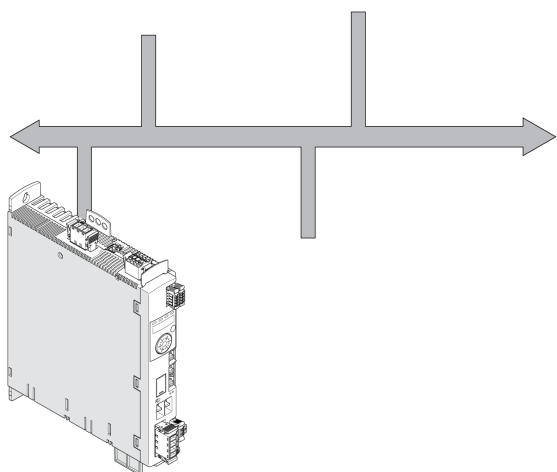


# LXM32A

## CANopen Interface

### Fieldbus manual

04/2018



---

The information provided in this documentation contains general descriptions and/or technical characteristics of the performance of the products contained herein. This documentation is not intended as a substitute for and is not to be used for determining suitability or reliability of these products for specific user applications. It is the duty of any such user or integrator to perform the appropriate and complete risk analysis, evaluation and testing of the products with respect to the relevant specific application or use thereof. Neither Schneider Electric nor any of its affiliates or subsidiaries shall be responsible or liable for misuse of the information contained herein. If you have any suggestions for improvements or amendments or have found errors in this publication, please notify us.

You agree not to reproduce, other than for your own personal, noncommercial use, all or part of this document on any medium whatsoever without permission of Schneider Electric, given in writing. You also agree not to establish any hypertext links to this document or its content. Schneider Electric does not grant any right or license for the personal and noncommercial use of the document or its content, except for a non-exclusive license to consult it on an "as is" basis, at your own risk. All other rights are reserved.

All pertinent state, regional, and local safety regulations must be observed when installing and using this product. For reasons of safety and to help ensure compliance with documented system data, only the manufacturer should perform repairs to components.

When devices are used for applications with technical safety requirements, the relevant instructions must be followed.

Failure to use Schneider Electric software or approved software with our hardware products may result in injury, harm, or improper operating results.

Failure to observe this information can result in injury or equipment damage.

© 2018 Schneider Electric. All Rights Reserved.

---

# Table of Contents

---



	<b>Safety Information</b> .....	<b>5</b>
	<b>About the Book</b> .....	<b>7</b>
<b>Chapter 1</b>	<b>Introduction</b> .....	<b>11</b>
	Fieldbus Devices on the CANopen Network .....	<b>11</b>
<b>Chapter 2</b>	<b>Basics</b> .....	<b>13</b>
	Communication Layers .....	<b>14</b>
	Objects .....	<b>15</b>
	CANopen Profiles .....	<b>16</b>
	Communication - Object Dictionary .....	<b>17</b>
	Communication - Objects .....	<b>18</b>
	Communication - Relationships .....	<b>21</b>
	SDO Data Exchange .....	<b>23</b>
	SDO Message .....	<b>24</b>
	SDO Reading and Writing Data .....	<b>25</b>
	SDO Reading Data Longer Than 4 Bytes .....	<b>27</b>
	PDO Data Exchange .....	<b>29</b>
	PDO Message .....	<b>30</b>
	PDO Events .....	<b>33</b>
	PDO Mapping .....	<b>34</b>
	Synchronization .....	<b>36</b>
	Emergency Object Service .....	<b>38</b>
	Network Management Services - Overview .....	<b>40</b>
	NMT Services for Device Control .....	<b>41</b>
	NMT Service Node Guarding/Life Guarding .....	<b>43</b>
	NMT Service Heartbeat .....	<b>45</b>
<b>Chapter 3</b>	<b>Commissioning</b> .....	<b>47</b>
	Preparation .....	<b>47</b>
<b>Chapter 4</b>	<b>Operating States and Operating Modes</b> .....	<b>49</b>
4.1	Operating States .....	<b>50</b>
	Indication of the Operating State .....	<b>51</b>
	Changing the Operating State .....	<b>53</b>
4.2	Operating Modes .....	<b>54</b>
	Starting and Changing an Operating Mode .....	<b>55</b>
	Operating Mode Jog .....	<b>56</b>
	Operating Mode Profile Torque .....	<b>57</b>
	Operating Mode Profile Velocity .....	<b>58</b>
	Operating Mode Profile Position .....	<b>59</b>
	Operating Mode Interpolated Position .....	<b>61</b>
	Operating Mode Homing .....	<b>63</b>
	Examples for Node Address 1 .....	<b>65</b>
<b>Chapter 5</b>	<b>Diagnostics and Troubleshooting</b> .....	<b>71</b>
	Fieldbus Communication Error Diagnostics .....	<b>72</b>
	Fieldbus Test .....	<b>73</b>
	Fieldbus Status LEDs .....	<b>74</b>
	Error Diagnostics via Fieldbus .....	<b>75</b>
	CANopen Error Messages .....	<b>76</b>

---

<b>Chapter 6 Object Dictionary</b> .....	<b>79</b>
Specifications for the Objects .....	<b>80</b>
Overview of object group 1000 <sub>h</sub> .....	<b>81</b>
Assignment Object Group 3000 <sub>h</sub> .....	<b>84</b>
Assignment Object Group 6000 <sub>h</sub> .....	<b>94</b>
Details of Object Group 1000 <sub>h</sub> .....	<b>96</b>
<b>Glossary</b> .....	<b>127</b>
<b>Index</b> .....	<b>131</b>



## Important Information

### NOTICE

Read these instructions carefully, and look at the equipment to become familiar with the device before trying to install, operate, service, or maintain it. The following special messages may appear throughout this documentation or on the equipment to warn of potential hazards or to call attention to information that clarifies or simplifies a procedure.



The addition of this symbol to a “Danger” or “Warning” safety label indicates that an electrical hazard exists which will result in personal injury if the instructions are not followed.



This is the safety alert symbol. It is used to alert you to potential personal injury hazards. Obey all safety messages that follow this symbol to avoid possible injury or death.

### **DANGER**

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

### **WARNING**

**WARNING** indicates a hazardous situation which, if not avoided, **could result in** death or serious injury.

### **CAUTION**

**CAUTION** indicates a hazardous situation which, if not avoided, **could result in** minor or moderate injury.

### **NOTICE**

**NOTICE** is used to address practices not related to physical injury.

### PLEASE NOTE

Electrical equipment should be installed, operated, serviced, and maintained only by qualified personnel. No responsibility is assumed by Schneider Electric for any consequences arising out of the use of this material.

A qualified person is one who has skills and knowledge related to the construction and operation of electrical equipment and its installation, and has received safety training to recognize and avoid the hazards involved.

### 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. 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 modifying the settings and by the mechanical, electrical and electronic equipment of the entire system in which the product is used.

The qualified person must be able to detect possible hazards that may arise from parameterization, modifying parameter values and generally from mechanical, electrical, or electronic equipment.

The qualified person must be familiar with the standards, provisions, and regulations for the prevention of industrial accidents, which they must observe when designing and implementing the system.

---

## INTENDED USE

The products described or affected by this document are, along with software, accessories and options, servo-drive systems for three-phase servo motors. The products are intended for industrial use according to the instructions, directions, examples and safety information contained in the present user guide and other supporting documentation.

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 products, you must perform a risk assessment in view of the planned application. Based on the results, the appropriate safety-related measures must be implemented.

Since the products are used as components in an overall machine or process, you must ensure the safety of persons by means of the design of this overall machine or process.

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

Any use other than the use explicitly permitted as described herein is prohibited and may result in unanticipated hazards.

# About the Book



## At a Glance

### Document Scope

The information provided in this manual supplements the product manual of the drive LXM32A.

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

### Validity Note

This manual applies to the fieldbus CANopen for the product LXM32A.

For product compliance and environmental information (RoHS, REACH, PEP, EOL, etc.), go to [www.schneider-electric.com/green-premium](http://www.schneider-electric.com/green-premium).

The technical characteristics of the devices described in this document also appear online. To access this information online:

Step	Action
1	Go to the Schneider Electric home page <a href="http://www.schneider-electric.com">www.schneider-electric.com</a> .
2	In the <b>Search</b> box type the reference of a product or the name of a product range. <ul style="list-style-type: none"><li>Do not include blank spaces in the reference or product range.</li><li>To get information on grouping similar modules, use asterisks (*).</li></ul>
3	If you entered a reference, go to the <b>Product Datasheets</b> search results and click on the reference that interests you. If you entered the name of a product range, go to the <b>Product Ranges</b> search results and click on the product range that interests you.
4	If more than one reference appears in the <b>Products</b> search results, click on the reference that interests you.
5	Depending on the size of your screen, you may need to scroll down to see the data sheet.
6	To save or print a data sheet as a .pdf file, click <b>Download XXX product datasheet</b> .

The characteristics that are presented in this manual should be the same as those characteristics that appear online. In line with our policy of constant improvement, we may revise content over time to improve clarity and accuracy. If you see a difference between the manual and online information, use the online information as your reference.

### Related Documents

Title of documentation	Reference number
LXM32A - CANopen interface - Fieldbus manual (this manual)	<a href="#">0198441113779 (eng)</a> <a href="#">0198441113780 (fre)</a> <a href="#">0198441113778 (ger)</a>
Lexium 32A - Servo Drive - User Guide	<a href="#">0198441113755 (eng)</a> <a href="#">0198441113756 (fre)</a> <a href="#">0198441113754 (ger)</a> <a href="#">0198441113758 (spa)</a> <a href="#">0198441113757 (ita)</a> <a href="#">0198441113759 (chi)</a>

You can download these technical publications and other technical information from our website at <http://www.schneider-electric.com/en/download>.

## Product Related Information

### WARNING

#### LOSS OF CONTROL

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

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

<sup>1</sup> For 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 and Operation of Adjustable-Speed Drive Systems” or their equivalent governing your particular location.

## Terminology Derived from Standards

The technical terms, terminology, symbols and the corresponding descriptions in this manual, or that appear in or on the products themselves, are generally derived from the terms or definitions of international standards.

In the area of functional safety systems, drives and general automation, this may include, but is not limited to, terms such as *safety*, *safety function*, *safe state*, *fault*, *fault reset*, *malfunction*, *failure*, *error*, *error message*, *dangerous*, etc.

Among others, these standards include:

Standard	Description
EN 61131-2:2007	Programmable controllers, part 2: Equipment requirements and tests.
ISO 13849-1:2008	Safety of machinery: Safety related parts of control systems. General principles for design.
EN 61496-1:2013	Safety of machinery: Electro-sensitive protective equipment. Part 1: General requirements and tests.
ISO 12100:2010	Safety of machinery - General principles for design - Risk assessment and risk reduction
EN 60204-1:2006	Safety of machinery - Electrical equipment of machines - Part 1: General requirements
EN 1088:2008 ISO 14119:2013	Safety of machinery - Interlocking devices associated with guards - Principles for design and selection
ISO 13850:2006	Safety of machinery - Emergency stop - Principles for design
EN/IEC 62061:2005	Safety of machinery - Functional safety of safety-related electrical, electronic, and electronic programmable control systems
IEC 61508-1:2010	Functional safety of electrical/electronic/programmable electronic safety-related systems: General requirements.
IEC 61508-2:2010	Functional safety of electrical/electronic/programmable electronic safety-related systems: Requirements for electrical/electronic/programmable electronic safety-related systems.
IEC 61508-3:2010	Functional safety of electrical/electronic/programmable electronic safety-related systems: Software requirements.
IEC 61784-3:2008	Digital data communication for measurement and control: Functional safety field buses.
2006/42/EC	Machinery Directive
2014/30/EU	Electromagnetic Compatibility Directive
2014/35/EU	Low Voltage Directive



---

In addition, terms used in the present document may tangentially be used as they are derived from other standards such as:

Standard	Description
IEC 60034 series	Rotating electrical machines
IEC 61800 series	Adjustable speed electrical power drive systems
IEC 61158 series	Digital data communications for measurement and control – Fieldbus for use in industrial control systems

Finally, the term *zone of operation* may be used in conjunction with the description of specific hazards, and is defined as it is for a *hazard zone* or *danger zone* in the *Machinery Directive (2006/42/EC)* and *ISO 12100:2010*.

**NOTE:** The aforementioned standards may or may not apply to the specific products cited in the present documentation. For more information concerning the individual standards applicable to the products described herein, see the characteristics tables for those product references.



---

# Chapter 1

## Introduction

---

### Fieldbus Devices on the CANopen Network

#### Overview

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.

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

#### Network Devices

Examples of CAN bus devices are

- Automation devices, for example, controllers
- PCs
- Input/output modules
- Drives
- Sensors and actuators



---

# Chapter 2

## Basics

---

The information contained in this chapter provides a general overview of the various protocols of the fieldbus as it applies to the equipment in the present document. It is not intended as a thorough treatment of the subject, nor is it a sufficient basis to design and deploy a fieldbus network in any given application.

The following information is intended to be consulted in an as needed, as is basis. Only appropriately trained persons who are familiar with and have the education and training necessary to understand the contents of this information, as well as all other pertinent product documentation, are authorized to work on and with this equipment.

### What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Communication Layers	14
Objects	15
CANopen Profiles	16
Communication - Object Dictionary	17
Communication - Objects	18
Communication - Relationships	21
SDO Data Exchange	23
SDO Message	24
SDO Reading and Writing Data	25
SDO Reading Data Longer Than 4 Bytes	27
PDO Data Exchange	29
PDO Message	30
PDO Events	33
PDO Mapping	34
Synchronization	36
Emergency Object Service	38
Network Management Services - Overview	40
NMT Services for Device Control	41
NMT Service Node Guarding/Life Guarding	43
NMT Service Heartbeat	45

## Communication Layers

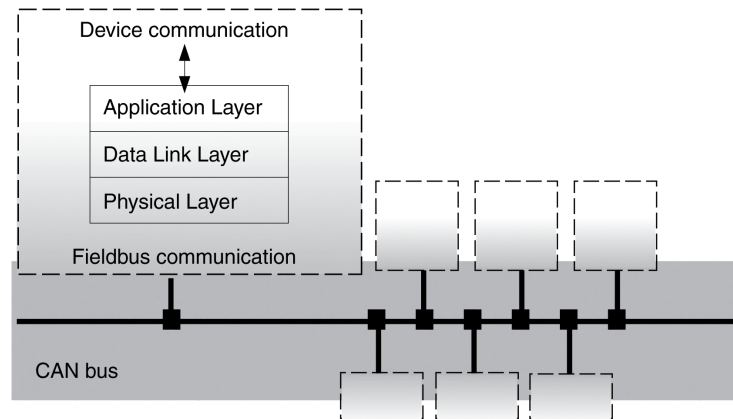
### Overview

CANopen uses the CAN bus technology for data communication.

CANopen is based on the network services for data communication as per the ISO-OSI model.

3 layers enable data communication via the CAN bus:

- Physical Layer
- Data Link Layer
- Application Layer



### Physical Layer

The physical layer defines the electrical properties of the CAN bus such as connectors, cable length and cable properties as well as bit assignment and bit timing.

### Data Link Layer

The data link layer connects the network devices. It assigns priorities to individual data packets and monitors and detects 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.

## Objects

### Overview

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 modified via these objects.

The product provides corresponding parameters for CANopen object groups 3000<sub>h</sub> and 6000<sub>h</sub>.

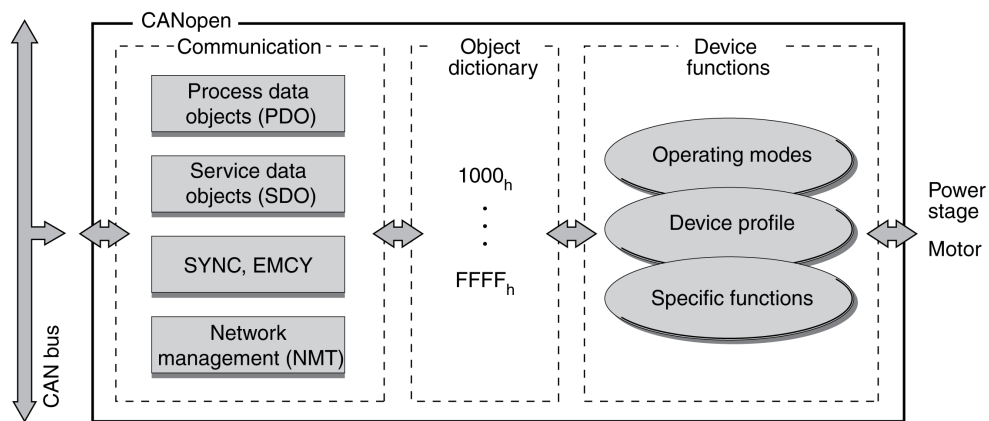
The names of the parameters and the data type of the parameters may be different from the DSP402 definition for object group 6000<sub>h</sub>. In this case, enter the data type according to the DS402.

A detailed description of the parameters can be found in the product user guide in the Parameters chapter.

### Object Dictionary

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

Device model with object dictionary



The object dictionary contains objects for describing the data types and executing the communication tasks and device functions under CANopen.

### 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 specifications.

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

See chapter Object Dictionary (*see page 79*) for a list of the CANopen objects.

## CANopen Profiles

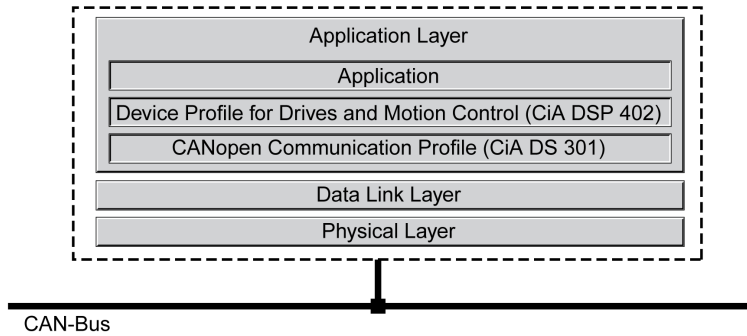
### Standardized Profiles

Standardized profiles describe objects that are used with different devices without additional configuration. The international users' and manufacturers' group, CAN in Automation (CiA), has standardized profiles.

These include:

- DS301 communication profile
- DSP402 device profile

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.

### DSP 402 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
- Change, monitoring and execution of operating modes

### Vendor-Specific Profiles

The core functions of a device can be used with objects of standardized device profiles. Vendor-specific device profiles offer an extended 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.



## Communication - Object Dictionary

### Overview

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 send and/or receive object data.

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

- Exchange parameter values
- Start movement functions of individual devices
- Request status information

Each CANopen device manages an object dictionary which contains the 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>	minimum position limit	Negative software limit switch
607D <sub>h</sub>	02 <sub>h</sub>	maximum position limit	Positive software limit switch

### Object Descriptions in the Manual

For CANopen 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 Operating States and Operating Modes (*see page 49*).
- 6xxx<sub>h</sub> objects: Standardized objects of the device profile in chapter Operating States and Operating Modes (*see page 49*).

### Standardized Objects

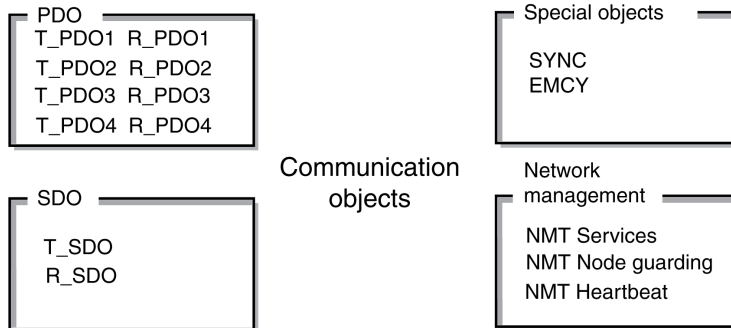
Standardized objects allow you to use the same application program for different network devices of the same device type. This requires these objects to be contained in the object dictionary of the network devices. Standardized objects are defined in the DS301 communication profile and the DSP402 device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

## 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.

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

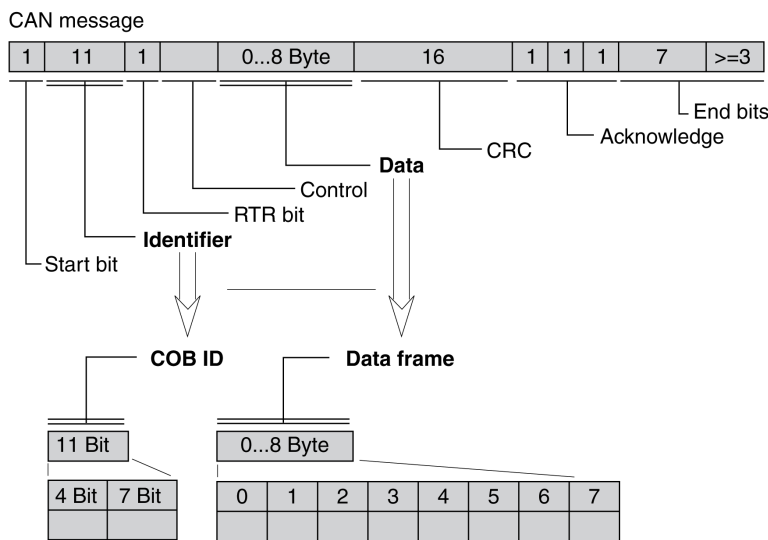


- PDOs (process data objects) for real-time transmission of process data
- SDOs (service data object) for read and write access to the object dictionary
- Objects for controlling CAN messages:
  - SYNC object (synchronization object) for synchronization of network devices
  - EMCY object (emergency object), 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.

CAN message and simplified representation of CANopen message



CANopen message (simplified)

### CANopen Message

To 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 detection. 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

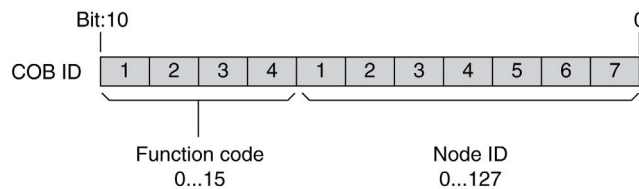
The COB ID (**C**ommunication **O**bject **I**dentifier) has 2 tasks for controlling communication objects:

- 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.

COB ID with function code and node address:



### COB IDs of the Communication Objects

The following table shows the COB IDs of the communication objects with the factory settings. The column "Index of object parameters" shows the index of special SDOs 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 (0 <sub>h</sub> )	-
SYNC object	0 0 0 1	0 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 x	128 (80 <sub>h</sub> ) + node ID	1014 <sub>h</sub> , 1015 <sub>h</sub>
T_PDO1	0 0 1 1	x x x x x x x x	384 (180 <sub>h</sub> ) + node ID	1800 <sub>h</sub>
R_PDO1	0 1 0 0	x x x x x x x x	512 (200 <sub>h</sub> ) + node ID	1400 <sub>h</sub>
T_PDO2	0 1 0 1	x x x x x x x x	640 (280 <sub>h</sub> ) + node ID	1801 <sub>h</sub>
R_PDO2	0 1 1 0	x x x x x x x x	768 (300 <sub>h</sub> ) + node ID	1401 <sub>h</sub>
T_PDO3	0 1 1 1	x x x x x x x x	896 (380 <sub>h</sub> ) + node ID	1802 <sub>h</sub>
R_PDO3	1 0 0 0	x 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 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 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 x	1408 (580 <sub>h</sub> ) + node ID	-
R_SDO	1 1 0 0	x 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 x	1792 (700 <sub>h</sub> ) + node ID	-

COB IDs of PDOs can be changed if required. The assignment pattern for COB IDs shown corresponds to the factory settings.

### 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 also 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 has to be 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 the reachable 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.

## Communication - Relationships

### Overview

CANopen uses 3 relationships for communication between network devices:

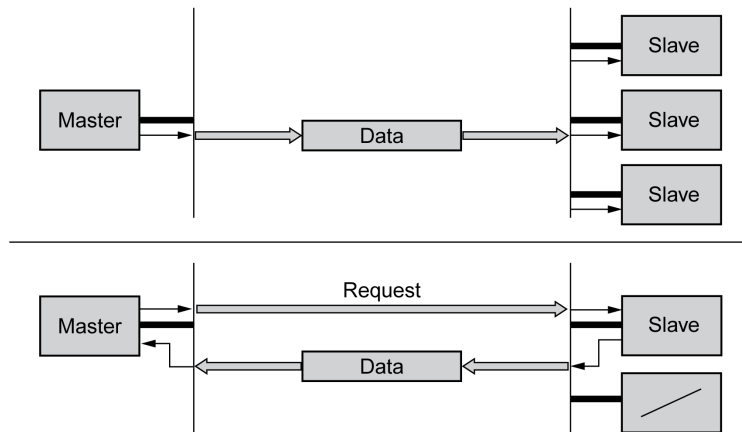
- Master-slave relationship
- Client-server relationship
- Producer-consumer relationship

### Master-Slave Relationship

A network master controls the message traffic. A slave can respond only 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.

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 slave or by the reachable slaves or by no slave.

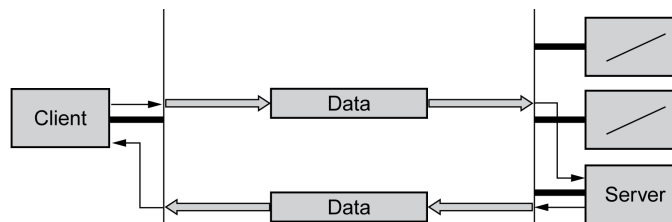
To confirm the message, the master requests a message from a specific slave, which then responds with the required data.

### 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.

Client-server relationship



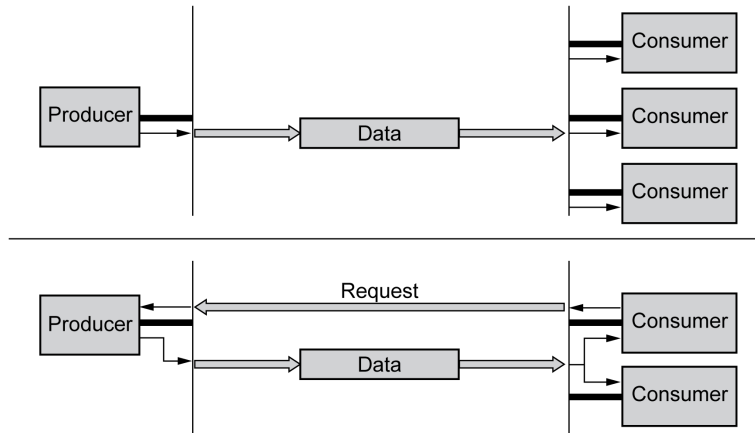
The client addresses and sends a CAN message to a server. The server evaluates the message and sends the response data as an acknowledgement.

### 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.

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 that the message was received.

The message transmission can be triggered by:

- An internal event, for example, "target position reached"
- The synchronization object SYNC
- A request of a consumer

See chapter PDO Data Exchange ([see page 29](#)) for details on the function of the producer-consumer relationship and on requesting messages.

## SDO Data Exchange

### 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 modified.

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.

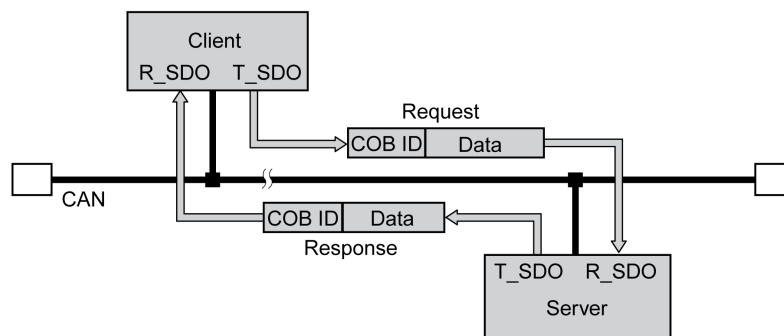
The T\_SDO of an SDO client is used to send the request for data exchange; the R\_SDO is used to receive. The data frame of an SDO consist of 8 bytes.

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

### 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.

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.

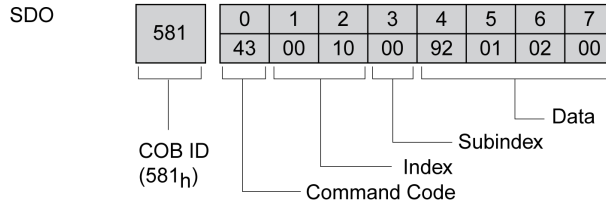
## SDO Message

### Overview

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 blocks of 7 bytes.

The following illustration shows an example of an SDO message:



### COB ID and Data Frame

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

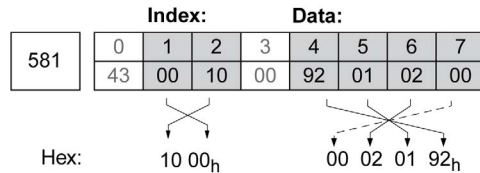
The data frame of an SDO messages consists of:

- Command Code: The command code contains the SDO message type and the data length of the transmitted value.
- Index: Index of the object.
- Subindex: Subindex of the object.
- Data: Data of up to 4 bytes of the object.

### 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.

Rearranging numeric values greater than 1 byte:





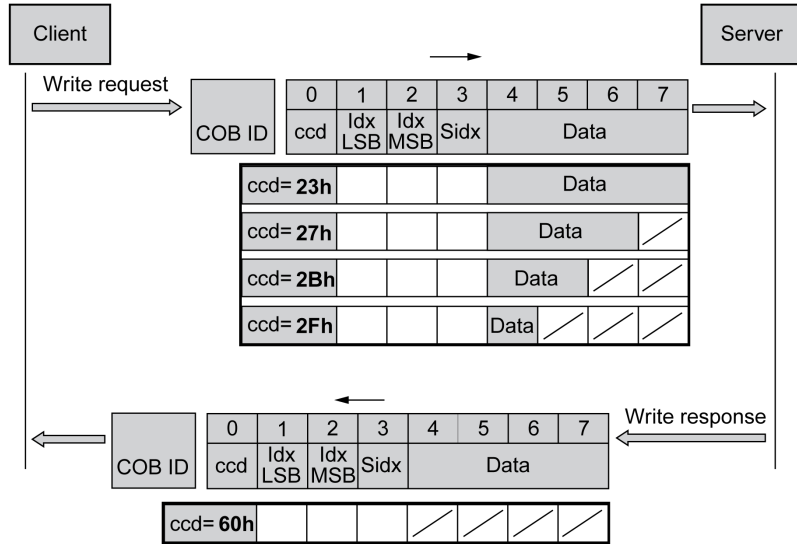
## SDO Reading and Writing Data

### 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.

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.

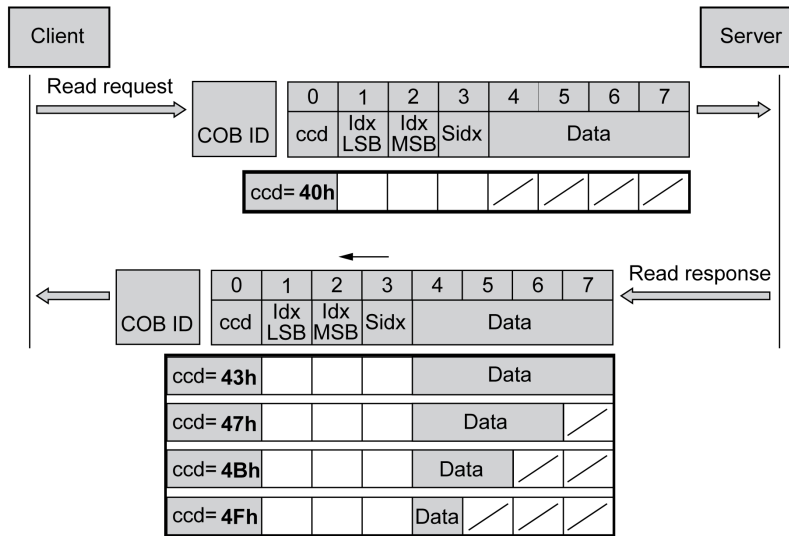
Message type	Data length used				Description
	4 byte	3 byte	2 byte	1 byte	
Write request	23 <sub>h</sub>	27 <sub>h</sub>	2B <sub>h</sub>	2F <sub>h</sub>	Transmitting parameters
Write response	60 <sub>h</sub>	60 <sub>h</sub>	60 <sub>h</sub>	60 <sub>h</sub>	Confirmation
Error response	80 <sub>h</sub>	80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

### Reading Data

The client starts a read request by transmitting the index and subindex that point to the object or part of the object to read.

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

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

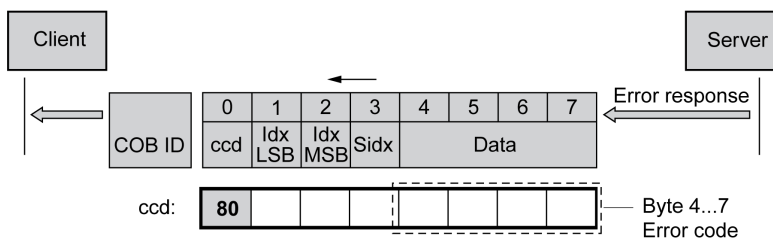
The table below shows the command code for transmitting a read value. It depends on the message type and the transmitted data length.

Message type	Data length used				Description
	4 byte	3 byte	2 byte	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

### Error Response

If a message could not be evaluated, the server sends an error message. See chapter SDO Error Message ABORT (see page 76) for details on the evaluation of the error message.

Response with error message (error response):



## SDO Reading Data Longer Than 4 Bytes

### Overview

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

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

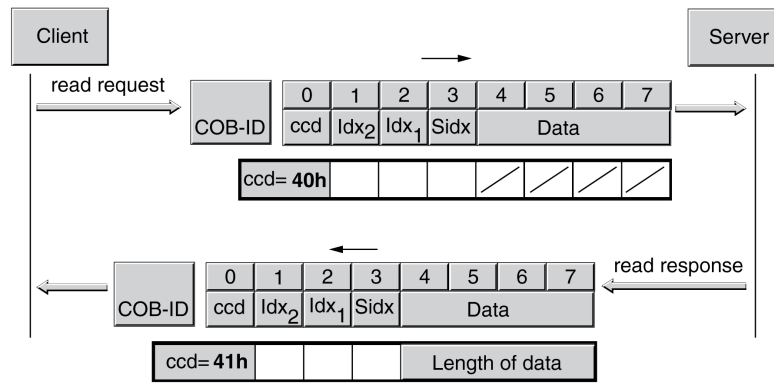
The read request by the SDO client contains the command code "ccd" with the toggle bit and a data segment. The confirmation also contains a toggle bit in the command code "ccd". In the first read request, the toggle bit has the value "0", in the subsequent read requests it toggles between 1 and 0.

### Reading Data

The client starts a read request by transmitting the index and subindex that point to the object to read.

The server confirms the read request with the command code 41<sub>h</sub>, the index, the subindex and the data length of the object to be read. The command code 41<sub>h</sub> indicates that the object has data with a length of more than 4 bytes.

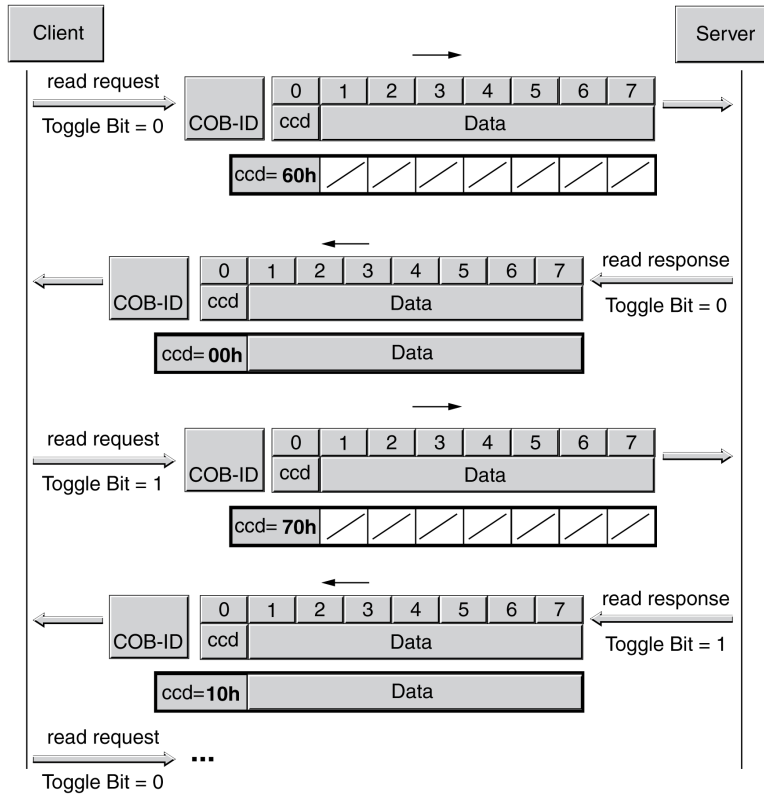
First read request:



The data is requested by means of further read requests. The data is transmitted in messages with 7 bytes each.

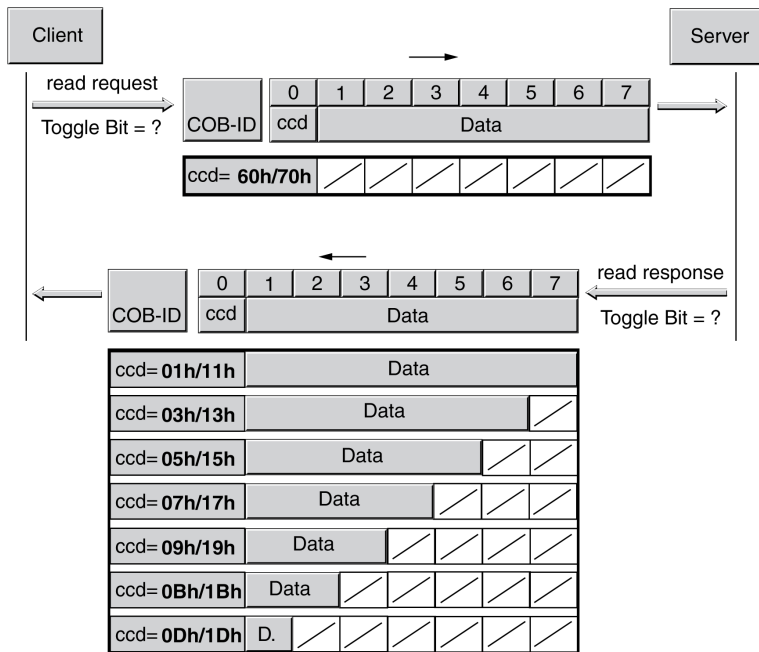
The client must continue to start read requests until the data is transmitted.

Additional read requests:



It is possible to detect whether the data has been transmitted on the basis of the command code of the server. Once the data has been transmitted, the command code of the server indicates the length of the remaining response data and, by the same token, the end of the transmission.

Final read request:



## PDO Data Exchange

### Overview

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

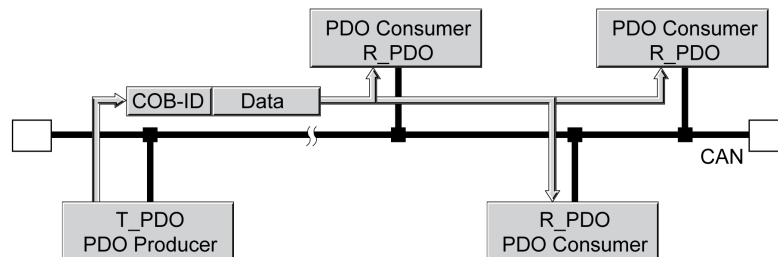
The flexible data length of a PDO message also increases the data throughput. A PDO message can transmit up to 8 bytes of data. If only 2 bytes are assigned, only 2 data bytes are sent.

The length of a PDO message and the assignment of the data fields are specified by PDO mapping. See chapter PDO Mapping (*see page 34*) for additional information.

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

### Data Exchange

PDO data exchange:



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

- Synchronized
- Event-driven, asynchronous

The SYNC object controls synchronized data processing. Synchronous PDO messages are transmitted immediately like the other PDO messages, but are only evaluated on the next SYNC transmission. 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.

## PDO Message

### Overview

The device uses 8 PDOs, 4 receive PDOs and 4 transmit PDOs.

- R\_PDO to receive PDO messages (R: Receive)
- T\_PDO to transmit the PDO message (T: Transmit)

By default, the PDOs are evaluated or transmitted in an event-driven way.

The settings of the PDOs can be read and modified with 8 communication objects:

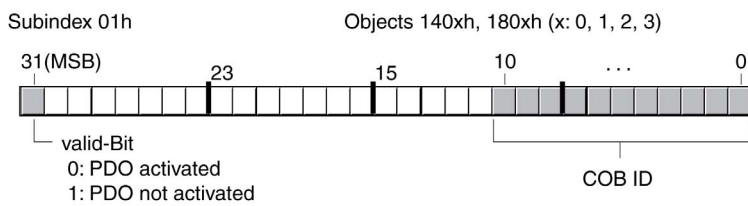
PDO	Object
Settings for R_PDO1	1st receive PDO parameter (1400 <sub>h</sub> )
Settings for R_PDO2	2nd receive PDO parameter (1401 <sub>h</sub> )
Settings for R_PDO3	3rd receive PDO parameter (1402 <sub>h</sub> )
Settings for R_PDO4	4th receive PDO parameter (1403 <sub>h</sub> )
Settings for T_PDO1	1st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

### Activating PDOs

With the default PDO settings, R\_PDO1 and T\_PDO1 are activated. The other PDOs must be activated manually in order to be used.

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

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

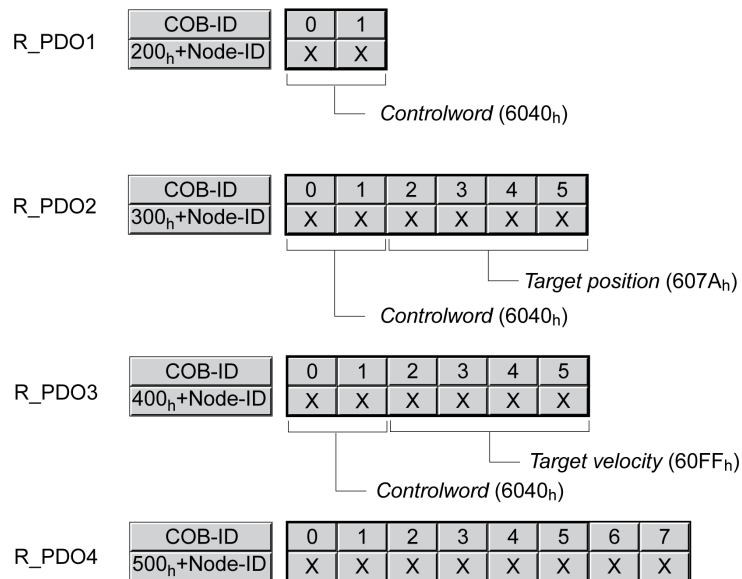
The time intervals "inhibit time" and "event timer" can be set for each transmit PDO.

- The time interval "inhibit time" can be used to reduce the CAN bus load, 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 interval has elapsed, the device transmits the event-controlled T\_PDO. The value of the time interval is set with subindex 05<sub>h</sub>.

## Receive PDOs

The R\_PDOs can be used to map various vendor-specific objects by means of PDO mapping. The objects for R\_PDO1, R\_PDO2, R\_PDO3 and R\_PDO4 are preset.

Receive PDOs



### R\_PDO1

R\_PDO1 contains the control word, object `controlword` (6040<sub>h</sub>), 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 are received for a movement in the operating mode "Profile Position" in the object `target position` (607A<sub>h</sub>).

R\_PDO2 is evaluated asynchronously, i.e. it is event-driven. R\_PDO2 is preset.

For details on the SYNC object see chapter Synchronization ([see page 36](#)).

### R\_PDO3

R\_PDO3 contains the control word and the target velocity, object `Target velocity` (60FF<sub>h</sub>), for the operating mode "Profile Velocity".

R\_PDO3 is evaluated asynchronously, i.e. it is event-driven. R\_PDO3 is preset.

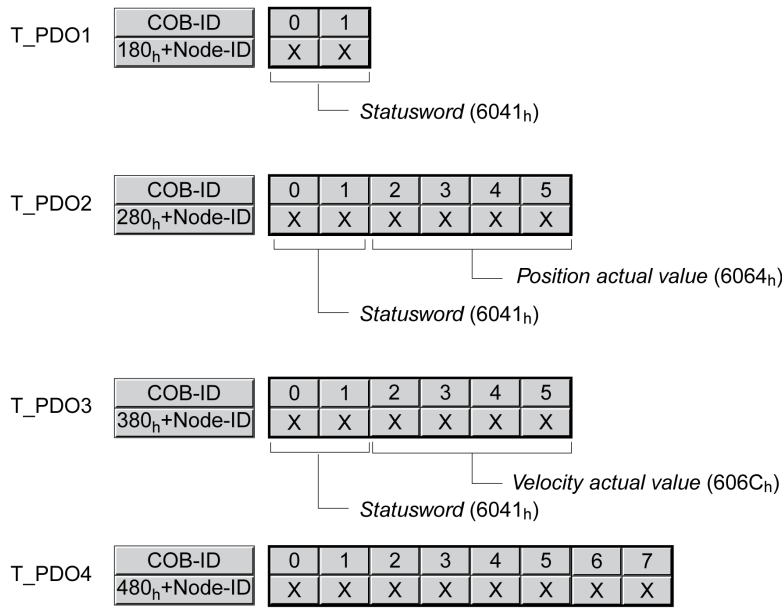
### R\_PDO4

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.

**Transmit PDOs**

The objects for T\_PDO1, T\_PDO2, T\_PDO3 and T\_PDO4 can be changed by means of PDO mapping. Transmit PDOs



**T\_PDO1**

T\_PDO1 contains the status word, object `statusword (6041h)`, 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 actual position of the motor, object `Position actual value (6064h)`, 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 actual velocity, object `Velocity actual value (606Ch)`, for monitoring the actual velocity 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 T\_PDOs can be used to map various vendor-specific objects via PDO mapping.



## PDO Events

### Overview

The parameters `CANpdo1Event` ... `CANpdo4Event` are used to specify the objects which are to trigger an event.

Example: If `CANpdo1Event` = 1 only a change to the first PDO object triggers an event. If `CANpdo1Event` = 15, each change to a PDO object triggers an event.

Parameter name HMI menu HMI name	Description	Unit Minimum value Factory setting Maximum value	Data type R/W Persistent Expert	Parameter address via fieldbus
<code>CANpdo1Event</code>	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 R/W - -	CANopen 3041:B <sub>h</sub> Modbus 16662
<code>CANpdo2Event</code>	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 R/W - -	CANopen 3041:C <sub>h</sub> Modbus 16664
<code>CANpdo3Event</code>	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 R/W - -	CANopen 3041:D <sub>h</sub> Modbus 16666
<code>CANpdo4Event</code>	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 R/W - -	CANopen 3041:E <sub>h</sub> Modbus 16668

## PDO Mapping

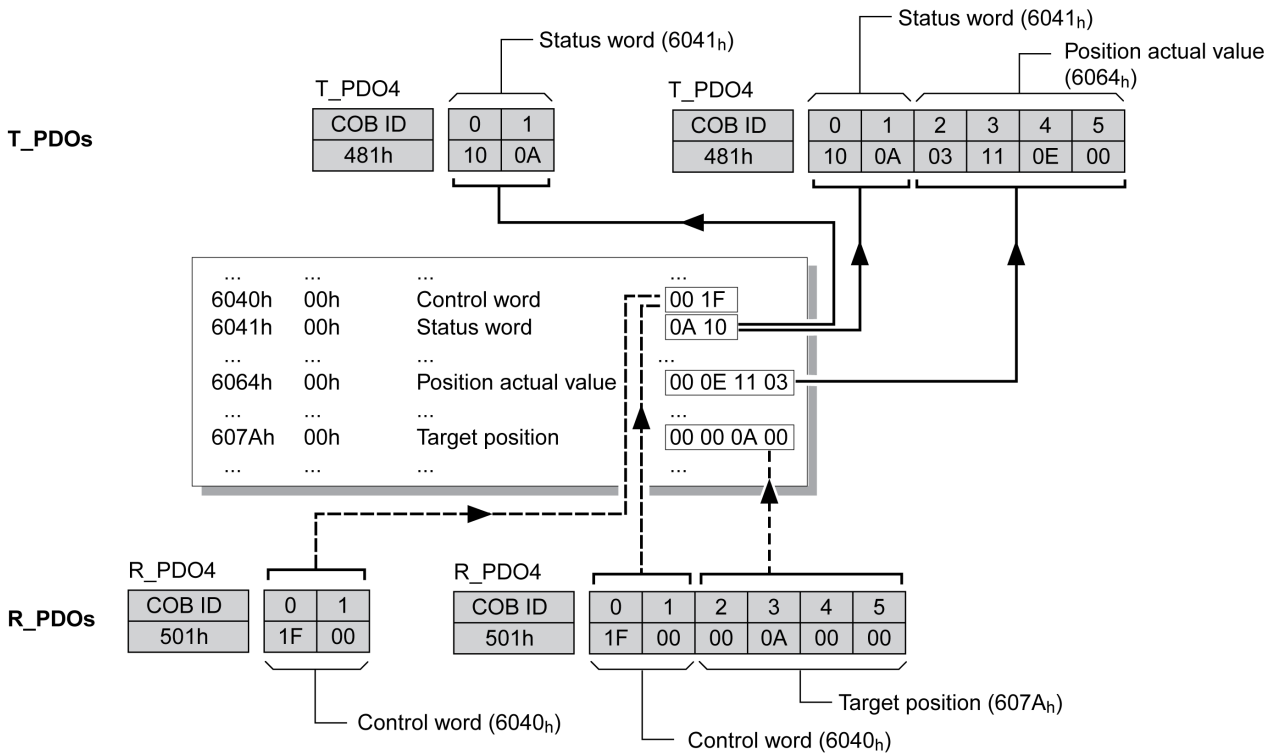
### Overview

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 Assignment Object Group 3000h (see page 84) and Assignment Object Group 6000h (see page 94) contain lists of vendor-specific objects with the information whether they 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.

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.

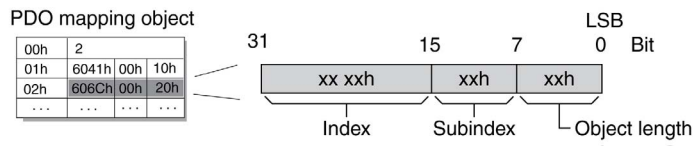
The settings for PDO mapping are defined in an assigned communication object for each PDO.

Object	PDO mapping for	Type
1st receive PDO mapping (1600h)	R_PDO1	Dynamic
2nd receive PDO mapping (1601h)	R_PDO2	Dynamic
3rd receive PDO mapping (1602h)	R_PDO3	Dynamic
4th receive PDO mapping (1603h)	R_PDO4	Dynamic
1st transmit PDO mapping (1A00h)	T_PDO1	Dynamic
2nd transmit PDO mapping (1A01h)	T_PDO2	Dynamic
3rd transmit PDO mapping (1A02h)	T_PDO3	Dynamic
4th transmit PDO mapping (1A03h)	T_PDO4	Dynamic

**Structure of the 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.

Structure of entries for PDO mapping:



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

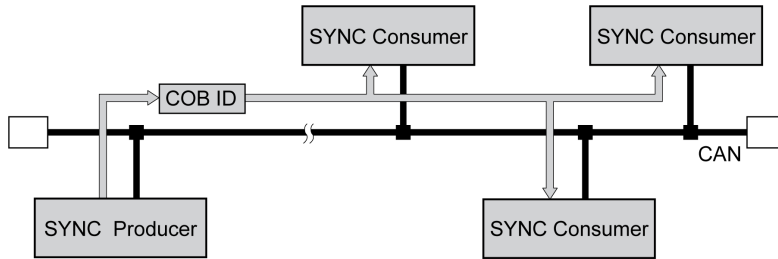
Object length	Bit value
08 <sub>h</sub>	8 bits
10 <sub>h</sub>	16 bits
20 <sub>h</sub>	32 bits

## Synchronization

### Overview

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 the reachable devices by a network device and can be evaluated by the devices that support synchronous PDOs.

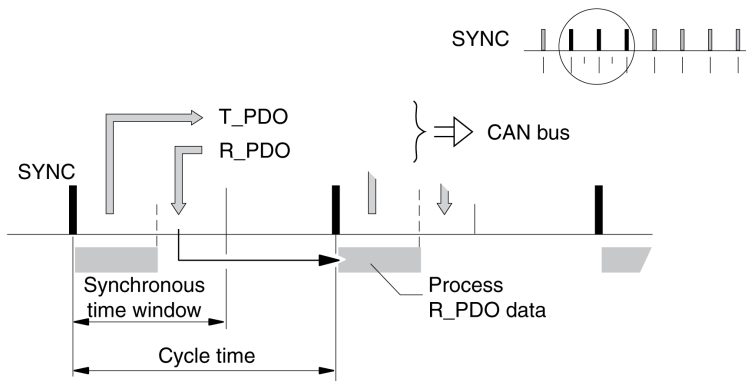


### Time Values for Synchronization

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 set with the object `Synchronous window length (1007h)`.

Synchronization times:



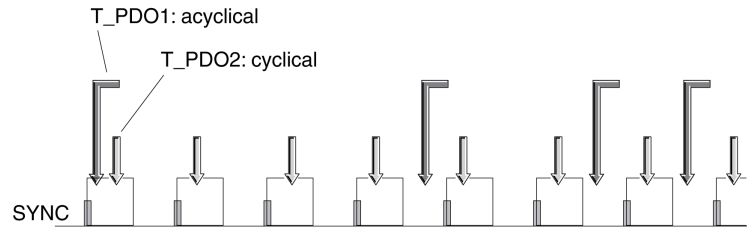
### Synchronous Data Transmission

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.

Cyclic and acyclic transmission:



In the case of cyclic transmission, PDO messages are exchanged continuously in a specified cycle, for example with each 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 (02h)` of the corresponding PDO parameter, for example, in the object `1st receive PDO parameter (1400h:02h)` 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 modified after initialization of the network with the object `COB-ID SYNC Message (1005h)`.

### "Start" PDO

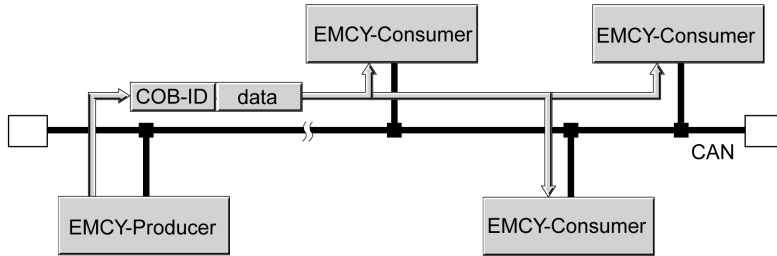
With the default settings of the PDOs, R\_PDO1 ... R\_PDO4 and T\_PDO1 ... T\_PDO4 are received and transmitted asynchronously. T\_PDO2 ... T\_PDO3 are transmitted additionally after the event timer has elapsed. The synchronization allows an operating mode to be started simultaneously on multiple devices so that, for example, the feed of a portal drive with several motors can be synchronized.

## Emergency Object Service

### Overview

The emergency object service signals internal errors via the CAN bus. The error message is transmitted to the network devices with an EMCY object according to the Consumer-Producer relationship.

Error message via EMCY objects:

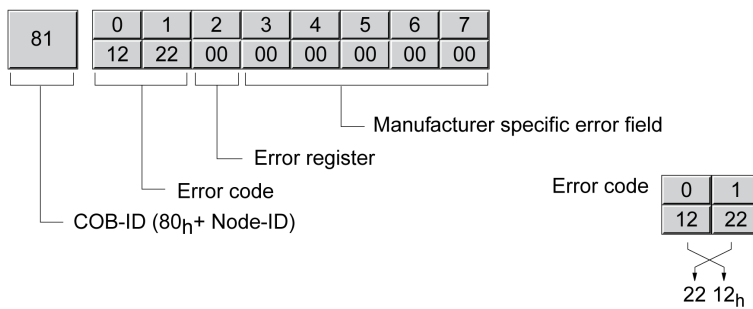


### Boot-Up Message

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

### EMCY Message

If an error occurs, the device switches to the operating state 9 Fault as per the CANopen state machine. At the same time, it transmits an EMCY message with error register and error code.



Bytes 0 ... 1: Error code (as per DS301)

The value is also saved in the object `Predefined error field (1003:1h)`.

Byte 2: Error register

The value is also saved in the object `Error register (1001h)`.

Bytes 3 ... 4: Reserved

Byte 5: PDO: Number of the PDO

Bytes 6 ... 7: Vendor-specific error number

The value is also saved in the object `Error code (603Fh)`.

### COB ID

The COB ID for each device on the network supporting an EMCY object is determined on the basis of the node address:

COB ID = EMCY object (80<sub>h</sub>) + node ID

The function code of the COB ID can be modified with the object `COB-ID emergency (1014h)`.

### Error Register and Error Code

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 cause of error can be determined using the error code. The error code is transmitted in Intel format as a 2 byte value; the bytes must be reversed for evaluation.

### Error Memory

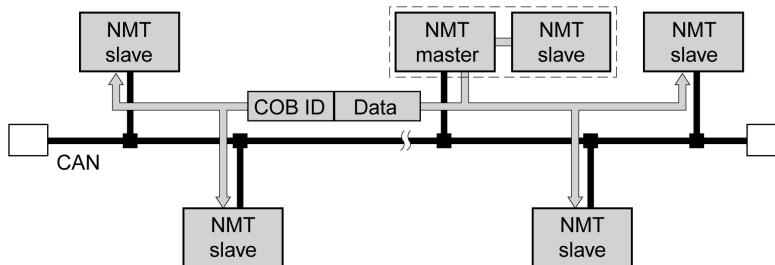
The device saves the error register in the object `Error register (1001h)` and the last error that was detected in the object `Error code (603Fh)`.

## Network Management Services - Overview

### Description

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 reachable NMT slaves simultaneously.



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

### NMT Services

NMT services can be divided into 2 groups:

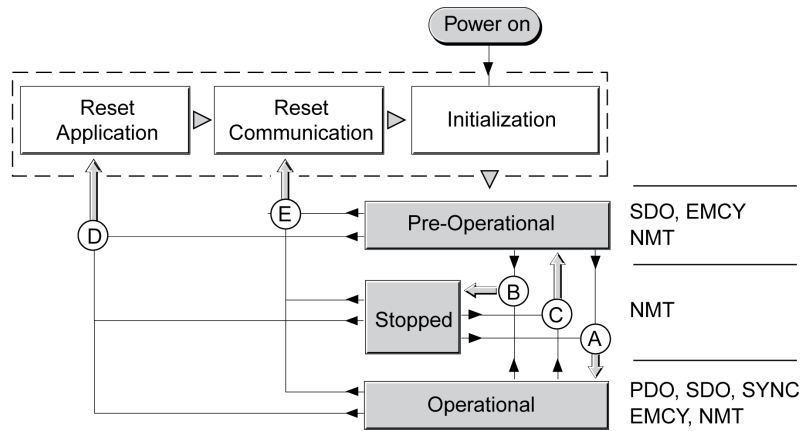
- Services for device control, to initialize devices for CANopen communication and to control the behavior of devices during operation on the network.
- Services for connection monitoring to monitor the communication status of network devices.
  - "Node guarding" for monitoring the connection of an NMT slave
  - "Life guarding" for monitoring the connection of an NMT master
  - "Heartbeat" for unconfirmed connection messages from network devices.



## 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.



To the right, the graphic shows the 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 applied (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. After this, 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 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. <b>NOTE:</b> If the power stage is enabled (operating state "Operation Enabled" or "Quick Stop"), an error of error class 2 is triggered. The motor is stopped and the power stage disabled.
Enter Pre-Operational (Transition to "Pre-Operational")	C	Transition to operating state "Pre-Operational" The 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 transition to operating state "Pre-Operational". <b>NOTE:</b> If the power stage is enabled (operating state "Operation Enabled" or "Quick Stop"), an error of error class 2 is triggered. The motor is stopped and the power stage disabled.

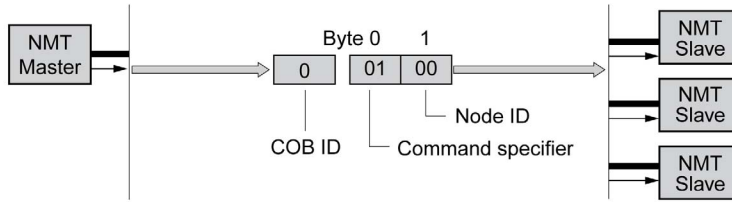
### Persistent Data Memory

When the supply voltage is applied (Power on), the device loads the saved object data from the non-volatile memory 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.



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 the reachable NMT slaves.

## NMT Service Node Guarding/Life Guarding

### COB ID

The communication object NMT error control (700<sub>h</sub>+Node-ID) is used for connection monitoring.

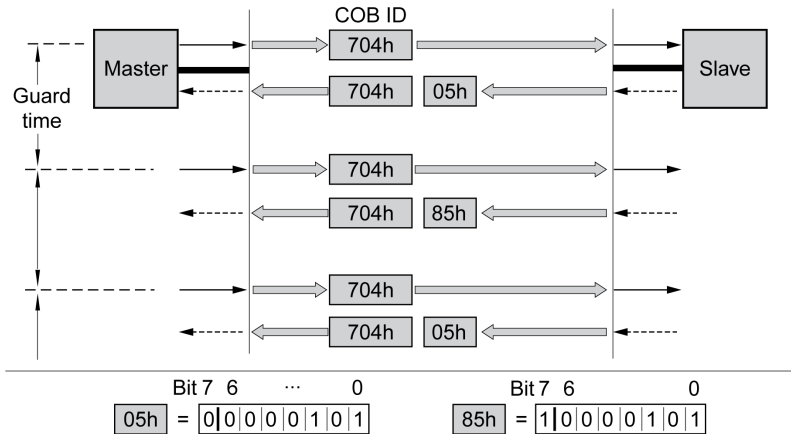
The COB ID for each NMT slave is determined on the basis of the node address:

COB ID = function code NMT error control (700<sub>h</sub>) + Node-ID.

### Structure of the NMT Message

After a request from the NMT master, the NMT slave responds with one data byte.

Acknowledgement of the NMT slave:



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

- 4 (04<sub>h</sub>): "Stopped"
- 5 (05<sub>h</sub>): "Operational"
- 127 (7F<sub>h</sub>): "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<sub>h</sub>)
- Life time factor (100D<sub>h</sub>)

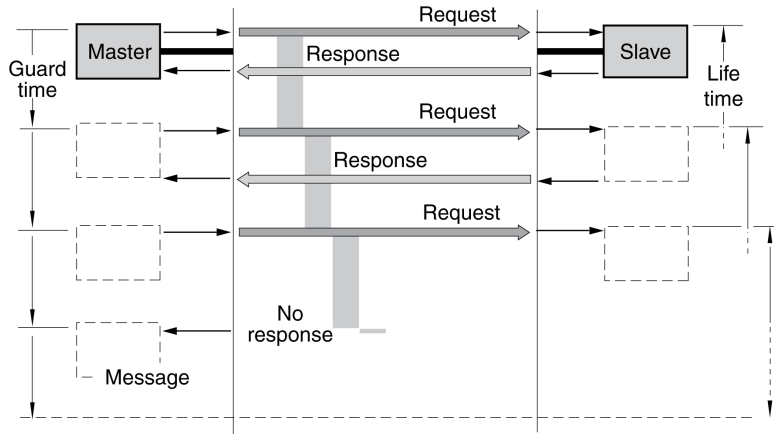
### Connection Error

The NMT master signals a connection error to the master program in the following cases:

- 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.

The illustration below shows an error message after the end of the third cycle because no response was received from an NMT slave.

"Node Guarding" and "Life Guarding" with time intervals:



## NMT Service Heartbeat

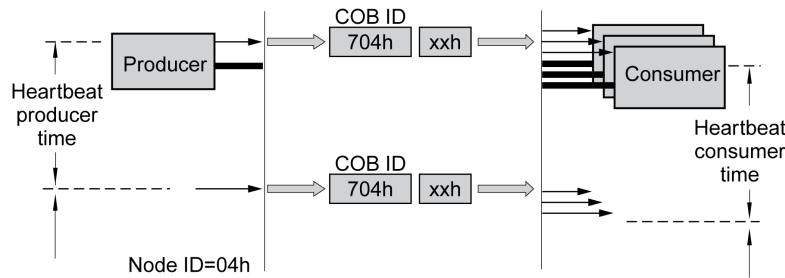
### Description

The optional Heartbeat protocol replaces the Node Guarding/Life Guarding protocol.

A Heartbeat producer transmits a Heartbeat message cyclically at the frequency defined in the object `Producer heartbeat time (1017h)`. One or several consumers can receive this message.

`Producer heartbeat time (1017h) = 0` deactivates sending of Heartbeat messages.

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 (1016h)`, it generates an error message (Heartbeat event). `Consumer heartbeat time (1016h) = 0` deactivates monitoring by a consumer.



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

- 0 (00<sub>h</sub>): "Boot-Up"
- 4 (04<sub>h</sub>): "Stopped"
- 5 (05<sub>h</sub>): "Operational"
- 127 (7F<sub>h</sub>): "Pre-Operational"

### Time Intervals

The time intervals are specified in increments of 1 ms. The values for the producer must be greater than the values for the consumer. Each time the "Heartbeat" message is received, the time interval of the consumer starts again.

### Start of Monitoring

"Heartbeat" monitoring of the producer starts as soon as a time interval is set.

"Heartbeat" monitoring of the consumer starts as soon as the consumer receives the first "Heartbeat" message. A time interval must have been set before.

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



---

# Chapter 3

## Commissioning

---

### Preparation

This chapter describes how to commission the product.

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

#### WARNING

##### LOSS OF CONTROL

- Ensure that connection monitoring is enabled.
- Set the shortest, practical monitoring time cycles to detect communication interruptions as quickly as possible.

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

**NOTE:** Connection monitoring include the heartbeat or node guarding features of CANopen.

#### WARNING

##### UNINTENDED EQUIPMENT OPERATION

- Only start the system if there are no persons or obstructions in the zone of 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 correct word order for 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.**

### Required Components

The following is required for commissioning:

- Commissioning software “Lexium32 DTM Library”  
[http://www.schneider-electric.com/en/download/document/Lexium\\_DTM\\_Library/](http://www.schneider-electric.com/en/download/document/Lexium_DTM_Library/)
- Fieldbus converter for the commissioning software for connection via the commissioning interface
- Lexium 32A - Servo Drive - User Guide

### Address and Baud Rate

Up to 64 devices can be addressed in a CAN bus network segment 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 a unique node address, i.e. a node address can be assigned only once in the network.

After the initialization, the CAN interface must be configured. You must assign a unique network address (node address) to each device. The transmission rate (baud rate) must be the same for all devices in the network.

- Enter the network address. The network address is stored in the parameter `CANaddress (C D R D)`.
- Set the transmission rate in the parameter `CANbaud (C a b d)` to meet the requirements of your network.

The settings are valid for CANopen and for CANmotion.

Parameter name HMI menu HMI name	Description	Unit Minimum value Factory setting Maximum value	Data type R/W Persistent Expert	Parameter address via fieldbus
CANaddress <i>C o n F</i> → <i>C o n -</i> <i>C o n F</i> → <i>F S u -</i> <i>C o R d</i>	CANopen address (node number) Changed settings become active the next time the product is powered on.	- 1 - 127	UINT16 R/W per. -	CANopen 3041:2 <sub>n</sub> Modbus 16644
CANbaud <i>C o n F</i> → <i>C o n -</i> <i>C o n F</i> → <i>F S u -</i> <i>C o b d</i>	CANopen baud rate <b>50 kBaud / ? 5 0</b> : 50 kBaud <b>125 kBaud / ? 1 2 5</b> : 125 kBaud <b>250 kBaud / ? 2 5 0</b> : 250 kBaud <b>500 kBaud / ? 5 0 0</b> : 500 kBaud <b>1 MBaud / 1 0 0 0</b> : 1 MBaud Changed settings become active the next time the product is powered on.	- 50 250 1000	UINT16 R/W per. -	CANopen 3041:3 <sub>n</sub> Modbus 16646



---

# Chapter 4

## Operating States and Operating Modes

---

### What Is in This Chapter?

This chapter contains the following sections:

Section	Topic	Page
4.1	Operating States	50
4.2	Operating Modes	54

## Section 4.1

### Operating States

---

#### What Is in This Section?

This section contains the following topics:

Topic	Page
Indication of the Operating State	51
Changing the Operating State	53

## Indication of the Operating State

### Status Word

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
<code>_DCOMstatus</code>	DriveCom status word Bit assignments: Bit 0: Operating state Ready To Switch On Bit 1: Operating state Switched On Bit 2: Operating state Operation Enabled Bit 3: Operating state Fault Bit 4: Voltage Enabled Bit 5: Operating state Quick Stop Bit 6: Operating state Switch On Disabled Bit 7: Error of error class 0 Bit 8: HALT request active Bit 9: Remote Bit 10: Target Reached Bit 11: Internal Limit Active Bit 12: Operating mode-specific Bit 13: <code>x_err</code> Bit 14: <code>x_end</code> Bit 15: <code>ref_ok</code>	- - - -	UINT16 R/- - -	CANopen 6041:0 <sub>h</sub> Modbus 6916

### 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 Enabled	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	0	0	0

### Bit 4

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

### Bit 7

Bit 7 is 1 if parameter `_WarnActive` contains an error message of error class 0. The movement is not interrupted. The bit remains set to 1 as long as the message is contained in parameter `_WarnActive`. The bit remains set to 1 for at least 100 ms, even if an error message of error class 0 is active for a shorter time. The bit is immediately reset to 0 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 to 1, the device carries out commands via the fieldbus. If Bit 9 is reset to 0, the device is controlled via a different access channel. In such a case, it is still possible to read or write parameters via the fieldbus.

**Bit 10**

Bit 10 is used for monitoring the operating mode. Details can be found in the chapters on the individual operating modes.

**Bit 11**

The meaning of bit 11 can be set via the parameter `DS402intLim`.

**Bit 12**

Bit 12 is used for monitoring the operating mode. Details can be found in the chapters on the individual operating modes.

**Bit 13**

Bit 13 is only set to 1 in the case of an error which needs to be corrected prior to further processing. The device responds corresponding to the 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 set to 1 if the motor has a valid zero point, for example as a result of a reference movement. A valid zero point remains valid even if the power stage is disabled.

## Changing the Operating State

### Control Word

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
<code>DCOMcontrol</code>	DriveCom control word See chapter Operation, Operating States, for bit assignment information. Bit 0: Operating state Switch On Bit 1: Enable Voltage Bit 2: Operating state Quick Stop Bit 3: Enable Operation Bits 4 ... 6: Operating mode-specific Bit 7: Fault Reset Bit 8: Halt Bit 9: Operating mode-specific Bits 10 ... 15: Reserved (must be 0) Changed settings become active immediately.	- - - -	UINT16 R/W - -	CANopen 6040:0 <sub>n</sub> Modbus 6914

### 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	<b>4</b> Ready To Switch On	0	X	1	1	0
Switch On	T3	<b>5</b> Switched On	0	0	1	1	1
Disable Voltage	T7, T9, T10, T12	<b>3</b> Switch On Disabled	0	X	X	0	X
Quick Stop	T7, T10 T11	<b>3</b> Switch On Disabled <b>7</b> Quick Stop Active	0	X	0	1	X
Disable Operation	T5	<b>5</b> Switched On	0	0	1	1	1
Enable Operation	T4, T16	<b>6</b> Operation Enabled	0	1	1	1	1
Fault Reset	T15	<b>3</b> 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

Bit 8 is used to trigger a "Halt". Set bit 8 to 1 to stop a movement with "Halt".

### Bit 9

Bit 9 is used for the operating mode-specific settings. Details can be found in the descriptions of the individual operating modes in this chapter.

### Bits 10 ... 15

Reserved.

## Section 4.2

### Operating Modes

---

#### What Is in This Section?

This section contains the following topics:

Topic	Page
Starting and Changing an Operating Mode	55
Operating Mode Jog	56
Operating Mode Profile Torque	57
Operating Mode Profile Velocity	58
Operating Mode Profile Position	59
Operating Mode Interpolated Position	61
Operating Mode Homing	63
Examples for Node Address 1	65

## Starting and Changing an Operating Mode

The parameter `DCOMopmode` is used to set 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
<code>DCOMopmode</code>	Operating mode <b>-6 / Manual Tuning / Autotuning:</b> Manual Tuning or Autotuning <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>7 / Interpolated Position:</b> Interpolated Position <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 Changed settings become active immediately. * Datatype for CANopen: INT8	- -6 - 7	INT16* R/W - -	CANopen 6060:0 <sub>h</sub> Modbus 6918

The parameter `_DCOMopmode_act` can be used to read 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
<code>_DCOMopmd_act</code>	Active operating mode <b>-6 / Manual Tuning / Autotuning:</b> Manual Tuning / Autotuning <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>7 / Interpolated Position:</b> Interpolated Position <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 * Datatype for CANopen: INT8	- -6 - 10	INT16* R/- - -	CANopen 6061:0 <sub>h</sub> Modbus 6920

## 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 R/W - -	CANopen 301B:9 <sub>h</sub> Modbus 6930

### Control Word

The operating mode-specific bits 4, 5, 6 and 9 are reserved in this operating mode and must be set to 0. For the common bits of the Control Word see chapter Changing the Operating State ([see page 53](#)).

### Status Word

The operating mode-specific bits 10 and 12 are reserved in this operating mode.

For the common bits of the Status Word see chapter Indication of the Operating State ([see page 51](#)).

### 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 a detected error



## 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 if the reference value source (parameter `PTtq_reference`) is set to **Parameter PTtq\_target**.

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 R/W - -	CANopen 6071:0 <sub>h</sub> Modbus 6944

See the product user guide of the drive if the reference value source is set to **Analog Input** or **PTI Interface**.

### Control Word

The operating mode-specific bits 4, 5, 6 and 9 are reserved in this operating mode and must be set to 0. For the common bits of the Control Word see chapter Changing the Operating State (*see page 53*).

### Status Word

Parameter <code>DCOMstatus</code>	Meaning
Bit 10	0: Target torque not reached 1: Target torque reached
Bit 12	Reserved

For the common bits of the Status Word see chapter Indication of the Operating State (*see page 51*).

### 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 a detected error

## 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 if the reference value source (parameter `PVv_reference`) is set to **Parameter PVv\_target**.

Parameter name HMI menu HMI name	Description	Unit Minimum value Factory setting Maximum value	Data type R/W Persistent Expert	Parameter address via fieldbus
<code>PVv_target</code>	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.	<code>usr_v</code> - 0 -	INT32 R/W - -	CANopen 60FF:0 <sub>h</sub> Modbus 6938

See the product user guide of the drive if the reference value source is set to **Analog Input**.

### Control Word

The operating mode-specific bits 4, 5, 6 and 9 are reserved in this operating mode and must be set to 0. For the common bits of the Control Word see chapter Changing the Operating State (*see page 53*).

### Status Word

Parameter <code>DCOMstatus</code>	Meaning
Bit 10	0: Target velocity not reached 1: Target velocity reached
Bit 12	0: Velocity = >0 1: Velocity = 0

For the common bits of the Status Word see chapter Indication of the Operating State (*see page 51*).

### 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 a detected error

## 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.

Parameter name HMI menu HMI name	Description	Unit Minimum value Factory setting Maximum value	Data type R/W Persistent Expert	Parameter address via fieldbus
<code>PPp_target</code>	Target position for operating mode Profile Position Minimum/maximum values depend on: - Scaling factor - Software limit switches (if they are activated) Changed settings become active immediately.	<code>usr_p</code> - - -	INT32 R/W - -	CANopen 607A:0 <sub>h</sub> Modbus 6940
<code>PVv_target</code>	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.	<code>usr_v</code> - 0 -	INT32 R/W - -	CANopen 60FF:0 <sub>h</sub> Modbus 6938

### Control Word

Bit 9: Change on setpoint	Bit 5: Change setpoint immediately	Bit 4: New setpoint	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 target position.
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 target position.
x	1	0->1	Starts a movement to a target position. Target values transmitted during a movement become immediately effective and are immediately executed.

Parameter value	Meaning
Bit 6: Absolute / relative	0: Absolute movement 1: Relative movement

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

For the common bits of the Control Word see chapter Changing the Operating State (*see page 53*).

### Status Word

Parameter <code>DCOMstatus</code>	Meaning
Bit 10	0: Target position not reached 1: Target position reached
Bit 12	0: New position possible 1: New target position accepted

For the common bits of the Status Word see chapter Indication of the Operating State (*see page 51*).

### 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 a detected error

## Operating Mode Interpolated Position

### Starting the Operating Mode

An initialization sequence must be written to start the operating mode. After the initialization sequence, the operating mode can be started via the control word.

In the operating mode Interpolated Position, the scaling factor of the user-defined unit `usr_p` must be set to 1 rpm/131072. Among other things, this scaling factor is written by means of the initialization sequence.

Index	Subindex	Length in bytes	Value	Meaning
1400 <sub>h</sub>	1 <sub>h</sub>	4	80000200 <sub>h</sub> + node id	Deactivate R_PDO1
1800 <sub>h</sub>	1 <sub>h</sub>	4	80000180 <sub>h</sub> + node id	Deactivate T_PDO1
1401 <sub>h</sub>	1 <sub>h</sub>	4	00000300 <sub>h</sub> + node id	Activate R_PDO2
1801 <sub>h</sub>	1 <sub>h</sub>	4	00000280 <sub>h</sub> + node id	Activate T_PDO2
1402 <sub>h</sub>	1 <sub>h</sub>	4	80000400 <sub>h</sub> + node id	Deactivate R_PDO3
1802 <sub>h</sub>	1 <sub>h</sub>	4	80000380 <sub>h</sub> + node id	Deactivate T_PDO3
1403 <sub>h</sub>	1 <sub>h</sub>	4	80000500 <sub>h</sub> + node id	Deactivate R_PDO4
1803 <sub>h</sub>	1 <sub>h</sub>	4	80000480 <sub>h</sub> + node id	Deactivate T_PDO4
1401 <sub>h</sub>	2 <sub>h</sub>	1	1 <sub>h</sub>	Activate cyclic transmission of R_PDO2
1801 <sub>h</sub>	2 <sub>h</sub>	1	1 <sub>h</sub>	Activate cyclic transmission of T_PDO2
6040 <sub>h</sub>	0 <sub>h</sub>	2	0 <sub>h</sub>	Control word = 0
6040 <sub>h</sub>	0 <sub>h</sub>	2	80 <sub>h</sub>	Perform Fault Reset
1601 <sub>h</sub>	0 <sub>h</sub>	1	0 <sub>h</sub>	Change PDO mapping for R_PDO2
1601 <sub>h</sub>	1 <sub>h</sub>	4	60400010 <sub>h</sub>	Map control word
1601 <sub>h</sub>	2 <sub>h</sub>	4	60C10120 <sub>h</sub>	Map reference position for Interpolated Position
1601 <sub>h</sub>	0 <sub>h</sub>	1	2 <sub>h</sub>	Finalize mapping for R_PDO2
1a01 <sub>h</sub>	0 <sub>h</sub>	1	0 <sub>h</sub>	Change PDO mapping for T_PDO2
1a01 <sub>h</sub>	1 <sub>h</sub>	4	60410010 <sub>h</sub>	Map status word
1a01 <sub>h</sub>	2 <sub>h</sub>	4	60640020 <sub>h</sub>	Map Position actual Value
1a01 <sub>h</sub>	0 <sub>h</sub>	1	2 <sub>h</sub>	Finalize mapping for T_PDO2
3006 <sub>h</sub>	7 <sub>h</sub>	4	20000 <sub>h</sub>	Position scaling: denominator
3006 <sub>h</sub>	8 <sub>h</sub>	4	1 <sub>h</sub>	Position scaling: numerator
6060 <sub>h</sub>	0 <sub>h</sub>	1	7 <sub>h</sub>	Select operating mode Interpolated Position
3006 <sub>h</sub>	3D <sub>h</sub>	2	1 <sub>h</sub>	Must be written for reasons of compatibility
60C2 <sub>h</sub>	1 <sub>h</sub>	1	2 <sub>h</sub>	Cycle time 2 ms (example)
3012 <sub>h</sub>	6 <sub>h</sub>	2	3E8 <sub>h</sub>	Velocity feed-forward control 100% CTRL1
3013 <sub>h</sub>	6 <sub>h</sub>	2	3E8 <sub>h</sub>	Velocity feed-forward control 100% CTRL2
3006 <sub>h</sub>	6 <sub>h</sub>	2	1 <sub>h</sub>	Suppress error message for LIMP or LIMN when the power stage is enabled
3022 <sub>h</sub>	4 <sub>h</sub>	2	1 <sub>h</sub>	Tolerance for synchronization mechanism (example)
3022 <sub>h</sub>	5 <sub>h</sub>	2	2 <sub>h</sub>	Activate synchronization mechanism

### Control Word

Parameter DCOMcontrol	Meaning
Bit 4	0: Terminate operating mode 1: Start operating mode  <b>NOTE:</b> If the control word is transmitted via SDO, the power stage must be enabled first. After that, the operating mode can be started with a rising edge.
Bits 5, 6 and 9	Reserved (must be set to 0)

For the common bits of the Control Word see chapter Changing the Operating State (*see page 53*).

### Status Word

Parameter DCOMstatus	Meaning
Bit 10	0: Halt = 0: Position not (yet) reached Halt = 1: Motor decelerates 1: Halt = 0: Position reached Halt = 1: Motor standstill
Bit 12	0: Operating mode terminated 1: Operating mode started

For the common bits of the Status Word see chapter Indication of the Operating State (*see page 51*).

### Terminating the Operating Mode

The operating mode is terminated under the following conditions is met:

- Bit 4 of the control word = 0
- Stop caused by "Halt" or "Quick Stop"
- Stop caused by a detected error

## 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.

Parameter name HMI menu HMI name	Description	Unit Minimum value Factory setting Maximum value	Data type R/W Persistent Expert	Parameter address via fieldbus
HMmethod	Homing method 1: LIMN with index pulse 2: LIMP with index pulse 7: REF+ with index pulse, inv., outside 8: REF+ with index pulse, inv., inside 9: REF+ with index pulse, not inv., inside 10: REF+ with index pulse, not inv., outside 11: REF- with index pulse, inv., outside 12: REF- with index pulse, inv., inside 13: REF- with index pulse, not inv., inside 14: REF- with index pulse, not inv., outside 17: LIMN 18: LIMP 23: REF+, inv., outside 24: REF+, inv., inside 25: REF+, not inv., inside 26: REF+, not inv., outside 27: REF-, inv., outside 28: REF-, inv., inside 29: REF-, not inv., inside 30: REF-, not inv., outside 33: Index pulse neg. direction 34: Index pulse pos. direction 35: Position setting  Abbreviations: REF+: Search movement in pos. direction REF-: Search movement in neg. direction inv.: Invert direction in switch not inv.: Direction not inverted in switch outside: Index pulse / distance outside switch inside: Index pulse / distance inside switch Changed settings become active immediately. * Datatype for CANopen: INT8	- 1 18 35	INT16* R/W - -	CANopen 6098:0 <sub>h</sub> Modbus 6936

### Control Word

Parameter <code>DCOMcontrol</code>	Meaning
Bit 4	Start Homing
Bits 5, 6 and 9	Reserved (must be set to 0)

For the common bits of the Control Word see chapter Changing the Operating State (*see page 53*).

## Status Word

Parameter DCOMstatus	Meaning
Bit 10	0: Homing not completed 1: Homing completed
Bit 12	1: Homing successfully completed

For the common bits of the Status Word see chapter Indication of the Operating State (*see page 51*).

## 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 a detected error



## Examples for Node Address 1

### Operating Mode Jog

Work step COB ID / data	Object Value
Slow velocity to 100 →601 / 23 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 / 23 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 / 23 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 / 23 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 / 23 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>
<b>(1) The operating mode must be checked until the device has activated the specified operating mode.</b>	

**Operating Mode Profile Torque**

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: 6 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>
Target torque set to 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	
<b>(1)</b> The operating mode must be checked until the device has activated the specified operating mode.	

## Operating Mode Profile Velocity

Work step COB ID / data	Object Value
Activate 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>
Activate 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	
<b>(1)</b> The operating mode must be checked until the device has activated the specified operating mode.	

**Operating Mode Profile Position**

Work step COB ID / data	Object Value
<b>Activate R_PDO2</b> →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>
<b>Activate T_PDO2</b> →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>
<b>Set acceleration to 2000</b> →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>
<b>Set deceleration to 4000</b> →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>
<b>Set target velocity to 4000</b> →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>
<b>NMT Start remote node</b> → 0 / 01 00 <b>T_PDO2 with status word</b> ←281 / 31 66 00 00 00 00	
<b>Enable power stage with R_PDO2</b> →301 / 00 00 00 00 00 00 →301 / 06 00 00 00 00 00 →301 / 0F 00 00 00 00 00 <b>T_PDO2 (operating state: 6 Operation Enabled)</b> ←281 / 37 42 00 00 00 00	
<b>Starting the operating mode</b> →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>
<b>Check operating mode<sup>(1)</sup></b> →601 / 40 61 60 00 00 00 00 00 <b>Operating mode active</b> ←581 / 4F 61 60 00 01 61 01 00	6061 <sub>h</sub>  01 <sub>h</sub>
<b>R_PDO2: Start relative movement with NewSetpoint=1</b> →301 / 5F 00 30 75 00 00 <b>T_PDO2 with status word and position actual value</b> ←281 / 37 12 00 00 00 00 <b>Target position reached</b> ←281 / 37 56 30 75 00 00	
<b>R_PDO2: NewSetpoint=0</b> →301 / 4F 00 30 75 00 00	
<b>(1) The operating mode must be checked until the device has activated the specified operating mode.</b>	

## Operating Mode Homing

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	
<b>(1)</b> The operating mode must be checked until the device has activated the specified operating mode.	



---

# Chapter 5

## Diagnostics and Troubleshooting

---

### What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Fieldbus Communication Error Diagnostics	72
Fieldbus Test	73
Fieldbus Status LEDs	74
Error Diagnostics via Fieldbus	75
CANopen Error Messages	76

## Fieldbus Communication Error Diagnostics

### Checking Connections

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

If the product cannot be addressed via the fieldbus, first verify the connections.

Verify the following connections:

- System power supply
- Supply connections
- Fieldbus cables and wiring
- Fieldbus connection

### Fieldbus Function Test

If the connections are correct, verify that you can address the product on the fieldbus.



## Fieldbus Test

### Baud Rate and Address

If it is impossible to connect to a device, verify 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 Commissioning (*see page 47*).

### Fieldbus Function Test

After configuration of the transmission data, test the 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:

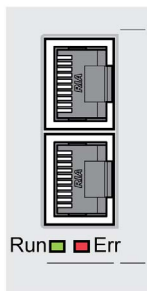
- Power cycle the drive.
- Observe the network messages. After initialization of the bus, the device sends a boot-up message (COB ID  $700_{\text{h}}$  + node ID and 1 data byte with the content  $00_{\text{h}}$ ).

If operation on the network cannot be started, contact your Schneider Electric representative.

## Fieldbus Status LEDs

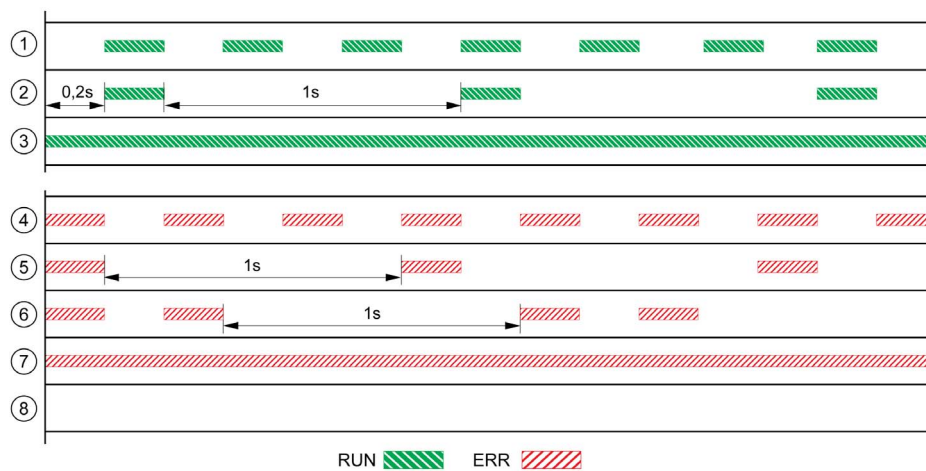
### Overview

The fieldbus status LEDs visualize the status of the fieldbus.



The illustration below shows the fieldbus communication states.

Signals of the CAN bus status LEDs (Run = green, Err = red)



- 1 NMT state PRE-OPERATIONAL
- 2 NMT state STOPPED
- 3 NMT state OPERATIONAL
- 4 Incorrect settings, for example, invalid node address
- 5 Limit for error of error class 0 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.

## Error Diagnostics via Fieldbus

### Message Objects

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

- Object `Statusword` (6041<sub>h</sub>)  
Operating states, see chapter Operating States (*see page 50*)
- Object `EMCY` (80<sub>h</sub>+ Node-ID)  
Error message from a device with error and error code, see chapter Emergency Object Service (*see page 38*)
- Object `Error register` (1001<sub>h</sub>)
- Object `Error code` (603F<sub>h</sub>)  
Vendor-specific error code, see user guide of the drive.
- Devices use the special SDO error message ABORT to signal errors in exchanging messages via SDO.

### Messages on the Device Status

Synchronous and asynchronous errors are distinguished in terms of 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.
- Asynchronous errors  
Asynchronous errors are signaled by the monitoring units in the device as soon as a device error occurs. An asynchronous error is signaled via bit 3, Fault, of the object `statusword` (6041<sub>h</sub>). In the case of errors that cause an interruption of the movement, the device transmits an EMCY message.

## CANopen Error Messages

### Description

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 Emergency Object Service (*see page 38*).

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

### Error Register

The object `Error register (1001h)` indicates the error of a device in bit-coded form. The cause of error can be determined with the error code table. Bit 0 is set to 1 as soon as an error is detected.

bit	Message	Meaning
0	Generic Error	An error has been detected
1	-	Reserved
2	-	Reserved
3	-	Reserved
4	Communication	Network communication error
5	Device Profile Specific	Error during execution as per device profile
6	-	Reserved
7	Manufacturer Specific	Vendor-specific error number

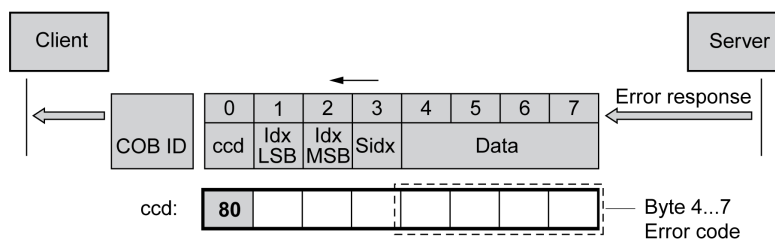
### Error Code Table

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 user guide for the meaning of the error code.

### 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`, byte 4 to byte 7.

SDO error message as a response to an SDO message



The table below lists the error messages that may be detected 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 indeterminable
0504 0005 <sub>h</sub>	No memory available
0601 0000 <sub>h</sub>	Access to object not possible
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

Error code	Meaning
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 above maximum value
0609 0032 <sub>h</sub>	Parameter values below minimum value
0609 0036 <sub>h</sub>	Upper value is less than lower value
0800 0000 <sub>h</sub>	General error. See parameter <code>_ManuSdoAbort</code> after this table. This parameter contains the drive specific error code.
0800 0020 <sub>h</sub>	Data can neither be transmitted to the application nor saved.
0800 0021 <sub>h</sub>	Local control mode, data can neither be transmitted nor saved.
0800 0022 <sub>h</sub>	Data can neither be transmitted nor saved in this device state.
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 0024 <sub>h</sub>	Data not available.

Parameter name HMI menu HMI name	Description	Unit Minimum value Factory setting Maximum value	Data type R/W Persistent Expert	Parameter address via fieldbus
<code>_ManuSdoAbort</code>	CANopen manufacturer-specific SDO abort code Provides more detailed information on a general SDO abort code (0800 0000).	- - - -	UINT16 R/- - -	CANopen 3041:A <sub>h</sub> Modbus 16660



---

# Chapter 6

## Object Dictionary

---

### What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Specifications for the Objects	80
Overview of object group 1000 <sub>h</sub>	81
Assignment Object Group 3000 <sub>h</sub>	84
Assignment Object Group 6000 <sub>h</sub>	94
Details of Object Group 1000 <sub>h</sub>	96

## 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 the entries have 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

### Factory Setting

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 powered off.



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

### Overview

Index	Subindex	Name	Object code	Data type	Access	Description
1000 <sub>h</sub>		Device type	VAR	Unsigned32	RO	Device type and profile
1001 <sub>h</sub>		Error register	VAR	Unsigned8	RO	Error register
1003 <sub>h</sub>		Predefined error field	ARR		RW	Error history, memory for error messages
1003 <sub>h</sub>	00 <sub>h</sub>	Number of errors	VAR	Unsigned8	RW	Number of error entries
1003 <sub>h</sub>	01 <sub>h</sub>	Error field	VAR	Unsigned32	RO	Error number
1005 <sub>h</sub>		COB-ID SYNC	VAR	Unsigned32	RW	Identifier of the synchronization object
1008 <sub>h</sub>		Manufacturer device name	VAR	Visible String8	RO	Manufacturer's designation
1009 <sub>h</sub>		Manufacturer hardware version	VAR	Visible String8	RO	Hardware version
100A <sub>h</sub>		Manufacturer software version	VAR	Visible String8	RO	Software version
100C <sub>h</sub>		Guard time	VAR	Unsigned16	RW	Time span for Node Guarding [ms]
100D <sub>h</sub>		Life time factor	VAR	Unsigned8	RW	Repeat factor for the Node Guarding protocol
1014 <sub>h</sub>		COB-ID EMCY	VAR	Unsigned32	RW	Unsigned16
1015 <sub>h</sub>		Inhibit time EMCY	VAR	Unsigned16	RW	Unsigned16
1016 <sub>h</sub>		Consumer Heartbeat Time	ARR	Unsigned32	RW	Unsigned16
1016 <sub>h</sub>	01 <sub>h</sub>	Consumer Heartbeat Time	VAR	Unsigned32	RW	Time interval and node ID of the "Heartbeat" recipient
1017 <sub>h</sub>		Producer Heartbeat Time	VAR	Unsigned16	RW	Time interval for producer "Heartbeat"
1018 <sub>h</sub>		Identity Object	REC	Identity	RO	Identification object:
1018 <sub>h</sub>	01 <sub>h</sub>	Vendor ID	VAR	Unsigned32	RO	Vendor ID
1018 <sub>h</sub>	02 <sub>h</sub>	Product code	VAR	Unsigned32	RO	Product code
1018 <sub>h</sub>	03 <sub>h</sub>	Revision number	VAR	Unsigned32	RO	Revision number
1029 <sub>h</sub>		Number of elements	ARR	Unsigned8	RO	Number of values for the object
1029 <sub>h</sub>	01 <sub>h</sub>	Communication error	ARR	Unsigned8	RW	Communication error
1200 <sub>h</sub>		1st server SDO parameter	REC	SDO server param.	RO	First server SDO, settings
1200 <sub>h</sub>	01 <sub>h</sub>	COB-ID Client -> Server	VAR	Unsigned32	RO	Identifier client -> server
1200 <sub>h</sub>	02 <sub>h</sub>	COB-ID Server -> Client	VAR	Unsigned32	RO	Identifier server -> client
1201 <sub>h</sub>		2nd server SDO parameter	REC	SDO server param.	RW	Second server SDO, settings
1201 <sub>h</sub>	01 <sub>h</sub>	COB-ID Client -> Server	VAR	Unsigned32	RW	Identifier client -> server
1201 <sub>h</sub>	02 <sub>h</sub>	COB-ID Server -> Client	VAR	Unsigned32	RW	Identifier server -> client
1201 <sub>h</sub>	03 <sub>h</sub>	Node-ID SDO Client	VAR	Unsigned32	RW	Node ID SDO client
1400 <sub>h</sub>		1st receive PDO parameter	REC	PDO comm. param.	RW	First receive PDO (R_PDO1), settings
1400 <sub>h</sub>	01 <sub>h</sub>	COB-ID R_PDO1	VAR	Unsigned32	RW	Identifier of the R_PDO1
1400 <sub>h</sub>	02 <sub>h</sub>	Transmission type R_PDO1	VAR	Unsigned8	RW	Transmission type

Index	Subindex	Name	Object code	Data type	Access	Description
1401 <sub>h</sub>		2nd receive PDO parameter	REC	PDO comm. param.	RW	Second receive PDO (R_PDO2), settings
1401 <sub>h</sub>	01 <sub>h</sub>	COB-ID R_PDO2	VAR	Unsigned32	RW	Identifier of the R_PDO2
1401 <sub>h</sub>	02 <sub>h</sub>	Transmission type R_PDO2	VAR	Unsigned8	RW	Transmission type
1402 <sub>h</sub>		3rd receive PDO parameter	REC	PDO comm. param.	RW	Third receive PDO (R_PDO3), settings
1402 <sub>h</sub>	01 <sub>h</sub>	COB-ID R_PDO3	VAR	Unsigned32	RW	Identifier of the R_PDO3
1402 <sub>h</sub>	02 <sub>h</sub>	Transmission type R_PDO3	VAR	Unsigned8	RW	Transmission type
1403 <sub>h</sub>		4th receive PDO parameter	REC	PDO comm. param.	RW	Fourth receive PDO (R_PDO4), settings
1403 <sub>h</sub>	01 <sub>h</sub>	COB-ID R_PDO4	VAR	Unsigned32	RW	Identifier of the R_PDO4
1403 <sub>h</sub>	02 <sub>h</sub>	Transmission type R_PDO4	VAR	Unsigned8	RW	Transmission type
1600 <sub>h</sub>		1st receive PDO mapping	REC	PDO mapping	RO	PDO mapping for R_PDO1, settings
1600 <sub>h</sub>	01 <sub>h</sub>	1st mapped object R_PDO1	VAR	Unsigned32	RO	First object for the mapping in R_PDO1
1601 <sub>h</sub>		2nd receive PDO mapping	REC	PDO mapping	RO	PDO mapping for R_PDO2, settings
1601 <sub>h</sub>	01 <sub>h</sub>	1st mapped object R_PDO2	VAR	Unsigned32	RO	First object for the mapping in R_PDO2
1601 <sub>h</sub>	02 <sub>h</sub>	2nd mapped object R_PDO2	VAR	Unsigned32	RO	Second object for the mapping in R_PDO2
1602 <sub>h</sub>		3rd receive PDO mapping	REC	PDO mapping	RO	PDO mapping for R_PDO3, settings
1602 <sub>h</sub>	01 <sub>h</sub>	1st mapped object R_PDO3	VAR	Unsigned32	RO	First object for the mapping in R_PDO3
1602 <sub>h</sub>	02 <sub>h</sub>	2nd mapped object R_PDO3	VAR	Unsigned32	RO	Second object for the mapping in R_PDO3
1603 <sub>h</sub>		4th receive PDO mapping	REC	PDO mapping	RW	PDO mapping for R_PDO3, settings
1603 <sub>h</sub>	01 <sub>h</sub>	1st mapped object R_PDO4	VAR	Unsigned32	RW	First object for the mapping in R_PDO4
1603 <sub>h</sub>	02 <sub>h</sub>	2nd mapped object R_PDO4	VAR	Unsigned32	RW	Second object for the mapping in R_PDO4
1603 <sub>h</sub>	03 <sub>h</sub>	3rd mapped object R_PDO4	VAR	Unsigned32	RW	Third object for mapping in R_PDO4
1800 <sub>h</sub>		1st transmit PDO parameter	REC	PDO comm. param.	RW	First transmit PDO (T_PDO1), settings
1800 <sub>h</sub>	01 <sub>h</sub>	COB-ID T_PDO1	VAR	Unsigned32	RW	Identifier of the T_PDO1
1800 <sub>h</sub>	02 <sub>h</sub>	Transmission type T_PDO1	VAR	Unsigned8	RW	Transmission type
1800 <sub>h</sub>	03 <sub>h</sub>	Inhibit time T_PDO1	VAR	Unsigned16	RW	Inhibit time for locking bus access (1=100µs)
1800 <sub>h</sub>	04 <sub>h</sub>	Reserved T_PDO1	VAR	Unsigned8	RW	Priority for CAN bus arbitration ([0-7]).
1800 <sub>h</sub>	05 <sub>h</sub>	Event timer T_PDO1	VAR	Unsigned16	RW	Time span for event triggering (1=1 ms)
1801 <sub>h</sub>		2nd transmit PDO parameter	REC	PDO comm. param.	RW	Second transmit PDO (T_PDO2), settings
1801 <sub>h</sub>	01 <sub>h</sub>	COB-ID T_PDO2	VAR	Unsigned32	RW	Identifier of the T_PDO2
1801 <sub>h</sub>	02 <sub>h</sub>	Transmission type T_PDO2	VAR	Unsigned8	RW	Transmission type
1801 <sub>h</sub>	03 <sub>h</sub>	Inhibit time T_PDO2	VAR	Unsigned16	RW	Inhibit time for locking bus access (1=100µs)

Index	Subindex	Name	Object code	Data type	Access	Description
1801 <sub>h</sub>	04 <sub>h</sub>	Reserved T_PDO2	VAR	Unsigned8	RW	Reserved
1801 <sub>h</sub>	05 <sub>h</sub>	Event timer T_PDO2	VAR	Unsigned16	RW	Time span for event triggering (1=1 ms)
1802 <sub>h</sub>		3rd transmit PDO parameter	REC	PDO comm. param.	RW	Third transmit PDO (T_PDO3), settings
1802 <sub>h</sub>	01 <sub>h</sub>	COB-ID T_PDO3	VAR	Unsigned32	RW	Identifier of the T_PDO3
1802 <sub>h</sub>	02 <sub>h</sub>	Transmission type T_PDO3	VAR	Unsigned8	RW	Transmission type
1802 <sub>h</sub>	03 <sub>h</sub>	Inhibit time T_PDO3	VAR	Unsigned16	RW	Inhibit time for locking bus access (1=100µs)
1802 <sub>h</sub>	04 <sub>h</sub>	Reserved T_PDO3	VAR	Unsigned8	RW	Reserved
1802 <sub>h</sub>	05 <sub>h</sub>	Event timer T_PDO3	VAR	Unsigned16	RW	Time span for event triggering (1=1 ms)
1803 <sub>h</sub>		4th transmit PDO parameter	REC	PDO comm. param.	RW	Fourth transmit PDO (T_PDO4), settings
1803 <sub>h</sub>	01 <sub>h</sub>	COB-ID T_PDO4	VAR	Unsigned32	RW	Identifier of the T_PDO4
1803 <sub>h</sub>	02 <sub>h</sub>	Transmission type T_PDO4	VAR	Unsigned8	RW	Transmission type
1803 <sub>h</sub>	03 <sub>h</sub>	Inhibit time T_PDO4	VAR	Unsigned16	RW	Inhibit time for locking bus access (1=100µs)
1803 <sub>h</sub>	04 <sub>h</sub>	Reserved T_PDO4	VAR	Unsigned8	RO	Reserved
1803 <sub>h</sub>	05 <sub>h</sub>	Event timer T_PDO4	VAR	Unsigned16	RW	Time span for event triggering (1=1 ms)
1A00 <sub>h</sub>		1st transmit PDO mapping	REC	PDO mapping	RW	PDO mapping for T_PDO1, settings
1A00 <sub>h</sub>	01 <sub>h</sub>	1st mapped object T_PDO1	VAR	Unsigned32	RO	First object for the mapping in T_PDO1
1A01 <sub>h</sub>		2nd transmit PDO mapping	REC	PDO mapping	RW	PDO mapping for T_PDO2, settings
1A01 <sub>h</sub>	01 <sub>h</sub>	1st mapped object T_PDO2	VAR	Unsigned32	RO	First object for the mapping in T_PDO2
1A01 <sub>h</sub>	02 <sub>h</sub>	2nd mapped object T_PDO2	VAR	Unsigned32	RO	Second object for the mapping in T_PDO2
1A02 <sub>h</sub>		3rd transmit PDO mapping	REC	PDO mapping	RW	PDO mapping for T_PDO3, settings
1A02 <sub>h</sub>	01 <sub>h</sub>	1st mapped object T_PDO3	VAR	Unsigned32	RO	First object for the mapping in T_PDO3
1A02 <sub>h</sub>	02 <sub>h</sub>	2nd mapped object T_PDO3	VAR	Unsigned32	RO	Second object for the mapping in T_PDO3
1A03 <sub>h</sub>		4th transmit PDO mapping	REC	PDO mapping	RW	PDO mapping for T_PDO4, settings
1A03 <sub>h</sub>	01 <sub>h</sub>	1st mapped object T_PDO4	VAR	Unsigned32	RW	First object for the mapping in T_PDO4
1A03 <sub>h</sub>	02 <sub>h</sub>	2nd mapped object T_PDO4	VAR	Unsigned32	RW	Second object for the mapping in T_PDO4
1A03 <sub>h</sub>	03 <sub>h</sub>	3rd mapped object T_PDO4	VAR	Unsigned32	RW	Third object for the mapping in T_PDO4
1A03 <sub>h</sub>	04 <sub>h</sub>	4th mapped object T_PDO4	VAR	Unsigned32	RW	Fourth object for the mapping in T_PDO4

## Assignment Object Group 3000<sub>h</sub>

### Overview

The product provides corresponding parameters for the CANopen object group 3000<sub>h</sub>.

Address	Object	PDO	Data type	Parameter name
3001:1 <sub>h</sub>	Firmware number of device	-	UINT32	_prgNoDEV
3001:2 <sub>h</sub>	Firmware version of device	-	UINT16	_prgVerDEV
3001:4 <sub>h</sub>	Firmware revision of device	-	UINT16	_prgRevDEV
3001:C <sub>h</sub>	Access channel information	T_PDO	UINT16	_AccessInfo
3001:E <sub>h</sub>	Locking other access channels	-	UINT16	AccessLock
3001:43 <sub>h</sub>	User-specific data 1	-	UINT32	UsrAppDataMem1
3001:44 <sub>h</sub>	User-specific data 2	-	UINT32	UsrAppDataMem2
3002:12 <sub>h</sub>	Hardware version of control board	T_PDO	UINT16	_hwVersCPU
3002:14 <sub>h</sub>	Hardware version of power stage	T_PDO	UINT16	_hwVersPS
3004:1 <sub>h</sub>	Save parameter values to the non-volatile memory	-	UINT16	PAReprSave
3004:2 <sub>h</sub>	Restore factory settings (default values)	-	UINT16	PARfactorySet
3004:7 <sub>h</sub>	Reset control loop parameters	-	UINT16	PAR_CTRLreset
3004:8 <sub>h</sub>	Reset user parameters	-	UINT16	PARuserReset
3004:14 <sub>h</sub>	Recalculation of parameters with user-defined units	-	UINT16	PAR_ScalingStart
3004:15 <sub>h</sub>	Status of recalculation of the parameters with user-defined units	T_PDO	UINT16	_PAR_ScalingState
3004:16 <sub>h</sub>	Additional information on error detected during recalculation	T_PDO	UINT32	_PAR_ScalingError
3004:1D <sub>h</sub>	Configuration modification monitoring	-	UINT16	MON_ConfModification
3005:4 <sub>h</sub>	Enabling the power stage as set via IO_AutoEnable even after error	-	UINT16	IO_AutoEnaConfig
3005:5 <sub>h</sub>	Commutation monitoring	-	UINT16	MON_commutat
3005:6 <sub>h</sub>	Enabling the power stage at PowerOn	-	UINT16	IO_AutoEnable
3005:7 <sub>h</sub>	Additional time delay for releasing the holding brake	-	INT16	BRK_AddT_release
3005:8 <sub>h</sub>	Additional time delay for applying the holding brake	-	INT16	BRK_AddT_apply
3005:9 <sub>h</sub>	Selection of type of braking resistor	-	UINT16	RESint_ext
3005:A <sub>h</sub>	Error response to missing mains phase	-	UINT16	ErrorResp_Flt_AC
3005:B <sub>h</sub>	Error response to excessively high load-dependent position deviation	-	UINT16	ErrorResp_p_dif
3005:F <sub>h</sub>	Detection and monitoring of mains phases	-	UINT16	MON_MainsVolt
3005:10 <sub>h</sub>	Ground fault monitoring	-	UINT16	MON_GroundFault
3005:11 <sub>h</sub>	Maximum permissible switch-on time of external braking resistor	-	UINT16	RESext_ton
3005:12 <sub>h</sub>	Nominal power of external braking resistor	-	UINT16	RESext_P
3005:13 <sub>h</sub>	Resistance value of external braking resistor	-	UINT16	RESext_R
3005:16 <sub>h</sub>	Adjustment of absolute position of encoder 1	-	INT32	ENC1_adjustment
3005:20 <sub>h</sub>	Mains reactor	-	UINT16	Mains_reactor
3005:21 <sub>h</sub>	Shifting of the encoder working range	-	UINT16	ShiftEncWorkRang
3005:22 <sub>h</sub>	Error response to 100% I2t braking resistor	-	UINT16	ErrorResp_I2tRES
3005:23 <sub>h</sub>	Simulation of absolute position at power cycling	-	UINT16	SimAbsolutePos
3005:26 <sub>h</sub>	DC bus compatibility LXM32 and ATV32	-	UINT16	DCbus_compat
3005:34 <sub>h</sub>	Additional 'Fault Reset' for the signal input function 'Enable'	-	UINT16	IO_FaultResOnEnaInp
3005:3A <sub>h</sub>	Error response to detected error with quasi absolute position	-	UINT16	ErrorResp_QuasiAbs
3005:3C <sub>h</sub>	Error response to excessively high load-dependent velocity deviation	-	UINT16	ErrorResp_v_dif
3005:3D <sub>h</sub>	DC bus overvoltage monitoring threshold	-	UINT16	MON_DCbusVdcThresh

Address	Object	PDO	Data type	Parameter name
3006:1 <sub>h</sub>	Acceleration and deceleration of the motion profile for velocity	R_PDO	UINT16	RAMP_v_sym
3006:2 <sub>h</sub>	Acceleration and deceleration for the Drive Profile Lexium	-	UINT32	RAMPaccdec
3006:3 <sub>h</sub>	Activation of software limit switches	-	UINT16	MON_SW_Limits
3006:6 <sub>h</sub>	Response to active limit switch during enabling of power stage	-	UINT16	IOsigRespOfPS
3006:7 <sub>h</sub>	Position scaling: Denominator	-	INT32	ScalePOSdenom
3006:8 <sub>h</sub>	Position scaling: Numerator	-	INT32	ScalePOSnum
3006:C <sub>h</sub>	Inversion of direction of movement	-	UINT16	InvertDirOfMove
3006:D <sub>h</sub>	Jerk limitation of the motion profile for velocity	-	UINT16	RAMP_v_jerk
3006:E <sub>h</sub>	Signal evaluation for reference switch	-	UINT16	IOsigREF
3006:F <sub>h</sub>	Signal evaluation for negative limit switch	-	UINT16	IOsigLIMN
3006:10 <sub>h</sub>	Signal evaluation for positive limit switch	-	UINT16	IOsigLIMP
3006:12 <sub>h</sub>	Deceleration ramp for Quick Stop	-	UINT32	RAMPquickstop
3006:16 <sub>h</sub>	Absolute positioning only after homing	-	UINT16	AbsHomeRequest
3006:18 <sub>h</sub>	Quick Stop option code	-	INT16	LIM_QStopReact
3006:19 <sub>h</sub>	Monitoring of position deviation	-	UINT16	MON_p_DiffWin
3006:1A <sub>h</sub>	Monitoring of velocity deviation	-	UINT32	MON_v_DiffWin
3006:1B <sub>h</sub>	Monitoring of velocity threshold	R_PDO	UINT32	MON_v_Threshold
3006:1C <sub>h</sub>	Monitoring of current threshold	R_PDO	UINT16	MON_I_Threshold
3006:1D <sub>h</sub>	Monitoring of time window	-	UINT16	MON_ChkTime
3006:1E <sub>h</sub>	Velocity limitation via input	-	UINT32	IO_v_limit
3006:21 <sub>h</sub>	Velocity scaling: Denominator	-	INT32	ScaleVELdenom
3006:22 <sub>h</sub>	Velocity scaling: Numerator	-	INT32	ScaleVELnum
3006:26 <sub>h</sub>	Timeout time for standstill window monitoring	-	UINT16	MON_p_winTout
3006:27 <sub>h</sub>	Current limitation via input	-	UINT16	IO_I_limit
3006:28 <sub>h</sub>	Velocity limit for Zero Clamp	-	UINT32	MON_v_zeroclamp
3006:29 <sub>h</sub>	Maximum load-dependent position deviation (error class 0)	-	UINT16	MON_p_dif_warn
3006:2B <sub>h</sub>	Activation of the motion profile for velocity	-	UINT16	RAMP_v_enable
3006:2C <sub>h</sub>	Activation of the motion profile for torque	-	UINT16	RAMP_tq_enable
3006:2D <sub>h</sub>	Torque window, permissible deviation	-	UINT16	MON_tq_win
3006:2E <sub>h</sub>	Torque window, time	-	UINT16	MON_tq_winTime
3006:30 <sub>h</sub>	Ramp scaling: Denominator	-	INT32	ScaleRAMPdenom
3006:31 <sub>h</sub>	Ramp scaling: Numerator	-	INT32	ScaleRAMPnum
3006:38 <sub>h</sub>	Activation of Modulo function	-	UINT16	MOD_Enable
3006:39 <sub>h</sub>	Minimum position of modulo range	-	INT32	MOD_Min
3006:3A <sub>h</sub>	Maximum position of modulo range	-	INT32	MOD_Max
3006:3B <sub>h</sub>	Direction of absolute movement with Modulo	-	UINT16	MOD_AbsDirection
3006:3C <sub>h</sub>	Multiple ranges for absolute movement with Modulo	-	UINT16	MOD_AbsMultiRng
3006:3E <sub>h</sub>	Maximum load-dependent position deviation	-	INT32	MON_p_dif_load_usr
3006:3F <sub>h</sub>	Monitoring of position deviation	-	INT32	MON_p_DiffWin_usr
3006:40 <sub>h</sub>	Standstill window, permissible control deviation	-	INT32	MON_p_win_usr
3006:41 <sub>h</sub>	Processing mode of backlash compensation	-	UINT16	BLSH_Mode
3006:42 <sub>h</sub>	Position value for backlash compensation	-	INT32	BLSH_Position
3006:44 <sub>h</sub>	Processing time for backlash compensation	-	UINT16	BLSH_Time
3006:47 <sub>h</sub>	Behavior when position limit is reached	-	UINT16	MON_SWLimMode
3006:49 <sub>h</sub>	Response to write command (operating state is not Operation Enabled)	-	UINT16	ResWriComNotOpEn

Address	Object	PDO	Data type	Parameter name
3006:4B <sub>h</sub>	Maximum load-dependent velocity deviation	-	UINT32	MON_VelDiff
3006:4C <sub>h</sub>	Time window for maximum load-dependent velocity deviation	-	UINT16	MON_VelDiff_Time
3007:1 <sub>h</sub>	Function Input DI0	-	UINT16	IOfuncnt_DI0
3007:2 <sub>h</sub>	Function Input DI1	-	UINT16	IOfuncnt_DI1
3007:3 <sub>h</sub>	Function Input DI2	-	UINT16	IOfuncnt_DI2
3007:4 <sub>h</sub>	Function Input DI3	-	UINT16	IOfuncnt_DI3
3007:9 <sub>h</sub>	Function Output DQ0	-	UINT16	IOfuncnt_DQ0
3007:A <sub>h</sub>	Function Output DQ1	-	UINT16	IOfuncnt_DQ1
3008:1 <sub>h</sub>	Physical status of the digital inputs and outputs	T_PDO	UINT16	_IO_act
3008:A <sub>h</sub>	Manual operation of the holding brake	-	UINT16	BRK_release
3008:F <sub>h</sub>	Status of digital inputs	T_PDO	UINT16	_IO_DI_act
3008:10 <sub>h</sub>	Status of digital outputs	T_PDO	UINT16	_IO_DQ_act
3008:11 <sub>h</sub>	Setting the digital outputs directly	R_PDO	UINT16	IO_DQ_set
3008:20 <sub>h</sub>	Debounce time of DI0	-	UINT16	DI_0_Debounce
3008:21 <sub>h</sub>	Debounce time of DI1	-	UINT16	DI_1_Debounce
3008:22 <sub>h</sub>	Debounce time of DI2	-	UINT16	DI_2_Debounce
3008:23 <sub>h</sub>	Debounce time of DI3	-	UINT16	DI_3_Debounce
3008:26 <sub>h</sub>	Status of the inputs for the safety-related function STO	T_PDO	UINT16	_IO_STO_act
3008:27 <sub>h</sub>	Signal evaluation for signal input function Velocity Limitation	-	UINT16	IOsigVelLim
3008:28 <sub>h</sub>	Signal evaluation for signal input function Current Limitation	-	UINT16	IOsigCurrLim
300A:1 <sub>h</sub>	Status of the capture inputs	T_PDO	UINT16	_CapStatus
300A:2 <sub>h</sub>	Capture input 1 configuration	-	UINT16	Cap1Config
300A:3 <sub>h</sub>	Capture input 2 configuration	-	UINT16	Cap2Config
300A:4 <sub>h</sub>	Capture input 1 start/stop	-	UINT16	Cap1Activate
300A:5 <sub>h</sub>	Capture input 2 start/stop	-	UINT16	Cap2Activate
300A:6 <sub>h</sub>	Capture input 1 captured position	T_PDO	INT32	_Cap1Pos
300A:7 <sub>h</sub>	Capture input 2 captured position	T_PDO	INT32	_Cap2Pos
300A:8 <sub>h</sub>	Capture input 1 event counter	T_PDO	UINT16	_Cap1Count
300A:9 <sub>h</sub>	Capture input 2 event counter	T_PDO	UINT16	_Cap2Count
300A:A <sub>h</sub>	Capture input 1 encoder source	-	UINT16	Cap1Source
300A:B <sub>h</sub>	Capture input 2 encoder source	-	UINT16	Cap2Source
300A:17 <sub>h</sub>	Capture input 1 event counter (consistent)	T_PDO	UINT16	_Cap1CountCons
300A:18 <sub>h</sub>	Capture input 1 captured position (consistent)	T_PDO	INT32	_Cap1PosCons
300A:19 <sub>h</sub>	Capture input 2 event counter (consistent)	T_PDO	UINT16	_Cap2CountCons
300A:1A <sub>h</sub>	Capture input 2 captured position (consistent)	T_PDO	INT32	_Cap2PosCons
300A:2B <sub>h</sub>	Capture input 1 event counter at rising edges	T_PDO	UINT16	_Cap1CntRise
300A:2C <sub>h</sub>	Capture input 1 event counter at falling edges	T_PDO	UINT16	_Cap1CntFall
300A:2D <sub>h</sub>	Capture input 2 event counter at rising edges	T_PDO	UINT16	_Cap2CntRise
300A:2E <sub>h</sub>	Capture input 2 event counter at falling edges	T_PDO	UINT16	_Cap2CntFall
300A:2F <sub>h</sub>	Capture inputs 1 and 2 summary of event counters	T_PDO	UINT16	_CapEventCounters
300B:1 <sub>h</sub>	Status of the position register channels	T_PDO	UINT16	_PosRegStatus
300B:2 <sub>h</sub>	Start/stop of position register channel 1	R_PDO	UINT16	PosReg1Start
300B:3 <sub>h</sub>	Start/stop of position register channel 2	R_PDO	UINT16	PosReg2Start
300B:4 <sub>h</sub>	Selection of comparison criterion for position register channel 1	-	UINT16	PosReg1Mode
300B:5 <sub>h</sub>	Selection of comparison criterion for position register channel 2	-	UINT16	PosReg2Mode

Address	Object	PDO	Data type	Parameter name
300B:6 <sub>h</sub>	Selection of source for position register channel 1	-	UINT16	PosReg1Source
300B:7 <sub>h</sub>	Selection of source for position register channel 2	-	UINT16	PosReg2Source
300B:8 <sub>h</sub>	Comparison value A for position register channel 1	R_PDO	INT32	PosReg1ValueA
300B:9 <sub>h</sub>	Comparison value B for position register channel 1	R_PDO	INT32	PosReg1ValueB
300B:A <sub>h</sub>	Comparison value A for position register channel 2	R_PDO	INT32	PosReg2ValueA
300B:B <sub>h</sub>	Comparison value B for position register channel 2	R_PDO	INT32	PosReg2ValueB
300B:C <sub>h</sub>	Start/stop of position register channel 3	R_PDO	UINT16	PosReg3Start
300B:D <sub>h</sub>	Start/stop of position register channel 4	R_PDO	UINT16	PosReg4Start
300B:E <sub>h</sub>	Selection of comparison criterion for position register channel 3	-	UINT16	PosReg3Mode
300B:F <sub>h</sub>	Selection of comparison criterion for position register channel 4	-	UINT16	PosReg4Mode
300B:10 <sub>h</sub>	Selection of source for position register channel 3	-	UINT16	PosReg3Source
300B:11 <sub>h</sub>	Selection of source for position register channel 4	-	UINT16	PosReg4Source
300B:12 <sub>h</sub>	Comparison value A for position register channel 3	R_PDO	INT32	PosReg3ValueA
300B:13 <sub>h</sub>	Comparison value B for position register channel 3	R_PDO	INT32	PosReg3ValueB
300B:14 <sub>h</sub>	Comparison value A for position register channel 4	R_PDO	INT32	PosReg4ValueA
300B:15 <sub>h</sub>	Comparison value B for position register channel 4	R_PDO	INT32	PosReg4ValueB
300B:16 <sub>h</sub>	Start/stop of position register channels	-	UINT16	PosRegGroupStart
300D:2 <sub>h</sub>	Motor type	T_PDO	UINT32	_M_Type
300D:3 <sub>h</sub>	Type of motor encoder	T_PDO	UINT16	_M_Encoder
300D:4 <sub>h</sub>	Maximum permissible speed of rotation/velocity of motor	T_PDO	UINT16	_M_n_max
300D:5 <sub>h</sub>	Nominal speed of rotation/velocity of motor	T_PDO	UINT16	_M_n_nom
300D:6 <sub>h</sub>	Maximum current of motor	T_PDO	UINT16	_M_I_max
300D:7 <sub>h</sub>	Nominal current of motor	T_PDO	UINT16	_M_I_nom
300D:8 <sub>h</sub>	Nominal torque/force of motor	T_PDO	UINT16	_M_M_nom
300D:9 <sub>h</sub>	Maximum torque of motor	T_PDO	UINT16	_M_M_max
300D:A <sub>h</sub>	Nominal voltage of motor	T_PDO	UINT16	_M_U_nom
300D:B <sub>h</sub>	Voltage constant kE of motor	T_PDO	UINT32	_M_kE
300D:C <sub>h</sub>	Moment of inertia of motor	T_PDO	UINT32	_M_Jrot
300D:D <sub>h</sub>	Winding resistance of motor	T_PDO	UINT16	_M_R_UV
300D:E <sub>h</sub>	Inductance q component of motor	T_PDO	UINT16	_M_L_q
300D:F <sub>h</sub>	Inductance d component of motor	T_PDO	UINT16	_M_L_d
300D:10 <sub>h</sub>	Maximum temperature of motor	T_PDO	INT16	_M_T_max
300D:11 <sub>h</sub>	Maximum permissible time for maximum current of motor	T_PDO	UINT16	_M_I2t
300D:13 <sub>h</sub>	Continuous stall current of motor	T_PDO	UINT16	_M_I_0
300D:14 <sub>h</sub>	Number of pole pairs of motor	T_PDO	UINT16	_M_Polepair
300D:16 <sub>h</sub>	Continuous stall torque of motor	T_PDO	UINT16	_M_M_0
300D:19 <sub>h</sub>	Maximum voltage of motor	T_PDO	UINT16	_M_U_max
300D:20 <sub>h</sub>	Holding brake identification	T_PDO	UINT16	_M_HoldingBrake
300D:21 <sub>h</sub>	Holding brake application time	T_PDO	UINT16	_M_BRK_T_apply
300D:22 <sub>h</sub>	Holding brake release time	T_PDO	UINT16	_M_BRK_T_release
300D:23 <sub>h</sub>	Pole pair pitch of motor	T_PDO	UINT16	_M_PolePairPitch
3010:1 <sub>h</sub>	Nominal current of power stage	T_PDO	UINT16	_PS_I_nom
3010:2 <sub>h</sub>	Maximum current of power stage	T_PDO	UINT16	_PS_I_max
3010:3 <sub>h</sub>	Maximum permissible DC bus voltage	T_PDO	UINT16	_PS_U_maxDC
3010:4 <sub>h</sub>	Minimum permissible DC bus voltage	T_PDO	UINT16	_PS_U_minDC

Address	Object	PDO	Data type	Parameter name
3010:6 <sub>h</sub>	Maximum temperature of power stage (error class 0)	T_PDO	INT16	_PS_T_warn
3010:7 <sub>h</sub>	Maximum temperature of power stage	T_PDO	INT16	_PS_T_max
3010:8 <sub>h</sub>	Resistance value of internal braking resistor	T_PDO	UINT16	_RESint_R
3010:9 <sub>h</sub>	Nominal power of internal braking resistor	T_PDO	UINT16	_RESint_P
3010:A <sub>h</sub>	DC bus voltage low threshold for Quick Stop	T_PDO	UINT16	_PS_U_minStopDC
3011:1 <sub>h</sub>	Current controller d component P gain	-	UINT16	_CTRL_KPid
3011:2 <sub>h</sub>	Current controller d component integral action time	-	UINT16	_CTRL_TNid
3011:3 <sub>h</sub>	Current controller q component P gain	-	UINT16	_CTRL_KPiq
3011:4 <sub>h</sub>	Current controller q component integral action time	-	UINT16	_CTRL_TNiq
3011:5 <sub>h</sub>	PID velocity controller: Time constant of D term smoothing filter	-	UINT16	CTRL_vPIDDTime
3011:6 <sub>h</sub>	PID velocity controller: D gain	-	UINT16	CTRL_vPIDDPart
3011:8 <sub>h</sub>	Filter time constant to smooth velocity of motor	-	UINT16	CTRL_TAUnact
3011:9 <sub>h</sub>	Speed of rotation up to which the friction compensation is linear	-	UINT32	CTRL_SpdFric
3011:A <sub>h</sub>	Acceleration feed-forward control	-	UINT16	CTRL_KFAcc
3011:C <sub>h</sub>	Current limitation	R_PDO	UINT16	CTRL_I_max
3011:D <sub>h</sub>	Current for Quick Stop	-	UINT16	LIM_I_maxQSTP
3011:E <sub>h</sub>	Current for Halt	-	UINT16	LIM_I_maxHalt
3011:F <sub>h</sub>	Maximum current for field weakening (d component)	-	UINT16	CTRL_I_max_fw
3011:10 <sub>h</sub>	Velocity limitation	R_PDO	UINT32	CTRL_v_max
3011:14 <sub>h</sub>	Period of time for control loop parameter set switching	-	UINT16	CTRL_ParChgTime
3011:15 <sub>h</sub>	Global gain factor (affects control loop parameter set 1)	-	UINT16	CTRL_GlobGain
3011:16 <sub>h</sub>	Control loop parameter set copying	-	UINT16	CTRL_ParSetCopy
3011:17 <sub>h</sub>	Active control loop parameter set	T_PDO	UINT16	_CTRL_ActParSet
3011:18 <sub>h</sub>	Selection of control loop parameter set at power up	-	UINT16	CTRL_PwrUpParSet
3011:19 <sub>h</sub>	Selection of control loop parameter set (non-persistent)	-	UINT16	CTRL_SelParSet
3011:1A <sub>h</sub>	Condition for parameter set switching	-	UINT16	CLSET_ParSwiCond
3011:1B <sub>h</sub>	Time window for parameter set switching	-	UINT16	CLSET_winTime
3011:1C <sub>h</sub>	Position deviation for control loop parameter set switching	-	UINT16	CLSET_p_DiffWin
3011:1D <sub>h</sub>	Velocity threshold for control loop parameter set switching	-	UINT32	CLSET_v_Threshold
3011:22 <sub>h</sub>	Activation of velocity observer	-	UINT16	CTRL_VelObsActiv
3011:23 <sub>h</sub>	Dynamics of velocity observer	-	UINT16	CTRL_VelObsDyn
3011:24 <sub>h</sub>	Inertia value for velocity observer	-	UINT32	CTRL_VelObsInert
3011:25 <sub>h</sub>	Position deviation for control loop parameter set switching	-	INT32	CLSET_p_DiffWin_usr
3011:26 <sub>h</sub>	Smoothing factor for current controller	-	UINT16	CTRL_SmoothCurr
3012:1 <sub>h</sub>	Velocity controller P gain	-	UINT16	CTRL1_KPn
3012:2 <sub>h</sub>	Velocity controller integral action time	-	UINT16	CTRL1_TNn
3012:3 <sub>h</sub>	Position controller P gain	-	UINT16	CTRL1_KPp
3012:4 <sub>h</sub>	Filter time constant of the reference velocity value filter	-	UINT16	CTRL1_TAUnref
3012:5 <sub>h</sub>	Filter time constant of the reference current value filter	-	UINT16	CTRL1_TAUiref
3012:6 <sub>h</sub>	Velocity feed-forward control	-	UINT16	CTRL1_KFPp
3012:8 <sub>h</sub>	Notch filter 1: Damping	-	UINT16	CTRL1_Nf1damp
3012:9 <sub>h</sub>	Notch filter 1: Frequency	-	UINT16	CTRL1_Nf1freq
3012:A <sub>h</sub>	Notch filter 1: Bandwidth	-	UINT16	CTRL1_Nf1bandw
3012:B <sub>h</sub>	Notch filter 2: Damping	-	UINT16	CTRL1_Nf2damp
3012:C <sub>h</sub>	Notch filter 2: Frequency	-	UINT16	CTRL1_Nf2freq



Address	Object	PDO	Data type	Parameter name
3012:D <sub>h</sub>	Notch filter 2: Bandwidth	-	UINT16	CTRL1_Nf2bandw
3012:E <sub>h</sub>	Overshoot suppression filter: Damping	-	UINT16	CTRL1_Osupdamp
3012:F <sub>h</sub>	Overshoot suppression filter: Time delay	-	UINT16	CTRL1_Osupdelay
3012:10 <sub>h</sub>	Friction compensation: Gain	-	UINT16	CTRL1_Kfric
3013:1 <sub>h</sub>	Velocity controller P gain	-	UINT16	CTRL2_KPn
3013:2 <sub>h</sub>	Velocity controller integral action time	-	UINT16	CTRL2_TNn
3013:3 <sub>h</sub>	Position controller P gain	-	UINT16	CTRL2_KPp
3013:4 <sub>h</sub>	Filter time constant of the reference velocity value filter	-	UINT16	CTRL2_TAU <sub>n</sub> ref
3013:5 <sub>h</sub>	Filter time constant of the reference current value filter	-	UINT16	CTRL2_TAU <sub>i</sub> ref
3013:6 <sub>h</sub>	Velocity feed-forward control	-	UINT16	CTRL2_KFPp
3013:8 <sub>h</sub>	Notch filter 1: Damping	-	UINT16	CTRL2_Nf1damp
3013:9 <sub>h</sub>	Notch filter 1: Frequency	-	UINT16	CTRL2_Nf1freq
3013:A <sub>h</sub>	Notch filter 1: Bandwidth	-	UINT16	CTRL2_Nf1bandw
3013:B <sub>h</sub>	Notch filter 2: Damping	-	UINT16	CTRL2_Nf2damp
3013:C <sub>h</sub>	Notch filter 2: Frequency	-	UINT16	CTRL2_Nf2freq
3013:D <sub>h</sub>	Notch filter 2: Bandwidth	-	UINT16	CTRL2_Nf2bandw
3013:E <sub>h</sub>	Overshoot suppression filter: Damping	-	UINT16	CTRL2_Osupdamp
3013:F <sub>h</sub>	Overshoot suppression filter: Time delay	-	UINT16	CTRL2_Osupdelay
3013:10 <sub>h</sub>	Friction compensation: Gain	-	UINT16	CTRL2_Kfric
3016:3 <sub>h</sub>	Modbus baud rate	-	UINT32	MBbaud
3016:4 <sub>h</sub>	Modbus address	-	UINT16	MBaddress
301B:5 <sub>h</sub>	Bit shift for RefA16 for Drive Profile Lexium	T_PDO	UINT16	_DPL_BitShiftRefA16
301B:6 <sub>h</sub>	Error response to detected data error (DE bit)	-	INT16	ErrorResp_bit_DE
301B:7 <sub>h</sub>	Error response to detected mode error (ME bit)	-	INT16	ErrorResp_bit_ME
301B:8 <sub>h</sub>	Activation of Drive Profile Lexium	-	UINT16	DPL_Activate
301B:9 <sub>h</sub>	Activation of operating mode Jog	R_PDO	UINT16	JOGactivate
301B:13 <sub>h</sub>	DS402 state machine: State transition from 3 to 4	-	UINT16	DS402compatib
301B:16 <sub>h</sub>	Position for Position Setting	-	INT32	Hmp_setP
301B:19 <sub>h</sub>	Error code for detected synchronous errors (ME bit)	T_PDO	UINT16	_ModeError
301B:1B <sub>h</sub>	Error code for detected synchronous errors (DE bit)	T_PDO	UINT16	_DataError
301B:1C <sub>h</sub>	Additional error information of a detected ModeError (ME bit)	T_PDO	UINT16	_ModeErrorInfo
301B:1D <sub>h</sub>	Additional error information of a detected DataError (DE bit)	T_PDO	UINT16	_DataErrorInfo
301B:1E <sub>h</sub>	DS402 status word: Setting for bit 11 (internal limit)	-	UINT16	DS402intLim
301B:1F <sub>h</sub>	Drive Profile Lexium dmControl	R_PDO	UINT16	DPL_dmControl
301B:21 <sub>h</sub>	Drive Profile Lexium RefB32	R_PDO	INT32	DPL_RefB32
301B:22 <sub>h</sub>	Drive Profile Lexium RefA16	R_PDO	INT16	DPL_RefA16
301B:25 <sub>h</sub>	Drive Profile Lexium driveStat	T_PDO	UINT16	_DPL_driveStat
301B:26 <sub>h</sub>	Drive Profile Lexium mfStat	T_PDO	UINT16	_DPL_mfStat
301B:27 <sub>h</sub>	Drive Profile Lexium motionStat	T_PDO	UINT16	_DPL_motionStat
301B:28 <sub>h</sub>	Drive Profile Lexium driveInput	T_PDO	UINT16	_DPL_driveInput
301B:35 <sub>h</sub>	Setting for bit 9 of _DPL_motionStat and _actionStatus	-	UINT16	DPL_intLim
301C:4 <sub>h</sub>	Action word	T_PDO	UINT16	_actionStatus
301C:6 <sub>h</sub>	Modbus address of parameter with invalid value	T_PDO	UINT16	_InvalidParam
301C:7 <sub>h</sub>	Status of monitoring signals	T_PDO	UINT32	_SigActive
301C:8 <sub>h</sub>	Saved status of monitoring signals	T_PDO	UINT32	_SigLatched

Address	Object	PDO	Data type	Parameter name
301C:9 <sub>h</sub>	Code of most recent error of error class 0	T_PDO	UINT16	_LastWarning
301C:A <sub>h</sub>	Operating hours counter	T_PDO	UINT32	_OpHours
301C:B <sub>h</sub>	Active errors of error class 0, bit-coded	T_PDO	UINT32	_WarnActive
301C:C <sub>h</sub>	Saved errors of error class 0, bit-coded	T_PDO	UINT32	_WarnLatched
301C:D <sub>h</sub>	Output power	T_PDO	INT32	_Power_act
301C:E <sub>h</sub>	Mean output power	T_PDO	UINT16	_Power_mean
301C:F <sub>h</sub>	Voltage at DC bus	T_PDO	UINT16	_UDC_act
301C:10 <sub>h</sub>	Temperature of power stage	T_PDO	INT16	_PS_T_current
301C:11 <sub>h</sub>	Temperature of motor	T_PDO	INT16	_M_T_current
301C:12 <sub>h</sub>	Temperature of device	T_PDO	INT16	_DEV_T_current
301C:13 <sub>h</sub>	Overload of braking resistor (I2t)	T_PDO	INT16	_RES_overload
301C:14 <sub>h</sub>	Load of braking resistor	T_PDO	INT16	_RES_load
301C:15 <sub>h</sub>	Maximum value of overload of braking resistor	T_PDO	INT16	_RES_maxoverload
301C:16 <sub>h</sub>	Overload of power stage (I2t)	T_PDO	INT16	_PS_overload_I2t
301C:17 <sub>h</sub>	Load of power stage	T_PDO	INT16	_PS_load
301C:18 <sub>h</sub>	Maximum value of overload of power stage	T_PDO	INT16	_PS_maxoverload
301C:19 <sub>h</sub>	Overload of motor (I2t)	T_PDO	INT16	_M_overload
301C:1A <sub>h</sub>	Load of motor	T_PDO	INT16	_M_load
301C:1B <sub>h</sub>	Maximum value of overload of motor	T_PDO	INT16	_M_maxoverload
301C:1E <sub>h</sub>	Maximum possible value for operating mode Profile Torque	T_PDO	INT16	_PT_max_val
301C:1F <sub>h</sub>	Additional info on most recent error	T_PDO	UINT16	_LastError_Qual
301C:22 <sub>h</sub>	Overload of power stage (chip temperature)	T_PDO	INT16	_PS_overload_cte
301C:23 <sub>h</sub>	Overload of power stage (power squared)	T_PDO	INT16	_PS_overload_psq
301C:24 <sub>h</sub>	Overload of power stage	T_PDO	INT16	_PS_overload
301C:26 <sub>h</sub>	Conditions for transition to operating state Ready To Switch On	T_PDO	UINT16	_Cond_State4
301C:27 <sub>h</sub>	Current limitation of the system	T_PDO	UINT16	_Imax_system
301C:28 <sub>h</sub>	Currently effective current limitation	T_PDO	UINT16	_Imax_act
301C:29 <sub>h</sub>	Currently effective velocity limitation	T_PDO	UINT32	_Vmax_act
301C:2B <sub>h</sub>	Voltage of cosine signal of encoder	-	INT16	_M_Enc_Cosine
301C:2C <sub>h</sub>	Voltage of sine signal of encoder	-	INT16	_M_Enc_Sine
301E:1 <sub>h</sub>	Actual motor current (q component, generating torque)	T_PDO	INT16	_Iq_act_rms
301E:2 <sub>h</sub>	Actual motor current (d component, field weakening)	T_PDO	INT16	_Id_act_rms
301E:3 <sub>h</sub>	Total motor current	T_PDO	INT16	_I_act
301E:4 <sub>h</sub>	Reference motor voltage q component	T_PDO	INT16	_Uq_ref
301E:5 <sub>h</sub>	Reference motor voltage d component	T_PDO	INT16	_Ud_ref
301E:6 <sub>h</sub>	Total motor voltage (vector sum d components and q components)	T_PDO	INT16	_Udq_ref
301E:7 <sub>h</sub>	Reference speed of rotation	T_PDO	INT16	_n_ref
301E:8 <sub>h</sub>	Actual speed of rotation	T_PDO	INT16	_n_act
301E:9 <sub>h</sub>	Reference position in internal units	T_PDO	INT32	_p_ref_int
301E:C <sub>h</sub>	Reference position	T_PDO	INT32	_p_ref
301E:E <sub>h</sub>	Absolute position with reference to internal resolution in internal units	T_PDO	UINT32	_p_absmodulo
301E:F <sub>h</sub>	Absolute position with reference to the encoder range	T_PDO	UINT32	_p_absENC
301E:10 <sub>h</sub>	Reference motor current (q component, generating torque)	T_PDO	INT16	_Iq_ref_rms
301E:11 <sub>h</sub>	Reference motor current (d component, field weakening)	T_PDO	INT16	_Id_ref_rms
301E:13 <sub>h</sub>	Degree of utilization of DC bus voltage	T_PDO	INT16	_VoltUtil

Address	Object	PDO	Data type	Parameter name
301E:14 <sub>h</sub>	Position deviation including dynamic position deviation	T_PDO	INT32	_p_dif_usr
301E:15 <sub>h</sub>	Maximum value of the load-dependent position deviation	-	INT32	_p_dif_load_peak_usr
301E:16 <sub>h</sub>	Load-dependent position deviation between reference and actual positions	T_PDO	INT32	_p_dif_load_usr
301E:1B <sub>h</sub>	Maximum value of the load-dependent position deviation	-	UINT32	_p_dif_load_peak
301E:1C <sub>h</sub>	Load-dependent position deviation between reference and actual positions	T_PDO	INT32	_p_dif_load
301E:1F <sub>h</sub>	Reference velocity	T_PDO	INT32	_v_ref
301E:26 <sub>h</sub>	Actual position of encoder 1 in internal units	T_PDO	INT32	_p_act_ENC1_int
301E:27 <sub>h</sub>	Actual position of encoder 1	T_PDO	INT32	_p_act_ENC1
301E:28 <sub>h</sub>	Actual speed of rotation of encoder 1	T_PDO	INT16	_n_act_ENC1
301E:29 <sub>h</sub>	Actual velocity of encoder 1	T_PDO	INT32	_v_act_ENC1
301E:2C <sub>h</sub>	Current load-dependent velocity deviation	T_PDO	INT32	_v_dif_usr
301F:1 <sub>h</sub>	Target position of profile generator	T_PDO	INT32	_RAMP_p_target
301F:2 <sub>h</sub>	Actual position of profile generator	T_PDO	INT32	_RAMP_p_act
301F:5 <sub>h</sub>	Target velocity of profile generator	T_PDO	INT32	_RAMP_v_target
301F:7 <sub>h</sub>	Velocity of reference value for velocity feed-forward control	T_PDO	INT32	_pref_v
301F:9 <sub>h</sub>	Acceleration of reference value for acceleration feed-forward control	T_PDO	INT32	_pref_acc
301F:A <sub>h</sub>	Maximum user-defined value for positions	T_PDO	INT32	_ScalePOSmax
301F:B <sub>h</sub>	Maximum user-defined value for velocity	T_PDO	INT32	_ScaleVELmax
301F:C <sub>h</sub>	Maximum user-defined value for acceleration and deceleration	T_PDO	INT32	_ScaleRAMPmax
3022:4 <sub>h</sub>	Synchronization tolerance	-	UINT16	SyncMechTol
3022:5 <sub>h</sub>	Activation of synchronization mechanism	-	UINT16	SyncMechStart
3022:6 <sub>h</sub>	Status of synchronization mechanism	T_PDO	UINT16	SyncMechStatus
3023:7 <sub>h</sub>	Absolute movement beyond movement range	-	UINT16	PP_ModeRangeLim
3023:9 <sub>h</sub>	Change to operating mode Profile Position during movements	-	UINT16	PP_OpmChgType
3023:C <sub>h</sub>	Activation of relative movement after capture	-	UINT16	RMAC_Activate
3023:D <sub>h</sub>	Target position of relative movement after capture	-	INT32	RMAC_Position
3023:E <sub>h</sub>	Velocity of relative movement after capture	-	UINT32	RMAC_Velocity
3023:F <sub>h</sub>	Response if target position is overtraveled	-	UINT16	RMAC_Response
3023:10 <sub>h</sub>	Edge of capture signal for relative movement after capture	-	UINT16	RMAC_Edge
3023:11 <sub>h</sub>	Status of relative movement after capture	T_PDO	UINT16	_RMAC_Status
3023:12 <sub>h</sub>	Detailed status of relative movement after capture (RMAC)	T_PDO	UINT16	_RMAC_DetailStatus
3028:6 <sub>h</sub>	Maximum distance for search for switching point	-	INT32	HMoutdis
3028:7 <sub>h</sub>	Distance from switching point	-	INT32	HMdis
3028:A <sub>h</sub>	Preferred homing method	-	INT16	HMprefmethod
3028:B <sub>h</sub>	Position at reference point	R_PDO	INT32	HMp_home
3028:C <sub>h</sub>	Distance from switching point to index pulse	T_PDO	INT32	_HMdisREFtoIDX
3028:D <sub>h</sub>	Maximum search distance after overtravel of switch	-	INT32	HMsrchdis
3028:F <sub>h</sub>	Distance from switching point to index pulse	T_PDO	INT32	_HMdisREFtoIDX_usr
3029:3 <sub>h</sub>	Selection of jog method	R_PDO	UINT16	JOGmethod
3029:4 <sub>h</sub>	Velocity for slow movement	R_PDO	UINT32	JOGv_slow
3029:5 <sub>h</sub>	Velocity for fast movement	R_PDO	UINT32	JOGv_fast
3029:7 <sub>h</sub>	Distance for step movement	-	INT32	JOGstep
3029:8 <sub>h</sub>	Wait time for step movement	-	UINT16	JOGtime

Address	Object	PDO	Data type	Parameter name
302E:3 <sub>h</sub>	Maximum permissible distance	-	UINT16	MT_dismax
302E:A <sub>h</sub>	Maximum permissible distance	-	INT32	MT_dismax_usr
302F:1 <sub>h</sub>	Autotuning start	-	UINT16	AT_start
302F:2 <sub>h</sub>	Autotuning status	T_PDO	UINT16	_AT_state
302F:3 <sub>h</sub>	Movement range for Autotuning	-	UINT32	AT_dis
302F:4 <sub>h</sub>	Direction of movement for Autotuning	-	UINT16	AT_dir
302F:6 <sub>h</sub>	Velocity jump for Autotuning	-	UINT32	AT_n_ref
302F:7 <sub>h</sub>	Friction torque of the system	T_PDO	UINT16	_AT_M_friction
302F:8 <sub>h</sub>	Constant load torque	T_PDO	INT16	_AT_M_load
302F:9 <sub>h</sub>	Waiting time between Autotuning steps	-	UINT16	AT_wait
302F:B <sub>h</sub>	Progress of Autotuning	T_PDO	UINT16	_AT_progress
302F:C <sub>h</sub>	Moment of inertia of the system	T_PDO	UINT16	_AT_J
302F:E <sub>h</sub>	Type of coupling of the system	-	UINT16	AT_mechanical
302F:12 <sub>h</sub>	Movement range for Autotuning	-	INT32	AT_dis_usr
302F:13 <sub>h</sub>	Velocity jump for Autotuning	-	INT32	AT_v_ref
303A:1 <sub>h</sub>	Lock HMI	-	UINT16	HMIlocked
303A:2 <sub>h</sub>	HMI display when motor moves	-	UINT16	HMIDispPara
303B:2 <sub>h</sub>	Number of power on cycles	T_PDO	UINT32	_ERR_powerOn
303B:4 <sub>h</sub>	Clear error memory	-	UINT16	ERR_clear
303B:5 <sub>h</sub>	Reset error memory read pointer	-	UINT16	ERR_reset
303B:6 <sub>h</sub>	Signal output function Selected Error (error classes 1 to 4): First error code	-	UINT16	MON_IO_SelErr1
303B:7 <sub>h</sub>	Signal output function Selected Error (error classes 1 to 4): Second error code	-	UINT16	MON_IO_SelErr2
303B:8 <sub>h</sub>	Signal output function Selected Warning (error class 0): First error code	-	UINT16	MON_IO_SelWar1
303B:9 <sub>h</sub>	Signal output function Selected Warning (error class 0): Second error code	-	UINT16	MON_IO_SelWar2
303C:1 <sub>h</sub>	Error code	-	UINT16	_ERR_number
303C:2 <sub>h</sub>	Error class	-	UINT16	_ERR_class
303C:3 <sub>h</sub>	Time of detection of error	-	UINT32	_ERR_time
303C:4 <sub>h</sub>	Additional information on detected error	-	UINT16	_ERR_qual
303C:5 <sub>h</sub>	Number of cycles of enabling the power stage at error time	-	UINT16	_ERR_enable_cycl
303C:6 <sub>h</sub>	Time between enabling of power stage and detection of the error	-	UINT16	_ERR_enable_time
303C:7 <sub>h</sub>	DC bus voltage at the time the error was detected	-	UINT16	_ERR_DCbus
303C:8 <sub>h</sub>	Motor velocity at the time the error was detected	-	INT32	_ERR_motor_v
303C:9 <sub>h</sub>	Motor current at the time the error was detected	-	UINT16	_ERR_motor_I
303C:A <sub>h</sub>	Temperature of power stage at the time the error was detected	-	INT16	_ERR_temp_ps
303C:B <sub>h</sub>	Temperature of device at the time the error was detected	-	INT16	_ERR_temp_dev
303F:5D <sub>h</sub>	Value of the SinCos amplitude	-	UINT16	_ENC_AmplVal
303F:5E <sub>h</sub>	Mean value of the SinCos amplitude	-	UINT16	_ENC_AmplMean
303F:5F <sub>h</sub>	Minimum value of the SinCos amplitude	-	UINT16	_ENC_AmplMin
303F:60 <sub>h</sub>	Maximum value of the SinCos amplitude	-	UINT16	_ENC_AmplMax
303F:61 <sub>h</sub>	Activation of monitoring of SinCos amplitude	-	UINT16	MON_ENC_Ampl
3040:43 <sub>h</sub>	Last error code of fieldbus parameter services	-	UINT16	_ErrNumFbParSvc
3041:2 <sub>h</sub>	CANopen address (node number)	-	UINT16	CANaddress
3041:3 <sub>h</sub>	CANopen baud rate	-	UINT16	CANbaud

Address	Object	PDO	Data type	Parameter name
3041:6 <sub>h</sub>	CANopen diagnostics word	-	UINT16	_CanDiag
3041:A <sub>h</sub>	CANopen manufacturer-specific SDO abort code	-	UINT16	_ManuSdoAbort
3041:B <sub>h</sub>	PDO 1 event mask	-	UINT16	CANpdo1Event
3041:C <sub>h</sub>	PDO 2 event mask	-	UINT16	CANpdo2Event
3041:D <sub>h</sub>	PDO 3 event mask	-	UINT16	CANpdo3Event
3041:E <sub>h</sub>	PDO 4 event mask	-	UINT16	CANpdo4Event
305C:17 <sub>h</sub>	Transformation ratio	-	UINT16	Mfb_ResRatio

## Assignment Object Group 6000<sub>h</sub>

### Overview

The product provides corresponding parameters for the CANopen object group 6000<sub>h</sub>.

Address	Object	PDO	Data type	Parameter name
603F:0 <sub>h</sub>	Detected error causing a stop (error classes 1 to 4)	T_PDO	UINT16	_LastError
6040:0 <sub>h</sub>	DriveCom control word	R_PDO	UINT16	DCOMcontrol
6041:0 <sub>h</sub>	DriveCom status word	T_PDO	UINT16	_DCOMstatus
605B:0 <sub>h</sub>	Behavior for disabling the power stage during movement	-	INT16	DSM_ShutDownOption
605D:0 <sub>h</sub>	Halt option code	-	INT16	LIM_HaltReaction
6060:0 <sub>h</sub>	Operating mode	R_PDO	INT8	DCOMopmode
6061:0 <sub>h</sub>	Active operating mode	T_PDO	INT8	_DCOMopmd_act
6063:0 <sub>h</sub>	Actual position in internal units	T_PDO	INT32	_p_act_int
6064:0 <sub>h</sub>	Actual position	T_PDO	INT32	_p_act
6065:0 <sub>h</sub>	Maximum load-dependent position deviation	R_PDO	UINT32	MON_p_dif_load
6067:0 <sub>h</sub>	Standstill window, permissible control deviation	R_PDO	UINT32	MON_p_win
6068:0 <sub>h</sub>	Standstill window, time	-	UINT16	MON_p_winTime
606B:0 <sub>h</sub>	Actual velocity of profile generator	T_PDO	INT32	_RAMP_v_act
606C:0 <sub>h</sub>	Actual velocity	T_PDO	INT32	_v_act
606D:0 <sub>h</sub>	Velocity window, permissible deviation	-	UINT16	MON_v_win
606E:0 <sub>h</sub>	Velocity window, time	-	UINT16	MON_v_winTime
6071:0 <sub>h</sub>	Target torque for operating mode Profile Torque	R_PDO	INT16	PTtq_target
6077:0 <sub>h</sub>	Actual torque	T_PDO	INT16	_tq_act
607A:0 <sub>h</sub>	Target position for operating mode Profile Position	R_PDO	INT32	PPp_target
607D:1 <sub>h</sub>	Negative position limit for software limit switch	-	INT32	MON_swLimN
607D:2 <sub>h</sub>	Positive position limit for software limit switch	-	INT32	MON_swLimP
607F:0 <sub>h</sub>	Maximum velocity of the motion profile for velocity	-	UINT32	RAMP_v_max
6081:0 <sub>h</sub>	Target velocity for operating mode Profile Position	R_PDO	UINT32	PPv_target
6083:0 <sub>h</sub>	Acceleration of the motion profile for velocity	R_PDO	UINT32	RAMP_v_acc
6084:0 <sub>h</sub>	Deceleration of the motion profile for velocity	R_PDO	UINT32	RAMP_v_dec
6087:0 <sub>h</sub>	Slope setting of the motion profile for torque	R_PDO	UINT32	RAMP_tq_slope
6098:0 <sub>h</sub>	Homing method	R_PDO	INT8	HMmethod
6099:1 <sub>h</sub>	Target velocity for searching the switch	-	UINT32	HMv
6099:2 <sub>h</sub>	Target velocity for moving away from switch	-	UINT32	HMv_out
60B8:0 <sub>h</sub>	Touch Probe function	R_PDO	UINT16	TouchProbeFct
60B9:0 <sub>h</sub>	Touch Probe status	T_PDO	UINT16	_TouchProbeStat
60BA:0 <sub>h</sub>	Capture input 1 captured position at rising edge	T_PDO	INT32	_Cap1PosRisEdge
60BB:0 <sub>h</sub>	Capture input 1 captured position at falling edge	T_PDO	INT32	_Cap1PosFallEdge
60BC:0 <sub>h</sub>	Capture input 2 captured position at rising edge	T_PDO	INT32	_Cap2PosRisEdge
60BD:0 <sub>h</sub>	Capture input 2 captured position at falling edge	T_PDO	INT32	_Cap2PosFallEdge
60C1:1 <sub>h</sub>	Position reference value for operating mode Interpolated Position	R_PDO	INT32	IPp_target
60C2:1 <sub>h</sub>	Interpolation time period value	-	UINT8	IP_IntTimPerVal
60C2:2 <sub>h</sub>	Interpolation time index	-	INT8	IP_IntTimInd
60F2:0 <sub>h</sub>	Options for operating mode Profile Position	-	UINT16	PPoption
60F4:0 <sub>h</sub>	Position deviation including dynamic position deviation	T_PDO	INT32	_p_dif

---

Address	Object	PDO	Data type	Parameter name
60FF:0 <sub>h</sub>	Target velocity for operating mode Profile Velocity	R_PDO	INT32	PVv_target
6502:0 <sub>h</sub>	Supported operating modes as per DSP402	T_PDO	UINT32	_SuppDriveModes

## Details of Object Group 1000<sub>h</sub>

### 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 device profile
Access	RO
PDO mapping	–
Value range	–
Default value	0042 0192 <sub>h</sub>
Can be saved	–

Bit assignment, subindex 00h

bit	Access	Value	Meaning
0 ... 15	RO	0192 <sub>h</sub>	Device profile DS-402 (192 <sub>h</sub> )
16 ... 23	RO	42 <sub>h</sub>	Bit 17 = 1: AC servo drive
24 ... 31	RO	00 <sub>h</sub>	Not used

### 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 fieldbus profiles - with the object Error code (603F<sub>h</sub>).

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

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	RO
PDO mapping	–
Value range	–
Default value	–
Can be saved	–



Bit assignment, subindex 00h

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

### 1003<sub>h</sub> Predefined Error Field

The object contains 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 most recent 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	RW
PDO mapping	–
Value range	0...1
Default value	1
Can be saved	–

Subindex	01 <sub>h</sub> , error field
Meaning	Error number
Access	RO
PDO mapping	–
Value range	–
Default value	0
Can be saved	–

Bit assignment, subindex 00<sub>h</sub> ... 05<sub>h</sub>

Bits 0 ... 15: Error code (as per DS301).

Bits 16 ... 31: Error code 1000<sub>h</sub>; Vendor-specific error number.

### 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	RW
PDO mapping	–
Value range	0...4294967295
Default value	8000 0080 <sub>h</sub>
Can be saved	–

Bit assignment, 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 is not used by the device.
10-7	RW	0001 <sub>b</sub>	Function code, bits 10 ... 7 of COB ID
6-0	RO	7F <sub>h</sub>	Node address, bit 6 ... 0 of COB ID

### 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	RO
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

**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	RO
PDO mapping	–
Value range	–
Default value	–
Can be saved	–

**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	RO
PDO mapping	–
Value range	–
Default value	–
Can be saved	–

**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(100Dh)`.

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	RW
PDO mapping	–
Value range	0...65535
Default value	0
Can be saved	–

**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	RW
PDO mapping	–
Value range	0...255
Default value	0
Can be saved	–

**1014<sub>h</sub> COB ID Emergency Object 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	RW
PDO mapping	–
Value range	0...4294967295

Default value	4000 0080 <sub>h</sub> + node ID
Can be saved	–

Bit assignment, subindex 00h

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"

### 1015<sub>h</sub> Inhibit Time Emergency Object 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	RW
PDO mapping	–
Value range	0...65535
Default value	0
Can be saved	–

### 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	RO
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	RW
PDO mapping	–
Value range	0...4294967295
Default value	0
Can be saved	–

Bit assignment, subindex 01<sub>h</sub> ... 03<sub>h</sub>

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 (1017h)`. If the time interval is zero, the device specified via the node ID is not monitored.

### 1017<sub>h</sub> 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 (1016h)`. A time interval of zero deactivates monitoring.

Object description

Index	1017 <sub>h</sub>
Object name	Producer Heartbeat Time
Object code	VAR
Data type	Unsigned16

Value description

Subindex	00 <sub>h</sub> , Producer Heartbeat Time
Meaning	Time interval for producer "Heartbeat"
Access	RW
PDO mapping	–
Value range	0...65535
Default value	0
Can be saved	–

### 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

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	RO
PDO mapping	–
Value range	–
Default value	3
Can be saved	–

Subindex	01 <sub>h</sub> , vendor ID
Meaning	Vendor ID
Access	RO
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	RO
PDO mapping	–
Value range	–
Default value	–
Can be saved	–

Subindex	03 <sub>h</sub> , revision number
Meaning	Revision number
Access	RO
PDO mapping	–
Value range	–
Default value	–
Can be saved	–

**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	RO
PDO mapping	–

Value range	–
Default value	1
Can be saved	–

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

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

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

### 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	RO
PDO mapping	–
Value range	–
Default value	2
Can be saved	–

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

Subindex	02 <sub>h</sub> , COB ID server -> client
Meaning	Identifier server -> client
Access	RO
PDO mapping	–
Value range	0...4294967295



Default value	1408 + node ID
Can be saved	–

### 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	RO
PDO mapping	–
Value range	–
Default value	3
Can be saved	–

Subindex	01 <sub>h</sub> , COB ID client -> server
Meaning	Identifier client -> server
Access	RW
PDO mapping	–
Value range	0...4294967295
Default value	8000 0000 <sub>h</sub>
Can be saved	–

Subindex	02 <sub>h</sub> , COB ID server -> client
Meaning	Identifier server -> client
Access	RW
PDO mapping	–
Value range	0...4294967295
Default value	8000 0000 <sub>h</sub>
Can be saved	–

Subindex	03 <sub>h</sub> , node ID SDO client
Meaning	Node ID SDO client
Access	RW
PDO mapping	–
Value range	1...127
Default value	–
Can be saved	–

**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	RO
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	RW
PDO mapping	–
Value range	0...4294967295
Default value	0200 <sub>h</sub> + node ID
Can be saved	–

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

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

bit	Access	Value	Meaning
31	RW	0 <sub>b</sub>	0: PDO is active 1: PDO is inactive
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

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

Bit assignment, subindex 02<sub>h</sub>

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	–

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

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".

### 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	RO
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	RW
PDO mapping	–
Value range	0 ... .4294967295
Default value	8000 0300 <sub>h</sub> + node ID
Can be saved	–

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

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

**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	RO
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	RW
PDO mapping	–
Value range	0 ... 4294967295
Default value	8000 0400 <sub>h</sub> + node ID
Can be saved	–

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

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

### 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	RO
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	RW
PDO mapping	–
Value range	0...4294967295
Default value	8000 0500 <sub>h</sub> + node ID
Can be saved	–

Subindex	02 <sub>h</sub> , transmission type
Meaning	Transmission type
Access	RO
PDO mapping	–
Value range	–
Default value	254
Can be saved	–

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

#### 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".

### 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	RW
PDO mapping	–
Value range	0 ... 4
Default value	1
Can be saved	–

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

Subindex	02 <sub>h</sub>
Meaning	Second object for mapping
Access	RW
PDO mapping	–
Value range	0...4294967295
Default value	–
Can be saved	–

Subindex	03 <sub>h</sub>
Meaning	Third object for mapping
Access	RW
PDO mapping	–
Value range	0...4294967295
Default value	–
Can be saved	–

Subindex	04 <sub>h</sub>
Meaning	Fourth object for mapping
Access	RW
PDO mapping	–
Value range	0...4294967295
Default value	–
Can be saved	–

Bit assignment, starting at subindex 01h

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

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.

#### Settings:

The following default assignment is used:

- Subindex 01<sub>h</sub>: controlword (6040<sub>h</sub>)

### 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	RW
PDO mapping	–
Value range	0 ... 4
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
Access	RW
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
Access	RW
PDO mapping	–
Value range	0...4294967295
Default value	607A 0020 <sub>h</sub>
Can be saved	–

Subindex	03 <sub>h</sub>
Meaning	Third object for mapping
Access	RW
PDO mapping	–
Value range	0...4294967295
Default value	–
Can be saved	–

Subindex	04 <sub>h</sub>
Meaning	Fourth object for mapping
Access	RW
PDO mapping	–
Value range	0...4294967295
Default value	–
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>: controlword (6040<sub>h</sub>)
- Subindex 02<sub>h</sub>: target position (607A<sub>h</sub>)



**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	RW
PDO mapping	–
Value range	0 ... 4
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
Access	RW
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
Access	RW
PDO mapping	–
Value range	0...4294967295
Default value	60FF 0020 <sub>h</sub>
Can be saved	–

Subindex	03 <sub>h</sub>
Meaning	Third object for mapping
Access	RW
PDO mapping	–
Value range	0...4294967295
Default value	–
Can be saved	–

Subindex	04 <sub>h</sub>
Meaning	Fourth object for mapping
Access	RW
PDO mapping	–

Value range	0...4294967295
Default value	–
Can be saved	–

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

**Settings:**

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

- Subindex 01<sub>h</sub>: `controlword (6040h)`
- Subindex 02<sub>h</sub>: `target velocity (60FFh)`

**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	RW
PDO mapping	–
Value range	0 ... 4
Default value	0
Can be saved	–

Subindex	01 <sub>h</sub>
Meaning	First object for mapping
Access	RW
PDO mapping	–
Value range	0...4294967295
Default value	–
Can be saved	–

Subindex	02 <sub>h</sub>
Meaning	Second object for mapping
Access	RW
PDO mapping	–
Value range	0...4294967295
Default value	–
Can be saved	–

Subindex	03 <sub>h</sub>
Meaning	Third object for mapping
Access	RW
PDO mapping	–

Value range	0...4294967295
Default value	–
Can be saved	–

Subindex	04 <sub>h</sub>
Meaning	Fourth object for mapping
Access	RW
PDO mapping	–
Value range	0...4294967295
Default value	–
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 R\_PDO4 can be modified.

### 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	RO
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	RW
PDO mapping	–
Value range	0...4294967295
Default value	4000 0180 <sub>h</sub> + node ID
Can be saved	–

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

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

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	RW
PDO mapping	–
Value range	0...65535
Default value	0
Can be saved	–

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".

### 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	RO
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	RW
PDO mapping	–
Value range	0..4294967295
Default value	C000 0280 <sub>h</sub> + node ID
Can be saved	–

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

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

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	RW
PDO mapping	–
Value range	0..65535
Default value	100
Can be saved	–

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: Actual position `position actual value` (6064<sub>h</sub>).

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

**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	RO
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	RW
PDO mapping	–
Value range	0...4294967295
Default value	C000 0380 <sub>h</sub> + node ID
Can be saved	–

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

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

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	RW
PDO mapping	–
Value range	0...65535
Default value	100
Can be saved	–

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: Actual velocity `velocity actual value` (606C<sub>h</sub>).

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

### 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	RO
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	RW
PDO mapping	–
Value range	0...4294967295

Default value	C000 0480 <sub>h</sub> + node ID
Can be saved	–

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

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

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	RW
PDO mapping	–
Value range	0...65535
Default value	0
Can be saved	–

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".



## 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	RW
PDO mapping	–
Value range	0 ... 4
Default value	1
Can be saved	–

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

Subindex	02 <sub>h</sub>
Meaning	Second object for mapping
Access	RW
PDO mapping	–
Value range	0...4294967295
Default value	–
Can be saved	–

Subindex	03 <sub>h</sub>
Meaning	Third object for mapping
Access	RW
PDO mapping	–
Value range	0...4294967295
Default value	–
Can be saved	–

Subindex	04 <sub>h</sub>
Meaning	Fourth object for mapping
Access	RW
PDO mapping	–

Value range	0...4294967295
Default value	–
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 used:

- Subindex 01<sub>h</sub>: statusword (6041<sub>h</sub>)

### 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	RW
PDO mapping	–
Value range	0 ... 4
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 mapping
Access	RW
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 mapping
Access	RW
PDO mapping	–
Value range	0...4294967295
Default value	6064 0020 <sub>h</sub>
Can be saved	–

Subindex	03 <sub>h</sub>
Meaning	Third object for mapping
Access	RW

PDO mapping	–
Value range	0...4294967295
Default value	–
Can be saved	–

Subindex	04 <sub>h</sub>
Meaning	Fourth object for mapping
Access	RW
PDO mapping	–
Value range	0...4294967295
Default value	–
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>: statusword (6041<sub>h</sub>)
- Subindex 02<sub>h</sub>: position actual value (6064<sub>h</sub>)

### 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	RW
PDO mapping	–
Value range	0 ... 4
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 mapping
Access	RW
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 mapping

Access	RW
PDO mapping	–
Value range	0...4294967295
Default value	606C 0020 <sub>h</sub>
Can be saved	–

Subindex	03 <sub>h</sub>
Meaning	Third object for mapping
Access	RW
PDO mapping	–
Value range	0...4294967295
Default value	–
Can be saved	–

Subindex	04 <sub>h</sub>
Meaning	Fourth object for mapping
Access	RW
PDO mapping	–
Value range	0...4294967295
Default value	–
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>: statusword (6041<sub>h</sub>)
- Subindex 02<sub>h</sub>: velocity actual value (606C<sub>h</sub>)

### 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	RW
PDO mapping	–
Value range	0 ... 4
Default value	0
Can be saved	–

Subindex	01 <sub>h</sub>
Meaning	First object for mapping
Access	RW
PDO mapping	–
Value range	0...4294967295
Default value	–
Can be saved	–

Subindex	02 <sub>h</sub>
Meaning	Second object for mapping
Access	RW
PDO mapping	–
Value range	0...4294967295
Default value	–
Can be saved	–

Subindex	03 <sub>h</sub>
Meaning	Third object for mapping
Access	RW
PDO mapping	–
Value range	0...4294967295
Default value	–
Can be saved	–

Subindex	04 <sub>h</sub>
Meaning	Fourth object for mapping
Access	RW
PDO mapping	–
Value range	0...4294967295
Default value	–
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\_PDO4 can be modified.





## C

<b>CAN</b>	(Controller <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.
<b>CANopen</b>	Device- and manufacturer-independent description language for communication via the CAN bus
<b>CiA</b>	CAN in <b>A</b> utomation, CAN interest group, standardization group for CAN and CANopen.
<b>COB</b>	Communication <b>O</b> bject, transport unit in a CAN network.
<b>COB ID</b>	Communication <b>O</b> bject <b>I</b> Dentifier; uniquely identifies each communication object in a CAN network

## D

<b>DOM</b>	<b>Date of manufacturing:</b> The nameplate of the product shows the date of manufacture in the format DD.MM.YY or in the format DD.MM.YYYY. For example: 31.12.11 corresponds to December 31, 2011 31.12.2011 corresponds to December 31, 2011
<b>DriveCom</b>	Specification of the DSP402 state machine was created in accordance with the DriveCom specification.
<b>DS301</b>	Standardizes the CANopen communication profile
<b>DSP402</b>	Standardizes the CANopen device profile for drives

## E

<b>EDS</b>	( <b>E</b> lectronic <b>D</b> ata <b>S</b> heet); contains the specific properties of a product.
<b>Electronic gear</b>	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.
<b>EMC</b>	Electromagnetic compatibility
<b>EMCY object</b>	Emergency Object
<b>Encoder</b>	Sensor that converts a measured distance or angle into an electrical signal. This signal is evaluated by the drive to determine the actual position of a shaft (rotor) or a driving unit.
<b>Error</b>	Discrepancy between a detected (computed, measured or signaled) value or condition and the specified or theoretically correct value or condition.
<b>Error class</b>	Classification of errors into groups. The different error classes allow for specific responses to errors, for example by severity.

**F****Factory setting**

Factory settings when the product is shipped

**Fault**

Fault is an operating state. If the monitoring functions detect an error, a transition to this operating state is triggered, depending on the error class. A "Fault Reset" or a power cycle are required to exit this operating state. Prior to this, the cause of the detected error must be removed. Further information can be found in the pertinent standards such as IEC 61800-7, ODVA Common Industrial Protocol (CIP).

**Fault reset**

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.

**H****Heartbeat**

Used for unconfirmed connection acknowledgment messages from network devices.

**HMI**

Human Machine Interface

**I****I/O**

Inputs/outputs

**Input device**

A device that can be connected via the RS232 interface; either the HMI or a PC with commissioning software.

**L****Life guarding**

For monitoring the connection of an NMT master

**Limit switch**

Switches that signal overtravel of the permissible movement range.

**M****Mapping**

Assignment of object dictionary entries to PDOs

**N****NMT**

Network Management (NMT), part of the CANopen communication profile; tasks include initialization of the network and devices, starting, stopping and monitoring of devices

**Node guarding**

Monitoring of the connection to the slave at an interface for cyclic data traffic.

**Node ID**

Node address assigned to a device on the network.

**O****Object dictionary**

List of the parameters, values and functions available in the device. Each entry is uniquely referenced via index (16 bit) and subindex (8 bit).



**P**

<b>Parameter</b>	Device data and values that can be read and set (to a certain extent) by the user.
<b>PDO</b>	Process Data Object
<b>Persistent</b>	Indicates whether the value of the parameter remains in the memory after power to the drive has been removed.
<b>Power stage</b>	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.

**Q**

<b>Quick Stop</b>	The Quick Stop function can be used for fast deceleration of a movement in the case of an error or via a command.
-------------------	---

**R**

<b>R_PDO</b>	Receive PDO
--------------	-------------

**S**

<b>SDO</b>	Service Data Object
<b>SYNC object</b>	Synchronization object

**T**

<b>T_PDO</b>	Transmit PDO
--------------	--------------





## B

before you begin  
  safety information, 5  
boot-up message, 38

## C

CAN message, 18  
CANopen message, 19  
client-server relationship, 21  
COB ID, 19

## D

data frame, 20

## E

EMCY message, 38  
EMCY object, 18  
emergency object service, 38  
error memory, 39  
error register, 76  
error register and error code, 39

## F

function code, 20

## H

heartbeat, 40

## I

intended use, 6

## L

life guarding, 40

## M

master-slave relationship, 21  
message objects, 75

## N

NMT, 18  
node address, 20  
node guarding, 40

## O

operating states, 50

## P

parameter \_DCOMopmd\_act, 55

parameter \_DCOMstatus, 51  
parameter \_ManuSdoAbort, 77  
parameter CANaddress, 48  
parameter CANbaud, 48  
parameter CANpdo1Event, 33  
parameter CANpdo2Event, 33  
parameter CANpdo3Event, 33  
parameter CANpdo4Event, 33  
parameter DCOMcontrol, 53  
parameter DCOMopmode, 55  
parameter HMmethod, 63  
parameter JOGactivate, 56  
parameter PPp\_target, 59  
parameter PTtq\_target, 57  
parameter PVv\_target, 58, 59  
PDO, 18  
PDO mapping, 34  
producer-consumer relationship, 22

## Q

qualification of personnel, 5

## S

SDO, 18  
SYNC object, 18  
synchronization, 36

