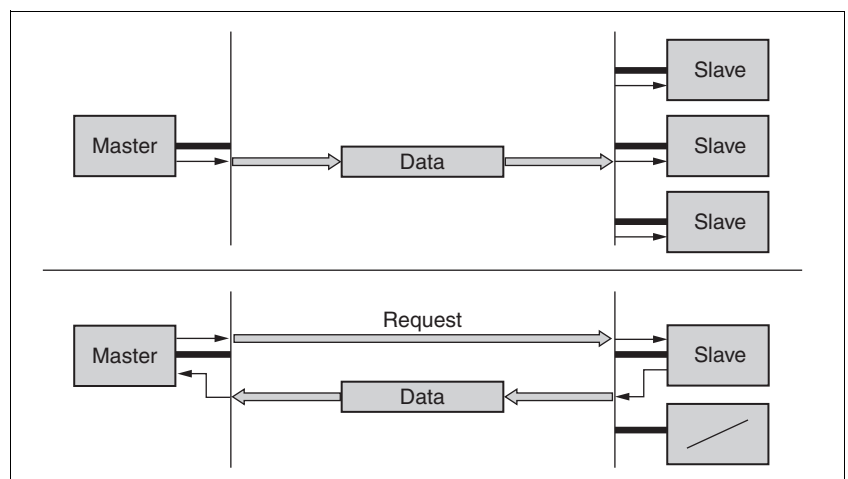


Figure 3.4 Master - slave relationships

Messages can be interchanged with and without confirmation. If the master sends an unconfirmed CAN message, it can be received by a single or by several slaves or by no slave.

To confirm the message, the master requests a message from a specific slave, which then responds with the desired data.

*Client-server relationship*

A client-server relationship is established between 2 devices. The "server" is the device whose object dictionary is used during data exchange. The "client" addresses and starts the exchange of messages and waits for a confirmation from the server.

A client-server relationship with SDOs is used to send configuration data and long messages.

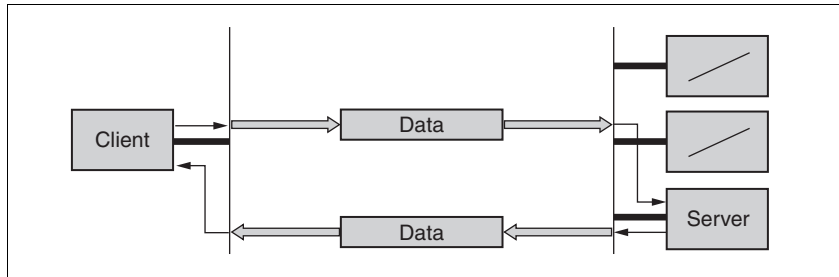


Figure 3.5 Client-server relationship

The client addresses and sends a CAN message to a server. The server evaluates the message and sends the response data as an acknowledgement.

*Producer-consumer relationship*

The producer-consumer relationship is used for exchanging messages with process data, because this relationship enables fast data exchange without administration data.

A "Producer" sends data, a "Consumer" receives data.

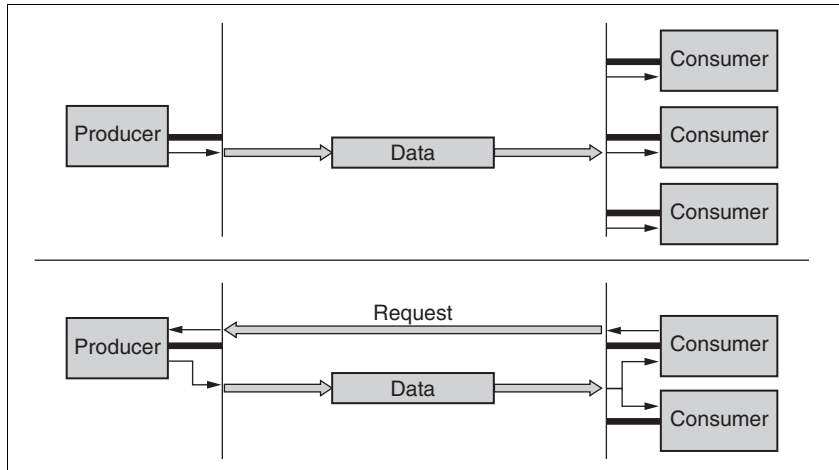


Figure 3.6 Producer-consumer relationships

The producer sends a message that can be received by one or more network devices. The producer does not receive an acknowledgement to the effect that the message was received. The message transmission can be triggered by

- An internal event, e.g. "target position reached"
- The synchronization object SYNC
- A request of a consumer

For details on the function of the producer-consumer relationship and on requesting messages see chapter 3.3 "Process data communication".

## 3.2 Service data communication

### 3.2.1 Overview

Service Data Objects (SDO: **S**ervice **D**ata **O**bject) can be used to access the entries of an object dictionary via index and subindex. The values of the objects can be read and, if permissible, also be changed.

Every network device has at least one server SDO to be able to respond to read and write requests from a different device. A client SDO is only required to request SDO messages from the object dictionary of a different device or to change them in the dictionary.

The T\_SDO of an SDO client is used to send the request for data exchange; the R\_SDO is used to receive. The data frame of an SDO consists of 8 bytes.

SDOs have a higher COB ID than PDOs; therefore, they are transmitted over the CAN bus at a lower priority.

### 3.2.2 SDO data exchange

A service data object (SDO) transmits parameter data between 2 devices. The data exchange conforms to the client-server relationship. The server is the device to whose object dictionary an SDO message refers.

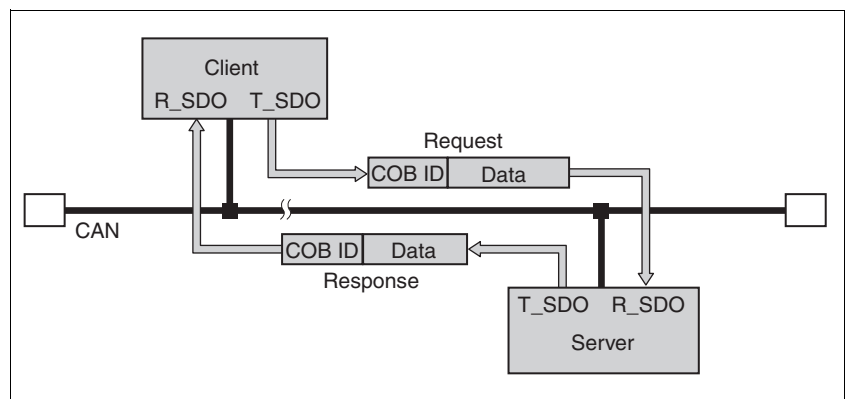


Figure 3.7 SDO message exchange with request and response

#### Message types

Client-server communication is triggered by the client to send parameter values to the server or to get them from the server. In both cases, the client starts the communication with a request and receives a response from the server.

### 3.2.3 SDO message

Put simply, an SDO message consists of the COB ID and the SDO data frame, in which up to 4 bytes of data can be sent. Longer data sequences are distributed over multiple SDO messages with a special protocol.

The device transmits SDOs with a data length of up to 4 bytes. Greater amounts of data such as 8 byte values of the data type "Visible String 8" can be distributed over multiple SDOs and are transmitted successively in 7 byte blocks.

*Example* The following illustration shows an example of an SDO message.

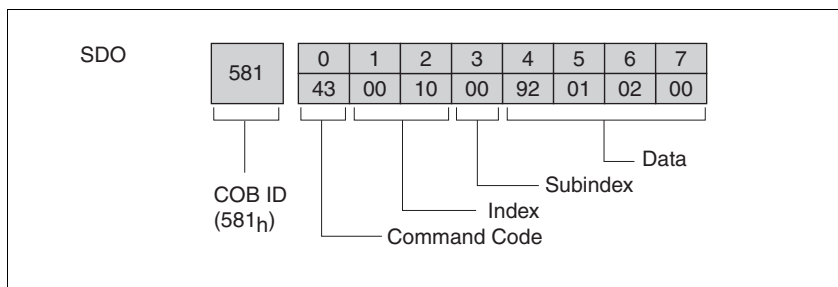


Figure 3.8 SDO message, example

#### COB ID and data frame

R\_SDO and T\_SDO have different COB IDs.

The data frame of an SDO messages consists of:

- Command code (ccd) in which the SDO message type and the data length of the transmitted value are encrypted
- Index and subindex which point to the object whose data is transported with the SDO message
- Data of up to 4 bytes

#### Evaluation of numeric values

Index and data are transmitted left-aligned in Intel format. If the SDO contains numerical values of more than 1 byte in length, the data must be rearranged byte-by-byte before and after a transmission.

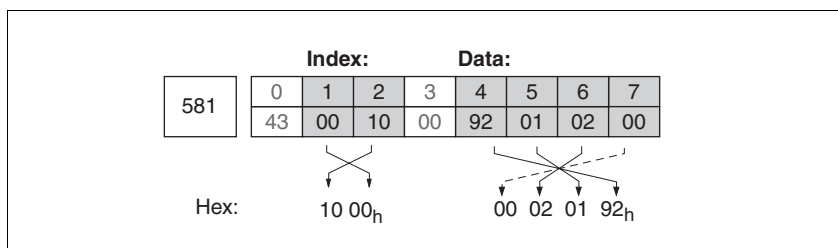


Figure 3.9 Rearranging numeric values greater than 1 byte



### 3.2.4 Reading and writing data

**Writing data** The client starts a write request by sending index, subindex, data length and value.

The server sends a confirmation indicating whether the data was correctly processed. The confirmation contains the same index and subindex, but no data.

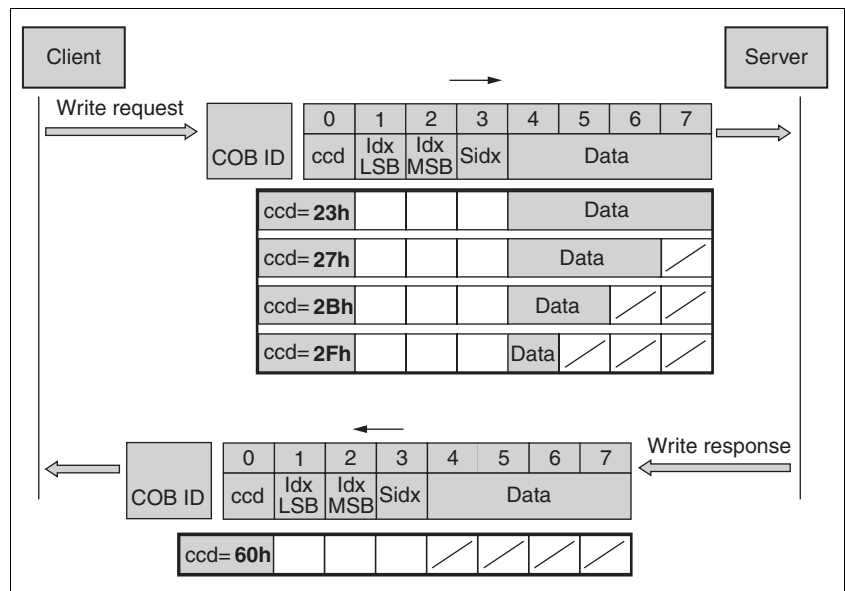


Figure 3.10 Writing parameter values

Unused bytes in the data field are shown with a slash in the graphic. The content of these data fields is not defined.

**ccd coding** The table below shows the command code for writing parameter values. It depends on the message type and the transmitted data length.

Message type	Data length used				
	4 bytes	3 bytes	2 bytes	1 byte	
Write request	23 <sub>h</sub>	27 <sub>h</sub>	2B <sub>h</sub>	2F <sub>h</sub>	Transmitting parameters
Write response	60 <sub>h</sub>	60 <sub>h</sub>	60 <sub>h</sub>	60 <sub>h</sub>	Confirmation
Error response	80 <sub>h</sub>	80 <sub>h</sub>	80 <sub>h</sub>	80 <sub>h</sub>	Error

Table 3.3 Command code for writing parameter values

**Reading data** The client starts a read request by transmitting the index and subindex that point to the object or part of the object whose value it wants to read.

The server confirms the request by sending the desired data. The SDO response contains the same index and subindex. The length of the response data is specified in the command code "ccd".

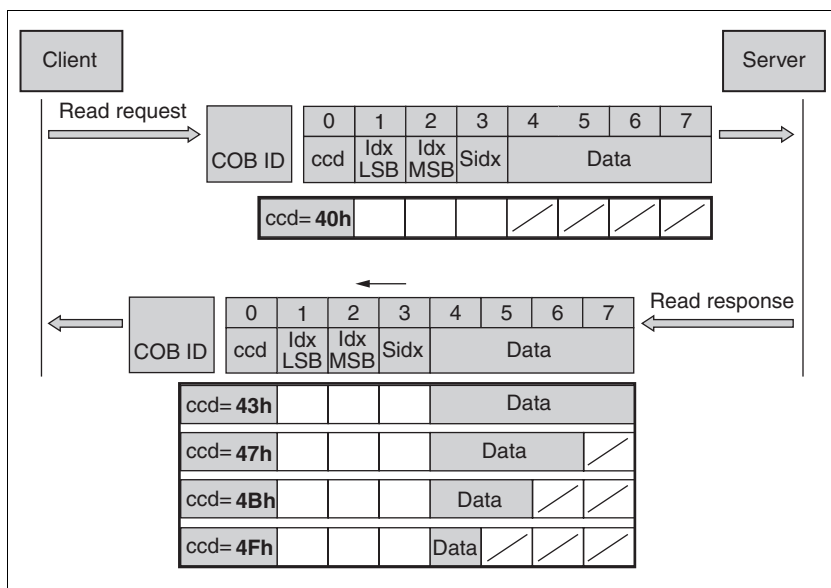


Figure 3.11 Reading a parameter value

Unused bytes in the data field are shown with a slash in the graphic. The content of these data fields is not defined.

**ccd coding** The table below shows the command code for transmitting a read value. It depends on the message type and the transmitted data length.

Message type	Data length used				
	4 bytes	3 bytes	2 bytes	1 byte	
read request	40 <sub>h</sub>	40 <sub>h</sub>	40 <sub>h</sub>	40 <sub>h</sub>	Request read value
Read response	43 <sub>h</sub>	47 <sub>h</sub>	4B <sub>h</sub>	4F <sub>h</sub>	Return read value
Error response	80 <sub>h</sub>	80 <sub>h</sub>	80 <sub>h</sub>	80 <sub>h</sub>	Error

Table 3.4 Command code for transmitting a read value

**Error response** If a message could not be evaluated, the server sends an error message. See chapter 7.4.3 "SDO error message ABORT" for details on the evaluation of the error message.

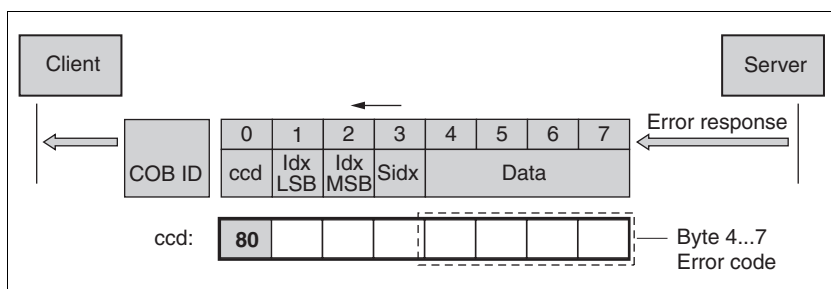


Figure 3.12 Response with error message (error response)

### 3.2.5 Reading data longer than 4 bytes

If values of more than 4 bytes are to be transmitted with an SDO message, the message must be divided into several frames. Each frame consists of 2 parts:

- Request by the SDO client,
- Confirmation by the SDO server.

The request by the SDO client contains the command code "ccd" with the toggle bit and a data segment. The confirmation frame also contains a toggle bit in the segment "ccd". In the first frame, the toggle bit has the value "0", in the subsequent frames, it toggles between 1 and 0.

*Reading data* The client starts a read request by transmitting the index and subindex that point to the object or the object value whose value it wants to read.

The server confirms the request by transmitting index, subindex, data length and the first 4 bytes of the requested data. The command code specifies that data of more than 4 bytes are transmitted. The command code of the read response from the server to the first message is 41<sub>h</sub>.

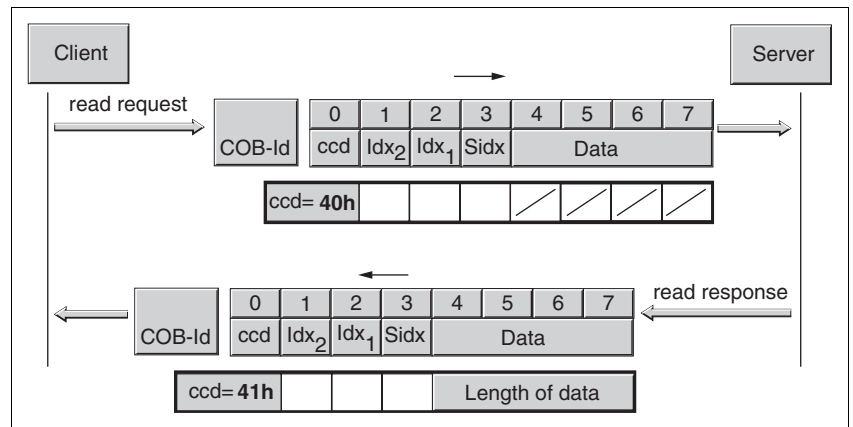


Figure 3.13 Transmitting the first message

In the next frames, the remaining data is requested and transmitted in packets of 7 bytes from the server. Refer to the DS301 of the CiA for additional information on this procedure.

### 3.3 Process data communication

#### 3.3.1 Overview

Process data objects (PDO: **P**rocess **D**ata **O**bject) are used for real-time data exchange of process data such as actual and reference or operating state of the device. Transmission is very fast because the data is sent without additional administration data and a response from the recipient is not required.

The flexible data length of a PDO message also increases the data throughput. A PDO message can transmit up to 8 bytes of data. If only 2 bytes are assigned, only 2 data bytes are sent.

The length of a PDO message and the assignment of the data fields are specified by PDO mapping. For more information see chapter 3.3.4 "PDO mapping".

PDO messages can be exchanged between devices that generate or process process data.

#### 3.3.2 PDO data exchange

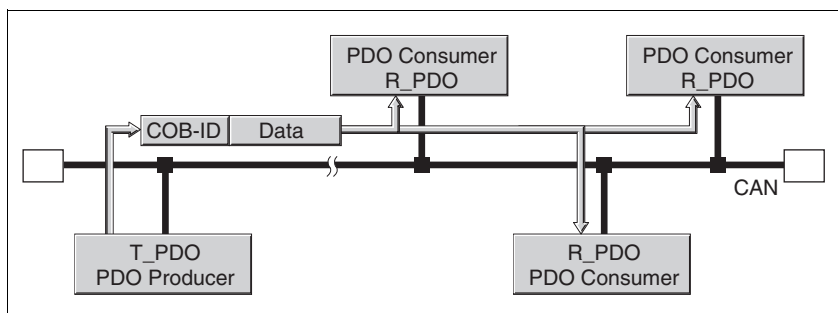


Figure 3.14 PDO data exchange

Data exchange with PDOs follows to the producer-consumer relationship and can be triggered in 3 ways

- Synchronized
- Event-driven, asynchronous
- On request of a consumer, asynchronous

The SYNC object controls synchronized data processing. Synchronous PDO messages are transmitted immediately like the standard PDO messages, but are only evaluated on the next SYNC. For example, several drives can be started simultaneously via synchronized data exchange.

The device immediately evaluates PDO messages that are called on request or in an event-driven way.

The transmission type can be specified separately for each PDO with subindex 02<sub>h</sub> (transmission type) of the PDO communication parameter. The objects are listed in Table 3.5.

### 3.3.3 PDO message

*T\_PDO, R\_PDO* One PDO each is available for sending and receiving a PDO message:

- T\_PDO to transmit PDO messages (T: Transmit),
- R\_PDO to receive PDO messages (R: Receive).



*The following settings for PDOs correspond to the defaults for the device, unless otherwise specified. They can be read and set via objects of the communication profile.*

The device uses 8 PDOs, 4 receive PDOs and 4 transmit PDOs. By default, all PDOs are evaluated or transmitted in an event-driven way.

*PDO settings* The PDO settings can be read and changed with 8 communication objects:

Object	Meaning
1st receive PDO parameter (1400 <sub>h</sub> )	Settings for R_PDO1
2nd receive PDO parameter (1401 <sub>h</sub> )	Settings for R_PDO2
3rd receive PDO parameter (1402 <sub>h</sub> )	Settings for R_PDO3
4th receive PDO parameter (1403 <sub>h</sub> )	Settings for R_PDO4
1st transmit PDO parameter (1800 <sub>h</sub> )	Settings for T_PDO1
2nd transmit PDO parameter (1801 <sub>h</sub> )	Settings for T_PDO2
3rd transmit PDO parameter (1802 <sub>h</sub> )	Settings for T_PDO3
4th transmit PDO parameter (1803 <sub>h</sub> )	Settings for T_PDO4

Table 3.5 Communication objects for PDO

*Activating PDOs* With the default PDO settings, R\_PDO1 and T\_PDO1 are activated. The other PDOs must be activated first.

A PDO is activated with bit 31 (valid bit) in subindex 01<sub>h</sub> of the respective communication object:

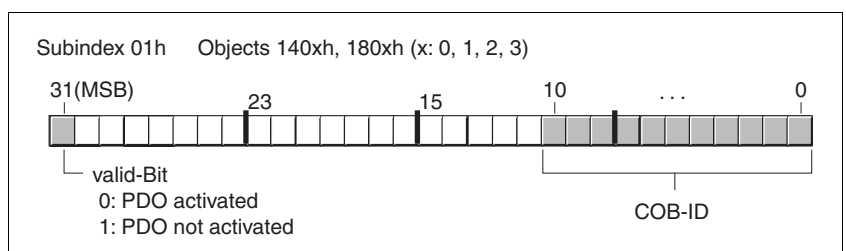


Figure 3.15 Activating PDOs via subindex 01<sub>h</sub>, bit 31

*Example* **Setting for R\_PDO3 in object 1402<sub>h</sub>**

- Subindex 01<sub>h</sub> = 8000 04xx<sub>h</sub>: R\_PDO3 not activated
- Subindex 01<sub>h</sub> = 0000 04xx<sub>h</sub>: R\_PDO3 activated.

Values for "x" in the example depend on the COB ID setting.

- PDO time intervals* The time intervals "inhibit time" and "event timer" can be set for each transmit PDO.
- The time interval "inhibit time" can be used to reduce the load on the CAN bus, which can be the result of continuous transmission of T\_PDOs. If an inhibit time not equal to zero is entered, a transmitted PDO will only be re-transmitted after the inhibit time has elapsed. The time is set with subindex 03<sub>h</sub>.
  - The time interval "event timer" cyclically triggers an event message. After the time intervals has elapsed, the device transmits the event-controlled T\_PDO. The time is set with subindex 05<sub>h</sub>.
- Receive PDOs* The objects for R\_PDO1, R\_PDO2, R\_PDO3 and R\_PDO4 are preset.

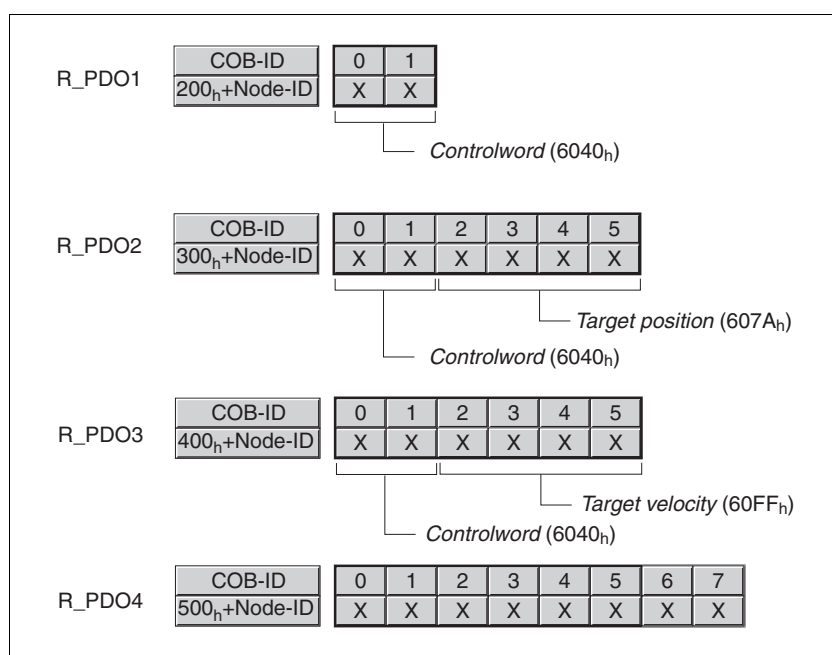


Figure 3.16 Receive PDOs

- R\_PDO1* R\_PDO1 contains the control word, object `controlword (6040h)`, of the state machine which can be used to set the operating state of the device.
- R\_PDO1 is evaluated asynchronously, i.e. it is event-driven. R\_PDO1 is preset.
- R\_PDO2* With R\_PDO2, the control word and the target position of a motion command, object `target position (607Ah)`, are received for a movement in the operating mode "Profile Position".
- R\_PDO2 is evaluated asynchronously, i.e. it is event-driven. R\_PDO2 is preset.
- For details on the SYNC object see chapter 3.4 "Synchronization".
- R\_PDO3* R\_PDO3 contains the control word and the target velocity, object `Target velocity (60FFh)`, for the operating mode "Profile Velocity".
- R\_PDO3 is evaluated asynchronously, i.e. it is event-driven. R\_PDO3 is preset.

**R\_PDO4** R\_PDO4 is used to transmit vendor-specific object values. By default, R\_PDO4 is empty.

R\_PDO4 is evaluated asynchronously, i.e. it is event-driven.

All R\_PDOs can be used to map various vendor-specific objects by means of PDO mapping.

**Transmit PDOs** The objects for T\_PDO1, T\_PDO2 and T\_PDO3 are permanently set. The object that is mapped to PDO T\_PDO4 can be modified by PDO mapping.

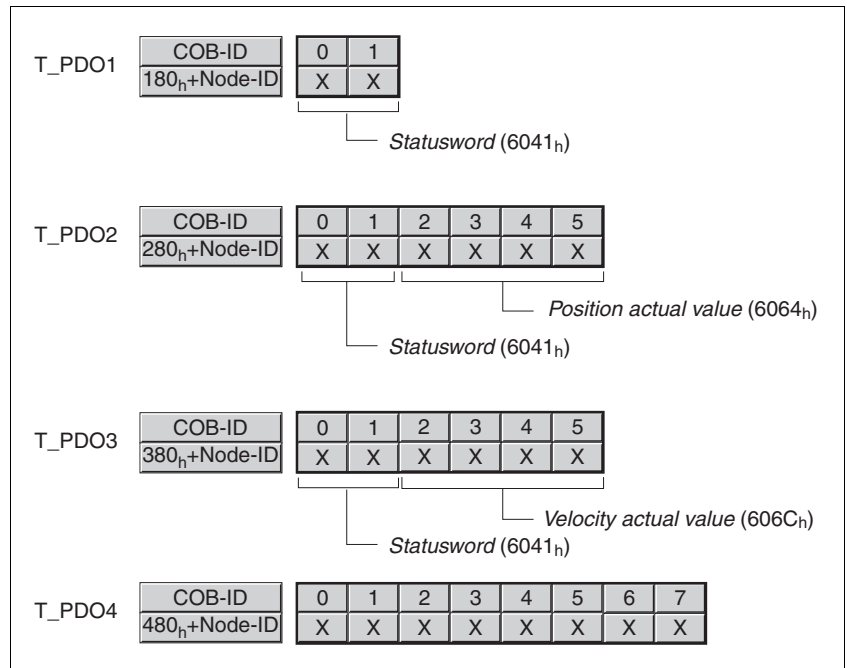


Figure 3.17 Transmit PDOs

**T\_PDO1** T\_PDO1 contains the status word, object `statusword` (6041<sub>h</sub>), of the state machine.

T\_PDO1 is transmitted asynchronously and in an event-driven way whenever the status information changes.

**T\_PDO2** T\_PDO2 contains the status word and the current position of the motor, object `Position actual value` (6064<sub>h</sub>), to monitor movements in the operating mode "Profile Position".

T\_PDO2 is transmitted after receipt of a SYNC object and in an event-driven way.

**T\_PDO3** T\_PDO3 contains the status word and the current velocity, object `Velocity actual value` (606C<sub>h</sub>), for monitoring the velocity profile in the operating mode "Profile Velocity".

T\_PDO3 is transmitted asynchronously and in an event-driven way whenever the status information changes.

**T\_PDO4** Vendor-specific object values (for monitoring) are transmitted with T\_PDO4. By default, T\_PDO4 is empty.

T\_PDO4 is transmitted asynchronously and in an event-driven way whenever the data changes. The parameter `CANpdo4Event` is used to specify the objects which are to trigger an event. With the default setting of the parameter, all mapped objects trigger an event.

All T\_PDOs can be used to map various vendor-specific objects via PDO mapping.

Parameter name HMI menu HMI name	Description	Unit Minimum value Factory setting Maximum value	Data type R/W Persistent Expert	Parameter address via fieldbus
CANpdo1Event	PDO 1 event mask Changes of values in the object trigger an event: Bit 0: First PDO object Bit 1: Second PDO object Bit 2: Third PDO object Bit 3: Fourth PDO object Changed settings become active immediately.	- 0 1 15	UINT16 UINT16 UINT16 UINT16 R/W - -	CANopen 3041:B <sub>h</sub> Modbus 16662 Profibus 16662 CIP 165.1.11
CANpdo2Event	PDO 2 event mask Changes of values in the object trigger an event: Bit 0: First PDO object Bit 1: Second PDO object Bit 2: Third PDO object Bit 3: Fourth PDO object Changed settings become active immediately.	- 0 1 15	UINT16 UINT16 UINT16 UINT16 R/W - -	CANopen 3041:C <sub>h</sub> Modbus 16664 Profibus 16664 CIP 165.1.12
CANpdo3Event	PDO 3 event mask Changes of values in the object trigger an event: Bit 0: First PDO object Bit 1: Second PDO object Bit 2: Third PDO object Bit 3: Fourth PDO object Changed settings become active immediately.	- 0 1 15	UINT16 UINT16 UINT16 UINT16 R/W - -	CANopen 3041:D <sub>h</sub> Modbus 16666 Profibus 16666 CIP 165.1.13
CANpdo4Event	PDO 4 event mask Changes of values in the object trigger an event: Bit 0: First PDO object Bit 1: Second PDO object Bit 2: Third PDO object Bit 3: Fourth PDO object Changed settings become active immediately.	- 0 15 15	UINT16 UINT16 UINT16 UINT16 R/W - -	CANopen 3041:E <sub>h</sub> Modbus 16668 Profibus 16668 CIP 165.1.14



### 3.3.4 PDO mapping

Up to 8 bytes of data from different areas of the object dictionary can be transmitted with a PDO message. Mapping of data to a PDO message is referred to as PDO mapping.

Chapter 8.3 "Assignment object group 3000h" contains a list of vendor-specific objects that are available for PDO mapping.

The picture below shows the data exchange between PDOs and object dictionary on the basis of two examples of objects in T\_PDO4 and R\_PDO4 of the PDOs.

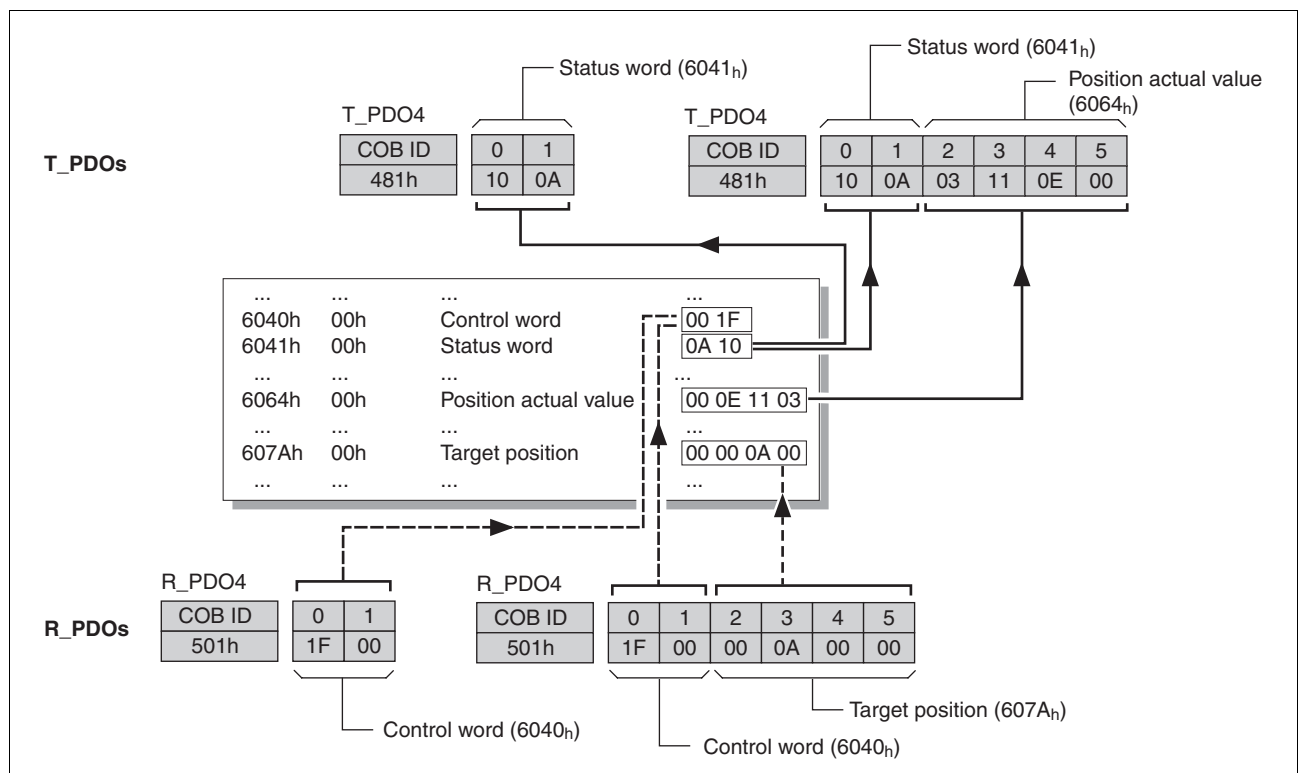


Figure 3.18 PDO mapping, in this case for a device with node address 1

#### Dynamic PDO mapping

The device uses dynamic PDO mapping. Dynamic PDO mapping means that objects can be mapped to the corresponding PDO using adjustable settings.

The settings for PDO mapping are defined in an assigned communication object for each PDO.

Object	PDO mapping for	Type
1st receive PDO mapping (1600 <sub>h</sub> )	R_PDO1	Dynamic
2nd receive PDO mapping (1601 <sub>h</sub> )	R_PDO2	Dynamic
3rd receive PDO mapping (1602 <sub>h</sub> )	R_PDO3	Dynamic
4th receive PDO mapping (1603 <sub>h</sub> )	R_PDO4	Dynamic
1st transmit PDO mapping (1A00 <sub>h</sub> )	T_PDO1	Dynamic
2nd transmit PDO mapping (1A01 <sub>h</sub> )	T_PDO2	Dynamic
3rd transmit PDO mapping (1A02 <sub>h</sub> )	T_PDO3	Dynamic
4th transmit PDO mapping (1A03 <sub>h</sub> )	T_PDO4	Dynamic

*Structure of entries* Up to 8 bytes of 8 different objects can be mapped in a PDO. Each communication object for setting the PDO mapping provides 4 subindex entries. A subindex entry contains 3 pieces of information on the object: the index, the subindex and the number of bits that the object uses in the PDO.

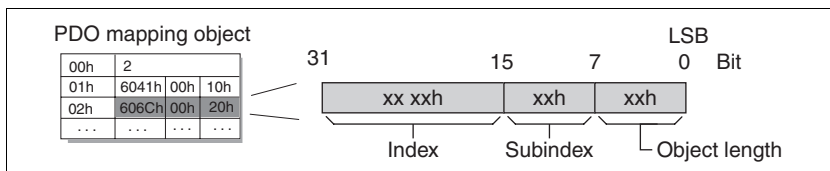


Figure 3.19 Structure of entries for PDO mapping

Subindex 00<sub>h</sub> of the communication object contains the number of valid subindex entries.

Object length	Bit value
10 <sub>h</sub>	16 bits
20 <sub>h</sub>	32 bits

#### *PDO mapping objects*

Index	Subindex	Object	PDO	Data type	Parameter name	Takes effect
3006 <sub>h</sub>	1	Acceleration and deceleration of the motion profile for velocity	R_PDO	UINT16	RAMP_v_sym	Next movement
3008 <sub>h</sub>	1	Physical status of the digital inputs and outputs	T_PDO	UINT16	_IO_act	-
3008 <sub>h</sub>	15	Status of digital inputs	T_PDO	UINT16	_IO_DI_act	-
3008 <sub>h</sub>	16	Status of digital outputs	T_PDO	UINT16	_IO_DQ_act	-
3008 <sub>h</sub>	17	Setting the digital outputs directly	R_PDO	UINT16	IO_DQ_set	-
300A <sub>h</sub>	6	Capture input 1 captured position	T_PDO	UINT32	_Cap1Pos	-
300A <sub>h</sub>	8	Capture input 1 event counter	T_PDO	UINT16	_Cap1Count	-
301B <sub>h</sub>	9	Activation of operating mode Jog	R_PDO	UINT16	JOGactivate	Immediately
301C <sub>h</sub>	4	Action word	T_PDO	UINT16	_actionStatus	-
301E <sub>h</sub>	3	Total motor current	T_PDO	INT16	_I_act	-
301F <sub>h</sub>	2	Actual position of profile generator	T_PDO	INT32	_RAMP_p_act	-

Index	Subindex	Object	PDO	Data type	Parameter name	Takes effect
6040 <sub>h</sub>	0	Control word	R_PDO	UINT16	DCOMcontrol	Immediately
6041 <sub>h</sub>	0	Status word	T_PDO	UINT16	_DCOMstatus	-
6060 <sub>h</sub>	0	Modes of Operation	R_PDO	INT8	DCOMopmode	Immediately
6061 <sub>h</sub>	0	Modes of Operation Display	T_PDO	INT8	_DCOMopmd_act	-
6063 <sub>h</sub>	0	Position actual Value	T_PDO	INT32	_p_actInt	-
6064 <sub>h</sub>	0	Position actual Value	T_PDO	INT32	_p_act	-
606C <sub>h</sub>	0	Velocity actual value	T_PDO	INT32	_v_act	-
6071 <sub>h</sub>	0	Target Torque	R_PDO	INT16	PTtq_target	Immediately
6077 <sub>h</sub>	0	Torque actual Value	T_PDO	INT16	_tq_act	-
607A <sub>h</sub>	0	Target position	R_PDO	INT32	PPp_target	Immediately
6081 <sub>h</sub>	0	Profile velocity	R_PDO	UINT32	PPv_target	Next movement
6083 <sub>h</sub>	0	Profile acceleration	R_PDO	UINT32	RAMP_v_acc	Next movement
6084 <sub>h</sub>	0	Profile deceleration	R_PDO	UINT32	RAMP_v_dec	Next movement
6087	0	Torque slope	R_PDO	UINT32	RAMP_tq_slope	Immediately
60FF <sub>h</sub>	0	Target velocity	R_PDO	INT32	PVv_target	Immediately

### 3.4 Synchronization

The synchronization object SYNC controls the synchronous exchange of messages between network devices for purposes such as the simultaneous start of multiple drives.

The data exchange conforms to the producer-consumer relationship. The SYNC object is transmitted to all devices by a network device and can be evaluated by all devices that support synchronous PDOs.

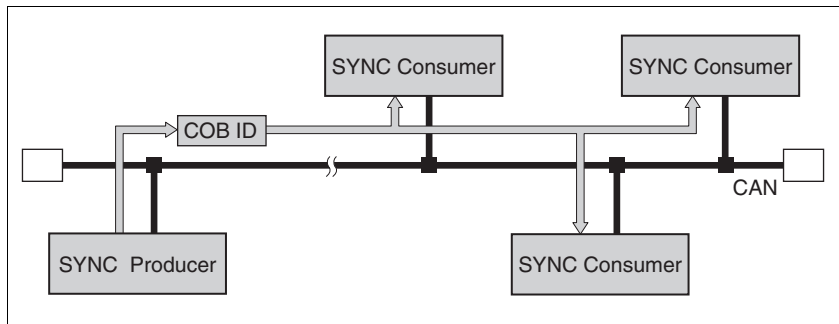


Figure 3.20 SYNC message

#### Time values for synchronization

Two time values define the behavior of synchronous data transmission:

- The cycle time specifies the time intervals between 2 SYNC messages. It is set with the object `Communication cycle period(1006h)`.
- The synchronous time window specifies the time span during which the synchronous PDO messages must be received and transmitted. The time window is defined with the object `Synchronous window length (1007h)`.

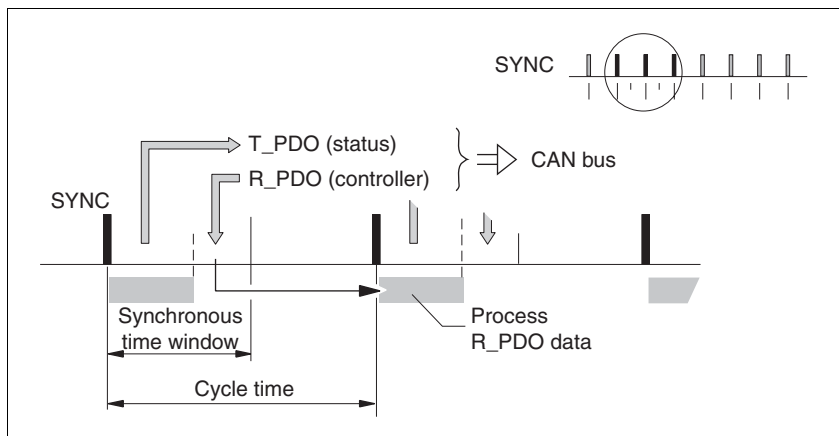


Figure 3.21 Synchronization times

#### Synchronous data transmission

From the perspective of a SYNC recipient, in one time window the status data is transmitted first in a T\_PDO, then new control data is received via an R\_PDO. However, the control data is only processed when the next SYNC message is received. The SYNC object itself does not transmit data.

*Cyclic and acyclic data transmission* Synchronous exchange of messages can be cyclic or acyclic.

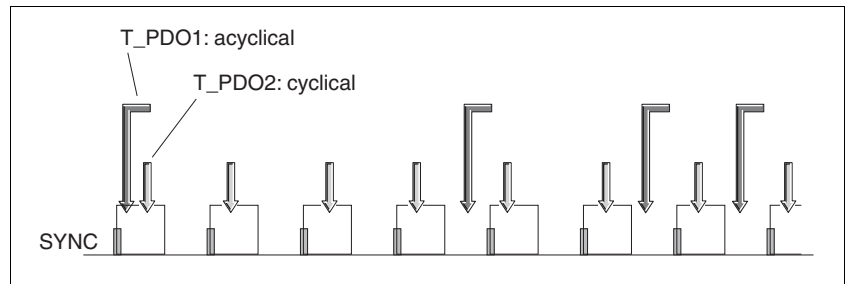


Figure 3.22 Cyclic and acyclic transmission

In the case of cyclic transmission, PDO messages are exchanged continuously in a specified cycle, e.g. with every SYNC message.

If a synchronous PDO message is transmitted acyclically, it can be transmitted or received at any time; however, it will not be valid until the next SYNC message.

Cyclic or acyclic behavior of a PDO is specified in the subindex transmission type (02<sub>h</sub>) of the corresponding PDO parameter, e.g. in the object 1st receive PDO parameter (1400<sub>h</sub>:02<sub>h</sub>) for R\_PDO1.

*COB ID, SYNC object* For fast transmission, the SYNC object is transmitted unconfirmed and with high priority.

The COB ID of the SYNC object is set to the value 128 (80<sub>h</sub>) by default. The value can be changed after initialization of the network with the object COB-ID SYNC Message (1005<sub>h</sub>) .

*"Start" PDO* With the default settings of the PDOs, R\_PDO2/T\_PDO2 and R\_PDO3/T\_PDO3 are received and transmitted synchronously. Both PDOs are used for starting and monitoring operating modes. The synchronization allows an operating mode to be started simultaneously on multiple devices so that, for example, the feed of a portal drive with several motors can be synchronized.

### 3.5 Emergency service

The Emergency Service signals internal device errors via the CAN bus. The error is transmitted to all network devices with an EMCY object according to the Consumer-Producer relationship.

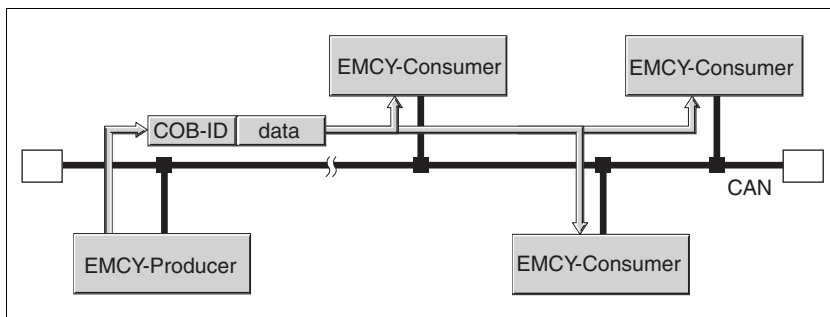


Figure 3.23 Error message via EMCY objects

#### *Boot-up message*

The communication profile DS301, version 3.0, defines an additional task for the EMCY object: sending a boot-up message. A boot-up message informs all network devices that the device that transmitted the message is ready for operation in the CAN network.

The boot-up message is transmitted with the COB ID 700h + node ID and one data byte (00h).

#### 3.5.1 Error evaluation and handling

##### *EMCY message*

If an internal device error occurs, the device switches to the operating state Fault as per the CANopen state machine. At the same time, it transmits an EMCY message with error register and error code.

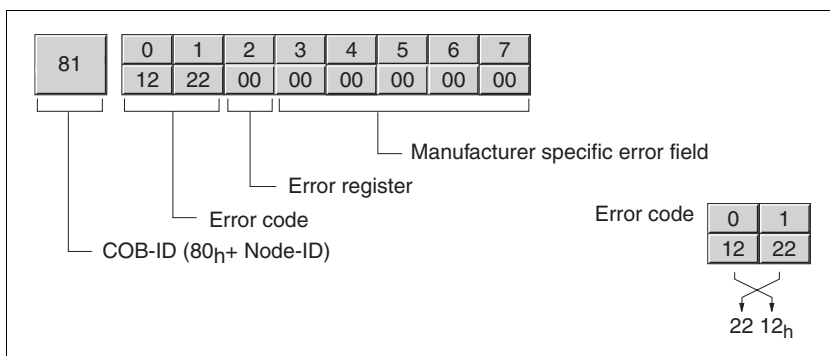


Figure 3.24 EMCY message

Bytes 0, 1 - Error code, value is also saved in the object `Error code` (603F<sub>h</sub>)

Byte 2 - Error register, value is also saved in the object `Error register` (1001<sub>h</sub>), see 7.4.1 "Error register".

Bytes 3, 4 - Vendor-specific error code of the mapped object

Bytes 5, 6 - Index of the mapped object

Byte 7 - Subindex of the mapped object

<i>COB ID</i>	<p>The COB ID for each device on the network supporting an EMCY object is determined on the basis of the node address:</p> <p>COB ID = Function code EMCY object (80<sub>h</sub>) + node ID</p> <p>The function code of the COB ID can be changed with the object COB-ID emergency(1014<sub>h</sub>).</p>
<i>Error register and error code</i>	<p>The error register contains bit-coded information on the error. Bit 0 remains set as long as an error is active. The remaining bits identify the error type. The precise cause of error can be determined on the basis of the error code. The error code is transmitted in Intel format as a 2 byte value; the bytes must be reversed for evaluation.</p> <p>A list of all error messages and error responses by the device as well as remedies can be found in chapter 7 "Diagnostics and troubleshooting".</p>
<i>Error memory</i>	<p>The device saves the error register in the object Error register (1001<sub>h</sub>) and the last error that occurred in the object ErrorError code (603F<sub>h</sub>). The last 20 error messages are stored in the object FLT_err_num (303C:1<sub>h</sub>) in the order in which the errors occurred. FLT_MemReset (303B:5<sub>h</sub>) resets the read pointer of the error memory to the oldest error.</p>

### 3.6 Network management services

Network management (NMT) is part of the CANopen communication profile; it is used to initialize the network and the network devices and to start, stop and monitor the network devices during operation on the network.

NMT services are executed in a master-slave relationship. The NMT master addresses individual NMT slaves via their node address. A message with node address "0" is broadcast to all NMT slaves simultaneously.

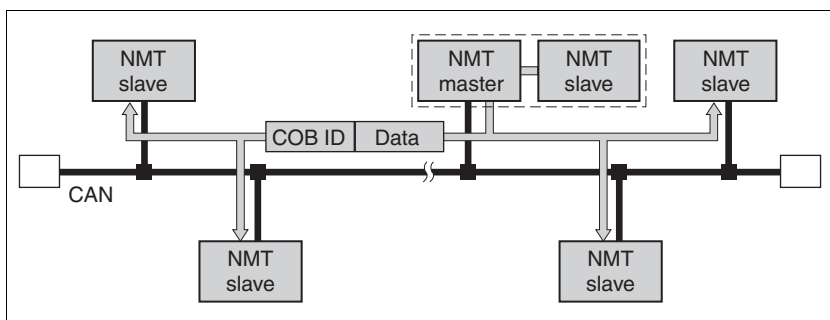


Figure 3.25 NMT services via the master-slave relationship

The device can only take on the function of an NMT slave.

#### *NMT services*

NMT services can be divided into 2 groups:

- Services for device control, to initialize devices for CANopen communication and to control the behavior of devices during operation on the network
- Services:for connection monitoring



### 3.6.1 NMT services for device control

*NMT state machine* The NMT state machine describes the initialization and states of an NMT slave during operation on the network.

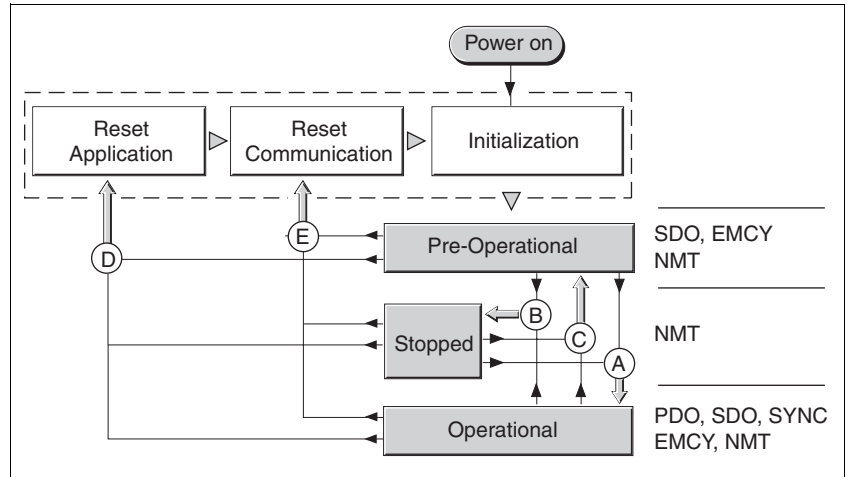


Figure 3.26 NMT state machine and available communication objects

To the right, the graphic shows all communication objects that can be used in the specific network state.

*Initialization* An NMT slave automatically runs through an initialization phase after the supply voltage is switched on (power on) to prepare it for CAN bus operation. On completion of the initialization, the slave switches to the operating state "Pre Operational" and sends a boot-up message. From now on, an NMT master can control the operational behavior of an NMT slave on the network via 5 NMT services, represented in the above illustration by the letters A to E.

NMT service	Transition	Meaning
Start remote node (Start network node)	A	Transition to operating state "Operational" Start normal operation on the network
Stop remote node (Stop network node)	B	Transition to operating state "Stopped" Stops communication of the network device on the network. If connection monitoring is active, it remains on. If the power stage is active (operating state "Operation Enabled" or "Quick Stop"), an error of error class 2 is triggered. The motor is stopped and the power stage deactivated.
Enter Pre-Operational (Transition to "Pre-Operational")	C	Transition to operating state "Pre-Operational" All communication objects except for PDOs can be used.  The operating state "Pre-Operational" can be used for configuration via SDOs: - PDO mapping - Start of synchronization - Start of connection monitoring
Reset node (Reset node)	D	Transition to operating state "Reset application" Load stored data of the device profiles and automatically switch via operating state "Reset communication" to "Pre-Operational".
Reset communication (Reset communication data)	E	Transition to operating state "Reset communication" Load stored data of the communication profile and automatically to operating state "Pre-Operational". If the power stage is active (operating state "Operation Enabled" or "Quick Stop"), an error of error class 2 is triggered. The motor is stopped and the power stage deactivated.

*Persistent data memory* When the supply voltage is switched on (power on), the device loads the saved object data from the non-volatile EEPROM for persistent data to the RAM.

*NMT message* The NMT services for device control are transmitted as unconfirmed messages with the COB ID = 0. By default, they have the highest priority on the CAN bus.

The data frame of the NMT device service consists of 2 bytes.

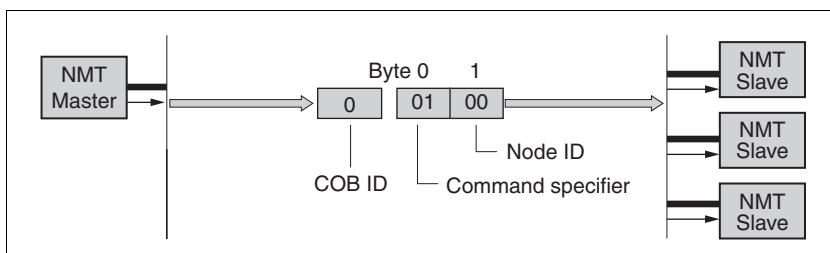


Figure 3.27 NMT message

The first byte, the "Command specifier", indicates the NMT service used.

Command Specifier	NMT service	Transition
1 (01 <sub>h</sub> )	Start remote node	A
2 (02 <sub>h</sub> )	Stop remote node	B
128 (80 <sub>h</sub> )	Enter Pre-Operational	C
129 (81 <sub>h</sub> )	Reset node	D
130 (82 <sub>h</sub> )	Reset communication	E

The second byte addresses the recipient of an NMT message with a node address between 1 and 127 (7F<sub>h</sub>). A message with node address "0" is broadcast to all NMT slaves.

### 3.6.2 NMT services for connection monitoring

Connection monitoring monitors the communication status of network devices.

3 NMT services for connection monitoring are available:

- "Node guarding" for monitoring the connection of an NMT slave
- "Life guarding" for monitoring the connection of an NMT master
- "Heartbeat" for unconfirmed connection messages from network devices.

#### 3.6.2.1 Node guarding/life guarding

**COB ID** Communication object NMT error control ( $700_h + \text{node-Id}$ ) is used for connection monitoring. The COB ID for each NMT slave is determined on the basis of the node address:

COB ID = function code NMT error control ( $700_h$ ) + node-Id.

**Structure of the NMT message** After a request from the NMT master, the NMT slave responds with one data byte.

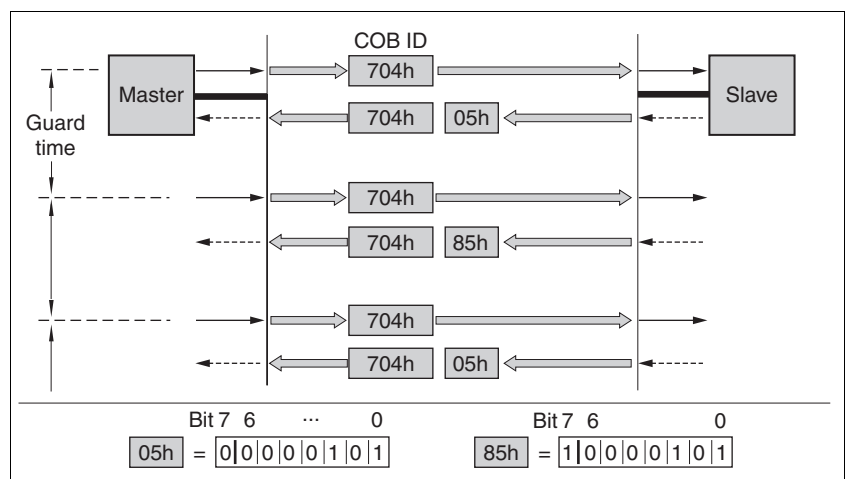


Figure 3.28 Acknowledgement of the NMT slave

Bits 0 to 6 identify the NMT state of the slave:

- 4 ( $04_h$ ): "Stopped"
- 5 ( $05_h$ ): "Operational"
- 127 ( $7F_h$ ): "Pre-Operational"

After each "guard time" interval, bit 7 switches toggles between "0" and "1", so the NMT master can detect and ignore a second response within the "guard time" interval. The first request when connection monitoring is started begins with bit 7 = 0.

Connection monitoring must not be active during the initialization phase of a device. The status of bit 7 is reset as soon as the device runs through the NMT state "Reset communication".

Connection monitoring remains active in the NMT state "Stopped".

*Configuration* Node guarding/life guarding is configured via:

- Guard time ( $100C_h$ )
- Life time factor ( $100D_h$ )

*Connection error* The NMT master signals a connection error to the master program if:

- the slave does not respond within the "guard time" period
- the NMT state of the slave has changed without a request by the NMT master.

Figure 3.29 shows an error message after the end of the third cycle because of a missing response from an NMT slave.

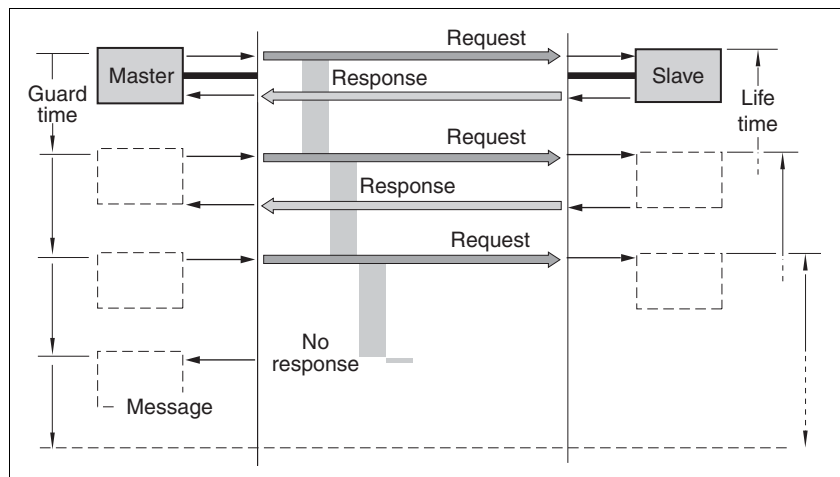


Figure 3.29 "Node Guarding" and "Life Guarding" with time intervals

### 3.6.2.2 Heartbeat

The optional Heartbeat protocol replaces the node guarding/life guarding protocol. It is recommended for new device versions.

A heartbeat producer transmits a heartbeat message cyclically at the frequency defined in the object `Producer heartbeat time` ( $1017_h$ ). One or several consumers can receive this message. `Producer heartbeat time` ( $1016_h$ ) = 0 deactivates heartbeat monitoring.

The relationship between producer and consumer can be configured with objects. If a consumer does not receive a signal within the period of time set with `Consumer heartbeat time` ( $1016_h$ ), it generates an error message (heartbeat event). `Consumer heartbeat time` ( $1016_h$ ) = 0 deactivates monitoring by a consumer.

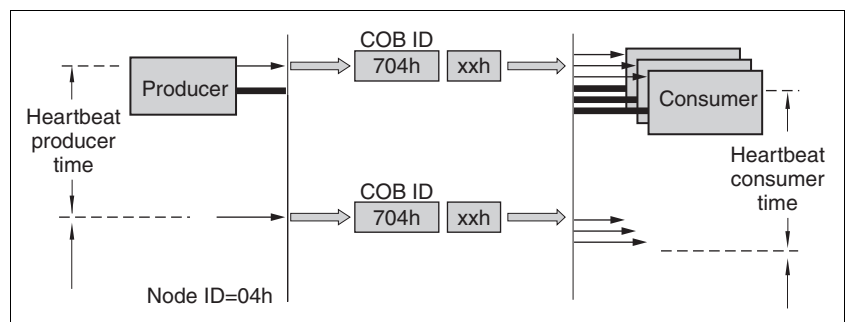


Figure 3.30 "Heartbeat" monitoring

Data byte for NMT state evaluation of the "Heartbeat" producer:

- 0 ( $00_h$ ): "Boot-Up"
- 4 ( $04_h$ ): "Stopped"
- 5 ( $05_h$ ): "Operational"
- 127 ( $7F_h$ ): "Pre-Operational"

#### *Time intervals*

The time intervals are set in increments of 1 ms steps; the values for the consumer must not be less than the values for the producer. Whenever the "Heartbeat" message is received, the time interval of the producer is restarted.

#### *Start of monitoring*

"Heartbeat" monitoring starts as soon as the time interval of the producer is greater than zero. If "Heartbeat" monitoring is already active during the NMT state transition to "Pre-Operational", "Heartbeat" monitoring starts with sending of the boot-up message. The boot-up message is a Heartbeat message with one data byte  $00_h$ .

Devices can monitor each other via "Heartbeat" messages. They assume the function of consumer and producer at the same time.



## 4 Installation

# 4

### ⚠ WARNING

#### SIGNAL AND DEVICE INTERFERENCE

Signal interference can cause unexpected responses of device.

- Install the wiring in accordance with the EMC requirements.
- Verify compliance with the EMC requirements.

**Failure to follow these instructions can result in death, serious injury or equipment damage.**

### 4.1 Installation of the module

### CAUTION

#### DESTRUCTION DUE TO ESD

Electrostatic discharge (ESD) can cause immediate or later destruction of the module or the device.

- Use suitable ESD measures (IEC 61340-5-2) when handling the module.
- Do not touch any internal components.

**Failure to follow these instructions can result in equipment damage.**

- Install the module according to the instructions in the product manual.

Description	Order no.
CAN fieldbus module with 2 x RJ45 connection	VW3A3608

## 4.2 Electrical installation

*D-Sub connection* The D-SUB connection is male and has 2 threads 4-40 UNC.

- Cable specifications*
- Shielded cable
  - Minimum cross section of the signal wires 0.14 mm<sup>2</sup>
  - Twisted pair
  - Shield grounded at both ends
  - Maximum length depends on the number of devices, the baud rate and signal propagation delay. The higher the baud rates the shorter the bus cable needs to be.

*Maximum bus length CAN* The maximum bus length depends on the selected baud rate. Table 4.1 shows the maximum recommended overall length of the CAN bus in the case of cables with D-SUB connectors.

Baud rate [kbit/s]	Maximum bus length [m]
50	1000
125	500
250	250
500	100
1000	20 <sup>1)</sup>

1) According to the CANopen specification, the maximum bus length is 4 m. However, in practice, 20 m have been possible in most cases. External interference may reduce this length.

Table 4.1 Maximum bus length for CAN with D-SUB connection

NOTE: If you use cables with RJ45 connectors, the maximum bus length is reduced by 50%.

At a baud rate of 1 Mbit/s, the drop lines are limited to 0.3m.

The reference potential CAN\_0V and the shield connection (connector housing) are galvanically isolated.

- Keep the galvanic isolation in order to avoid ground loops via the CAN bus.
- Use equipotential bonding conductors.
- Use pre-assembled cables to reduce the risk of wiring errors.
- Verify that wiring, cables and connected interfaces meet the PELV requirements.

*Terminating resistors* Both ends of a CAN bus line must be terminated. A 120Ω terminating resistor between CAN\_L and CAN\_H is used for this purpose.



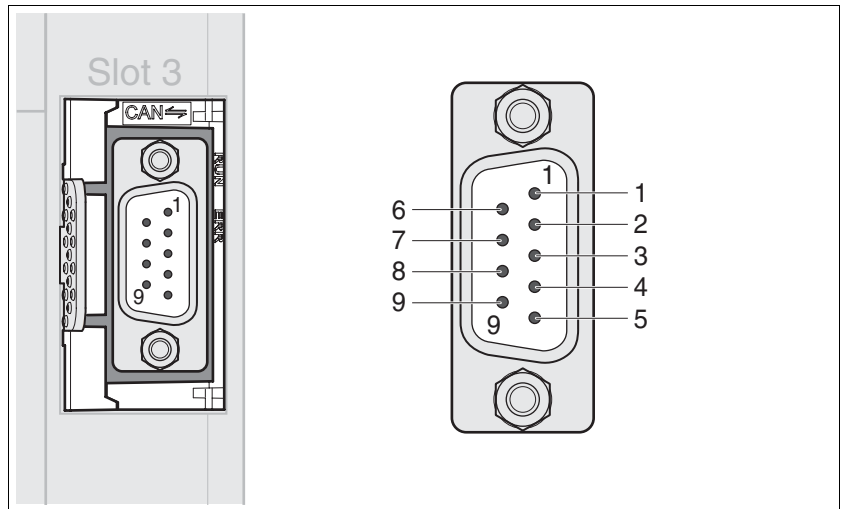
*Pin assignment*

Figure 4.1 Pin assignment module CANopen D-Sub

Pin	Signal	Meaning
1	-	Reserved
2	CAN_L	Fieldbus
3	CAN_0V	Reference potential CAN
4	-	Reserved
5	-	Reserved
6	CAN_0V	Reference potential CAN (connected to pin 3)
7	CAN_H	Fieldbus
8	-	Reserved
9	-	Reserved

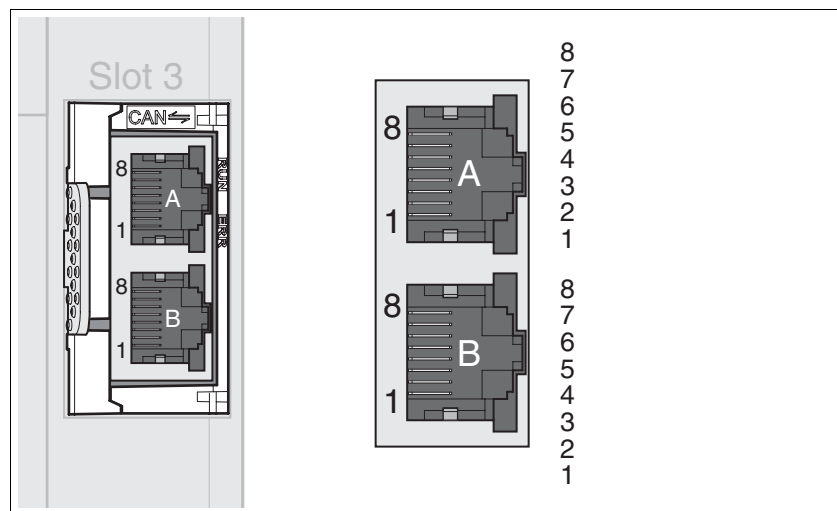


Figure 4.2 Pin assignment module CANopen RJ45

Pin	Signal	Meaning
1	CAN_H	Fieldbus
2	CAN_L	Fieldbus
3	CAN_0V	Reference potential CAN
4	-	Reserved
5	-	Reserved
6	-	Reserved
7	-	Reserved
8	-	Reserved
9	-	Reserved

## 5 Commissioning

# 5

### ⚠ WARNING

#### LOSS OF CONTROL

The product is unable to detect an interruption of the network link if connection monitoring is not active.

- Verify that connection monitoring is on.
- The shorter the time for monitoring, the faster the detection of the interruption.

**Failure to follow these instructions can result in death, serious injury or equipment damage.**

### ⚠ WARNING

#### UNINTENDED OPERATION

- Do not write values to reserved parameters.
- Do not write values to parameters unless you fully understand the function.
- Run initial tests without coupled loads.
- Verify that the system is free and ready for the movement before changing parameters.
- Verify the use of the word sequence with fieldbus communication.
- Do not establish a fieldbus connection unless you have fully understood the communication principles.

**Failure to follow these instructions can result in death, serious injury or equipment damage.**



*Using the library considerably facilitates controlling the device. The library is available for download from the Internet.*

<http://www.schneider-electric.com>

### 5.1 Commissioning the device

For installation in the network, the device must first be properly installed (mechanically and electrically) and commissioned.

Commission the device as per product manual. This prepares the device for operation in the network.

## 5.2 Address and baud rate

Up to 64 devices can be addressed in one CAN bus network branch and up to 127 devices in the extended network. Each device is identified by a unique address. The default node address for a device is 0.

The default baud rate is 250 kbaud.



*Each device must be assigned its own node address, i.e. any given node address may be assigned only once in the network.*

### *Setting address and baud rate*

The address is set directly at the device via parameter `canAddr` and the baud rate via parameter `canBaud`.

The baud rate must be the same for all devices on the fieldbus.

## 6 Operation

# 6

### WARNING

#### UNINTENDED OPERATION

- Do not write values to reserved parameters.
- Do not write values to parameters unless you fully understand the function.
- Run initial tests without coupled loads.
- Verify that the system is free and ready for the movement before changing parameters.
- Verify the use of the word sequence with fieldbus communication.
- Do not establish a fieldbus connection unless you have fully understood the communication principles.

**Failure to follow these instructions can result in death, serious injury or equipment damage.**

The chapter "Operation" describes the basic operating states, operating modes and functions of the device.



*Using the library considerably facilitates controlling the device. The library is available for download from the Internet.*

<http://www.schneider-electric.com>

## 6.1 Operating states

### 6.1.1 Indication of the operating state

The parameter `DCOMstatus` provides information on the operating state of the device and the processing status of the operating mode.

Parameter name HMI menu HMI name	Description	Unit Minimum value Factory setting Maximum value	Data type R/W Persistent Expert	Parameter address via fieldbus
_DCOMstatus	DriveCom status word  Refer to chapter Operation, State Machine for bit coding information. Bits 0 ... 3: Status bits Bit 4: Voltage enabled Bits 5 ... 6: Status bits Bit 7: Warning Bit 8: HALT request active Bit 9: Remote Bit 10: Target reached Bit 11: Reserved Bit 12: Operating mode specific Bit 13: x_err Bit 14: x_end Bit 15: ref_ok	- - - -	UINT16 UINT16 UINT16 UINT16 R/- - -	CANopen 6041:0h Modbus 6916 Profibus 6916 CIP 127.1.2

*Bits 0, 1, 2, 3, 5 and 6* Bits 0, 1, 2, 3, 5 and 6 of the `DCOMstatus` parameter provide information on the operating state.

Operating state	Bit 6 Switch On Disabled	Bit 5 Quick Stop	Bit 3 Fault	Bit 2 Operation Enabled	Bit 1 Switch On	Bit 0 Ready To Switch On
2 Not Ready To Switch On	0	X	0	0	0	0
3 Switch On Disabled	1	X	0	0	0	0
4 Ready To Switch On	0	1	0	0	0	1
5 Switched On	0	1	0	0	1	1
6 Operation Enable	0	1	0	1	1	1
7 Quick Stop Active	0	0	0	1	1	1
8 Fault Reaction Active	0	X	1	1	1	1
9 Fault	0	X	1	1	1	1

**Bit 4** Bit 4=1 indicates whether the DC bus voltage is correct. If the voltage is missing or is too low, the device does not transition from operating state 3 to operating state 4.

**Bit 7** Bit 7 becomes 1 if parameter `_WarnActive` contains a warning message. The movement is not interrupted. The bit remains set as long as a warning message is contained in parameter `_WarnActive`. The bit remains set for at least 100ms, even if a warning message is active for a shorter time. The bit is reset immediately in the case of a "Fault Reset".

**Bit 8** Bit 8=1 indicates that a "Halt" is active.

**Bit 9** If Bit 9 is set, the device carries out commands via the fieldbus. If Bit 9 is reset, the device is controlled via a different interface. In such a case, it is still possible to read or write parameters via the fieldbus.

- Bit 10* Bit 10 becomes "1" under the following conditions:
- A specified reference value in an operating mode was not reached. The reference value depends on the operating mode; see chapter 6.2 "Operating modes" for details.
  - The operating mode has been changed.
  - The operating mode has been interrupted by "Quick Stop" or "Halt".
- When the reference value changes, the bit changes.
- Bit 11* Reserved.
- Bit 12* Bit 12 is used for monitoring the current operating mode. Details can be found in the chapters on the individual operating modes.
- Bit 13* Bit 13 only becomes "1" in the case of an error which needs to be remedied prior to further processing. The device responds corresponding to an error class.
- Bit 14* Bit 14 changes to "0" if an operating mode is started. When processing is terminated or interrupted, for example by a "Halt", bit 14 toggles back to "1" once the motor has come to a standstill. The signal change of bit 14 to "1" is suppressed if one process is followed immediately by a new process in a different operating mode.
- Bit 15* Bit 15 is "1" if the motor has a valid reference point, for example as a result of a reference movement. A valid reference point remains valid even if the power stage is disabled.

### 6.1.2 Changing the operating state

It is possible to switch between operating states via the parameter `DCOMcontrol`.

Parameter name HMI menu HMI name	Description	Unit Minimum value Factory setting Maximum value	Data type R/W Persistent Expert	Parameter address via fieldbus
DCOMcontrol	DriveCom control word  Refer to chapter Operation, Operating States, for bit coding information. Bit 0: Switch on Bit 1: Enable Voltage Bit 2: Quick Stop Bit 3: Enable Operation Bits 4 ... 6: Operating mode specific Bit 7: Fault Reset Bit 8: Halt Bit 9: Change on setpoint Bits 10 ... 15: Reserved (must be 0)  Changed settings become active immediately.	- - - -	UINT16 UINT16 UINT16 UINT16 R/W - -	CANopen 6040:0 <sub>h</sub> Modbus 6914 Profibus 6914 CIP 127.1.1

*Bits 0, 1, 2, 3 and 7* Bits 0, 1, 2, 3 and 7 of the parameter `DCOMcontrol` allow you to switch between the operating states.

Fieldbus command	State transitions	State transition to	Bit 7, Fault Reset	Bit 3, Enable operation	Bit 2, Quick Stop	Bit 1, Enable Voltage	Bit 0, Switch On
Shutdown	T2, T6, T8	4: Ready To Switch On	X	X	1	1	0
Switch On	T3	5 Switched On	X	X	1	1	1
Disable Voltage	T7, T9, T10, T12	3 Switch On Disabled	X	X	X	0	X
Quick Stop	T7, T10 T11	3 Switch On Disabled 7 Quick Stop Active	X	X	0	1	X
Disable Operation	T5	5 Switched On	X	0	1	1	1
Enable Operation	T4, T16	6 Operation Enable	X	1	1	1	1
Fault Reset	T15	3 Switch On Disabled	0->1	X	X	X	X

*Bits 4 ... 6* Bits 4 to 6 are used for the operating mode-specific settings. Details can be found in the descriptions of the individual operating modes in this chapter.

*Bit 8* A "Halt" can be triggered with bit 8=1.

*Bits 9 ... 15* Reserved.



## 6.2 Operating modes

### 6.2.1 Starting and changing an operating mode

The parameter `DCOMopmode` is used to set the desired operating mode.

Parameter name HMI menu HMI name	Description	Unit Minimum value Factory setting Maximum value	Data type R/W Persistent Expert	Parameter address via fieldbus
DCOMopmode	Operating mode <b>-6 / Manual Tuning / Autotuning:</b> Manual Tuning / Autotuning <b>-3 / Motion Sequence:</b> Motion Sequence <b>-2 / Electronic Gear:</b> Electronic Gear <b>-1 / Jog:</b> Jog <b>0 / Reserved:</b> Reserved <b>1 / Profile Position:</b> Profile Position <b>3 / Profile Velocity:</b> Profile Velocity <b>4 / Profile Torque:</b> Profile Torque <b>6 / Homing:</b> Homing Changed settings become active immediately.	- -6 - 6	INT8 INT16 INT16 INT16 R/W - -	CANopen 6060:0 <sub>h</sub> Modbus 6918 Profibus 6918 CIP 127.1.3

► Set the operating mode with the parameter `DCOMopmode`.

The parameter `_DCOMopmode_act` can be used to read the current operating mode.

Parameter name HMI menu HMI name	Description	Unit Minimum value Factory setting Maximum value	Data type R/W Persistent Expert	Parameter address via fieldbus
_DCOMopmd_act	Active operating mode <b>-6 / Manual Tuning / Autotuning:</b> Manual Tuning / Autotuning <b>-3 / Motion Sequence:</b> Motion Sequence <b>-2 / Electronic Gear:</b> Electronic Gear <b>-1 / Jog:</b> Jog <b>0 / Reserved:</b> Reserved <b>1 / Profile Position:</b> Profile Position <b>3 / Profile Velocity:</b> Profile Velocity <b>4 / Profile Torque:</b> Profile Torque <b>6 / Homing:</b> Homing <b>8 / Cyclic Synchronous Position:</b> Cyclic Synchronous Position <b>9 / Cyclic Synchronous Velocity:</b> Cyclic Synchronous Velocity <b>10 / Cyclic Synchronous Torque:</b> Cyclic Synchronous Torque	- -6 - 10	INT8 INT16 INT16 INT16 R/- - -	CANopen 6061:0 <sub>h</sub> Modbus 6920 Profibus 6920 CIP 127.1.4

## 6.2.2 Operating mode Jog

*Starting the operating mode* The operating mode must be set in the parameter `DCOMopmode`. Writing the parameter value causes the operating mode to start.

The parameter `JOGactivate` starts the movement.

Parameter name HMI menu HMI name	Description	Unit Minimum value Factory setting Maximum value	Data type R/W Persistent Expert	Parameter address via fieldbus
JOGactivate	Activation of operating mode Jog Bit 0: Positive direction of movement Bit 1: Negative direction of movement Bit 2: 0=slow 1=fast  Changed settings become active immediately.	- 0 0 7	UINT16 UINT16 UINT16 UINT16 R/W - -	CANopen 301B:9 <sub>h</sub> Modbus 6930 Profibus 6930 CIP 127.1.9

*Control word* Bit 8 in parameter `DCOMcontrol` is used to stop a movement with "Halt".

Parameter value	Meaning
Bit 4: Reserved	Not relevant for this operating mode
Bit 5: Reserved	Not relevant for this operating mode
Bit 6: Reserved	Not relevant for this operating mode
Bit 8: Halt	Stop movement with "Halt"
Bit 9: Change on setpoint	Not relevant for this operating mode

*Terminating the operating mode* The operating mode is terminated when the motor is at a standstill and one of the following conditions is met:

- Value of the parameter `JOGactivate` is 0
- Stop caused by "Halt" or "Quick Stop"
- Stop caused by an error

*Status word* Information on the operating state and the current movement is available via bits 10 and 12 ... 15 in the parameter `DCOMstatus`.

Parameter value	Meaning
Bit 10: Reserved	Not relevant for this operating mode
Bit 12: Reserved	Not relevant for this operating mode
Bit 13: x_err	1: Error
Bit 14: x_end	0: Operating mode started 1: Operating mode terminated
Bit 15: ref_ok	1: Drive has valid reference point

## 6.2.2.1 Example Node address 1

Work step COB ID / data	Object Value
► Slow velocity to 100 601 / 2B 29 30 04 64 00 00 00 ◁ 581 / 60 29 30 04 00 00 00 00	3029:4 <sub>h</sub> 0064 <sub>h</sub>
► Fast velocity to 250 601 / 2B 29 30 05 FA 00 00 00 ◁ 581 / 60 29 30 05 00 00 00 00	3029:5 <sub>h</sub> 00FA <sub>h</sub>
► NMT Start remote node 0 / 01 00 ◁ T_PDO1 with status word 181 / 31 62	
► Enable power stage with R_PDO1 201 / 00 00 201 / 06 00 201 / 0F 00 ◁ T_PDO1 (operating state: 6 Operation Enabled) 181 / 37 42	
► Starting the operating mode 601 / 2F 60 60 00 FF 00 00 00 ◁ 581 / 60 60 60 00 00 00 00 00	6060 <sub>h</sub> FF <sub>h</sub>
► Check operating mode <sup>1)</sup> 601 / 40 61 60 00 00 00 00 00 ◁ Operating mode active 581 / 4F 61 60 00 FF 61 01 00	6061 <sub>h</sub>  FF <sub>h</sub>
► Start movement (positive direction, slow) 601 / 2B 1B 30 09 01 00 00 00 ◁ 581 / 60 1B 30 09 00 00 00 00 ◁ T_PDO1 with status word 181 / 37 02	301B:9 <sub>h</sub> 01 <sub>h</sub>
► Start movement (positive direction, fast) 601 / 2B 1B 30 09 05 00 00 00 ◁ 581 / 60 1B 30 09 00 00 00 00 ◁ T_PDO1 with status word 181 / 37 02	301B:9 <sub>h</sub> 05 <sub>h</sub>
► Terminate movement 601 / 2B 1B 30 09 00 00 00 00 ◁ 581 / 60 1B 30 09 00 00 00 00 ◁ T_PDO1 with status word 181 / 37 42	301B:9 <sub>h</sub> 00 <sub>h</sub>

1) The operating mode must be checked until the device has activated the specified operating mode.

### 6.2.3 Operating mode Electronic Gear

*Starting the operating mode* The operating mode must be set in the parameter `DCOMopmode`. Writing the parameter value causes the operating mode to start.

The parameter `GEARreference` starts the movement.

Parameter name HMI menu HMI name	Description	Unit Minimum value Factory setting Maximum value	Data type R/W Persistent Expert	Parameter address via fieldbus
GEARreference	Processing mode for operating mode Electronic Gear  <b>0 / Deactivated:</b> Deactivated <b>1 / Position Synchronization Immediate:</b> Position synchronization without compensation movement <b>2 / Position Synchronization Compensated:</b> Position synchronization with compensation movement <b>3 / Velocity Synchronization:</b> Velocity synchronization  Changed settings become active immediately.	- 0 0 3	UINT16 UINT16 UINT16 UINT16 R/W - -	CANopen 301B:12 <sub>h</sub> Modbus 6948 Profibus 6948 CIP 127.1.18

*Control word* Bit 8 in parameter `DCOMcontrol` is used to stop a movement with "Halt".

Parameter value	Meaning
Bit 4: Reserved	Not relevant for this operating mode
Bit 5: Reserved	Not relevant for this operating mode
Bit 6: Reserved	Not relevant for this operating mode
Bit 8: Halt	Stop movement with "Halt"
Bit 9: Change on setpoint	Not relevant for this operating mode

*Terminating the operating mode* The operating mode is terminated when the motor is at a standstill and one of the following conditions is met:

- Value of the parameter `GEARreference` is 0
- Stop caused by "Halt" or "Quick Stop"
- Stop caused by an error

*Status word* Information on the operating state and the current movement is available via bits 10 and 12 ... 15 in the parameter `DCOMstatus`.

Parameter value	Meaning
Bit 10: Reserved	Not relevant for this operating mode
Bit 12: Reserved	Not relevant for this operating mode
Bit 13: x_err	1: Error
Bit 14: x_end	0: Operating mode started 1: Operating mode terminated
Bit 15: ref_ok	1: Drive has valid reference point

### 6.2.4 Operating mode Profile Torque

*Starting the operating mode* The operating mode must be set in the parameter `DCOMopmode`. Writing the parameter value causes the operating mode to start.

The parameter `PTtq_target` starts the movement.

Parameter name HMI menu HMI name	Description	Unit Minimum value Factory setting Maximum value	Data type R/W Persistent Expert	Parameter address via fieldbus
PTtq_target	Target torque for operating mode Profile Torque  100.0 % correspond to the continuous stall torque <code>_M_M_0</code> .  In increments of 0.1 %.  Changed settings become active immediately.	% -3000.0 0.0 3000.0	INT16 INT16 INT16 INT16 R/W - -	CANopen 6071:0 <sub>h</sub> Modbus 6944 Profibus 6944 CIP 127.1.16

*Control word* Bit 8 in parameter `DCOMcontrol` is used to stop a movement with "Halt".

Parameter value	Meaning
Bit 4: Reserved	Not relevant for this operating mode
Bit 5: Reserved	Not relevant for this operating mode
Bit 6: Reserved	Not relevant for this operating mode
Bit 8: Halt	Stop movement with "Halt"
Bit 9: Change on setpoint	Not relevant for this operating mode

*Terminating the operating mode* The operating mode is terminated when the motor is at a standstill and one of the following conditions is met:

- Stop caused by "Halt" or "Quick Stop"
- Stop caused by an error

*Status word* Information on the operating state and the current movement is available via bits 10 and 12 ... 15 in the parameter `DCOMstatus`.

Parameter value	Meaning
Bit 10: Target reached	0: Target torque not reached 1: Target torque reached
Bit 12: Reserved	Not relevant for this operating mode
Bit 13: x_err	1: Error
Bit 14: x_end	0: Operating mode started 1: Operating mode terminated
Bit 15: ref_ok	1: Drive has valid reference point

## 6.2.4.1 Example Node address 1

Work step COB ID / data	Object Value
► NMT Start remote node 0 / 01 00 ◁ T_PDO1 with status word 181 / 31 62	
► Enable power stage with R_PDO1 201 / 00 00 201 / 06 00 201 / 0F 00 ◁ T_PDO1 (operating state: <b>6</b> Operation Enabled) 181 / 31 62	
► Starting the operating mode 601 / 2F 60 60 00 04 00 00 00 ◁ 581 / 60 60 60 00 00 00 00 00	6060 <sub>h</sub> 04 <sub>h</sub>
► Check operating mode <sup>1)</sup> 601 / 40 61 60 00 00 00 00 00 ◁ Operating mode active 581 / 4F 61 60 00 04 61 01 00	6061 <sub>h</sub>  04 <sub>h</sub>
► Specification of target torque 100 (10.0%) 601 / 2B 71 60 00 64 00 00 00 ◁ 581 / 60 71 60 00 00 00 00 00 ◁ Target torque reached 181 / 37 06	6071 <sub>h</sub> 64 <sub>h</sub>
► Terminate operating mode with "Quick Stop" with R_PDO1 201 / 0B 00 ◁ T_PDO1 with status word 181 / 17 66	
► Clear "Quick Stop" with R_PDO1 201 / 0F 00 ◁ T_PDO1 with status word 181 / 37 46	

1) The operating mode must be checked until the device has activated the specified operating mode.

### 6.2.5 Operating mode Profile Velocity

*Starting the operating mode* The operating mode must be set in the parameter `DCOMopmode`. Writing the parameter value causes the operating mode to start.

The parameter `PVv_target` starts the movement.

Parameter name HMI menu HMI name	Description	Unit Minimum value Factory setting Maximum value	Data type R/W Persistent Expert	Parameter address via fieldbus
PVv_target	Target velocity for operating mode Profile Velocity  The target velocity is limited to the setting in <code>CTRL_v_max</code> and <code>RAMP_v_max</code> .  Changed settings become active immediately.	usr_v - 0 -	INT32 INT32 INT32 INT32 R/W - -	CANopen 60FF:0h Modbus 6938 Profibus 6938 CIP 127.1.13

*Control word* Bit 8 in parameter `DCOMcontrol` is used to stop a movement with "Halt".

Parameter value	Meaning
Bit 4: Reserved	Not relevant for this operating mode
Bit 5: Reserved	Not relevant for this operating mode
Bit 6: Reserved	Not relevant for this operating mode
Bit 8: Halt	Stop movement with "Halt"
Bit 9: Change on setpoint	Not relevant for this operating mode

*Terminating the operating mode* The operating mode is terminated when the motor is at a standstill and one of the following conditions is met:

- Stop caused by "Halt" or "Quick Stop"
- Stop caused by an error

*Status word* Information on the operating state and the current movement is available via bits 10 and 12 ... 15 in the parameter `DCOMstatus`.

Parameter value	Meaning
Bit 10: Target reached	0: Target velocity not reached 1: Target velocity reached
Bit 12: Velocity	0: Velocity = >0 1: Velocity = 0
Bit 13: x_err	1: Error
Bit 14: x_end	0: Operating mode started 1: Operating mode terminated
Bit 15: ref_ok	1: Drive has valid reference point

## 6.2.5.1 Example Node address 1

Work step COB ID / data	Object Value
► Enable R_PDO3 601 / 23 02 14 01 01 04 00 04 ◁ 581 / 60 02 14 01 00 00 00 00	1402:1 <sub>h</sub> 0400 0401 <sub>h</sub>
► Enable T_PDO3 601 / 23 02 18 01 81 03 00 04 ◁ 581 / 60 02 18 01 00 00 00 00	1802:1 <sub>h</sub> 0400 0381 <sub>h</sub>
► Set acceleration to 2000 601 / 23 83 60 00 D0 07 00 00 ◁ 581 / 60 83 60 00 00 00 00 00	6083 <sub>h</sub> 0000 07D0 <sub>h</sub>
► NMT Start remote node 0 / 01 00 ◁ T_PDO3 with status word 381 / 31 66 00 00 00 00	
► Enable power stage with R_PDO3 401 / 00 00 00 00 00 00 401 / 06 00 00 00 00 00 401 / 0F 00 00 00 00 00 ◁ T_PDO3 (operating state: 6 Operation Enabled) 381 / 37 46 00 00 00 00	
► Starting the operating mode 601 / 2F 60 60 00 03 00 00 00 ◁ 581 / 60 60 60 00 00 00 00 00	6060 <sub>h</sub> 03 <sub>h</sub>
► Check operating mode <sup>1)</sup> 601 / 40 61 60 00 00 00 00 00 ◁ Operating mode active 581 / 4F 61 60 00 03 61 01 00	6061 <sub>h</sub>  03 <sub>h</sub>
► R_PDO3: Specification of target velocity 1000 401 / 0F 00 E8 03 00 00 ◁ T_PDO2 with status word and velocity actual value 381 / 37 02 00 00 00 00 ◁ Target velocity reached 381 / 37 06 E8 03 00 00	
► Terminate operating mode with "Quick Stop" with R_PDO3 401 / 0B 00 00 00 00 00 ◁ T_PDO3 with status word 381 / 17 66 00 00 00 00	
► Clear "Quick Stop" with R_PDO3 401 / 0F 00 00 00 00 00 ◁ T_PDO3 with status word 381 / 37 46 00 00 00 00	

1) The operating mode must be checked until the device has activated the specified operating mode.



## 6.2.6 Operating mode Profile Position

*Starting the operating mode* The operating mode must be set in the parameter `DCOMopmode`. Writing the parameter value causes the operating mode to start.

The movement is started via the control word.

The parameter `PVv_target` starts the movement.

*Control word* The bits 4 ... 6 and the bits 8 ... 9 in the parameter `DCOMcontrol` start a movement.

Bit 9: Change on setpoint	Bit 5: Change setpoint immediately	Bit 4: New target value	Meaning
0	0	0->1	Starts a movement to a target position. Target values transmitted during a movement become immediately effective and are executed at the target. The movement is stopped at the current target position. <sup>1)</sup>
1	0	0->1	Starts a movement to a target position. Target values transmitted during a movement become immediately effective and are executed at the target. The movement is not stopped at the current target position. <sup>1)</sup>
x	1	0->1	Starts a movement to a target position. Target values transmitted during a movement become immediately effective and are immediately executed. <sup>1)</sup>

1) Target values include target position, target velocity, acceleration and deceleration.

Parameter value	Meaning
Bit 6: Absolute / relative	0: Absolute movement 1: Relative movement
Bit 8: Halt	Stop movement with "Halt"

*Terminating the operating mode* The operating mode is terminated when the motor is at a standstill and one of the following conditions is met:

- Target position reached
- Stop caused by "Halt" or "Quick Stop"
- Stop caused by an error

*Status word* Information on the operating state and the current movement is available via bits 10 and 12 ... 15 in the parameter `DCOMstatus`.

Parameter value	Meaning
Bit 10: Target reached	0: Target position not reached 1: Target position reached
Bit 12: Target value acknowledge	0: New position possible 1: New target position accepted
Bit 13: x_err	1: Error
Bit 14: x_end	0: Operating mode started 1: Operating mode terminated
Bit 15: ref_ok	1: Drive has valid reference point

## 6.2.6.1 Example Node address 1

Work step COB ID / data	Object Value
▶ Enable R_PDO2 601 / 23 01 14 01 01 03 00 04 ◁ 581 / 60 01 14 01 00 00 00 00	1401:1 <sub>h</sub> 0400 0301 <sub>h</sub>
▶ Enable T_PDO2 601 / 23 01 18 01 81 02 00 04 ◁ 581 / 60 01 18 01 00 00 00 00	1801:1 <sub>h</sub> 0400 0281 <sub>h</sub>
▶ Set acceleration to 2000 601 / 23 83 60 00 D0 07 00 00 ◁ 581 / 60 83 60 00 00 00 00 00	6083 <sub>h</sub> 0000 07D0 <sub>h</sub>
▶ Set deceleration to 4000 601 / 23 84 60 00 A0 0F 00 00 ◁ 581 / 60 84 60 00 00 00 00 00	6084 <sub>h</sub> 0000 0FA0 <sub>h</sub>
▶ Set target velocity to 4000 601 / 23 81 60 00 A0 0F 00 00 ◁ 581 / 60 81 60 00 00 00 00 00	6081 <sub>h</sub> 0000 0FA0 <sub>h</sub>
▶ NMT Start remote node 0 / 01 00 ◁ T_PDO2 with status word 281 / 31 66 00 00 00 00	
▶ Enable power stage with R_PDO2 301 / 00 00 00 00 00 00 301 / 06 00 00 00 00 00 301 / 0F 00 00 00 00 00 ◁ T_PDO2 (operating state: 6 Operation Enabled) 281 / 37 42 00 00 00 00	
▶ Starting the operating mode 601 / 2F 60 60 00 01 00 00 00 ◁ 581 / 60 60 60 00 00 00 00 00	6060 <sub>h</sub> 01 <sub>h</sub>
▶ Check operating mode <sup>1)</sup> 601 / 40 61 60 00 00 00 00 00 ◁ Operating mode active 581 / 4F 61 60 00 01 61 01 00	6061 <sub>h</sub>  01 <sub>h</sub>
▶ R_PDO2: Start relative movement with NewSet- point=1 301 / 5F 00 30 75 00 00 ◁ T_PDO2 with status word and position actual value 281 / 37 12 00 00 00 00 ◁ Target position reached 281 / 37 56 30 75 00 00	
▶ R_PDO2: NewSetpoint=0 301 / 4F 00 30 75 00 00	

1) The operating mode must be checked until the device has activated the specified operating mode.

## 6.2.7 Operating mode Homing

*Starting the operating mode* The operating mode must be set in the parameter `DCOMopmode`. Writing the parameter value causes the operating mode to start.

The movement is started via the control word.

The parameter `HMmethod` lets you set the method.

*Control word* The bits 4 ... 6 and the bits 8 ... 9 in the parameter `DCOMcontrol` start a movement.

Parameter value	Meaning
Bit 4: Homing operation start	Start Homing
Bit 5: Reserved	Not relevant for this operating mode
Bit 6: Reserved	Not relevant for this operating mode
Bit 8: Halt	Stop movement with "Halt"
Bit 9: Change on setpoint	Not relevant for this operating mode

*Terminating the operating mode* The operating mode is terminated when the motor is at a standstill and one of the following conditions is met:

- Homing successful
- Stop caused by "Halt" or "Quick Stop"
- Stop caused by an error

*Status word* Information on the operating state and the current movement is available via bits 10 and 12 ... 15 in the parameter `DCOMstatus`.

Parameter value	Meaning
Bit 10: Target reached	0: Homing not completed 1: Homing completed
Bit 12: Homing attained	1: Homing successfully completed
Bit 13: x_err	1: Error
Bit 14: x_end	0: Operating mode started 1: Operating mode terminated
Bit 15: ref_ok	1: Drive has valid reference point

## 6.2.7.1 Example Node address 1

Work step COB ID / data	Object Value
► Velocity for searching the limit switch to 100 601 / 23 99 60 01 64 00 00 00 ◁ 581 / 60 99 60 01 00 00 00 00	6099:1 <sub>h</sub> 0000 0064 <sub>h</sub>
► Velocity for moving away from switch to 10 601 / 23 99 60 02 0A 00 00 00 ◁ 581 / 60 99 60 02 00 00 00 00	6099:2 <sub>h</sub> 0000 000A <sub>h</sub>
► NMT Start remote node 0 / 01 00 ◁ T_PDO1 with status word 181 / 31 62	
► Enable power stage with R_PDO1 201 / 00 00 201 / 06 00 201 / 0F 00 ◁ T_PDO1 (operating state: 6 Operation Enabled) 181 / 37 42	
► Starting the operating mode 601 / 2F 60 60 00 06 00 00 00 ◁ 581 / 60 60 60 00 00 00 00 00	6060 <sub>h</sub> 06 <sub>h</sub>
► Check operating mode <sup>1)</sup> 601 / 40 61 60 00 00 00 00 00 ◁ Operating mode active 581 / 4F 61 60 00 06 61 01 00	6061 <sub>h</sub>  06 <sub>h</sub>
► Select method 17 601 / 2F 98 60 00 11 00 00 00 ◁ 581 / 60 98 60 00 00 00 00 00	6098 <sub>h</sub> 11 <sub>h</sub>
► Start reference movement (Homing operation start) 201 / 1F 00 ◁ T_PDO1 reference movement active 181 / 37 02 ◁ T_PDO1 reference movement terminated 181 / 37 D6	

1) The operating mode must be checked until the device has activated the specified operating mode.

### 6.2.8 Operating mode Motion Sequence

*Starting the operating mode* The operating mode must be set in the parameter `DCOMopmode`. Writing the parameter value causes the operating mode to start.

The movement is started via the control word.

The parameter `MSM_start_ds` lets you set the data set to be started.

Parameter name HMI menu HMI name	Description	Unit Minimum value Factory setting Maximum value	Data type R/W Persistent Expert	Parameter address via fieldbus
MSM_start_ds	Selection of a data set to be started for operating mode Motion Sequence  Changed settings become active immediately.	- 0 0 31	UINT16 UINT16 UINT16 UINT16 R/W - -	CANopen 301B:A <sub>h</sub> Modbus 6932 Profibus 6932 CIP 127.1.10

*Control word* The bits 4 ... 6 and the bits 8 ... 9 in the parameter `DCOMcontrol` start a movement.

Parameter value	Meaning
Bit 4: Start movement	0 -> 1: Start data set
Bit 5: Movement type	0: Start individual data set 1: Start sequence
Bit 6: Data set for sequence	1: Use data set from parameter <code>MSM_start_ds</code> for starting a sequence
Bit 8: Halt	Stop movement with "Halt"
Bit 9: Change on setpoint	Not relevant for this operating mode

*Terminating the operating mode* The operating mode is terminated when the motor is at a standstill and one of the following conditions is met:

- Individual data set terminated
- Individual data set of a sequence terminated (waiting for transition condition to be fulfilled)
- Sequence terminated
- Stop caused by "Halt" or "Quick Stop"
- Stop caused by an error

*Status word* Information on the operating state and the current movement is available via bits 10 and 12 ... 15 in the parameter `DCOMstatus`.

Parameter value	Meaning
Bit 10: Sequence done	1: End of a sequence
Bit 12: Reserved	Not relevant for this operating mode
Bit 13: x_err	1: Error
Bit 14: x_end	0: Operating mode started 1: Operating mode terminated
Bit 15: ref_ok	1: Drive has valid reference point



## 7 Diagnostics and troubleshooting

# 7

### 7.1 Fieldbus communication error diagnostics

A properly operating fieldbus is essential for evaluating operating and error messages.

#### *Connections for fieldbus mode*

If the product cannot be addressed via the fieldbus, first check the connections. The product manual contains the technical data of the device and information on network and device installation. Check the following:

- 24V<sub>dc</sub> power supply
- Power connections to the device
- Fieldbus cable and fieldbus wiring
- Network connection to the device

You can also use the commissioning software for troubleshooting.

#### *Baud rate and address*

If it is impossible to connect to a device, check the baud rate and node address.

- The baud rate must be the same for all devices in the network.
- The node address of each device must be between 1 and 127 and unique for each device.

To set the baud rate and node address see chapter 5.2 "Address and baud rate".

#### *Fieldbus function test*

After correct configuration of the transmission data, test fieldbus mode. This requires installation of a CAN configuration tool that displays CAN messages. Feedback from the product is indicated in the form of a boot-up message:

- Switch the power supply off and on again.
- Observe the network messages after switching on. After initialization of the bus, the device sends a boot-up message (COB ID 700<sub>h</sub> + node ID and 1 data byte with the content 00<sub>h</sub>).
- With the factory setting 127 (7F<sub>h</sub>) for the node address, the boot-up message is sent via the bus. The device can then be put into operation via NMT services.



*If operation on the network cannot be started, the network function of the device must be checked by your local sales office. Contact your local sales office.*

## 7.2 Fieldbus status LEDs

*General* The fieldbus status LEDs visualize the fieldbus states.

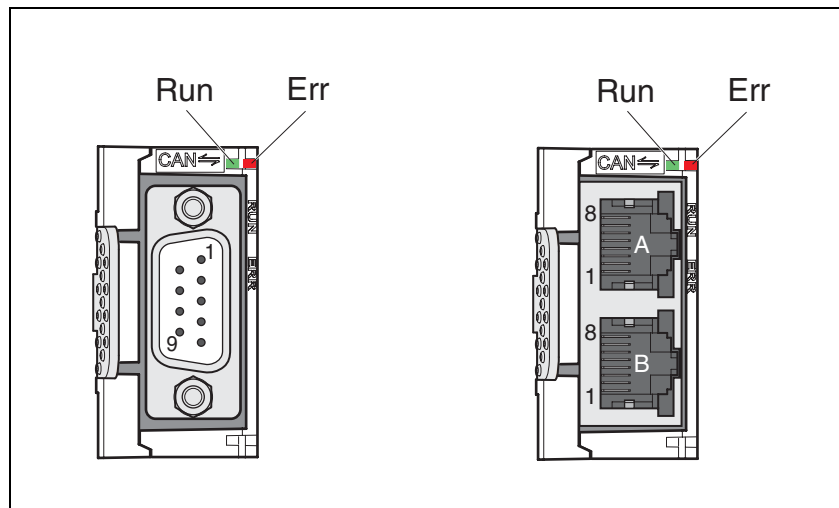


Figure 7.1 Fieldbus status LEDs



The illustration below shows the fieldbus communication states.

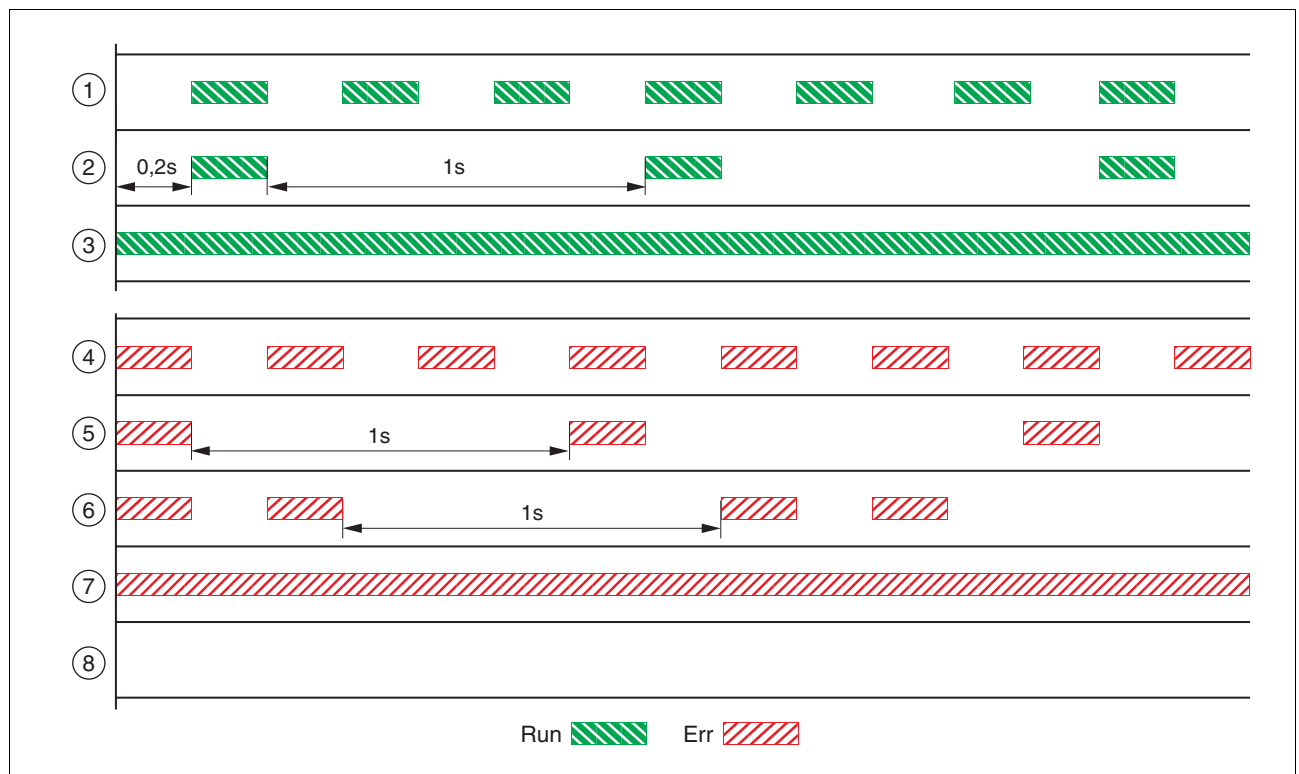


Figure 7.2 Signals of the CAN bus status LEDs (Run=GN; Err=RD)

- (1) NMT state PRE-OPERATIONAL
- (2) NMT state STOPPED
- (3) NMT state OPERATIONAL
- (4) Incorrect settings,  
for example, invalid node address
- (5) Warning limit reached,  
for example after 16 incorrect transmission attempts
- (6) Monitoring event (node guarding)
- (7) CAN is BUS-OFF,  
for example after 32 incorrect transmission attempts.
- (8) Fieldbus communication without error message.

## 7.3 Error diagnostics via fieldbus

### 7.3.1 Message objects

A number of objects provide information on the operating state and on errors:

- Object `Statusword` (`6041h`)  
Operating states, see product manual
- Object `EMCY` (`80h + Node-ID`)  
Error message from a device with error and error code, see chapter 3.5 "Emergency service"
- Object `Error register` (`1001h`)  
Error
- Object `Error code` (`603Fh`)  
Error code of the most recent error
- Devices use the special SDO error message ABORT to signal errors in exchanging messages via SDO.

### 7.3.2 Messages on the device status

Synchronous and asynchronous errors are distinguished in the evaluation and handling of errors.

*Synchronous errors* The device signals a synchronous error directly as a response to a message that cannot be evaluated. Possible causes comprise transmission errors or invalid data. For a list of synchronous errors see chapter 7.4.1 "Error register".

*Asynchronous errors* Asynchronous errors are signaled by the monitoring units in the device as soon as a device error occurs. An asynchronous error is signal via bit 3, "Fault", of the object `statusword` (`6041h`). In the case of errors that cause an interruption of the movement, the device transmits an EMCY message.

Asynchronous errors are also signaled via bits 5 ... 7 of the object `driveStat` (`2041h`).

## 7.4 CANopen error messages

CANopen error messages are signaled in the form of EMCY messages. They are evaluated via the objects `Error register (1001h)` and `Error code (603Fh)`. For information on the object EMCY see chapter 3.5 "Emergency service".

CANopen signals errors that occur during data exchange via SDO with the special SDO error message ABORT.

### 7.4.1 Error register

The object `Error register (1001h)` indicates the error of a device in bit-coded form. The exact cause of error can be determined with the error code table. Bit 0 is set as soon as an error occurs.

Bit	Message	Meaning
0	Generic error	An error has occurred
1	-	reserved
2	-	reserved
3	-	reserved
4	Communication	Network communication error
5	Device profile-specific	Error in execution as per device profile
6	-	reserved
7	Manufacturer-specific	Vendor-specific error message

### 7.4.2 Error code table

The error code is evaluated with the object `error code (603Fh)`, an object of the DSP402 device profile, and output as a four-digit hexadecimal value. The error code indicates the cause of the last interruption of movement. See the Troubleshooting chapter of the product manual for the meaning of the error code.

## 7.4.3 SDO error message ABORT

An SDO error message is generated as a response to an SDO transmission error. The cause of error is contained in `error code`, bytes 4 to byte 7.

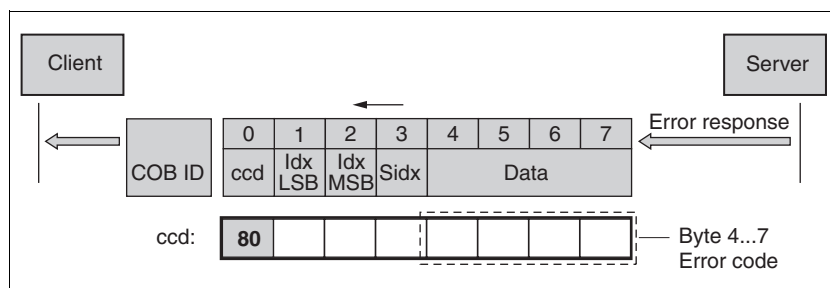


Figure 7.3 SDO error message as a response to an SDO message

The table below shows all error messages that may occur during data exchange with the product.

Error code	Meaning
0503 0000 <sub>h</sub>	Toggle bit not toggled
0504 0000 <sub>h</sub>	Time-out during SDO transfer
0504 0001 <sub>h</sub>	Command specifier CS incorrect or unknown
0504 0005 <sub>h</sub>	No memory available
0601 0000 <sub>h</sub>	Access to object impossible
0601 0001 <sub>h</sub>	No read access, because write-only object (wo)
0601 0002 <sub>h</sub>	No write access, because read object (ro)
0602 0000 <sub>h</sub>	Object does not exist in object dictionary
0604 0041 <sub>h</sub>	Object does not support PDO mapping
0604 0042 <sub>h</sub>	PDO mapping: number or length of objects exceed the byte length of the PDO
0604 0043 <sub>h</sub>	Parameters are incompatible
0604 0047 <sub>h</sub>	Device detects internal incompatibility
0606 0000 <sub>h</sub>	Hardware error, access denied
0607 0010 <sub>h</sub>	Data type and parameter length do not match
0607 0012 <sub>h</sub>	Data type does not match, parameter too long
0607 0013 <sub>h</sub>	Data type does not match, parameter too short
0609 0011 <sub>h</sub>	Subindex not supported
0609 0030 <sub>h</sub>	Value range of parameter too large (relevant only for write access)
0609 0031 <sub>h</sub>	Parameter values too great
0609 0032 <sub>h</sub>	Parameter values too small
0609 0036 <sub>h</sub>	Upper value is less than lower value
0800 0000 <sub>h</sub>	General error
0800 0020 <sub>h</sub>	Data can neither be transmitted to the application nor saved.
0800 0021 <sub>h</sub>	Device control is local, data cannot be transmitted or saved.
0800 0022 <sub>h</sub>	Data cannot be transmitted or saved in this device state.

Error code	Meaning
0800 0023 <sub>h</sub>	Object dictionary does not exist or cannot be generated, for example, if data error occurs during generation from file.
0800 xxxx <sub>h</sub>	Manufacturer-specific error, xxxx corresponds to the error number of the device. It is listed in the error code table of the device manual.



## 8 Object dictionary

# 8

### 8.1 Specifications for the objects

*Index* The index specifies the position of the object in the object dictionary. The index value is specified as a hexadecimal value.

*Object code* The object code specifies the data structure of the object.

Object code	Meaning	Coding
VAR	A simple value, for example of the type Integer8, Unsigned32 or Visible String8.	7
ARR (ARRAY)	A data field in which every entry is of the same data type.	8
REC (RECORD)	A data field that contains entries that are a combination of simple data types.	9

Data type	Value range	Data length	DS301 coding
Boolean	0 = false, 1 = true	1 byte	0001
Integer8	-128 ... +127	1 byte	0002
Integer16	-32768 ... +32767	2 byte	0003
Integer32	-2147483648 ... +2147483647	4 byte	0004
Unsigned8	0 ... 255	1 byte	0005
Unsigned16	0 ... 65535	2 byte	0006
Unsigned32	0 ... 4294967295	4 byte	0007
Visible String8	ASCII characters	8 byte	0009
Visible String16	ASCII characters	16 byte	0010

*RO/RW* Indicates read and/or write values  
 RO: values can only be read  
 RW: values can be read and written.

*PDO* R\_PDO: Mapping for R\_PDO possible  
 T\_PDO: Mapping for T\_PDO possible  
 No specification: PDO mapping not possible with the object

*Min/max values* Specifies the permissible range in which the object value is defined and valid.

*Factory setting* Factory settings when the product is shipped

*Persistent* "per." indicates whether the value of the parameter is persistent, i.e. whether it remains in the memory after the device is switched off .

8.2 Overview of object group 1000<sub>h</sub>

Index	Subindex	Name	Obj. code	Data type	Access	PDO	Description	Page
1000 <sub>h</sub>		Device type	VAR	Unsigned32	ro		Device type and profile	94
1001 <sub>h</sub>		Error register	VAR	Unsigned8	ro		Error register	95
1003 <sub>h</sub>		Predefined error field	ARR		rw		Error history, memory for error messages	96
1003 <sub>h</sub>	00 <sub>h</sub>	Number of errors	VAR	Unsigned8	rw		Number of error entries	96
1003 <sub>h</sub>	01 <sub>h</sub>	Error field	VAR	Unsigned32	ro		Error number	96
1005 <sub>h</sub>		COB-ID SYNC	VAR	Unsigned32	rw		Identifier of the synchronization object	97
1008 <sub>h</sub>		Manufacturer device name	VAR	Visible String8	ro		Manufacturer's designation	98
1009 <sub>h</sub>		Manufacturer hardware version	VAR	Visible String8	ro		Hardware version	98
100A <sub>h</sub>		Manufacturer software version	VAR	Visible String8	ro		Software version	99
100C <sub>h</sub>		Guard time	VAR	Unsigned16	rw		Time span for node guarding [ms]	99
100D <sub>h</sub>		Life time factor	VAR	Unsigned8	rw		Repeat factor for the node guarding protocol	100
1010 <sub>h</sub>		Save parameters	ARR	Unsigned32	rw		Saves parameters:	101
1010 <sub>h</sub>	01 <sub>h</sub>	Save all parameters	VAR	Unsigned32	rw		Saves all parameters	101
1010 <sub>h</sub>	02 <sub>h</sub>	Save communication parameters	VAR	Unsigned32	rw		Saves communication parameters	101
1010 <sub>h</sub>	03 <sub>h</sub>	Save application parameters	VAR	Unsigned32	rw		Saves application parameters	101
1011 <sub>h</sub>		Restore default of parameters	ARR	Unsigned32	rw		Resets parameter values to the default setting	102
1011 <sub>h</sub>	01 <sub>h</sub>	Restore default of all parameters	VAR	Unsigned32	rw		Resets all parameter values to the default setting	102
1011 <sub>h</sub>	02 <sub>h</sub>	Restore default of communication parameters	VAR	Unsigned32	rw		Resets communication parameter values to default	102
1011 <sub>h</sub>	03 <sub>h</sub>	Restore default of communication application parameters	VAR	Unsigned32	rw		Resets application parameter values to default	102
1014 <sub>h</sub>		COB-ID EMCY	VAR	Unsigned32	rw		Unsigned16	103
1015 <sub>h</sub>		Inhibit time EMCY	VAR	Unsigned16	rw		Unsigned16	104
1016 <sub>h</sub>		Consumer Heartbeat Time	ARR	Unsigned32	rw		Unsigned16	104
1016 <sub>h</sub>	01 <sub>h</sub>	Consumer Heartbeat Time	VAR	Unsigned32	rw		Time interval and node ID of the "Heartbeat" recipient	104
1017 <sub>h</sub>		Producer Heartbeat Time	VAR	Unsigned16	rw		Time interval for producer "Heartbeat"	105
1018 <sub>h</sub>		Identity Object	REC	Identity	ro		Identification object:	106
1018 <sub>h</sub>	01 <sub>h</sub>	Vendor ID	VAR	Unsigned32	ro		Vendor ID	106
1018 <sub>h</sub>	02 <sub>h</sub>	Product code	VAR	Unsigned32	ro		Product code	106
1018 <sub>h</sub>	03 <sub>h</sub>	Revision number	VAR	Unsigned32	ro		Revision number	106
1018 <sub>h</sub>	04 <sub>h</sub>	Serial number	VAR	Unsigned32	ro		Serial number	106



Index	Subindex	Name	Obj. code	Data type	Access	PDO	Description	Page
1020 <sub>h</sub>		Verify configuration	ARR	Unsigned32	rw		Checks configuration data	107
1020 <sub>h</sub>	01 <sub>h</sub>	Configuration date	VAR	Unsigned32	rw		Date of configuration	107
1020 <sub>h</sub>	02 <sub>h</sub>	Configuration time	VAR	Unsigned32	rw		Time of configuration	107
1029 <sub>h</sub>		Number of elements	ARR	Unsigned8	ro		Number of values for the object	107
1029 <sub>h</sub>	01 <sub>h</sub>	Communication error	ARR	Unsigned8	rw		Communication errors	107
1200 <sub>h</sub>		1st server SDO parameter	REC	SDO server param.	ro		First server SDO, settings	109
1200 <sub>h</sub>	01 <sub>h</sub>	COB-ID Client -> Server	VAR	Unsigned32	ro		Identifier client -> server	109
1200 <sub>h</sub>	02 <sub>h</sub>	COB-ID Server -> Client	VAR	Unsigned32	ro		Identifier server -> client	109
1201 <sub>h</sub>		2nd server SDO parameter	REC	SDO server param.	rw		Second server SDO, settings	110
1201 <sub>h</sub>	01 <sub>h</sub>	COB-ID Client -> Server	VAR	Unsigned32	rw		Identifier client -> server	110
1201 <sub>h</sub>	02 <sub>h</sub>	COB-ID Server -> Client	VAR	Unsigned32	rw		Identifier server -> client	110
1201 <sub>h</sub>	03 <sub>h</sub>	Node-ID SDO Client	VAR	Unsigned32	rw		Node ID SDO client	110
1400 <sub>h</sub>		1st receive PDO parameter	REC	PDO comm. param.	rw		First receive PDO (R_PDO1), settings	111
1400 <sub>h</sub>	01 <sub>h</sub>	COB-ID R_PDO1	VAR	Unsigned32	rw		Identifier of the R_PDO1	111
1400 <sub>h</sub>	02 <sub>h</sub>	Transmission type R_PDO1	VAR	Unsigned8	rw		Transmission type	111
1401 <sub>h</sub>		2nd receive PDO parameter	REC	PDO comm. param.	rw		Second receive PDO (R_PDO2), settings	113
1401 <sub>h</sub>	01 <sub>h</sub>	COB-ID R_PDO2	VAR	Unsigned32	rw		Identifier of the R_PDO2	113
1401 <sub>h</sub>	02 <sub>h</sub>	Transmission type R_PDO2	VAR	Unsigned8	rw		Transmission type	113
1402 <sub>h</sub>		3rd receive PDO parameter	REC	PDO comm. param.	rw		Third receive PDO (R_PDO3), settings	115
1402 <sub>h</sub>	01 <sub>h</sub>	COB-ID R_PDO3	VAR	Unsigned32	rw		Identifier of the R_PDO3	115
1402 <sub>h</sub>	02 <sub>h</sub>	Transmission type R_PDO3	VAR	Unsigned8	rw		Transmission type	115
1403 <sub>h</sub>		4th receive PDO parameter	REC	PDO comm. param.	rw		Fourth receive PDO (R_PDO4), settings	116
1403 <sub>h</sub>	01 <sub>h</sub>	COB-ID R_PDO4	VAR	Unsigned32	rw		Identifier of the R_PDO4	116
1403 <sub>h</sub>	02 <sub>h</sub>	Transmission type R_PDO4	VAR	Unsigned8	rw		Transmission type	116
1600 <sub>h</sub>		1st receive PDO mapping	REC	PDO mapping	ro		PDO mapping for R_PDO1, settings	117
1600 <sub>h</sub>	01 <sub>h</sub>	1st mapped object R_PDO1	VAR	Unsigned32	ro		First object for mapping in R_PDO1	117
1601 <sub>h</sub>		2nd receive PDO mapping	REC	PDO mapping	ro		PDO mapping for R_PDO2, settings	118
1601 <sub>h</sub>	01 <sub>h</sub>	1st mapped object R_PDO2	VAR	Unsigned32	ro		First object for mapping in R_PDO2	118
1601 <sub>h</sub>	02 <sub>h</sub>	2nd mapped object R_PDO2	VAR	Unsigned32	ro		Second object for mapping in R_PDO2	118

Index	Subindex	Name	Obj. code	Data type	Access	PDO	Description	Page
1602 <sub>h</sub>		3rd receive PDO mapping	REC	PDO mapping	ro		PDO mapping for R_PDO3, settings	119
1602 <sub>h</sub>	01 <sub>h</sub>	1st mapped object R_PDO3	VAR	Unsigned32	ro		First object for mapping in R_PDO3	119
1602 <sub>h</sub>	02 <sub>h</sub>	2nd mapped object R_PDO3	VAR	Unsigned32	ro		Second object for mapping in R_PDO3	119
1603 <sub>h</sub>		4th receive PDO mapping	REC	PDO mapping	rw		PDO mapping for R_PDO3, settings	120
1603 <sub>h</sub>	01 <sub>h</sub>	1st mapped object R_PDO4	VAR	Unsigned32	rw		First object for mapping in R_PDO4	120
1603 <sub>h</sub>	02 <sub>h</sub>	2nd mapped object R_PDO4	VAR	Unsigned32	rw		Second object for mapping in R_PDO4	120
1603 <sub>h</sub>	03 <sub>h</sub>	3rd mapped object R_PDO4	VAR	Unsigned32	rw		Third object for mapping in R_PDO4	120
1800 <sub>h</sub>		1st transmit PDO parameter	REC	PDO comm. param.	rw		First transmit PDO (T_PDO1), settings	121
1800 <sub>h</sub>	01 <sub>h</sub>	COB-ID T_PDO1	VAR	Unsigned32	rw		Identifier of the T_PDO1	121
1800 <sub>h</sub>	02 <sub>h</sub>	Transmission type T_PDO1	VAR	Unsigned8	rw		Transmission type	121
1800 <sub>h</sub>	03 <sub>h</sub>	Inhibit time T_PDO1	VAR	Unsigned16	rw		Inhibit time for locking bus access (1=100µs)	121
1800 <sub>h</sub>	04 <sub>h</sub>	Reserved T_PDO1	VAR	Unsigned8	rw		Priority for CAN bus arbitration ([0-7]).	121
1800 <sub>h</sub>	05 <sub>h</sub>	Event timer T_PDO1	VAR	Unsigned16	rw		Time span for event triggering (1=1 ms)	121
1801 <sub>h</sub>		2nd transmit PDO parameter	REC	PDO comm. param.	rw		Second transmit PDO (T_PDO2), settings	123
1801 <sub>h</sub>	01 <sub>h</sub>	COB-ID T_PDO2	VAR	Unsigned32	rw		Identifier of the T_PDO2	123
1801 <sub>h</sub>	02 <sub>h</sub>	Transmission type T_PDO2	VAR	Unsigned8	rw		Transmission type	123
1801 <sub>h</sub>	03 <sub>h</sub>	Inhibit time T_PDO2	VAR	Unsigned16	rw		Inhibit time for locking bus access (1=100µs)	123
1801 <sub>h</sub>	04 <sub>h</sub>	Reserved T_PDO2	VAR	Unsigned8	rw		Reserved	123
1801 <sub>h</sub>	05 <sub>h</sub>	Event timer T_PDO2	VAR	Unsigned16	rw		Time span for event triggering (1=1 ms)	123
1802 <sub>h</sub>		3rd transmit PDO parameter	REC	PDO comm. param.	rw		Third transmit PDO (T_PDO3), settings	125
1802 <sub>h</sub>	01 <sub>h</sub>	COB-ID T_PDO3	VAR	Unsigned32	rw		Identifier of the T_PDO3	125
1802 <sub>h</sub>	02 <sub>h</sub>	Transmission type T_PDO3	VAR	Unsigned8	rw		Transmission type	125
1802 <sub>h</sub>	03 <sub>h</sub>	Inhibit time T_PDO3	VAR	Unsigned16	rw		Inhibit time for locking bus access (1=100µs)	125
1802 <sub>h</sub>	04 <sub>h</sub>	Reserved T_PDO3	VAR	Unsigned8	rw		Reserved	125
1802 <sub>h</sub>	05 <sub>h</sub>	Event timer T_PDO3	VAR	Unsigned16	rw		Time span for event triggering (1=1 ms)	125
1803 <sub>h</sub>		4th transmit PDO parameter	REC	PDO comm. param.	rw		Fourth transmit PDO (T_PDO4), settings	127
1803 <sub>h</sub>	01 <sub>h</sub>	COB-ID T_PDO4	VAR	Unsigned32	rw		Identifier of the T_PDO4	127

Index	Subindex	Name	Obj. code	Data type	Access	PDO	Description	Page
1803 <sub>h</sub>	02 <sub>h</sub>	Transmission type T_PDO4	VAR	Unsigned8	rw		Transmission type	127
1803 <sub>h</sub>	03 <sub>h</sub>	Inhibit time T_PDO4	VAR	Unsigned16	rw		Inhibit time for locking bus access (1=100µs)	127
1803 <sub>h</sub>	04 <sub>h</sub>	Reserved T_PDO4	VAR	Unsigned8	ro		Reserved	127
1803 <sub>h</sub>	05 <sub>h</sub>	Event timer T_PDO4	VAR	Unsigned16	rw		Time span for event triggering (1=1 ms)	127
1A00 <sub>h</sub>		1st transmit PDO mapping	REC	PDO mapping	rw		PDO mapping for T_PDO1, settings	129
1A00 <sub>h</sub>	01 <sub>h</sub>	1st mapped object T_PDO1	VAR	Unsigned32	ro		First object for the mapping in T_PDO1	129
1A01 <sub>h</sub>		2nd transmit PDO mapping	REC	PDO mapping	rw		PDO mapping for T_PDO2, settings	130
1A01 <sub>h</sub>	01 <sub>h</sub>	1st mapped object T_PDO2	VAR	Unsigned32	ro		First object for the mapping in T_PDO2	130
1A01 <sub>h</sub>	02 <sub>h</sub>	2nd mapped object T_PDO2	VAR	Unsigned32	ro		Second object for the mapping in T_PDO2	130
1A02 <sub>h</sub>		3rd transmit PDO mapping	REC	PDO mapping	rw		PDO mapping for T_PDO3, settings	131
1A02 <sub>h</sub>	01 <sub>h</sub>	1st mapped object T_PDO3	VAR	Unsigned32	ro		First object for the mapping in T_PDO3	131
1A02 <sub>h</sub>	02 <sub>h</sub>	2nd mapped object T_PDO3	VAR	Unsigned32	ro		Second object for the mapping in T_PDO3	131
1A03 <sub>h</sub>		4th transmit PDO mapping	REC	PDO mapping	rw		PDO mapping for T_PDO4, settings	132
1A03 <sub>h</sub>	01 <sub>h</sub>	1st mapped object T_PDO4	VAR	Unsigned32	rw		First object for the mapping in T_PDO4	132
1A03 <sub>h</sub>	02 <sub>h</sub>	2nd mapped object T_PDO4	VAR	Unsigned32	rw		Second object for the mapping in T_PDO4	132
1A03 <sub>h</sub>	03 <sub>h</sub>	3rd mapped object T_PDO4	VAR	Unsigned32	rw		Third object for the mapping in T_PDO4	132
1A03 <sub>h</sub>	04 <sub>h</sub>	4th mapped object T_PDO4	VAR	Unsigned32	rw		Fourth object for the mapping in T_PDO4	132

8.3 Assignment object group 3000<sub>h</sub>

The product provides corresponding parameters for the CANopen object group 3000<sub>h</sub>. A detailed description of the parameters can be found in the product manual in the Parameters chapter.

Index	Subindex	Object	PDO	Data type	Parameter name	Takes effect
3001 <sub>h</sub>	1	Firmware program number	-	UINT32	_prgNoDEV	-
3001 <sub>h</sub>	2	Firmware version number	-	UINT16	_prgVerDEV	-
3001 <sub>h</sub>	4	Firmware revision number	-	UINT16	_prgRevDEV	-
3001 <sub>h</sub>	12	Current access channel	-	UINT16	_AccessInfo	-
3001 <sub>h</sub>	14	Locking other access channels	-	UINT16	AccessLock	Immediately
3004 <sub>h</sub>	7	Reset controller parameters	-	UINT16	PAR_CTRLreset	Immediately
3004 <sub>h</sub>	8	Reset user parameters	-	UINT16	PARuserReset	Enable power stage
3005 <sub>h</sub>	5	Commutation monitoring	-	UINT16	MON_commutat	Enable power stage
3005 <sub>h</sub>	6	Enabling the power stage at PowerOn	-	UINT16	IO_AutoEnable	Enable power stage
3005 <sub>h</sub>	7	Additional time delay for releasing the holding brake	-	INT16	BRK_AddT_release	Enable power stage
3005 <sub>h</sub>	8	Additional time delay for applying the holding brake	-	INT16	BRK_AddT_apply	Enable power stage
3005 <sub>h</sub>	9	Selection of internal or external braking resistor	-	UINT16	RESint_ext	Enable power stage
3005 <sub>h</sub>	10	Error response to missing mains phase of three-phase devices	-	UINT16	ErrorResp_Flt_AC	Enable power stage
3005 <sub>h</sub>	11	Error response to following error	-	UINT16	ErrorResp_p_dif	Enable power stage
3005 <sub>h</sub>	15	Detection and monitoring of mains phases	-	UINT16	MON_MainsVolt	Enable power stage
3005 <sub>h</sub>	16	Ground fault monitoring	-	UINT16	MON_GroundFault	Start-up
3005 <sub>h</sub>	17	Max. permissible switch-on time of external braking resistor	-	UINT16	RESex_ton	Enable power stage
3005 <sub>h</sub>	18	Nominal power of external braking resistor	-	UINT16	RESex_P	Enable power stage
3005 <sub>h</sub>	19	Resistance value of external braking resistor	-	UINT16	RESex_R	Enable power stage
3005 <sub>h</sub>	32	Mains reactor	-	UINT16	Mains_reactor	Immediately
3005 <sub>h</sub>	33	Shifting of the encoder working range	-	UINT16	ShiftEncWorkRang	Start-up
3005 <sub>h</sub>	34	Error response to 100% I <sub>2t</sub> braking resistor	-	UINT16	ErrorResp_I2tRES	Enable power stage
3006 <sub>h</sub>	1	Acceleration and deceleration of the motion profile for velocity	R_PDO	UINT16	RAMP_v_sym	Next movement
3006 <sub>h</sub>	3	Monitoring of software limit switches	-	UINT16	MON_SW_Limits	Immediately
3006 <sub>h</sub>	7	Position scaling: Denominator	-	INT32	ScalePOSdenom	Parameters
3006 <sub>h</sub>	8	Position scaling: Numerator	-	INT32	ScalePOSnum	Immediately
3006 <sub>h</sub>	12	Inversion of direction of movement	-	UINT16	InvertDirOfMove	Start-up

Index	Subindex	Object	PDO	Data type	Parameter name	Takes effect
3006 <sub>h</sub>	13	Jerk limitation of the motion profile for velocityv	-	UINT16	RAMP_v_jerk	Next movement
3006 <sub>h</sub>	14	Signal evaluation for reference switch	-	UINT16	IOsigREF	Enable power stage
3006 <sub>h</sub>	15	Signal evaluation for negative limit switch	-	UINT16	IOsigLIMN	Enable power stage
3006 <sub>h</sub>	16	Signal evaluation for positive limit switch	-	UINT16	IOsigLIMP	Enable power stage
3006 <sub>h</sub>	18	Deceleration ramp for Quick Stop	-	UINT32	RAMPquickstop	Next movement
3006 <sub>h</sub>	22	Absolute positioning only after homing	-	UINT16	AbsHomeRequest	Immediately
3006 <sub>h</sub>	24	Quick Stop option code	-	INT16	LIM_QStopReact	Immediately
3006 <sub>h</sub>	25	Monitoring of position deviation	-	UINT16	MON_p_DiffWin	Immediately
3006 <sub>h</sub>	26	Monitoring of velocity deviation	-	UINT32	MON_v_DiffWin	Immediately
3006 <sub>h</sub>	27	Monitoring of velocity threshold	-	UINT32	MON_v_Threshold	Immediately
3006 <sub>h</sub>	28	Monitoring of current threshold	-	UINT16	MON_I_Threshold	Immediately
3006 <sub>h</sub>	29	Monitoring of time window	-	UINT16	MON_ChkTime	Immediately
3006 <sub>h</sub>	30	Velocity limitation via input	-	UINT32	IO_v_limit	Immediately
3006 <sub>h</sub>	33	Velocity scaling: Denominator	-	INT32	ScaleVELdenom	Parameters
3006 <sub>h</sub>	34	Velocity scaling: Numerator	-	INT32	ScaleVELnum	Immediately
3006 <sub>h</sub>	38	Timeout time for standstill window monitoring	-	UINT16	MON_p_winTout	Immediately
3006 <sub>h</sub>	39	Current limitation via input	-	UINT16	IO_I_limit	Immediately
3006 <sub>h</sub>	40	Velocity limit for Zero Clamp	-	UINT32	MON_v_zeroclamp	Immediately
3006 <sub>h</sub>	41	Maximum load-dependent position deviation (warning)	-	UINT16	MON_p_dif_warn	Immediately
3006 <sub>h</sub>	45	Torque window, permissible deviation	-	UINT16	MON_tq_win	Immediately
3006 <sub>h</sub>	46	Torque window, time	-	UINT16	MON_tq_winTime	Immediately
3006 <sub>h</sub>	48	Ramp scaling: Denominator	-	INT32	ScaleRAMPdenom	Parameters
3006 <sub>h</sub>	49	Ramp scaling: Numerator	-	INT32	ScaleRAMPnum	Immediately
3007 <sub>h</sub>	1	Function Input DI0	-	UINT16	IOfuncn_DI0	Start-up
3007 <sub>h</sub>	2	Function Input DI1	-	UINT16	IOfuncn_DI1	Start-up
3007 <sub>h</sub>	3	Function Input DI2	-	UINT16	IOfuncn_DI2	Start-up
3007 <sub>h</sub>	4	Function Input DI3	-	UINT16	IOfuncn_DI3	Start-up
3007 <sub>h</sub>	9	Function Output DQ0	-	UINT16	IOfuncn_DQ0	Start-up
3007 <sub>h</sub>	10	Function Output DQ1	-	UINT16	IOfuncn_DQ1	Start-up
3008 <sub>h</sub>	1	Physical status of the digital inputs and outputs	T_PDO	UINT16	_IO_act	-
3008 <sub>h</sub>	15	Status of digital inputs	T_PDO	UINT16	_IO_DI_act	-
3008 <sub>h</sub>	16	Status of digital outputs	T_PDO	UINT16	_IO_DQ_act	-
3008 <sub>h</sub>	17	Setting the digital outputs directly	R_PDO	UINT16	IO_DQ_set	-
3008 <sub>h</sub>	32	Debounce time of DI0	-	UINT16	DI_0_Debounce	Immediately

Index	Subindex	Object	PDO	Data type	Parameter name	Takes effect
3008 <sub>h</sub>	33	Debounce time of DI1	-	UINT16	DI_1_Debounce	Immediately
3008 <sub>h</sub>	34	Debounce time of DI2	-	UINT16	DI_2_Debounce	Immediately
3008 <sub>h</sub>	35	Debounce time of DI3	-	UINT16	DI_3_Debounce	Immediately
3008 <sub>h</sub>	38	Status of the inputs for the safety function STO	-	UINT16	_IO_STO_act	-
300A <sub>h</sub>	1	Status of the capture inputs	-	UINT16	_CapStatus	-
300A <sub>h</sub>	2	Capture input 1 configuration	-	UINT16	Cap1Config	Immediately
300A <sub>h</sub>	4	Capture input 1 start/stop	-	UINT16	Cap1Activate	Immediately
300A <sub>h</sub>	6	Capture input 1 captured position	T_PDO	UINT32	_Cap1Pos	-
300A <sub>h</sub>	8	Capture input 1 event counter	T_PDO	UINT16	_Cap1Count	-
300D <sub>h</sub>	2	Motor type	-	UINT32	_M_Type	-
300D <sub>h</sub>	3	Motor encoder type	-	UINT16	_M_Sensor	-
300D <sub>h</sub>	4	Maximum permissible motor speed of rotation	-	UINT16	_M_n_max	-
300D <sub>h</sub>	5	Nominal motor speed of rotation	-	UINT16	_M_n_nom	-
300D <sub>h</sub>	6	Maximum motor current	-	UINT16	_M_I_max	-
300D <sub>h</sub>	7	Nominal motor current	-	UINT16	_M_I_nom	-
300D <sub>h</sub>	8	Nominal motor torque	-	UINT16	_M_M_nom	-
300D <sub>h</sub>	9	Maximum motor torque	-	UINT16	_M_M_max	-
300D <sub>h</sub>	10	Nominal motor voltage	-	UINT16	_M_U_nom	-
300D <sub>h</sub>	11	Motor EMF constant kE	-	UINT32	_M_kE	-
300D <sub>h</sub>	12	Moment of inertia of motor	-	UINT32	_M_Jrot	-
300D <sub>h</sub>	13	Motor winding resistance	-	UINT16	_M_R_UV	-
300D <sub>h</sub>	14	Motor inductance q component	-	UINT16	_M_L_q	-
300D <sub>h</sub>	15	Motor inductance d component	-	UINT16	_M_L_d	-
300D <sub>h</sub>	16	Maximum motor temperature	-	INT16	_M_T_max	-
300D <sub>h</sub>	17	Maximum permissible time for maximum motor current	-	UINT16	_M_I2t	-
300D <sub>h</sub>	19	Continuous stall current of motor	-	UINT16	_M_I_0	-
300D <sub>h</sub>	20	Number of pole pairs of motor	-	UINT16	_M_Polepair	-
3010 <sub>h</sub>	1	Nominal current of power stage	-	UINT16	_PS_I_nom	-
3010 <sub>h</sub>	2	Maximum current of power stage	-	UINT16	_PS_I_max	-
3010 <sub>h</sub>	3	Maximum permissible DC bus voltage	-	UINT16	_PS_U_maxDC	-
3010 <sub>h</sub>	4	Minimum permissible DC bus voltage	-	UINT16	_PS_U_minDC	-
3010 <sub>h</sub>	6	Temperature warning threshold of power stage	-	INT16	_PS_T_warn	-
3010 <sub>h</sub>	7	Maximum power stage temperature	-	INT16	_PS_T_max	-
3010 <sub>h</sub>	8	Resistance value of internal braking resistor	-	UINT16	_RESint_R	-
3010 <sub>h</sub>	9	Nominal power of internal braking resistor	-	UINT16	_RESint_P	-

Index	Subindex	Object	PDO	Data type	Parameter name	Takes effect
3010 <sub>h</sub>	10	DC bus voltage low threshold for Quick Stop	-	UINT16	_PS_U_minStopDC	-
3011 <sub>h</sub>	1	Current controller d component P gain	-	UINT16	_CTRL_KPd	Immediately
3011 <sub>h</sub>	2	Current controller d component integral action time	-	UINT16	_CTRL_TNid	Immediately
3011 <sub>h</sub>	3	Current controller q component P gain	-	UINT16	_CTRL_KPiq	Immediately
3011 <sub>h</sub>	4	Current controller q component integral action time	-	UINT16	_CTRL_TNiq	Immediately
3011 <sub>h</sub>	5	PID velocity controller time constant of D gain	-	UINT16	CTRL_nPIDTime	Immediately
3011 <sub>h</sub>	6	PID velocity controller amplification of D gain	-	UINT16	CTRL_nPIDDPart	Immediately
3011 <sub>h</sub>	8	Filter time constant to smooth velocity of motor	-	UINT16	CTRL_TAUact	Immediately
3011 <sub>h</sub>	9	Speed of rotation up to which the friction compensation is linear	-	UINT32	CTRL_SpdFric	Immediately
3011 <sub>h</sub>	10	Gain acceleration feed forward	-	UINT16	CTRL_KFAcc	Immediately
3011 <sub>h</sub>	12	Current limitation	-	UINT16	CTRL_I_max	Immediately
3011 <sub>h</sub>	13	Current value for Quick Stop	-	UINT16	LIM_I_maxQSTP	Immediately
3011 <sub>h</sub>	14	Current value for Halt	-	UINT16	LIM_I_maxHalt	Immediately
3011 <sub>h</sub>	15	Maximum field current of field weakness controller	-	UINT16	CTRL_I_max_fw	Enable power stage
3011 <sub>h</sub>	16	Velocity limitation	-	UINT32	CTRL_v_max	Immediately
3011 <sub>h</sub>	20	Period of time for parameter switching	-	UINT16	CTRL_ParChgTime	Immediately
3011 <sub>h</sub>	21	Global gain factor (affects parameter set 1)	-	UINT16	CTRL_GlobGain	Immediately
3011 <sub>h</sub>	22	Controller parameter set copying	-	UINT16	CTRL_ParSetCopy	Immediately
3011 <sub>h</sub>	23	Active controller parameter set	-	UINT16	_CTRL_ActParSet	-
3011 <sub>h</sub>	24	Selection of controller parameter set at power u	-	UINT16	CTRL_PwrUpParSet	Immediately
3011 <sub>h</sub>	25	Selection of controller parameter set (non-persistent)	-	UINT16	CTRL_SelParSet	Immediately
3011 <sub>h</sub>	26	Condition for parameter set switching	-	UINT16	CLSET_ParSwiCond	Immediately
3011 <sub>h</sub>	27	Time window for parameter set switching	-	UINT16	CLSET_winTime	Immediately
3011 <sub>h</sub>	28	Position deviation for parameter set switching	-	UINT16	CLSET_p_DiffWin	Immediately
3011 <sub>h</sub>	29	Velocity threshold for parameter set switching	-	UINT32	CLSET_v_Threshold	Immediately
3012 <sub>h</sub>	1	Velocity controller P gain	-	UINT16	CTRL1_KPn	Immediately
3012 <sub>h</sub>	2	Velocity controller integral action time	-	UINT16	CTRL1_TNn	Immediately
3012 <sub>h</sub>	3	Position controller P gain	-	UINT16	CTRL1_KPp	Immediately
3012 <sub>h</sub>	4	Filter time constant of the reference speed value filter	-	UINT16	CTRL1_TAUref	Immediately

Index	Subindex	Object	PDO	Data type	Parameter name	Takes effect
3012 <sub>h</sub>	5	Filter time constant of the reference current value filter	-	UINT16	CTRL1_TAUiref	Immediately
3012 <sub>h</sub>	6	Velocity feed-forward	-	UINT16	CTRL1_KFPp	Immediately
3012 <sub>h</sub>	8	Notch filter 1: Damping	-	UINT16	CTRL1_Nf1damp	Immediately
3012 <sub>h</sub>	9	Notch filter 1: Frequency	-	UINT16	CTRL1_Nf1freq	Immediately
3012 <sub>h</sub>	10	Notch filter 1: Bandwidth	-	UINT16	CTRL1_Nf1bandw	Immediately
3012 <sub>h</sub>	11	Notch filter 2: Damping	-	UINT16	CTRL1_Nf2damp	Immediately
3012 <sub>h</sub>	12	Notch filter 2: Frequency	-	UINT16	CTRL1_Nf2freq	Immediately
3012 <sub>h</sub>	13	Notch filter 2: Bandwidth	-	UINT16	CTRL1_Nf2bandw	Immediately
3012 <sub>h</sub>	14	Overshoot suppression filter: Damping	-	UINT16	CTRL1_Osupdamp	Immediately
3012 <sub>h</sub>	15	Overshoot suppression filter: Time delay	-	UINT16	CTRL1_Osupdelay	Immediately
3012 <sub>h</sub>	16	Friction compensation: Gain	-	UINT16	CTRL1_Kfric	Immediately
3013 <sub>h</sub>	1	Velocity controller P gain	-	UINT16	CTRL2_KPn	Immediately
3013 <sub>h</sub>	2	Velocity controller integral action time	-	UINT16	CTRL2_TNn	Immediately
3013 <sub>h</sub>	3	Position controller P gain	-	UINT16	CTRL2_KPp	Immediately
3013 <sub>h</sub>	4	Filter time constant of the reference speed value filter	-	UINT16	CTRL2_TAUunref	Immediately
3013 <sub>h</sub>	5	Filter time constant of the reference current value filter	-	UINT16	CTRL2_TAUiref	Immediately
3013 <sub>h</sub>	6	Velocity feed-forward	-	UINT16	CTRL2_KFPp	Immediately
3013 <sub>h</sub>	8	Notch filter 1: Damping	-	UINT16	CTRL2_Nf1damp	Immediately
3013 <sub>h</sub>	9	Notch filter 1: Frequency	-	UINT16	CTRL2_Nf1freq	Immediately
3013 <sub>h</sub>	10	Notch filter 1: Bandwidth	-	UINT16	CTRL2_Nf1bandw	Immediately
3013 <sub>h</sub>	11	Notch filter 2: Damping	-	UINT16	CTRL2_Nf2damp	Immediately
3013 <sub>h</sub>	12	Notch filter 2: Frequency	-	UINT16	CTRL2_Nf2freq	Immediately
3013 <sub>h</sub>	13	Notch filter 2: Bandwidth	-	UINT16	CTRL2_Nf2bandw	Immediately
3013 <sub>h</sub>	14	Overshoot suppression filter: Damping	-	UINT16	CTRL2_Osupdamp	Immediately
3013 <sub>h</sub>	15	Overshoot suppression filter: Time delay	-	UINT16	CTRL2_Osupdelay	Immediately
3013 <sub>h</sub>	16	Friction compensation: Gain	-	UINT16	CTRL2_Kfric	Immediately
301B <sub>h</sub>	6	Error response to data error (DE bit)	-	INT16	ErrorResp_bit_DE	-
301B <sub>h</sub>	7	Error response to mode error (ME bit)	-	INT16	ErrorResp_bit_ME	-
301B <sub>h</sub>	9	Activation of operating mode Jog	R_PDO	UINT16	JOGactivate	Immediately
301B <sub>h</sub>	19	DS402 state machine: State transition from 3 to 4	-	UINT16	DS402compatib	Immediately
301B <sub>h</sub>	22	Position for Position Setting	-	INT32	HMp_setP	Immediately
301B <sub>h</sub>	25	Error code for synchronous errors (ME bit)	-	UINT16	_ModeError	-
301B <sub>h</sub>	27	Error code for synchronous errors (DE bit)	-	UINT16	_DataError	-



Index	Subindex	Object	PDO	Data type	Parameter name	Takes effect
301B <sub>h</sub>	28	Additional error information of a ModeError (ME bit)	-	UINT16	_ModeErrorInfo	-
301B <sub>h</sub>	29	Additional error information of a DataError (DE bit)	-	UINT16	_DataErrorInfo	-
301B <sub>h</sub>	30	DS402 statusword: Selection for internal limit bit	-	UINT16	DS402intLim	Immediately
301C <sub>h</sub>	4	Action word	T_PDO	UINT16	_actionStatus	-
301C <sub>h</sub>	6	Modbus address of parameter with invalid value	-	UINT16	_InvalidParam	-
301C <sub>h</sub>	7	Current status of monitoring signals	-	UINT32	_SigActive	-
301C <sub>h</sub>	8	Saved status of monitoring signals	-	UINT32	_SigLatched	-
301C <sub>h</sub>	9	Number of last warning (error class 0)	-	UINT16	_LastWarning	-
301C <sub>h</sub>	10	Operating hours counter	-	UINT32	_OpHours	-
301C <sub>h</sub>	11	Active warnings, bit-coded	-	UINT32	_WarnActive	-
301C <sub>h</sub>	12	Saved warnings, bit-coded	-	UINT32	_WarnLatched	-
301C <sub>h</sub>	13	Current output power	-	INT16	_Power_act	-
301C <sub>h</sub>	14	Mean output power	-	INT16	_Power_mean	-
301C <sub>h</sub>	15	Voltage at DC bus	-	UINT16	_UDC_act	-
301C <sub>h</sub>	16	Current power stage temperature	-	INT16	_PS_T_current	-
301C <sub>h</sub>	17	Current motor temperature	-	INT16	_M_T_current	-
301C <sub>h</sub>	18	Current device temperature	-	INT16	_DEV_T_current	-
301C <sub>h</sub>	19	Current overload of braking resistor (I2t)	-	INT16	_RES_overload	-
301C <sub>h</sub>	20	Current load of braking resistor	-	INT16	_RES_load	-
301C <sub>h</sub>	21	Maximum value of overload of braking resistor	-	INT16	_RES_maxoverload	-
301C <sub>h</sub>	22	Current overload of power stage (I2t)	-	INT16	_PS_overload_I2t	-
301C <sub>h</sub>	23	Current load of power stage	-	INT16	_PS_load	-
301C <sub>h</sub>	24	Maximum value of overload of power stage	-	INT16	_PS_maxoverload	-
301C <sub>h</sub>	25	Current overload of motor (I2t)	-	INT16	_M_overload	-
301C <sub>h</sub>	26	Current load of motor	-	INT16	_M_load	-
301C <sub>h</sub>	27	Maximum value of overload of motor	-	INT16	_M_maxoverload	-
301C <sub>h</sub>	30	Maximum possible value for operating mode Profile Torque	-	INT16	_PT_max_val	-
301C <sub>h</sub>	31	Additional info of last error	-	UINT16	_LastError_Qual	-
301C <sub>h</sub>	34	Current overload of power stage (chip temperature)	-	INT16	_PS_overload_cte	-
301C <sub>h</sub>	35	Current overload of power stage (power squared)	-	INT16	_PS_overload_psq	-
301C <sub>h</sub>	36	Current overload of power stage	-	INT16	_PS_overload	-

Index	Subindex	Object	PDO	Data type	Parameter name	Takes effect
301C <sub>h</sub>	39	Current limitation of the system	-	UINT16	_lmax_system	-
301C <sub>h</sub>	40	Actual current limit	-	UINT16	_lmax_act	-
301C <sub>h</sub>	41	Actual velocity limit	-	UINT32	_Vmax_act	-
301E <sub>h</sub>	1	Actual motor current (q component, generating torque)	-	INT16	_lq_act_rms	-
301E <sub>h</sub>	2	Actual motor current (d component, field weakening)	-	INT16	_ld_act_rms	-
301E <sub>h</sub>	3	Total motor current	T_PDO	INT16	_l_act	-
301E <sub>h</sub>	4	Reference motor voltage q component	-	INT16	_Uq_ref	-
301E <sub>h</sub>	5	Reference motor voltage d component	-	INT16	_Ud_ref	-
301E <sub>h</sub>	6	Total motor voltage (vector sum d components and q components)	-	INT16	_Udq_ref	-
301E <sub>h</sub>	7	Reference speed of rotation of velocity controller	-	INT16	_n_ref	-
301E <sub>h</sub>	8	Actual speed of rotation of motor	-	INT16	_n_act	-
301E <sub>h</sub>	9	Reference position in internal units	-	INT32	_p_refInt	-
301E <sub>h</sub>	12	Reference position	-	INT32	_p_ref	-
301E <sub>h</sub>	14	Absolute position with reference to one motor revolution	-	UINT32	_p_absmodulo	-
301E <sub>h</sub>	15	Motor position with reference to encoder range	-	UINT32	_p_absENC	-
301E <sub>h</sub>	16	Reference motor current (q component, generating torque)	-	INT16	_lq_ref_rms	-
301E <sub>h</sub>	17	Reference motor current (d component, field weakening)	-	INT16	_ld_ref_rms	-
301E <sub>h</sub>	19	Degree of utilization of DC bus voltage	-	INT16	_VoltUtil	-
301E <sub>h</sub>	27	Maximum value of the load-dependent position deviation	-	UINT32	_p_dif_load_peak	Immediately
301E <sub>h</sub>	28	Current load-dependent position deviation between reference and actual position	-	INT32	_p_dif_load	-
301E <sub>h</sub>	31	Reference velocity of velocity controller	-	INT32	_v_ref	-
301F <sub>h</sub>	1	Target position of profile generator	-	INT32	_RAMP_p_target	-
301F <sub>h</sub>	2	Actual position of profile generator	T_PDO	INT32	_RAMP_p_act	-
301F <sub>h</sub>	5	Target velocity of profile generator	-	INT32	_RAMP_v_target	-
301F <sub>h</sub>	7	Velocity of reference value for profile generator	-	INT32	_pref_v	-
301F <sub>h</sub>	9	Acceleration of reference value for profile generator	-	INT32	_pref_acc	-
301F <sub>h</sub>	10	Maximum possible value of position scaling	-	INT32	_ScalePOSmaxusr	-

Index	Subindex	Object	PDO	Data type	Parameter name	Takes effect
301F <sub>h</sub>	11	Maximum possible value of velocity scaling	-	INT32	_ScaleVELmaxusr	-
301F <sub>h</sub>	12	Maximum possible value of ramp scaling	-	INT32	_ScaleRAMPmaxusr	-
3028 <sub>h</sub>	6	Maximum distance for search for switching point	-	INT32	HMoutdis	Next movement
3028 <sub>h</sub>	7	Distance from switching point	-	INT32	HMdis	Next movement
3028 <sub>h</sub>	10	Preferred homing method	-	INT16	HMprefmethod	Immediately
3028 <sub>h</sub>	11	Position at reference point	-	INT32	HMp_home	Next movement
3028 <sub>h</sub>	12	Distance from switching point to index pulse	-	INT32	_HMdisREFtoIDX	-
3028 <sub>h</sub>	13	Maximum search distance after overtravel of switch	-	INT32	HMsrchdis	Next movement
3029 <sub>h</sub>	3	Selection of jog method	-	UINT16	JOGmethod	Immediately
3029 <sub>h</sub>	4	Velocity for slow movement	-	UINT32	JOGv_slow	Immediately
3029 <sub>h</sub>	5	Velocity for fast movement	-	UINT32	JOGv_fast	Immediately
3029 <sub>h</sub>	7	Distance for step movement	-	INT32	JOGstep	Next movement
3029 <sub>h</sub>	8	Wait time for step movement	-	UINT16	JOGtime	Next movement
302F <sub>h</sub>	1	Autotuning start	-	UINT16	AT_start	Immediately
302F <sub>h</sub>	2	Autotuning status	-	UINT16	_AT_state	-
302F <sub>h</sub>	3	Movement range for Autotuning	-	UINT32	AT_dis	Next movement
302F <sub>h</sub>	4	Direction of movement for Autotuning	-	UINT16	AT_dir	Next movement
302F <sub>h</sub>	6	Jump of speed of rotation for Autotuning	-	UINT32	AT_n_ref	Next movement
302F <sub>h</sub>	7	Friction torque of the system	-	UINT16	_AT_M_friction	-
302F <sub>h</sub>	8	Constant load torque	-	INT16	_AT_M_load	-
302F <sub>h</sub>	9	Waiting time between Autotuning steps	-	UINT16	AT_wait	Next movement
302F <sub>h</sub>	11	Progress of Autotuning	-	UINT16	_AT_progress	-
302F <sub>h</sub>	12	Moment of inertia of the complete system	-	UINT16	_AT_J	Parameters
302F <sub>h</sub>	14	Type of coupling of the system	-	UINT16	AT_mechanical	Next movement
303A <sub>h</sub>	1	Lock HMI	-	UINT16	HMIlocked	Immediately
303A <sub>h</sub>	2	HMI display when motor moves	-	UINT16	HMIDispPara	Immediately
303B <sub>h</sub>	2	Number of power on cycles	-	UINT32	_ERR_powerOn	-
303B <sub>h</sub>	4	Clear error memory	-	UINT16	ERR_clear	Immediately
303B <sub>h</sub>	5	Reset error memory read pointer	-	UINT16	ERR_reset	Immediately
303B <sub>h</sub>	6	First number for the signal output function Selected Error	-	UINT16	MON_IO_SelErr1	Immediately
303B <sub>h</sub>	7	Second number for the signal output function Selected Error	-	UINT16	MON_IO_SelErr2	Immediately
303B <sub>h</sub>	8	First number for the signal output function Selected Warning	-	UINT16	MON_IO_SelWar1	Immediately

Index	Subindex	Object	PDO	Data type	Parameter name	Takes effect
303B <sub>h</sub>	9	Second number for the signal output function Selected Warning	-	UINT16	MON_IO_SelWar2	Immediately
303C <sub>h</sub>	1	Error number	-	UINT16	_ERR_number	-
303C <sub>h</sub>	2	Error class	-	UINT16	_ERR_class	-
303C <sub>h</sub>	3	Error time	-	UINT32	_ERR_time	-
303C <sub>h</sub>	4	Error additional information	-	UINT16	_ERR_qual	-
303C <sub>h</sub>	5	Number of cycles of enabling the power stage at error time	-	UINT16	_ERR_enable_cycl	-
303C <sub>h</sub>	6	Time between enabling of power stage and occurrence of the error	-	UINT16	_ERR_enable_time	-
303C <sub>h</sub>	7	DC bus voltage at error time	-	UINT16	_ERR_DCbus	-
303C <sub>h</sub>	8	Motor velocity at error time	-	INT32	_ERR_motor_v	-
303C <sub>h</sub>	9	Motor current at error time	-	UINT16	_ERR_motor_I	-
303C <sub>h</sub>	10	Temperature of power stage at error time	-	INT16	_ERR_temp_ps	-
303C <sub>h</sub>	11	Temperature of device at error time	-	INT16	_ERR_temp_dev	-
3041 <sub>h</sub>	6	CANopen diagnosis word	-	UINT16	_CanDiag	-
3041 <sub>h</sub>	10	CANopen Manufacturer-specific SDO Abort Code	-	UINT16	_ManuSdoAbort	-
3041 <sub>h</sub>	11	PDO 1 event mask	-	UINT16	CANpdo1Event	Immediately
3041 <sub>h</sub>	12	PDO 2 event mask	-	UINT16	CANpdo2Event	Immediately
3041 <sub>h</sub>	13	PDO 3 event mask	-	UINT16	CANpdo3Event	Immediately
3041 <sub>h</sub>	14	PDO 4 event mask	-	UINT16	CANpdo4Event	Immediately

## 8.4 Assignment object group 6000<sub>h</sub>



The product provides corresponding parameters for the CANopen object group 6000<sub>h</sub>. A detailed description of the parameters can be found in the product manual in the Parameters chapter.

Index	Subindex	Object	PDO	Data type	Parameter name	Takes effect
603F <sub>h</sub>	0	Error code	-	UINT16	_LastError	-
6040 <sub>h</sub>	0	Control word	R_PDO	UINT16	DCOMcontrol	Immediately
6041 <sub>h</sub>	0	Status word	T_PDO	UINT16	_DCOMstatus	-
605D	0	Halt option code	-	INT16	LIM_HaltReaction	Immediately
6060 <sub>h</sub>	0	Modes of Operation	R_PDO	INT8	DCOMopmode	Immediately
6061 <sub>h</sub>	0	Modes of Operation Display	T_PDO	INT8	_DCOMopmd_act	-
6063 <sub>h</sub>	0	Position actual Value	T_PDO	INT32	_p_actInt	-
6064 <sub>h</sub>	0	Position actual Value	T_PDO	INT32	_p_act	-
6065 <sub>h</sub>	0	Following error window	-	UINT32	MON_p_dif_load	Immediately
6067 <sub>h</sub>	0	Position Window	-	UINT32	MON_p_win	Immediately
6068 <sub>h</sub>	0	Position window time	-	UINT16	MON_p_winTime	Immediately
606B <sub>h</sub>	0	Velocity demand value	-	INT32	_RAMP_v_act	-
606C <sub>h</sub>	0	Velocity actual value	T_PDO	INT32	_v_act	-
606D <sub>h</sub>	0	Velocity Window	-	UINT16	MON_v_win	Immediately
606E <sub>h</sub>	0	Velocity Window Time	-	UINT16	MON_v_winTime	Immediately
6071 <sub>h</sub>	0	Target Torque	R_PDO	INT16	PTtq_target	Immediately
6077 <sub>h</sub>	0	Torque actual Value	T_PDO	INT16	_tq_act	-
607A <sub>h</sub>	0	Target position	R_PDO	INT32	PPp_target	Immediately
607D <sub>h</sub>	1	Negative software position limit	-	INT32	MON_swLimN	Enable power stage
607D <sub>h</sub>	2	Positive software position limit	-	INT32	MON_swLimP	Enable power stage
607F <sub>h</sub>	0	Maximum profile velocity	-	UINT32	RAMP_v_max	Next movement
6081 <sub>h</sub>	0	Profile velocity	R_PDO	UINT32	PPv_target	Next movement
6083 <sub>h</sub>	0	Profile acceleration	R_PDO	UINT32	RAMP_v_acc	Next movement
6084 <sub>h</sub>	0	Profile deceleration	R_PDO	UINT32	RAMP_v_dec	Next movement
6086 <sub>h</sub>	0	Motion profile type	-	UINT16	ProfileType	
6087	0	Torque slope	R_PDO	UINT32	RAMP_tq_slope	Immediately
6098 <sub>h</sub>	0	Homing method	-	INT8	HMmethod	Immediately
6099 <sub>h</sub>	1	Homing speed search	-	UINT32	HMv	Next movement
6099 <sub>h</sub>	2	Homing speed moving away	-	UINT32	HMv_out	Next movement
60C2 <sub>h</sub>	1	Interpolation Time Period value	-	UINT8	Cycle time	-
60C2 <sub>h</sub>	2	Interpolation Time Period index	-	INT8	time dimension	-
60F2 <sub>h</sub>	0	Position Option Code	-	UINT16	PPoption	Next movement
60F4 <sub>h</sub>	0	Following error actual value	-	INT32	_p_dif	-
60FF <sub>h</sub>	0	Target velocity	R_PDO	INT32	PVv_target	Immediately
6502 <sub>h</sub>	0	Supported Drive Modes	-	UINT32	_SuppDriveModes	-

## 8.5 Details of object group 1000h

### 8.5.1 1000<sub>h</sub> Device type

The object specifies the device profile used as well as the device type.

#### *Object description*

Index	1000 <sub>h</sub>
Object name	Device type
Object code	VAR
Data type	Unsigned32

#### *Value description*

Subindex	00 <sub>h</sub> , device type
Meaning	Device type and profile
Access	read-only
PDO mapping	–
Value range	–
Default value	
Can be saved	–

#### *Bit coding, subindex 00<sub>h</sub>*

bit	Access	Value	Meaning
31-24	ro	00 <sub>h</sub>	not used
15-0	ro	0192 <sub>h</sub>	Device profile DS-402 (192 <sub>h</sub> )

8.5.2 1001<sub>h</sub> Error register

The object specifies the error of the device. The detailed cause of error can be determined with the object `predefined error field` (1003<sub>h</sub>) and - for reasons of compatibility with devices with other field-bus profiles - the object `error code` (603F<sub>h</sub>).

Errors are signaled by an EMCY message as soon as they occur.

*Object description*

Index	1001 <sub>h</sub>
Object name	Error register
Object code	VAR
Data type	Unsigned8

*Value description*

Subindex	00 <sub>h</sub> , error register
Meaning	Error register
Access	read-only
PDO mapping	–
Value range	–
Default value	–
Can be saved	–

*Bit coding, subindex 00<sub>h</sub>*

bit	Access	Value	Meaning
0	ro	–	Error (generic error)
1	ro	–	Reserved
2	ro	–	Reserved
3	ro	–	Reserved
4	ro	–	Communication profile (communication error)
5	ro	–	Device profile (device profile error)
6	ro	–	Reserved
7	ro	–	Manufacturer-specific

8.5.3 1003<sub>h</sub> Predefined error field

The object saves the latest error messages that were shown as EMCY messages.

- The subindex 00<sub>h</sub> entry contains the number of saved error messages.
- The current error message is stored at subindex 01<sub>h</sub>, older messages are moved to higher subindex entries.
- Writing 0 to subindex 00<sub>h</sub> resets the error list.

*Object description*

Index	1003 <sub>h</sub>
Object name	Predefined error field
Object code	ARRAY
Data type	Unsigned32

*Value description*

Subindex	00 <sub>h</sub> , number of errors
Meaning	Number of error entries
Access	read-write
PDO mapping	–
Value range	0...1
Default value	1
Can be saved	–
Subindex	01 <sub>h</sub> , error field
Meaning	Error number
Access	read-only
PDO mapping	–
Value range	–
Default value	0
Can be saved	–

*Bit coding, subindex 00<sub>h</sub> ... 05<sub>h</sub>*

Bytes 0 ... 15: error code. Bytes 16 ... 31 additional error information, not assigned in the device.



8.5.4 1005<sub>h</sub> COB ID SYNC message

The object specifies the COB ID of the SYNC object and determines whether a device sends or receives SYNC messages.

The device can only receive SYNC messages.

For synchronization, a device in the network must send SYNC objects.

The COB ID can be changed in the NMT state "Pre-Operational"

*Object description*

Index	1005 <sub>h</sub>
Object name	COB ID SYNC
Object code	VAR
Data type	Unsigned32

*Value description*

Subindex	00 <sub>h</sub> , COB ID SYNC
Meaning	Identifier of the synchronization object
Access	read-write
PDO mapping	–
Value range	0...4294967295
Default value	8000 0080 <sub>h</sub>
Can be saved	Yes

*Bit coding, subindex 00<sub>h</sub>*

bit	Access	Value	Meaning
31	ro	0 <sub>b</sub>	1: Device can receive SYNC messages (SYNC consumer)
30	ro	1 <sub>b</sub>	1: Device can send SYNC messages (SYNC producer)
29	ro	0 <sub>b</sub>	0: 11 bit identifier (CAN 3.0A) 1: 29 bit identifier (CAN 3.0B)
28-11	ro	0000 <sub>h</sub>	Only relevant if bit 29=1, not used by the device.
10-7	rw	0001 <sub>b</sub>	Function code, bit 10 ... 7 of COB ID
6-0	ro	7F <sub>h</sub>	Node address, bit 6 ... 0 of COB ID

8.5.5 1008<sub>h</sub> Manufacturer device name

The object specifies the device name of the manufacturer.

*Object description*

Index	1008 <sub>h</sub>
Object name	Manufacturer device name
Object code	VAR
Data type	Visible String8

*Value description*

Subindex	00 <sub>h</sub> , manufacturer device name
Meaning	Manufacturer's designation
Access	read-only
PDO mapping	–
Value range	–
Default value	–
Can be saved	–

The following objects contain additional information on the device:- Objects 6404<sub>h</sub>, 6410<sub>h</sub>: Motor data

8.5.6 1009<sub>h</sub> Manufacturer hardware version

The object specifies the version of the device hardware.

*Object description*

Index	1009 <sub>h</sub>
Object name	Manufacturer hardware version
Object code	VAR
Data type	Visible String8

*Value description*

Subindex	00 <sub>h</sub> , manufacturer hardware version
Meaning	Hardware version
Access	read-only
PDO mapping	–
Value range	–
Default value	–
Can be saved	–

### 8.5.7 100A<sub>h</sub> Manufacturer software version

The object specifies the version of the device software.

#### Object description

Index	100A <sub>h</sub>
Object name	Manufacturer software version
Object code	VAR
Data type	Visible String8

#### Value description

Subindex	00 <sub>h</sub> , manufacturer software version
Meaning	Software version
Access	read-only
PDO mapping	–
Value range	–
Default value	–
Can be saved	–

### 8.5.8 100C<sub>h</sub> Guard time

The object specifies the time span for connection monitoring (node guarding) of an NMT slave.

The time span for connection monitoring of an NMT master results from the time span "guard time" multiplied by the factor "life time", object Life time factor(100D<sub>h</sub>) .

The time span can be changed in the NMT state "Pre-Operational".

#### Object description

Index	100C <sub>h</sub>
Object name	Guard time
Object code	VAR
Data type	Unsigned16

#### Value description

Subindex	00 <sub>h</sub> , guard time
Meaning	Time span for node guarding [ms]
Access	read-write
PDO mapping	–
Value range	0...65535
Default value	0
Can be saved	Yes

8.5.9 100D<sub>h</sub> Life time factor

The object specifies the factor that, together with the time span "guard time", results in the time interval for connection monitoring of an NMT master. Within this period, the NMT slave device expects a monitoring request via node guarding from the NMT master.

life time = guard time \* life time factor

The value "0" deactivates monitoring of the NMT master.

If there is no connection monitoring through the NMT master during the time interval "life time", the device signals an error and switches to the operating state Fault.

The time factor can be changed in the NMT state "Pre-Operational".

The time span "guard time" is set with the object `Guard time` (100C<sub>h</sub>).

*Object description*

Index	100D <sub>h</sub>
Object name	Life time factor
Object code	VAR
Data type	Unsigned8

*Value description*

Subindex	00 <sub>h</sub> , life time factor
Meaning	Repeat factor for the node guarding protocol.
Access	read-write
PDO mapping	–
Value range	0...255
Default value	0
Can be saved	Yes

8.5.10 1010<sub>h</sub> Save Parameters

The object is used to save parameters.

- Subindex 01<sub>h</sub>, all parameters
- Subindex 02<sub>h</sub>, communication parameters
- Subindex 03<sub>h</sub>, application parameters

*Object description*

Index	1010 <sub>h</sub>
Object name	Save parameters
Object code	ARRAY
Data type	Unsigned32

*Value description*

Subindex	00 <sub>h</sub> , number of elements
Meaning	Number of values for the object
Access	read-only
PDO mapping	–
Value range	–
Default value	3
Can be saved	–

Subindex	01 <sub>h</sub> , save all parameters
Meaning	saves all parameters
Access	read-write
PDO mapping	–
Value range	–
Default value	1
Can be saved	–

Subindex	02 <sub>h</sub> , save communication parameters
Meaning	Saves communication parameters
Access	read-write
PDO mapping	–
Value range	–
Default value	1
Can be saved	–

Subindex	03 <sub>h</sub> , save application parameters
Meaning	Saves application parameters
Access	read-write
PDO mapping	–
Value range	–
Default value	1
Can be saved	–

8.5.11 1011<sub>h</sub> Restore default parameters

The object is used to restore the default parameters.

*Switching off the product*

NOTE: If you use the object to reset application parameters or user parameter or both, the device also saves the data to the EEPROM. If the product is switched off while the data is saved, the product signals a CRC error the next time it is switched on. To avoid this, proceed as follows:

- ▶ Read the object `PAReeprSave 3004:01h` during the reset (see ).
- ▶ Do not switch off the product unless the object `PAReeprSave 3004:01h` has assumed the value 0.

- Subindex 01<sub>h</sub>, all parameters
- Subindex 02<sub>h</sub>, communication parameters
- Subindex 03<sub>h</sub>, application parameters

*Object description*

Index	1011 <sub>h</sub>
Object name	Restore Default Parameters
Object code	ARRAY
Data type	Unsigned32

*Value description*

Subindex	00 <sub>h</sub> , number of elements
Meaning	Number of values for the object
Access	read-only
PDO mapping	–
Value range	–
Default value	3
Can be saved	–
Subindex	01 <sub>h</sub> , restore default of all parameters
Meaning	Resets all parameter values to the default setting
Access	read-write
PDO mapping	–
Value range	–
Default value	1
Can be saved	–
Subindex	02 <sub>h</sub> , restore default of communication parameters
Meaning	Resets communication parameter values to default
Access	read-write
PDO mapping	–
Value range	–
Default value	1
Can be saved	–
Subindex	03 <sub>h</sub> , restore default of application parameters

Meaning	Resets application parameter values to default
Access	read-write
PDO mapping	–
Value range	–
Default value	1
Can be saved	–

### 8.5.12 1014<sub>h</sub> COB ID-Emergency message

The object specifies the COB ID of the emergency object "EMCY".

#### Object description

Index	1014 <sub>h</sub>
Object name	COB ID EMCY
Object code	VAR
Data type	Unsigned32

#### Value description

Subindex	00 <sub>h</sub> , COB ID EMCY
Meaning	Identifier of the emergency object
Access	read-write
PDO mapping	–
Value range	0...4294967295
Default value	4000 0080 <sub>h</sub> + node ID
Can be saved	Yes

#### Bit coding, subindex 00<sub>h</sub>

bit	Access	Value	Meaning
31, 30	ro	0 <sub>b</sub>	Reserved
29	ro	0 <sub>b</sub>	0: 11 bit identifier (CAN 3.0A) 1: 29 bit identifier (CAN 3.0B)
28-11	ro	0000 <sub>h</sub>	Only relevant if bit 29=1 is not used by the device.
10-7	rw	0001 <sub>b</sub>	Function code, bits 10-7 of the COB ID
6-0	ro	–	Node address, bits 6-0, of the COB ID

The COB ID can be changed in the NMT state "Pre-Operational".

8.5.13 1015<sub>h</sub> Inhibit time emergency message

The object specifies the waiting time for the repeated transmission of EMCY messages as a multiple of 100µs.

*Object description*

Index	1015 <sub>h</sub>
Object name	Inhibit time EMCY
Object code	VAR
Data type	Unsigned16

*Value description*

Subindex	00 <sub>h</sub> , inhibit time EMCY
Meaning	Waiting time for repeated transmission of an EMCY
Access	read-write
PDO mapping	–
Value range	0...65535
Default value	0
Can be saved	Yes

8.5.14 1016<sub>h</sub> Consumer Heartbeat Time

The object contains the settings of the "Heartbeat Consumers" for NMT monitoring by means of "Heartbeat" connection message.

*Object description*

Index	1016 <sub>h</sub>
Object name	Consumer Heartbeat Time
Object code	ARRAY
Data type	Unsigned32

*Value description*

Subindex	00 <sub>h</sub> , number of elements
Meaning	Number of values for the object
Access	read-only
PDO mapping	–
Value range	–
Default value	3
Can be saved	–
Subindex	01 <sub>h</sub> , Consumer Heartbeat Time
Meaning	Time interval and node ID of the "Heartbeat" recipient
Access	read-write
PDO mapping	–
Value range	0...4294967295
Default value	0
Can be saved	Yes



Bit coding subindex  $01_h \dots 03_h$

bit	Meaning
31 ... 24	Reserved
23 ... 16	Node ID
15 ... 0	Time interval for "Heartbeat" message

The time interval is specified as a multiple of 1 ms and must be greater than the producer "heartbeat" time, object `Producer Heartbeat Time` ( $1017_h$ ). If the time interval is zero, the device specified via the node ID is not monitored.

### 8.5.15 $1017_h$ Producer Heartbeat Time

The object contains the time interval of the "Heartbeat" producer for NMT monitoring by means of "Heartbeat" connection message as a multiple of 1 ms.

The producer "Heartbeat" time must be less than the time interval of the "Heartbeat" consumer, object `Consumer Heartbeat Time` ( $1016_h$ ). A time interval of zero deactivates monitoring.

#### Object description

Index	$1017_h$
Object name	Producer Heartbeat Time
Object code	VAR
Data type	Unsigned16

#### Value description

Subindex	$00_h$ , Producer Heartbeat Time
Meaning	Time interval for producer "Heartbeat"
Access	read-write
PDO mapping	–
Value range	0...65535
Default value	0
Can be saved	Yes

8.5.16 1018<sub>h</sub> Identity Object

The object provides information on the product.

- Subindex 01<sub>h</sub> (vendor ID) contains the manufacturer ID
- Subindex 02<sub>h</sub> (product ID) contains the manufacturer-specific product code
- Subindex 03<sub>h</sub> (revision number) identifies special CANopen properties for the device
- Subindex 04<sub>h</sub> (serial number) contains the serial number

*Object description*

Index	1018 <sub>h</sub>
Object name	Identity Object
Object code	RECORD
Data type	Identity

*Value description*

Subindex	00 <sub>h</sub> , number of elements
Meaning	Number of values for the object
Access	read-only
PDO mapping	–
Value range	–
Default value	4
Can be saved	–
Subindex	01 <sub>h</sub> , vendor ID
Meaning	Vendor ID
Access	read-only
PDO mapping	–
Value range	–
Default value	0800 005A <sub>h</sub>
Can be saved	–
Subindex	02 <sub>h</sub> , product code
Meaning	Product code
Access	read-only
PDO mapping	–
Value range	–
Default value	163DCH <sub>h</sub>
Can be saved	–
Subindex	03 <sub>h</sub> , revision number
Meaning	Revision number
Access	read-only
PDO mapping	–
Value range	–
Default value	1

Can be saved	–
Subindex	04 <sub>h</sub> , serial number
Meaning	Serial number
Access	read-only
PDO mapping	–
Value range	–
Default value	0
Can be saved	–

### 8.5.17 1020<sub>h</sub> data on configuration

The object is used to verify the configuration.

- Subindex 01<sub>h</sub>, date of configuration
- Subindex 02<sub>h</sub>, time of configuration

#### Object description

Index	1020 <sub>h</sub>
Object name	
Object code	RECORD
Data type	Identity

#### Value description

Subindex	00 <sub>h</sub> , verify configuration
Meaning	Checks configuration data
Access	read-only
PDO mapping	–
Value range	–
Default value	2
Can be saved	–

Subindex	01 <sub>h</sub> , configuration date
Meaning	Date of configuration
Access	read-write
PDO mapping	–
Value range	–
Default value	
Can be saved	Yes

Subindex	02 <sub>h</sub> , configuration time
Meaning	Time of configuration
Access	read-write
PDO mapping	–
Value range	–
Default value	
Can be saved	Yes

8.5.18 1029<sub>h</sub> error behavior

The object specifies the behavior of the NMT state machine in the event of a communication error.

*Object description*

Index	1029 <sub>h</sub>
Object name	Error behavior
Object code	ARRAY
Data type	Unsigned8

*Value description*

Subindex	00 <sub>h</sub> , number of elements
Meaning	Number of values for the object
Access	read-only
PDO mapping	–
Value range	–
Default value	1
Can be saved	–

Subindex	01 <sub>h</sub> , Communication Error
Meaning	Communication errors
Access	read-write
PDO mapping	–
Value range	0...2
Default value	0
Can be saved	Yes

*Settings, subindex 01<sub>h</sub>*

Value	Meaning
0	Pre-operational (with operational state only)
1	No state transition
2	stopped

8.5.19 1200<sub>h</sub> 1st server SDO parameter

The object contains the settings for the first server SDO.

*Object description*

Index	1200 <sub>h</sub>
Object name	1st server SDO parameter
Object code	RECORD
Data type	SDO server parameter

*Value description*

Subindex	00 <sub>h</sub> , number of elements
Meaning	Number of values for the object
Access	read-only
PDO mapping	–
Value range	–
Default value	2
Can be saved	–

Subindex	01 <sub>h</sub> , COB ID client -> server
Meaning	Identifier client -> server
Access	read-only
PDO mapping	–
Value range	0...4294967295
Default value	1536 + node ID
Can be saved	Yes

Subindex	02 <sub>h</sub> , COB ID server -> client
Meaning	Identifier server -> client
Access	read-only
PDO mapping	–
Value range	0...4294967295
Default value	1408 + node ID
Can be saved	Yes

8.5.20 1201<sub>h</sub> 2nd server SDO parameter

The object contains the settings for the second server SDO.

*Object description*

Index	1201 <sub>h</sub>
Object name	2nd server SDO parameter
Object code	RECORD
Data type	SDO server parameter

*Value description*

Subindex	00 <sub>h</sub> , number of elements
Meaning	Number of values for the object
Access	read-only
PDO mapping	–
Value range	–
Default value	3
Can be saved	–
Subindex	01 <sub>h</sub> , COB ID client -> server
Meaning	Identifier client -> server
Access	read-write
PDO mapping	–
Value range	0...4294967295
Default value	8000 0000 <sub>h</sub>
Can be saved	Yes
Subindex	02 <sub>h</sub> , COB ID server -> client
Meaning	Identifier server -> client
Access	read-write
PDO mapping	–
Value range	0...4294967295
Default value	8000 0000 <sub>h</sub>
Can be saved	Yes
Subindex	03 <sub>h</sub> , node ID SDO client
Meaning	Node ID SDO client
Access	read-write
PDO mapping	–
Value range	1...127
Default value	–
Can be saved	Yes

8.5.21 1400<sub>h</sub> 1st receive PDO parameter

The object contains the settings for the first receive PDO R\_PDO1.

*Object description*

Index	1400 <sub>h</sub>
Object name	1st receive PDO parameter
Object code	RECORD
Data type	PDO Communication Parameter

*Value description*

Subindex	00 <sub>h</sub> , number of entries
Meaning	Number of values for the object
Access	read-only
PDO mapping	–
Value range	–
Default value	2
Can be saved	–

Subindex	01 <sub>h</sub> , COB ID used by PDO
Meaning	Identifier of the R_PDO1
Access	read-write
PDO mapping	–
Value range	0...4294967295
Default value	0200 <sub>h</sub> + node ID
Can be saved	Yes

Subindex	02 <sub>h</sub> , transmission type = asynchronous
Meaning	Transmission type
Access	read-write
PDO mapping	–
Value range	0...255
Default value	255
Can be saved	Yes

*Bit assignment subindex 01<sub>h</sub>*

bit	Access	Value	Meaning
31	rw	0 <sub>b</sub>	0: PDO is enabled 1: PDO is disabled
30	ro	0 <sub>b</sub>	0: RTR (see below) is possible 1: RTR not permitted
29	ro	0 <sub>b</sub>	0: 11 bit identifier (CAN 3.0A) 1: 29 bit identifier (CAN 3.0B)
28-11	ro	0000 <sub>h</sub>	Only relevant if bit 29=1 is not used by the device.
10-7	rw	0100 <sub>b</sub>	Function code, bits 10-7 of the COB ID
6-0	ro	–	Node address, bits 6-0, of the COB ID

**Bit 31** A R\_PDO can only be used if bit 31="0".

**Bit 30: RTR Bit** If a device supports R\_PDOs with RTR (remote transmission request), it can request a PDO from a PDO producer with RTR = "0" in accordance with the producer-consumer relationship.

The device cannot request PDOs, but it can respond to the request for a PDO, see RTR bit for T\_PDO1 settings (1800<sub>h</sub>).

**Bit coding, subindex 02<sub>h</sub>** The type of control for evaluating R\_PDO data is specified via subindex 02<sub>h</sub>. The values 241 ... 251 are reserved.

Transmission type	cyclic	acyclic	synchronous	asynchronous	RTR-controlled
0	–	X	X	–	–
1-240	X	–	X	–	–
252	–	–	X	–	X
253	–	–	–	X	X
254	–	–	–	X	–
255	–	–	–	X	–

If an R\_PDO is transmitted synchronously (transmission type=0 ... 252), the device evaluates the received data depending on the SYNC object.

- In the case of acyclic transmission (transmission type=0), the evaluation depends on the SYNC object, but not the transmission of the PDO. A received PDO message is evaluated with the following SYNC.

A value between 1 and 240 specifies the number of SYNC cycles after which a received PDO is evaluated.

The values 252 to 254 are relevant for updating T\_PDOs, but not for sending them.

- 252: Updating of transmit data with receipt of the next SYNC
- 253: Updating of transmit data with receipt of a request from a PDO consumer
- 254: Updating of data in an event-driven way, the triggering event is specified in a manufacturer-specific way

R\_PDOs with the value 255 are updated immediately upon receipt of the PDOs. The triggering event is the data that is transmitted corresponding to the definition of the DSP402 device profile in the PDO.

**Settings** R\_PDO1 is processed asynchronously and in an event-driven way.

The byte assignment of the R\_PDO1 is specified via PDO mapping with the object `1st receive PDO mapping` (1600<sub>h</sub>). The following default assignment is used for R\_PDO1:

- Bytes 0 ... 1: Control word `controlword` (6040<sub>h</sub>).

The COB ID of the object can be changed in the NMT state "Pre-Operational".



8.5.22 1401<sub>h</sub> 2nd receive PDO\_parameter

The object contains settings for the second receive PDO R\_PDO2.

*Object description*

Index	1401 <sub>h</sub>
Object name	2nd receive PDO parameter
Object code	RECORD
Data type	PDO Communication Parameter

*Value description*

Subindex	00 <sub>h</sub> , largest subindex supported
Meaning	Largest subindex supported
Access	read-only
PDO mapping	–
Value range	–
Default value	2
Can be saved	–

Subindex	01 <sub>h</sub> , COB ID R_PDO2
Meaning	Identifier of the R_PDO2
Access	read-write
PDO mapping	–
Value range	0 ... 4294967295
Default value	8000 0300 <sub>h</sub> + node ID
Can be saved	Yes

Subindex	02 <sub>h</sub> , transmission type
Meaning	Transmission type
Access	read-write
PDO mapping	–
Value range	0 ... 255
Default value	255
Can be saved	Yes

The meaning of the bit states and subindex values is described with the object 1st receive PDO parameters (1400<sub>h</sub>).

*Settings* R\_PDO2 is processed synchronously, acyclically and in an event-driven way and must be activated with bit 31=1 in subindex 01<sub>h</sub> before it can be used.

The byte assignment of R\_PDO2 is specified via PDO mapping with the object 2nd Receive PDO mapping (1601<sub>h</sub>). The following default assignment is set for the operating mode "Profile Position":

- Bytes 0 ... 1: Control word `controlword` (6040<sub>h</sub>)
- Bytes 2 ... 5: Target position of the motion command `target position` (607A<sub>h</sub>)

The COB ID of the object can be changed in the NMT state "Pre-Operational".

The transmission type for the receive PDO can have 3 value ranges:

0	For an asynchronous cycle
1 to 240	Instructs the receive PDO to become active only if a SYNC object is received
255	Specifies that the PDO is executed when it is received

8.5.23 1402<sub>h</sub> 3rd receive PDO-Parameter

The object contains settings for the third receive PDO R\_PDO3.

*Object description*

Index	1402 <sub>h</sub>
Object name	3rd receive PDO parameter
Object code	RECORD
Data type	PDO Communication Parameter

*Value description*

Subindex	00 <sub>h</sub> , largest subindex supported
Meaning	Largest subindex supported
Access	read-only
PDO mapping	–
Value range	–
Default value	2
Can be saved	–

Subindex	01 <sub>h</sub> , COB ID used by PDO
Meaning	Identifier of the R_PDO3
Access	read-write
PDO mapping	–
Value range	0 ... 4294967295
Default value	8000 0400 <sub>h</sub> + node ID
Can be saved	Yes

Subindex	02 <sub>h</sub> , transmission type
Meaning	Transmission type
Access	read-write
PDO mapping	–
Value range	0...255
Default value	255
Can be saved	Yes

The meaning of the bit states and subindex values is described with the object 1st receive PDO-parameters (1400<sub>h</sub>).

*Settings*

R\_PDO3 is processed synchronously, acyclically and in an event-driven way and must be activated with bit 31=1 in subindex 01<sub>h</sub> before it can be used.

The byte assignment of the R\_PDO3 is specified via PDO mapping with the object 3rd Receive PDO mapping (1602<sub>h</sub>). The following default assignment is set for the operating mode "Profile Velocity":

- Bytes 0 ... 1: Control word `controlword` (6040<sub>h</sub>)
- Bytes 2 ... 5: Reference velocity of motion command `Target velocity` (60FF<sub>h</sub>)

The COB ID of the object can be changed in the NMT state "Pre-Operational".

The transmission type for the receive PDO can have 3 value ranges:

0	For an asynchronous cycle
1 to 240	Instructs the receive PDO to become active only if a SYNC object is received
255	Specifies that the PDO is executed when it is received

### 8.5.24 1403<sub>h</sub> 4th receive PDO parameter

The object stores settings for the fourth receive PDO R\_PDO4.

#### Object description

Index	1403 <sub>h</sub>
Object name	4th receive PDO parameter
Object code	RECORD
Data type	PDO Communication Parameter

#### Value description

Subindex	00 <sub>h</sub> , largest subindex supported
Meaning	Largest subindex supported
Access	read-only
PDO mapping	–
Value range	–
Default value	2
Can be saved	–
Subindex	01 <sub>h</sub> , COB ID used by PDO
Meaning	Identifier of the R_PDO4
Access	read-write
PDO mapping	–
Value range	0...4294967295
Default value	8000 0500 <sub>h</sub> + node ID
Can be saved	Yes
Subindex	02 <sub>h</sub> , transmission type
Meaning	Transmission type
Access	read-only
PDO mapping	–
Value range	–
Default value	254
Can be saved	Yes

The meaning of the bit states and subindex values is described under object 1st receive PDO-parameters (1400<sub>h</sub>).

#### PDO settings

R\_PDO4 is processed asynchronously and in an event-driven way and must be activated with bit 31=1 in subindex 01<sub>h</sub> before it can be used.

The COB ID of the object can be changed in the NMT state "Pre-Operational".

8.5.25 1600<sub>h</sub> 1st receive PDO mapping

The object specifies the objects mapped in R\_PDO1 and transmitted with the PDO. When the object is read, subindex 00<sub>h</sub>, the number of mapped objects is read.

*Object description*

Index	1600 <sub>h</sub>
Object name	1st receive PDO mapping
Object code	RECORD
Data type	PDO mapping

*Value description*

Subindex	00 <sub>h</sub> , number of mapped objects
Meaning	Number of values for the object
Access	read-only
PDO mapping	–
Value range	1...8
Default value	1
Can be saved	–

Subindex	01 <sub>h</sub> , CMD: Control word
Meaning	First object for mapping in R_PDO1
Access	read-write
PDO mapping	–
Value range	0...4294967295
Default value	6040 0010 <sub>h</sub>
Can be saved	–

*Bit coding starting at subindex 01<sub>h</sub>*

Each subindex entry from subindex 01<sub>h</sub> on specifies the object and the bit length of the object. The object is identified via the index and the subindex, which refer to the object dictionary of the device.

bit	Meaning
31 ... 16	Index
15 ... 8	Subindex
7 ... 0	Object length in bit

*Settings*

The following default assignment is used:

- Subindex 01<sub>h</sub>: PDO mapping of the control word, object controlword (6040<sub>h</sub>).

8.5.26 1601<sub>h</sub> 2nd receive PDO mapping

The object specifies the objects mapped in R\_PDO2 and transmitted with the PDO. When the object is read, subindex 00<sub>h</sub>, the number of mapped objects is read.

*Object description*

Index	1601 <sub>h</sub>
Object name	2nd receive PDO mapping
Object code	RECORD
Data type	PDO mapping

*Value description*

Subindex	00 <sub>h</sub> , number of mapped application objects in PDO
Meaning	Number of values for the object
Access	read-write
PDO mapping	–
Value range	1...8
Default value	2
Can be saved	–

Subindex	01 <sub>h</sub> , PDO mapping for the first application object to be mapped (control word)
Meaning	First object for mapping in R_PDO2
Access	read-write
PDO mapping	–
Value range	0...4294967295
Default value	6040 0010 <sub>h</sub>
Can be saved	–

Subindex	02 <sub>h</sub> , PDO mapping for the second application object to be mapped (target position)
Meaning	Second object for mapping in R_PDO2
Access	read-write
PDO mapping	–
Value range	0...4294967295
Default value	607A 0020 <sub>h</sub>
Can be saved	–

The meaning of the bit states is described with the object 1st receive PDO-mapping (1600<sub>h</sub>).

*Settings*

The following default assignment is set for the operating mode "Profile Position":

- Subindex 01<sub>h</sub>: PDO mapping of the control word, object controlword (6040<sub>h</sub>).
- Subindex 02<sub>h</sub>: target position of the motion command, object target position (607A<sub>h</sub>).

8.5.27 1602<sub>h</sub> 3rd receive PDO mapping

The object specifies the objects mapped in R\_PDO3 and transmitted with the PDO. When the object is read, subindex 00<sub>h</sub>, the number of mapped objects is read.

*Object description*

Index	1602 <sub>h</sub>
Object name	3rd receive PDO mapping
Object code	RECORD
Data type	PDO mapping

*Value description*

Subindex	00 <sub>h</sub> , number of mapped application objects in PDO
Meaning	Number of values for the object
Access	read-write
PDO mapping	–
Value range	1...8
Default value	2
Can be saved	–

Subindex	01 <sub>h</sub> , PDO mapping for the first application object to be mapped (control word)
Meaning	First object for mapping in R_PDO3
Access	read-write
PDO mapping	–
Value range	0...4294967295
Default value	6040 0010 <sub>h</sub>
Can be saved	–

Subindex	02 <sub>h</sub> , PDO mapping for the second application object to be mapped (target velocity)
Meaning	Second object for mapping in R_PDO3
Access	read-write
PDO mapping	–
Value range	0...4294967295
Default value	60FF 0020 <sub>h</sub>
Can be saved	–

The meaning of the bit states is described with the object 1st receive PDO-mapping (1600<sub>h</sub>).

*Settings*

The following defaults are set for the operating mode "Profile Velocity":

- Subindex 01<sub>h</sub>: PDO mapping of the control word, object controlword (6040<sub>h</sub>).
- Bytes 2 ... 5: Reference velocity of motion command Target velocity (60FF<sub>h</sub>)

**8.5.28 1603<sub>h</sub> 4th receive PDO mapping**

The object specifies the objects mapped in R\_PDO4 and transmitted with the PDO. When the object is read, subindex 00<sub>h</sub>, the number of mapped objects is read.

*Object description*

---

Index	1603 <sub>h</sub>
Object name	4th receive PDO mapping
Object code	RECORD
Data type	PDO mapping

---

*Value description*

---

Subindex	00 <sub>h</sub> , number of elements
Meaning	Number of values for the object
Access	read-write
PDO mapping	–
Value range	0...4
Default value	0
Can be saved	Yes

---

The meaning of the bit states is described with the object 1st receive PDO mapping (1600<sub>h</sub>).

*Settings*

The PDO assignment for R\_PDO4 can be modified.



8.5.29 1800<sub>h</sub> 1st transmit PDO parameter

The object contains settings for the first transmit PDO T\_PDO1.

*Object description*

Index	1800 <sub>h</sub>
Object name	1st transmit PDO parameter
Object code	RECORD
Data type	PDO Communication Parameter

*Value description*

Subindex	00 <sub>h</sub> , number of entries
Meaning	Number of values for the object
Access	read-only
PDO mapping	–
Value range	–
Default value	5
Can be saved	–
Subindex	01 <sub>h</sub> , COB ID used by PDO
Meaning	Identifier of the T_PDO1
Access	read-write
PDO mapping	–
Value range	0...4294967295
Default value	4000 0180 <sub>h</sub> + node ID
Can be saved	Yes
Subindex	02 <sub>h</sub> , transmission type = asynchronous
Meaning	Transmission type
Access	read-write
PDO mapping	–
Value range	0...255
Default value	255
Can be saved	Yes
Subindex	03 <sub>h</sub> , inhibit time
Meaning	Inhibit time for locking bus access (1=100μs)
Access	read-write
PDO mapping	–
Value range	0...65535
Default value	0
Can be saved	Yes
Subindex	04 <sub>h</sub> , reserved
Meaning	Reserved
Access	–
PDO mapping	–

---

Value range	0...255
-------------	---------

Default value	–
---------------	---

Can be saved	–
--------------	---

---

Subindex	05 <sub>h</sub> , event timer
----------	-------------------------------

Meaning	Time span for event triggering (1=1 ms)
---------	---

Access	read-write
--------	------------

PDO mapping	–
-------------	---

Value range	0...65535
-------------	-----------

Default value	0
---------------	---

Can be saved	Yes
--------------	-----

---

The meaning of the bit states and subindex values is described with the object `1st receive PDO-parameters` (1400<sub>h</sub>).

*Settings* T\_PDO1 is transmitted asynchronously and in an event-driven way whenever the PDO data changes.

The byte assignment of the T\_PDO1 is specified via PDO mapping with the object `1st transmit PDO mapping` (1A00<sub>h</sub>). The following default assignment is used:

- Bytes 0 ... 1: Status word `statusword` (6041<sub>h</sub>).

The COB ID of the object can be changed in the NMT state "Pre-Operational".

8.5.30 1801<sub>h</sub> 2nd transmit PDO parameter

The object contains settings for the second transmit PDO T\_PDO2.

*Object description*

Index	1801 <sub>h</sub>
Object name	2nd transmit PDO parameter
Object code	RECORD
Data type	PDO Communication Parameter

*Value description*

Subindex	00 <sub>h</sub> , largest subindex supported
Meaning	Largest subindex supported
Access	read-only
PDO mapping	–
Value range	–
Default value	5
Can be saved	–

Subindex	01 <sub>h</sub> , COB ID used by PDO
Meaning	Identifier of the T_PDO2
Access	read-write
PDO mapping	–
Value range	0...4294967295
Default value	C000 0280 <sub>h</sub> + node ID
Can be saved	Yes

Subindex	02 <sub>h</sub> , transmission type
Meaning	Transmission type
Access	read-write
PDO mapping	–
Value range	0...255
Default value	255
Can be saved	Yes

Subindex	03 <sub>h</sub> , inhibit time
Meaning	Inhibit time for locking bus access (1=100µs)
Access	read-write
PDO mapping	–
Value range	0...65535
Default value	0
Can be saved	Yes

Subindex	04 <sub>h</sub> , reserved
Meaning	Reserved
Access	–
PDO mapping	–

---

Value range	0...255
-------------	---------

Default value	–
---------------	---

Can be saved	–
--------------	---

---

Subindex	05 <sub>h</sub> , event timer
----------	-------------------------------

Meaning	Time span for event triggering (1=1 ms)
---------	---

Access	read-write
--------	------------

PDO mapping	–
-------------	---

Value range	0...65535
-------------	-----------

Default value	100
---------------	-----

Can be saved	Yes
--------------	-----

---

The meaning of the bit states and subindex values is described with the object `1st receive PDO-parameters` (1400<sub>h</sub>).

*Settings* T\_PDO2 is transmitted synchronously and acyclically.

The byte assignment of the T\_PDO2 is specified via PDO mapping with the object `2nd transmit PDO mapping` (1A01<sub>h</sub>). The following default assignment is set for the operating mode "Profile Position":

- Bytes 0 ... 1: Status word `statusword` (6041<sub>h</sub>)
- Bytes 2 ... 5: Current position `position actual value` (6064<sub>h</sub>).

The COB ID of the object can be changed in the NMT state "Pre-Operational".

8.5.31 1802<sub>h</sub> 3rd transmit PDO parameter

The object contains settings for the third transmit PDO T\_PDO3.

*Object description*

Index	1802 <sub>h</sub>
Object name	3rd transmit PDO parameter
Object code	RECORD
Data type	PDO Communication Parameter

*Value description*

Subindex	00 <sub>h</sub> , largest subindex supported
Meaning	Largest subindex supported
Access	read-only
PDO mapping	–
Value range	–
Default value	5
Can be saved	–

Subindex	01 <sub>h</sub> , COB ID used by PDO
Meaning	Identifier of the T_PDO3
Access	read-write
PDO mapping	–
Value range	0...4294967295
Default value	C000 0380 <sub>h</sub> + node ID
Can be saved	Yes

Subindex	02 <sub>h</sub> , transmission type
Meaning	Transmission type
Access	read-write
PDO mapping	–
Value range	0...255
Default value	255
Can be saved	Yes

Subindex	03 <sub>h</sub> , inhibit time
Meaning	Inhibit time for locking bus access (1=100µs)
Access	read-write
PDO mapping	–
Value range	0...65535
Default value	0
Can be saved	Yes

Subindex	04 <sub>h</sub> , reserved
Meaning	Reserved
Access	–
PDO mapping	–

---

Value range	0...255
-------------	---------

Default value	–
---------------	---

Can be saved	–
--------------	---

---

Subindex	05 <sub>h</sub> , event timer
----------	-------------------------------

Meaning	Time span for event triggering (1=1 ms)
---------	---

Access	read-write
--------	------------

PDO mapping	–
-------------	---

Value range	0...65535
-------------	-----------

Default value	100
---------------	-----

Can be saved	Yes
--------------	-----

---

The meaning of the bit states and subindex values is described with the object 1st receive PDO-parameters (1400<sub>h</sub>).

*Settings* T\_PDO3 is transmitted synchronously and acyclically.

The byte assignment of the T\_PDO3 is specified via PDO mapping with the object 3rd transmit PDO mapping (1A02<sub>h</sub>). The following default assignment is set for the operating mode "Profile Position":

- Bytes 0 ... 1: Status word `statusword` (6041<sub>h</sub>)
- Bytes 2 ... 5: Current velocity `velocity actual value` (606C<sub>h</sub>).

The COB ID of the object can be changed in the NMT state "Pre-Operational".

8.5.32 1803<sub>h</sub> 4th transmit PDO parameter

The object contains settings for the fourth transmit PDO T\_PDO4.

*Object description*

Index	1803 <sub>h</sub>
Object name	4th transmit PDO parameter
Object code	RECORD
Data type	PDO Communication Parameter

*Value description*

Subindex	00 <sub>h</sub> , largest subindex supported
Meaning	Largest subindex supported
Access	read-only
PDO mapping	–
Value range	–
Default value	5
Can be saved	–

Subindex	01 <sub>h</sub> , COB ID used by PDO
Meaning	Identifier of the T_PDO4
Access	read-write
PDO mapping	–
Value range	0...4294967295
Default value	C000 0480 <sub>h</sub> + node ID
Can be saved	Yes

Subindex	02 <sub>h</sub> , transmission type
Meaning	Transmission type
Access	read-only
PDO mapping	–
Value range	0...255
Default value	254
Can be saved	Yes

Subindex	03 <sub>h</sub> , inhibit time
Meaning	Inhibit time for locking bus access (1=100µs)
Access	read-write
PDO mapping	–
Value range	0...65535
Default value	0
Can be saved	Yes

Subindex	04 <sub>h</sub> , reserved
Meaning	Reserved
Access	–
PDO mapping	–

---

Value range	0...255
-------------	---------

Default value	–
---------------	---

Can be saved	–
--------------	---

---

Subindex	05 <sub>h</sub> , event timer
----------	-------------------------------

Meaning	Time span for event triggering (1=1 ms)
---------	---

Access	read-write
--------	------------

PDO mapping	–
-------------	---

Value range	0...65535
-------------	-----------

Default value	0
---------------	---

Can be saved	Yes
--------------	-----

---

The meaning of the bit states and subindex values is described with the object `1st receive PDO-parameters` (1400<sub>h</sub>).

*Settings* R\_PDO4 is transmitted asynchronously and in an event-driven way.

The COB ID of the object can be changed in the NMT state "Pre-Operational".



8.5.33 1A00<sub>h</sub> 1st transmit PDO mapping

The object specifies the objects mapped in T\_PDO1 and transmitted with the PDO. When the object is read, subindex 00<sub>h</sub>, the number of mapped objects is read.

*Object description*

Index	1A00 <sub>h</sub>
Object name	1st transmit PDO mapping
Object code	RECORD
Data type	PDO mapping

*Value description*

Subindex	00 <sub>h</sub> , number of mapped objects
Meaning	Number of values for the object
Access	read-write
PDO mapping	–
Value range	1...8
Default value	1
Can be saved	–

Subindex	01 <sub>h</sub> , ETA: status word
Meaning	First object for the mapping in T_PDO1
Access	read-write
PDO mapping	–
Value range	0...4294967295
Default value	6041 0010 <sub>h</sub>
Can be saved	–

The meaning of the bit states is described with the object `1st receive PDO mapping` (1600<sub>h</sub>).

*Settings*

The PDO assignment for T\_PDO1 can be modified. The following default assignment is used:

- Subindex 1: PDO mapping of the status word, object `statusword` (6041<sub>h</sub>)

8.5.34 1A01<sub>h</sub> 2nd transmit PDO mapping

The object specifies the objects mapped in T\_PDO2 and transmitted with the PDO. When the object is read, subindex 00<sub>h</sub>, the number of mapped objects is read.

*Object description*

Index	1A01 <sub>h</sub>
Object name	2nd transmit PDO mapping
Object code	RECORD
Data type	PDO mapping

*Value description*

Subindex	00 <sub>h</sub> , number of mapped application objects in PDO
Meaning	Number of values for the object
Access	read-write
PDO mapping	–
Value range	1...8
Default value	2
Can be saved	–

Subindex	01 <sub>h</sub> , PDO mapping for the first application object to be mapped (status word)
Meaning	First object for the mapping in T_PDO2
Access	read-write
PDO mapping	–
Value range	0...4294967295
Default value	6041 0010 <sub>h</sub>
Can be saved	–

Subindex	02 <sub>h</sub> , PDO mapping for the second application object to be mapped (actual position)
Meaning	Second object for the mapping in T_PDO2
Access	read-write
PDO mapping	–
Value range	0...4294967295
Default value	6064 0020 <sub>h</sub>
Can be saved	–

The meaning of the bit states is described with the object 1st receive PDO-mapping (1600<sub>h</sub>).

*Settings*

The PDO assignment for T\_PDO2 can be modified. The following default assignment is set for the operating mode "Profile Position":

- Subindex 1: PDO mapping of the status word, object `statusword` (6041<sub>h</sub>)
- Subindex 2: PDO mapping of the current position, object `position actual value` (6064<sub>h</sub>).

8.5.35 1A02<sub>h</sub> 3rd transmit PDO mapping

The object specifies the objects mapped in T\_PDO3 and transmitted with the PDO. When the object is read, subindex 00<sub>h</sub>, the number of mapped objects is read.

*Object description*

Index	1A02 <sub>h</sub>
Object name	3rd transmit PDO mapping
Object code	RECORD
Data type	PDO mapping

*Value description*

Subindex	00 <sub>h</sub> , number of mapped application objects in PDO
Meaning	Number of values for the object
Access	read-write
PDO mapping	–
Value range	1...8
Default value	2
Can be saved	–

Subindex	01 <sub>h</sub> , PDO mapping for the first application object to be mapped (status word)
Meaning	First object for the mapping in T_PDO3
Access	read-write
PDO mapping	–
Value range	0...4294967295
Default value	6041 0010 <sub>h</sub>
Can be saved	–

Subindex	02 <sub>h</sub> , PDO mapping for the second application object to be mapped (actual velocity)
Meaning	Second object for the mapping in T_PDO3
Access	read-write
PDO mapping	–
Value range	0...4294967295
Default value	606C 0020 <sub>h</sub>
Can be saved	–

The meaning of the bit states is described with the object 1st receive PDO-mapping (1600<sub>h</sub>).

*Settings*

The PDO assignment for T\_PDO3 can be modified. The following defaults are set for the operating mode "Profile Velocity":

- Bytes 0 ... 1: Status word `statusword` (6041<sub>h</sub>)
- Bytes 2 ... 5: Current velocity `velocity actual value` (606C<sub>h</sub>).

8.5.36 1A03<sub>h</sub> 4th transmit PDO mapping

The object specifies the objects mapped in T\_PDO4 and transmitted with the PDO. When the object is read, subindex 00<sub>h</sub>, the number of mapped objects is read.

*Object description*

Index	1A03 <sub>h</sub>
Object name	4th transmit PDO mapping
Object code	RECORD
Data type	PDO mapping

*Value description*

Subindex	00 <sub>h</sub> , number of elements
Meaning	Number of values for the object
Access	read-write
PDO mapping	–
Value range	0...4
Default value	0
Can be saved	Yes

The meaning of the bit states is described under object 1st receive PDO mapping (1600<sub>h</sub>) .

*Settings*

The PDO assignment for T\_PDO4 can be modified.

## 9 Accessories and spare parts

# 9

### 9.1 CANopen cable with connectors

Description	Order no.
CANopen cable, 0.3 m, 2 x RJ45	VW3CANCARR03
CANopen cable, 1 m, 2 x RJ45	VW3CANCARR1
2 m, 2 x RJ45, shielded twisted pair cable	490NTW00002
5 m, 2 x RJ45, shielded twisted pair cable	490NTW00005
12 m, 2 x RJ45, shielded twisted pair cable	490NTW00012
2 m, 2 x RJ45, shielded twisted pair cable with UL and CSA 22.1 certification	490NTW00002U
5 m, 2 x RJ45, shielded twisted pair cable with UL and CSA 22.1 certification	490NTW00005U
12 m, 2 x RJ45, shielded twisted pair cable with UL and CSA 22.1 certification	490NTW00012U
CANopen cable, 1 m, D9-SUB (female) to RJ45	TCSCCN4F3M1T
CANopen cable, 1 m, D9-SUB (female) with integrated terminating resistor to RJ45	VW3M3805R010
CANopen cable, 3 m, D9-SUB (female) with integrated terminating resistor to RJ45	VW3M3805R030
CANopen cable, 0.3 m, 2 x D9-SUB (female), LSZH standard cable (low-smoke, zero halogen, flame-retardant, tested as per IEC 60332-1)	TSXCANCADD03
CANopen cable, 1 m, 2 x D9-SUB (female), LSZH standard cable (low-smoke, zero halogen, flame-retardant, tested as per IEC 60332-1)	TSXCANCADD1
CANopen cable, 3 m, 2 x D9-SUB (female), LSZH standard cable (low-smoke, zero halogen, flame-retardant, tested as per IEC 60332-1)	TSXCANCADD3
CANopen cable, 5 m, 2 x D9-SUB (female), LSZH standard cable (low-smoke, zero halogen, flame-retardant, tested as per IEC 60332-1)	TSXCANCADD5
CANopen cable, 0.3 m, 2 x D9-SUB (female), flame-retardant, tested as per IEC 60332-2, UL certification	TSXCANCBDD03
CANopen cable, 1 m, 2 x D9-SUB (female), flame-retardant, tested as per IEC 60332-2, UL certification	TSXCANCBDD1
CANopen cable, 3 m, 2 x D9-SUB (female), flame-retardant, tested as per IEC 60332-2, UL certification	TSXCANCBDD3
CANopen cable, 5 m, 2 x D9-SUB (female), flame-retardant, tested as per IEC 60332-2, UL certification	TSXCANCBDD5

## 9.2 CANopen connectors, distributors, terminating resistors

Description	Order no.
CANopen terminating resistor, 120 Ohm, integrated in RJ45 connector	TCSCAR013M120
CANopen connector with PC interface, D9-SUB (female), with switchable terminating resistor and additional D9-SUB (male) to connect a PC to the bus, PC interface straight, bus cable angled 90°	TSXCANKCDF90TP
CANopen connector, D9-SUB (female), with switchable terminating resistor, angled 90°	TSXCANKCDF90T
CANopen connector, D9-SUB (female), with switchable terminating resistor, straight	TSXCANKCDF180T
Four-port tap, for connection of 4 drop lines to trunk line, 4 x D9-SUB (male) with switchable terminating resistor	TSXCANTDM4
Two-port tap for connection of 2 drop lines to trunk line, with additional commissioning interface, 3 x RJ45 (female), with switchable terminating resistor	VW3CANTAP2

## 9.3 CANopen cables

Cables with open cable ends are suitable for connection of D-SUB connectors. Observe the cable cross section and the connection cross section of the required connector.

Description	Order no.
CANopen cable, 50 m, [(2 x AWG 22) + (2 x AWG 24)], LSZH standard cable (low-smoke, zero halogen, flame-retardant, tested as per IEC 60332-1), both cable ends open	TSXCANCA50
CANopen cable, 100 m, [(2 x AWG 22) + (2 x AWG 24)], LSZH standard cable (low-smoke, zero halogen, flame-retardant, tested as per IEC 60332-1), both cable ends open	TSXCANCA100
CANopen cable, 300 m, [(2 x AWG 22) + (2 x AWG 24)], LSZH standard cable (low-smoke, zero halogen, flame-retardant, tested as per IEC 60332-1), both cable ends open	TSXCANCA300
CANopen cable, 50 m, [(2 x AWG 22) + (2 x AWG 24)], flame-retardant, tested as per IEC 60332-2, UL certification, both cable ends open	TSXCANCB50
CANopen cable, 100 m, [(2 x AWG 22) + (2 x AWG 24)], flame-retardant, tested as per IEC 60332-2, UL certification, both cable ends open	TSXCANCB100
CANopen cable, 300 m, [(2 x AWG 22) + (2 x AWG 24)], flame-retardant, tested as per IEC 60332-2, UL certification, both cable ends open	TSXCANCB300
CANopen cable, 50 m, [(2 x AWG 22) + (2 x AWG 24)], flexible LSZH HD standard cable (low-smoke, zero halogen, flame-retardant, tested as per IEC 60332-1), for heavy-duty or flexible installation, oil-resistant, both cable ends open	TSXCANCD50
CANopen cable, 100 m, [(2 x AWG 22) + (2 x AWG 24)], flexible LSZH HD standard cable (low-smoke, zero halogen, flame-retardant, tested as per IEC 60332-1), for heavy-duty or flexible installation, oil-resistant, both cable ends open	TSXCANCD100
CANopen cable, 300 m, [(2 x AWG 22) + (2 x AWG 24)], flexible LSZH HD standard cable (low-smoke, zero halogen, flame-retardant, tested as per IEC 60332-1), for heavy-duty or flexible installation, oil-resistant, both cable ends open	TSXCANCD300

## 9.4 Spare parts connectors, fans, cover plates

Description	Order no.
Connector kit LXM32M: 3 x AC power stage supply (230/400 V <sub>ac</sub> ), 1 x control supply, 3 x digital inputs/outputs (4-pin), 2 x motor (10 A / 24 A), 1 x holding brake	VW3M2203
Cover plate for module slot, spare part to replace broken/lost cover plates, 10 pieces	VW3M2405
Cooling fan kit 40 mm x 40 mm, plastic housing, with connection cable	VW3M2401
Cooling fan kit 60 mm x 60 mm, plastic housing, with connection cable	VW3M2402
Cooling fan kit 80 mm x 80 mm, plastic housing, with connection cable	VW3M2403





## 10 Glossary

# 10

### 10.1 Units and conversion tables

The value in the specified unit (left column) is calculated for the desired unit (top row) with the formula (in the field).

Example: conversion of 5 meters [m] to yards [yd]  
 $5 \text{ m} / 0.9144 = 5.468 \text{ yd}$

#### 10.1.1 Length

	in	ft	yd	m	cm	mm
in	-	/ 12	/ 36	* 0.0254	* 2.54	* 25.4
ft	* 12	-	/ 3	* 0.30479	* 30.479	* 304.79
yd	* 36	* 3	-	* 0.9144	* 91.44	* 914.4
m	/ 0.0254	/ 0.30479	/ 0.9144	-	* 100	* 1000
cm	/ 2.54	/ 30.479	/ 91.44	/ 100	-	* 10
mm	/ 25.4	/ 304.79	/ 914.4	/ 1000	/ 10	-

#### 10.1.2 Mass

	lb	oz	slug	kg	g
lb	-	* 16	* 0.03108095	* 0.4535924	* 453.5924
oz	/ 16	-	* $1.942559 \cdot 10^{-3}$	* 0.02834952	* 28.34952
slug	/ 0.03108095	/ $1.942559 \cdot 10^{-3}$	-	* 14.5939	* 14593.9
kg	/ 0.45359237	/ 0.02834952	/ 14.5939	-	* 1000
g	/ 453.59237	/ 28.34952	/ 14593.9	/ 1000	-

#### 10.1.3 Force

	lb	oz	p	dyne	N
lb	-	* 16	* 453.55358	* 444822.2	* 4.448222
oz	/ 16	-	* 28.349524	* 27801	* 0.27801
p	/ 453.55358	/ 28.349524	-	* 980.7	* $9.807 \cdot 10^{-3}$
dyne	/ 444822.2	/ 27801	/ 980.7	-	/ $100 \cdot 10^3$
N	/ 4.448222	/ 0.27801	/ $9.807 \cdot 10^{-3}$	* $100 \cdot 10^3$	-

#### 10.1.4 Power

	HP	W
HP	-	* 746
W	/ 746	-

## 10.1.5 Rotation

	min <sup>-1</sup> (RPM)	rad/s	deg./s
min <sup>-1</sup> (RPM)	-	* $\pi / 30$	* 6
rad/s	* $30 / \pi$	-	* 57.295
deg./s	/ 6	/ 57.295	-

## 10.1.6 Torque

	lb-in	lb-ft	oz-in	Nm	kp-m	kp-cm	dyne-cm
lb-in	-	/ 12	* 16	* 0.112985	* 0.011521	* 1.1521	* $1.129 \cdot 10^6$
lb-ft	* 12	-	* 192	* 1.355822	* 0.138255	* 13.8255	* $13.558 \cdot 10^6$
oz-in	/ 16	/ 192	-	* $7.0616 \cdot 10^{-3}$	* $720.07 \cdot 10^{-6}$	* $72.007 \cdot 10^{-3}$	* 70615.5
Nm	/ 0.112985	/ 1.355822	/ $7.0616 \cdot 10^{-3}$	-	* 0.101972	* 10.1972	* $10 \cdot 10^6$
kp-m	/ 0.011521	/ 0.138255	/ $720.07 \cdot 10^{-6}$	/ 0.101972	-	* 100	* $98.066 \cdot 10^6$
kp-cm	/ 1.1521	/ 13.8255	/ $72.007 \cdot 10^{-3}$	/ 10.1972	/ 100	-	* $0.9806 \cdot 10^6$
dyne-cm	/ $1.129 \cdot 10^6$	/ $13.558 \cdot 10^6$	/ 70615.5	/ $10 \cdot 10^6$	/ $98.066 \cdot 10^6$	/ $0.9806 \cdot 10^6$	-

## 10.1.7 Moment of inertia

	lb-in <sup>2</sup>	lb-ft <sup>2</sup>	kg-m <sup>2</sup>	kg-cm <sup>2</sup>	kp-cm-s <sup>2</sup>	oz-in <sup>2</sup>
lb-in <sup>2</sup>	-	/ 144	/ 3417.16	/ 0.341716	/ 335.109	* 16
lb-ft <sup>2</sup>	* 144	-	* 0.04214	* 421.4	* 0.429711	* 2304
kg-m <sup>2</sup>	* 3417.16	/ 0.04214	-	* $10 \cdot 10^3$	* 10.1972	* 54674
kg-cm <sup>2</sup>	* 0.341716	/ 421.4	/ $10 \cdot 10^3$	-	/ 980.665	* 5.46
kp-cm-s <sup>2</sup>	* 335.109	/ 0.429711	/ 10.1972	* 980.665	-	* 5361.74
oz-in <sup>2</sup>	/ 16	/ 2304	/ 54674	/ 5.46	/ 5361.74	-

## 10.1.8 Temperature

	°F	°C	K
°F	-	(°F - 32) * 5/9	(°F - 32) * 5/9 + 273.15
°C	°C * 9/5 + 32	-	°C + 273.15
K	(K - 273.15) * 9/5 + 32	K - 273.15	-

## 10.1.9 Conductor cross section

AWG	1	2	3	4	5	6	7	8	9	10	11	12	13
mm <sup>2</sup>	42.4	33.6	26.7	21.2	16.8	13.3	10.5	8.4	6.6	5.3	4.2	3.3	2.6

AWG	14	15	16	17	18	19	20	21	22	23	24	25	26
mm <sup>2</sup>	2.1	1.7	1.3	1.0	0.82	0.65	0.52	0.41	0.33	0.26	0.20	0.16	0.13

## 10.2 Terms and Abbreviations

See chapter 2.5 "Standards and terminology" for information on the pertinent standards on which many terms are based. Some terms and abbreviations may have specific meanings with regard to the standards.

<i>AC</i>	Alternating current
<i>CAN</i>	( <b>C</b> ontroller <b>A</b> rea <b>N</b> etwork), standardized open fieldbus as per ISO 11898, allows drives and other devices from different manufacturers to communicate.
<i>CANopen</i>	Device- and manufacturer-independent description language for communication via the CAN bus
<i>COB</i>	<b>C</b> ommunication <b>O</b> bject, transport unit in a CAN network.
<i>COB ID</i>	<b>C</b> ommunication <b>O</b> bject <b>I</b> Dentifier; uniquely identifies each communication object in a CAN network
<i>CiA</i>	<b>CAN in Automation</b> , CAN interest group, standardization group for CAN and CANopen.
<i>DC</i>	Direct current
<i>DS301</i>	Standardizes the CANopen communication profile
<i>DSP402</i>	Standardizes the CANopen device profile for drives
<i>DriveCom</i>	Specification of the DSP402 state machine was created in accordance with the DriveCom specification.
<i>E</i>	Encoder
<i>EDS</i>	( <b>E</b> lectronic <b>D</b> ata <b>S</b> heet); contains the specific properties of a product.
<i>EMC</i>	Electromagnetic compatibility
<i>EMCY object</i>	Emergency Object
<i>Electronic gear</i>	Calculation of a new output velocity for the motor movement based on the input velocity and the values of an adjustable gear ratio; calculated by the drive system.
<i>Encoder</i>	Sensor for detection of the angular position of a rotating component. Installed in a motor, the encoder shows the angular position of the rotor.
<i>Error</i>	Discrepancy between a computed, observed or measured value or condition and the specified or theoretically correct value or condition.
<i>Error class</i>	Classification of errors into groups. The different error classes allow for specific responses to errors, for example by severity.
<i>Factory setting</i>	Factory settings when the product is shipped
<i>Fatal error</i>	In the case of fatal error, the product is no longer able to control the motor so that the power stage must be immediately disabled.
<i>Fault</i>	Fault is a state that can be caused by an error. Further information can be found in the pertinent standards such as IEC 61800-7, ODVA Common Industrial Protocol (CIP).
<i>Fault reset</i>	A function used to restore the drive to an operational state after a detected error is cleared by removing the cause of the error so that the error is no longer active.
<i>HMI</i>	Human Machine Interface

<i>Heartbeat</i>	Used for unconfirmed connection acknowledgement messages from network devices.
<i>I/O</i>	Inputs/outputs
<i>Input device</i>	A device that can be connected via the RS232 interface; either the HMI or a PC with commissioning software.
<i>Life guarding</i>	For monitoring the connection of an NMT master
<i>Limit switch</i>	Switches that signal overtravel of the permissible range of travel.
<i>Mapping</i>	Assignment of object dictionary entries to PDOs
<i>NMT</i>	Network Management (NMT), part of the CANopen communication profile; tasks include initialization of the network and devices, starting, stopping and monitoring of devices
<i>Node ID</i>	Node address assigned to a device on the network.
<i>Node guarding</i>	Monitoring of the connection to the slave at an interface for cyclic data traffic.
<i>Object dictionary</i>	List of all parameters, values and functions available in the device. Each entry is uniquely referenced via index (16 bit) and subindex (8 bit).
<i>PDO</i>	Process Data Object
<i>Parameter</i>	Device data and values that can be read and set (to a certain extent) by the user.
<i>Persistent</i>	Indicates whether the value of the parameter remains in the memory after the device is switched off.
<i>Power stage</i>	The power stage controls the motor. The power stage generates current for controlling the motor on the basis of the positioning signals from the controller.
<i>Quick Stop</i>	Function which can be used for fast deceleration of the motor via a command or in the event of an error.
<i>R_PDO</i>	Receive PDO
<i>SDO</i>	Service Data Object
<i>SYNC object</i>	Synchronization object
<i>T_PDO</i>	Transmit PDO
<i>Warning</i>	If the term is used outside the context of safety instructions, a warning alerts to a potential problem that was detected by a monitoring function. A warning is not an error and does not cause a transition of the operating state.

# 11 Index

# 11

## A

- Abbreviations 139
- ABORT 74, 76
- Accessories and spare parts 133
- Activating
  - PDO 29
- Acyclic data transmission 37
- Address 52
  - Checking 71
- Asynchronous errors 74

## B

- Baud rate 52
  - Checking 71
- Before you begin
  - Safety information 13
- Bit field data 19
- Bit field identifier 19
- Bit fields
  - Data 19
  - Identifier 19
- Boot Up
  - Message 41, 45
- Boot-up
  - Message 38
- Bus arbitration 19

## C

- CAN
  - message 19
- CAN 3.0A 19
- CANopen
  - Communication profile, NMT 40
  - Error messages 75
  - Message 19
  - Standards 8
  - State machine 38
  - terminating resistors 48
- ccd
  - See command code
- Certifications 12
- Checking
  - Address 71
  - Baud rate 71
- Client-Server 22
- Client-server
  - SDO data exchange 23
- COB ID 19

- EMCY object 39
  - For node guarding 43
  - of communication objects 20
  - SDO 24
  - SYNC object 37
- COB Id
  - bus arbitration 19
  - Identification of communication objects 19
  - tasks 19
- Coding
  - Command code 25, 26
- Command code
  - Read value 26
  - SDO 24
  - Write value 25
- Command specifier 42
- Command-code
  - See command code
- Commissioning 51
- Commissioning the device 51
- Communication objects
  - COB IDs 20
  - Controlling 19
  - for PDO 29
  - Identification 19
  - Overview 18
- Communication profile
  - DS301 12
- Communication relationship
  - Client - server 21
  - Master - slave 21
  - Producer - consumer 21
- Connection error
  - Node guarding 44
- Connection monitoring
  - Heartbeat 45
  - NMT services 43
- Cyclic data transmission 37

## D

- Data
  - Persistent data 42
  - Reading 26
  - SDO 24
  - Writing 25
- Data frame 21
  - of the NMT device service 42
  - SDO 24
- Data length
  - Flexible 28
- Data transmission
  - Acyclic 37
  - Cyclic 37
  - Synchronous 36
- Device error
  - Internal 38
- Device profile

- DS402 12
- Diagnostics 71
- DS301
  - communication profile 12
- DS402
  - Device profile 12

## E

- Electrical installation 48
- Electronic Gear 60
- EMCY
  - COB ID of the object 39
  - Message 38
  - Object 38
  - object 18
- Emergency object
  - See EMCY object
- Emergency service 38
- EPLAN Macros 7
- Error
  - Evaluation 38
  - Response with SDO 26
- Error code 39
  - Table 75
- Error diagnostics
  - Connections to for fieldbus operation 71
  - Function test of fieldbus 71
- Error handling 38
- Error memory 39
- Error register 39, 75
- Errors
  - Messages for CANopen 75
- Example
  - Index and subindex entries 17
  - SDO message 24
  - Selection of a COB ID 20
  - Setting for R\_PDO3 29

## F

- Function code 20
- function code
  - See Function code
- Function test
  - Fieldbus 71
- Further reading 8

## G

- Glossary 137

## H

- Hazard categories 14
- Heartbeat 43, 45
  - Mutual monitoring 45
  - NMT state evaluation 45
  - Start of monitoring 45
- Homing 67

**I**

- Identification
  - of communication objects 19
- Index
  - SDO 24
- Installation
  - electrical 48
- Intended use 13
- Interruption of movement
  - Cause 75
- Introduction 9

**J**

- Jog 58

**L**

- Layer model
  - Application Layer 10
  - Data Link Layer 10
  - Physical Layer 10
- Life guarding 43

**M**

- Macros EPLAN 7
- Manuals
  - Source 7
- Master - Slave 21
- Message 19
  - Boot-up 38
  - CANopen 19
  - EMCY 38
  - NMT 42, 43
  - PDO 29
  - SDO 24
- Message objects 74
  - EMCY(80h+ node ID) 74
  - Error code (603Fh) 74
  - Error register (1001h) 74
  - Status word (6041h) 74
- Message-oriented communication 9
- Messages
  - Asynchronous errors 74
  - Error code (603Fh) 75
  - Error register (1001h) 75
  - On the device status 74
  - Synchronous errors 74
- Motion Sequence 69
- Multimaster capability 9

- N**
- Network management
  - See NMT
- NMT
  - Message 42
  - Network services 40



- Recipient of a message 42
- Services 40
  - For connection monitoring 43
  - For device control 41
  - Initialization 41
- services 18
- State machine 41
- State of slave 43
- Structure of a message 43
- Node address 19, 20, 42
- Node guarding 43
  - COB ID 43
  - Connection error 44
- Node ID 19

## O

- Object groups
  - Overview 11
- Objects
  - Standardized 17
- Operating mode
  - Electronic Gear 60
  - Homing 67
  - Jog 58
  - Motion Sequence 69
  - Profile Position 65
  - Profile Torque 61
  - Profile Velocity 63
- Operating modes 57
  - Operating modes, starting and changing 57
  - Starting and monitoring 37
- Operating modes, starting and changing 57
- Operating state, changing the 56
- Operating states 54
  - Indicating operating states 54
  - Operating state, changing the 56
- Operating states, indication 54
- Operation 53
- Overview
  - Communication objects 18
  - Object groups 11

## P

- PDO 18, 28
  - Activating 29
  - Communication objects 29
  - Message 29
  - Producer-consumer 28
  - Receive PDOs 30
  - Settings 29
  - Start PDO 37
  - Time intervals 30
  - Transmit PDOs 31
- PDO Mapping
  - Dynamic 33
- PDO mapping 33

- Structure of entries 34
- Prioritization of messages 9
- Process Data Object
  - see PDO
- process data objects
  - see PDO
- Producer-Consumer 22
- Producer-consumer
  - EMCY 38
  - Heartbeat 45
  - PDO 28
  - SYNC 36
- Profile Position 65
- Profile Torque 61
- Profile Velocity 63
- Profiles
  - standardized 12
  - Vendor-specific 12

## Q

- Qualification of personnel 13

## R

- R\_PDO
  - R\_PDO1 30
  - R\_PDO2 30
  - R\_PDO3 30
  - R\_PDO4 31
- Real-time data exchange 28
- Receive PDOs 30
- Recipient
  - of an NMT message 42
- Residual error probability 9
- Response
  - to SDO error 26

## S

- SDO 18, 23
  - COB ID 24
  - Command code 24
  - Data 24
  - Data frame 24
  - Error message 74, 76
  - Error response 26
  - Index, Subindex 24
  - Message 24
  - Message types 23
  - Response 26
  - Transmission error 76
- Service Data Object
  - See SDO
- Service data objects 18
  - See SDO
- Services
  - EMCY 38
  - For connection monitoring 40

- For device control 40
  - NMT 18, 40
- Source
  - EPLAN Macros 7
  - Manuals 7
- Specification
  - CAN 3.0A 19
- State machine
  - CANopen 38
  - NMT 41
- Subindex
  - SDO 24
- SYNC object 18, 36
  - COB ID 37
  - with PDO 28
- Synchronization 36
  - Time values 36
- Synchronization object
  - See SYNC object
- Synchronous
  - Data transmission 36
  - Errors 74

## T

- T\_PDO
  - T\_PDO1 31
  - T\_PDO2 31
  - T\_PDO3 31
  - T\_PDO4 32
- Tasks
  - of the COB Id 19
- Terminating resistors
  - CANopen 48
- Terms 139
- Time interval
  - Event timer 30
  - Heartbeat 45
  - Inhibit time 30
  - PDO 30
- Time values
  - For synchronization 36
- Transmit PDOs 31
- Troubleshooting 71

## U

- Units and conversion tables 137

## V

- Vendor-specific
  - Profiles 12

