

LXM32A

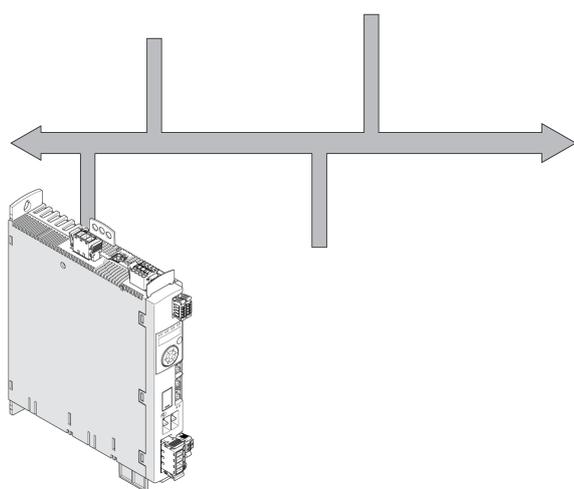
CANopen Interface

User Guide

Original instructions

0198441113779.06

06/2021



Legal Information

The Schneider Electric brand and any trademarks of Schneider Electric SE and its subsidiaries referred to in this guide are the property of Schneider Electric SE or its subsidiaries. All other brands may be trademarks of their respective owners.

This guide and its content are protected under applicable copyright laws and furnished for informational use only. No part of this guide may be reproduced or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), for any purpose, without the prior written permission of Schneider Electric.

Schneider Electric does not grant any right or license for commercial use of the guide or its content, except for a non-exclusive and personal license to consult it on an "as is" basis. Schneider Electric products and equipment should be installed, operated, serviced, and maintained only by qualified personnel.

As standards, specifications, and designs change from time to time, information contained in this guide may be subject to change without notice.

To the extent permitted by applicable law, no responsibility or liability is assumed by Schneider Electric and its subsidiaries for any errors or omissions in the informational content of this material or consequences arising out of or resulting from the use of the information contained herein.

© 2021 Schneider Electric. All Rights Reserved.

Table of Contents

Safety Information.....	5
Qualification of Personnel	5
Intended Use.....	6
About the Book.....	7
Introduction	10
Fieldbus Devices on the CANopen Network.....	10
Basics	11
Communication Layers	11
Objects	12
CANopen Profiles	13
Communication - Object Dictionary	14
Communication - Objects	15
Communication - Relationships	18
SDO Data Exchange.....	19
SDO Message.....	20
SDO Reading and Writing Data	21
SDO Reading Data Longer Than 4 Bytes	23
PDO Data Exchange.....	25
PDO Message.....	25
PDO Events	29
PDO Mapping	30
Synchronization.....	31
Emergency Object Service	33
Network Management Services - Overview	34
NMT Services for Device Control	35
NMT Service Node Guarding/Life Guarding	36
NMT Service Heartbeat.....	38
Commissioning.....	40
Preparation	40
Operating States and Operating Modes.....	42
Operating States	42
Indication of the Operating State	42
Changing the Operating State.....	44
Operating Modes.....	45
Starting and Changing an Operating Mode	45
Operating Mode Jog.....	46
Operating Mode Profile Torque	47
Operating Mode Profile Velocity	48
Operating Mode Profile Position.....	49
Operating Mode Interpolated Position.....	50
Operating Mode Homing	52
Examples for Node Address 1	53
Diagnostics and Troubleshooting	59
Fieldbus Communication Error Diagnostics	59
Fieldbus Test.....	59
Fieldbus Status LEDs	59
Error Diagnostics via Fieldbus	60

CANopen Error Messages.....	61
Object Dictionary.....	64
Specifications for the Objects	64
Overview of object group 1000 hex	65
Assignment Object Group 3000 hex.....	67
Assignment Object Group 6000 hex.....	77
Details of Object Group 1000 hex	78
Glossary	111
Index	115

Safety Information

Important Information

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

Document Scope

The information provided in this user guide supplements the user guide of the servo drive LXM32A.

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

Validity Note

This user guide applies to the fieldbus CANopen for the servo drive LXM32A.

For product compliance and environmental information (RoHS, REACH, PEP, EOL, etc.), go to www.se.com/ww/en/work/support/green-premium/.

The characteristics that are described in the present document, as well as those described in the documents included in the Related Documents section below, can be found online. To access the information online, go to the Schneider Electric home page www.se.com/ww/en/download/.

The characteristics that are described in the present document 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 document and online information, use the online information as your reference.

Related Documents

Title of documentation	Reference number
LXM32A - CANopen Interface - User Guide (this user guide)	0198441113779 (eng)
	0198441113780 (fre)
	0198441113778 (ger)
Lexium 32A - Servo Drive - User Guide	0198441113755 (eng)
	0198441113756 (fre)
	0198441113754 (ger)
	0198441113758 (spa)
	0198441113757 (ita)
	0198441113759 (chi)

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.¹
- 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.

¹ 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
IEC 61131-2:2007	Programmable controllers, part 2: Equipment requirements and tests.
ISO 13849-1:2015	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
ISO 14119:2013	Safety of machinery - Interlocking devices associated with guards - Principles for design and selection
ISO 13850:2015	Safety of machinery - Emergency stop - Principles for design
IEC 62061:2015	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 61784-3:2016	Industrial communication networks - Profiles - Part 3: Functional safety fieldbuses - General rules and profile definitions.
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.

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

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.

Communication Layers

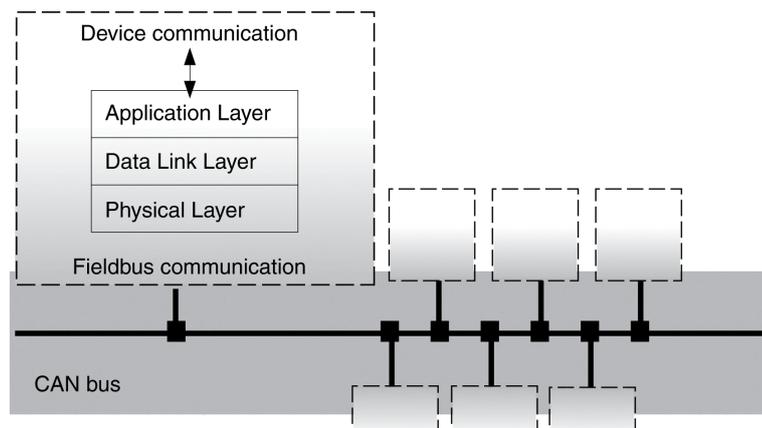
Overview

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 hex and 6000 hex.

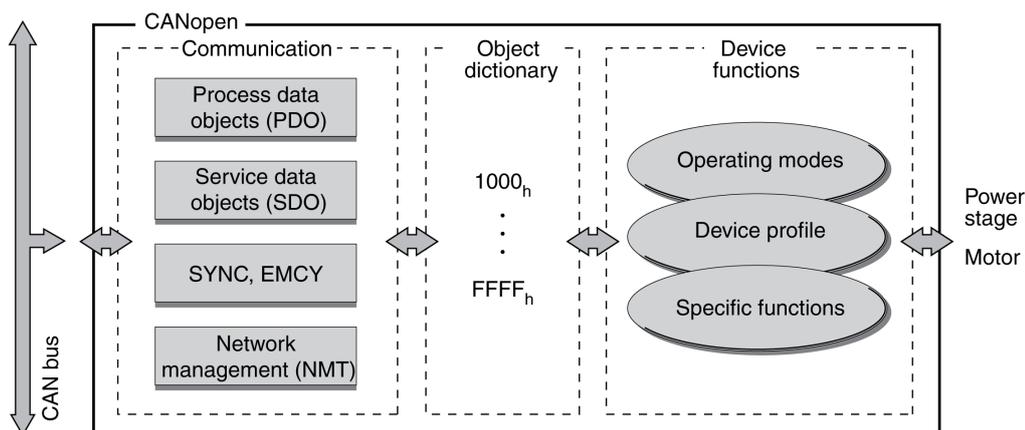
The names of the parameters and the data type of the parameters may be different from the DSP402 definition for object group 6000 hex. In this case, enter the data type according to the DS402.

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 Object Dictionary, page 64 for a list of the CANopen objects.

CANopen Profiles

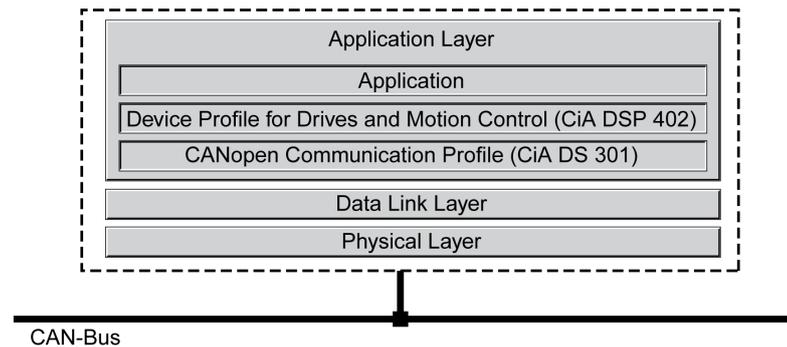
Standardized Profiles

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.

Example

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

Index	Subindex	Name	Meaning
607D hex	00 hex	-	Number of data fields
607D hex	01 hex	minimum position limit	Negative software limit switch
607D hex	02 hex	maximum position limit	Positive software limit switch

Object Descriptions in the User Guide

For CANopen programming, the objects of the following object groups are described in detail:

- 1xxx hex objects: Communication objects in this chapter.
- 3xxx hex objects: Vendor-specific objects required to control the device in chapter *Operating States and Operating Modes*, page 42.
- 6xxx hex objects: Standardized objects of the device profile in chapter *Operating States and Operating Modes*, page 42.

Standardized Objects

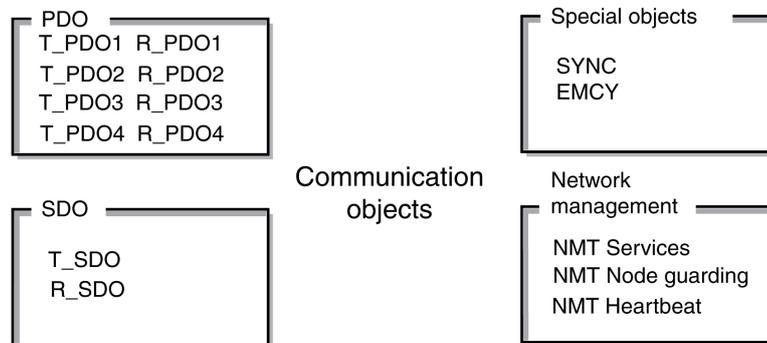
Standardized objects allow you to use the same application program for different network devices of the same device type. This requires these objects to be contained in the object dictionary of the network devices. Standardized objects are defined in the DS301 communication profile and the DSP402 device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

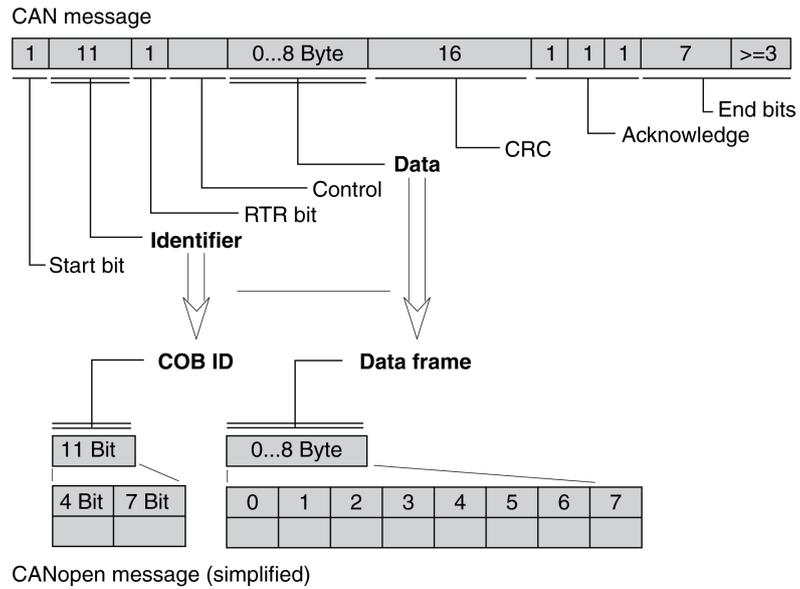


- PDOs (process data objects) for real-time transmission of process data
- SDOs (service data object) for read and write access to the object dictionary
- Objects for controlling CAN messages:
 - SYNC object (synchronization object) for synchronization of network devices
 - EMCY object (emergency object),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

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

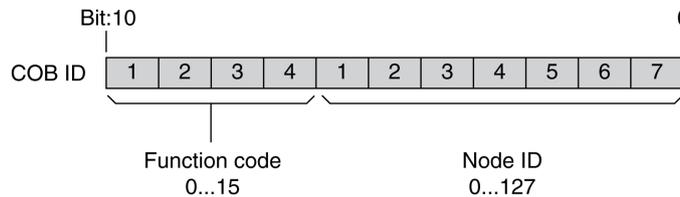
The COB ID (**C**ommunication **O**bject **I**dentifier) has 2 tasks for controlling communication objects:

- 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 objects with which the settings of the communication objects can be read or modified via an SDO.

Communication object	Function code	Node address, node ID [1...127]	COB ID decimal (hexadecimal)	Index of object parameters
NMT Start/Stop Service	0 0 0 0	0 0 0 0 0 0 0	0 (0 hex)	-
SYNC object	0 0 0 1	0 0 0 0 0 0 0	128 (80 hex)	1005 hex ... 1007 hex
EMCY object	0 0 0 1	x x x x x x x	128 (80 hex) + node ID	1014 hex, 1015 hex
T_PDO1	0 0 1 1	x x x x x x x	384 (180 hex) + node ID	1800 hex
R_PDO1	0 1 0 0	x x x x x x x	512 (200 hex) + node ID	1400 hex
T_PDO2	0 1 0 1	x x x x x x x	640 (280 hex) + node ID	1801 hex
R_PDO2	0 1 1 0	x x x x x x x	768 (300 hex) + node ID	1401 hex
T_PDO3	0 1 1 1	x x x x x x x	896 (380 hex) + node ID	1802 hex
R_PDO3	1 0 0 0	x x x x x x x	1024 (400 hex) + node ID	1402 hex
T_PDO4	1 0 0 1	x x x x x x x	1152 (480 hex) + node ID	1803 hex
R_PDO4	1 0 1 0	x x x x x x x	1280 (500 hex) + node ID	1403 hex
T_SDO	1 0 1 1	x x x x x x x	1408 (580 hex) + node ID	-
R_SDO	1 1 0 0	x x x x x x x	1536 (600 hex) + node ID	-
NMT error control	1 1 1 0	x x x x x x x	1792 (700 hex) + 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 prior to 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 hex) and 127 (7F hex). 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:

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

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:

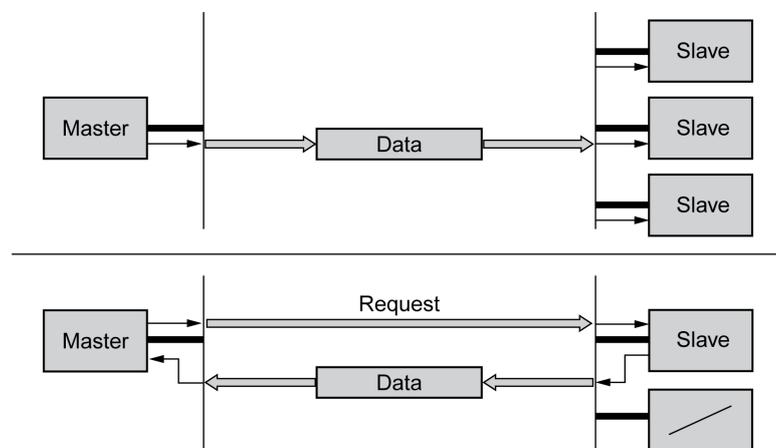
- Master-slave relationship
- Client-server relationship
- Producer-consumer relationship

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.

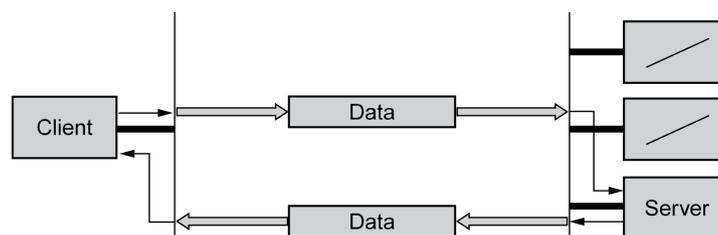
To confirm the message, the master requests a message from a specific slave, which then responds with the required data.

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



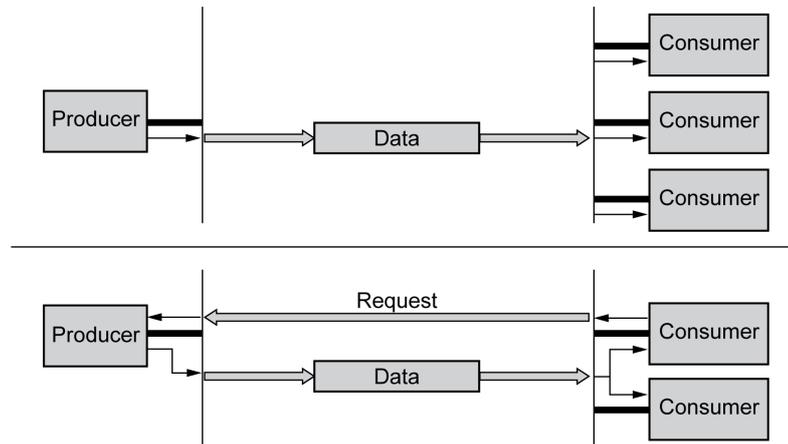
The client addresses and sends a CAN message to a server. The server evaluates the message and sends the response data as an acknowledgement.

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 [PDO Data Exchange](#), page 25 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.

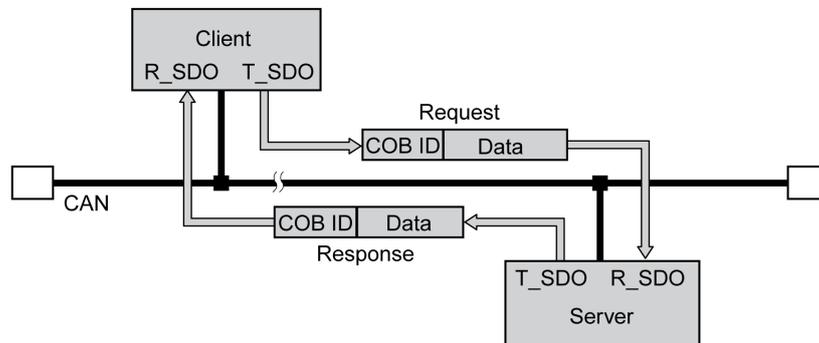
The T_SDO of an SDO client is used to send the request for data exchange; the R_SDO is used to receive. The data frame of an SDO consist of 8 bytes.

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.

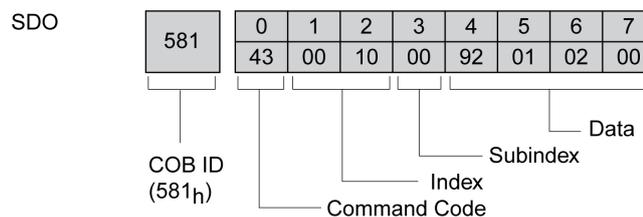
SDO Message

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.

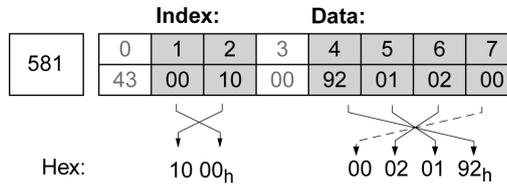
The data frame of an SDO messages consists of:

- 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:



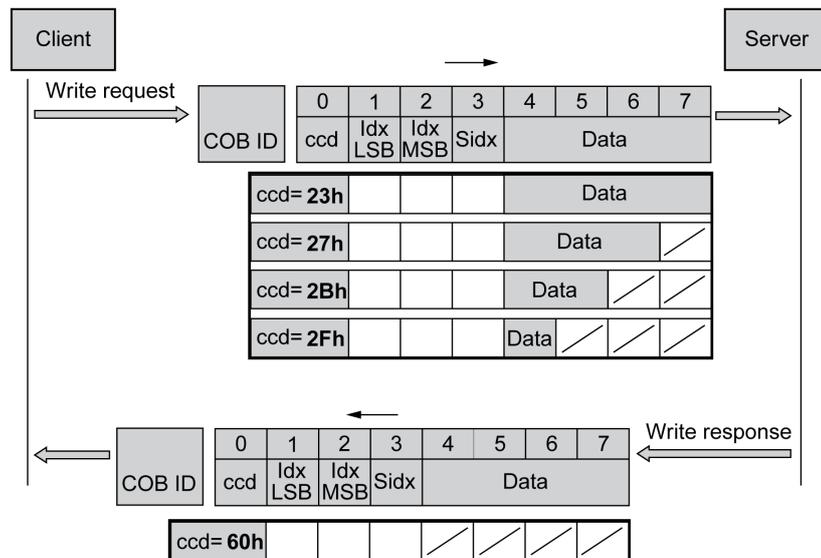
SDO Reading and Writing Data

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.

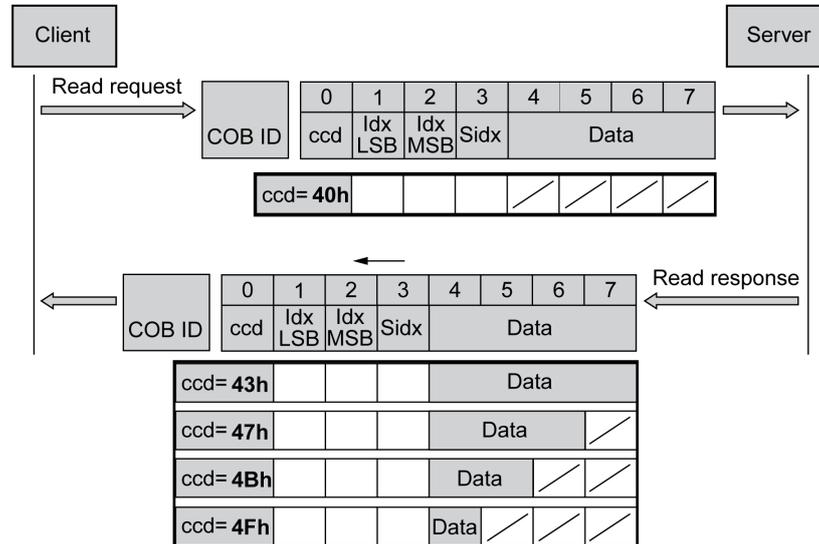
Message type	Data length used				Description
	4 byte	3 byte	2 byte	1 byte	
Write request	23 hex	27 hex	2B hex	2F hex	Transmitting parameters
Write response	60 hex	60 hex	60 hex	60 hex	Confirmation
Error response	80 hex	80 hex	80 hex	80 hex	Error

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

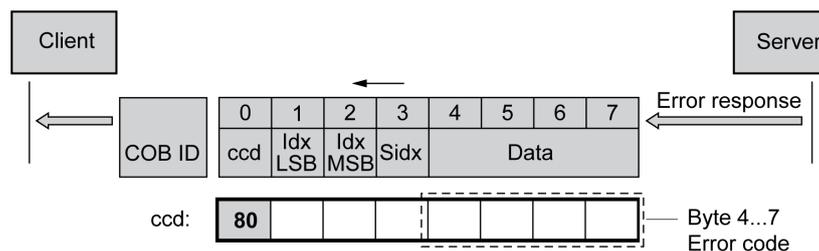
The table below shows the command code for transmitting a read value. It depends on the message type and the transmitted data length.

Message type	Data length used				Description
	4 byte	3 byte	2 byte	1 byte	
Read request	40 hex	40 hex	40 hex	40 hex	Request read value
Read response	43 hex	47 hex	4B hex	4F hex	Return read value
Error response	80 hex	80 hex	80 hex	80 hex	Error

Error Response

If a message could not be evaluated, the server sends an error message. See SDO Error Message ABORT, page 61 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.

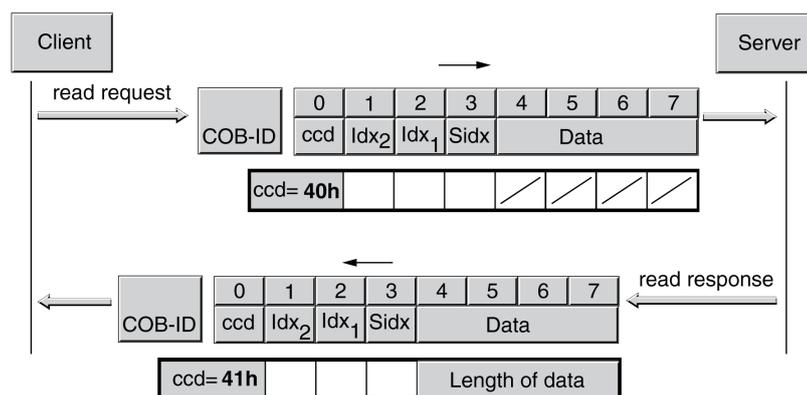
The read request by the SDO client contains the command code "ccd" with the toggle bit and a data segment. The confirmation also contains a toggle bit in the command code "ccd". In the first read request, the toggle bit has the value "0", in the subsequent read requests it toggles between 1 and 0.

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 hex, the index, the subindex and the data length of the object to be read. The command code 41 hex indicates that the object has data with a length of more than 4 bytes.

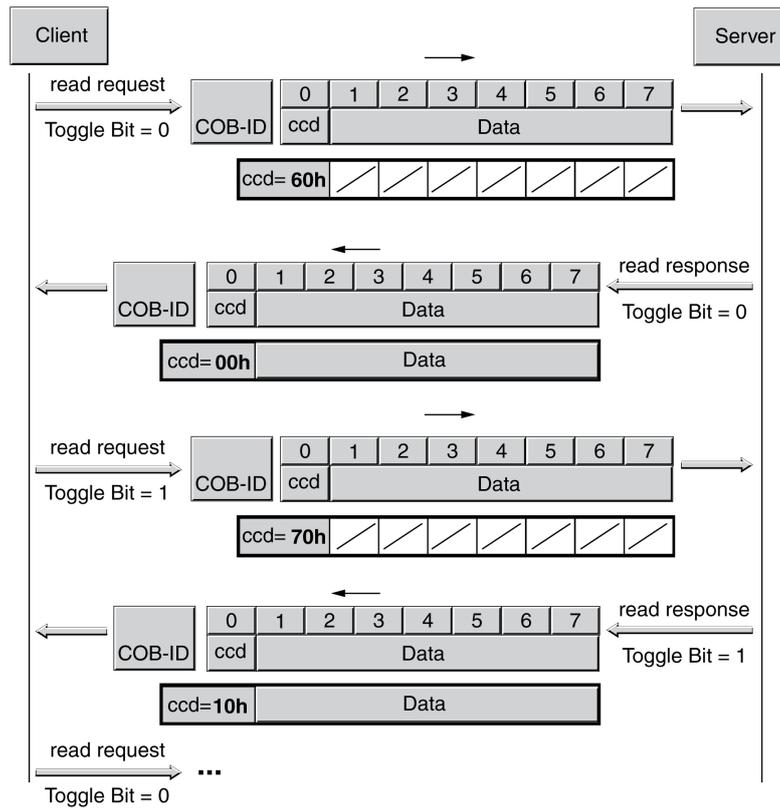
First read request:



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

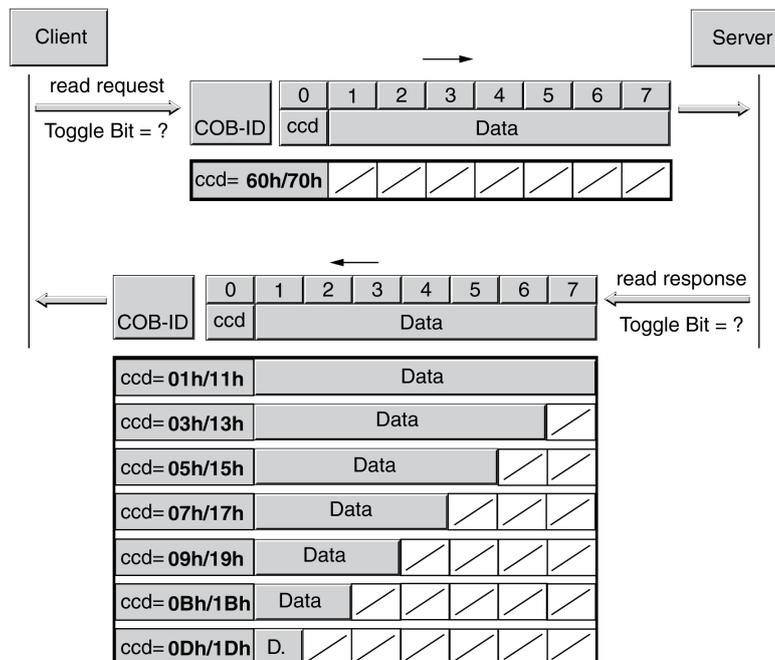
The client must continue to start read requests until the data is transmitted.

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.

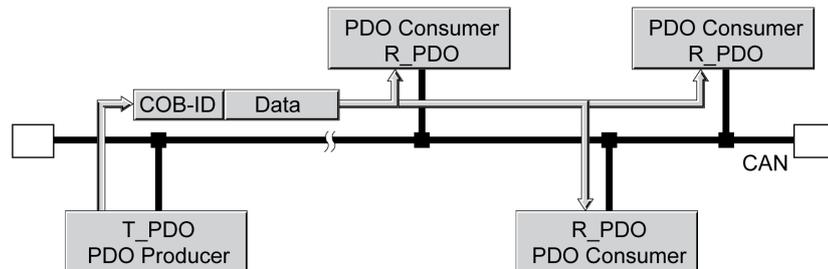
The flexible data length of a PDO message also increases the data throughput. A PDO message can transmit up to 8 bytes of data. If only 2 bytes are assigned, only 2 data bytes are sent.

The length of a PDO message and the assignment of the data fields are specified by PDO mapping. See [PDO Mapping](#), page 30 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_h (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:

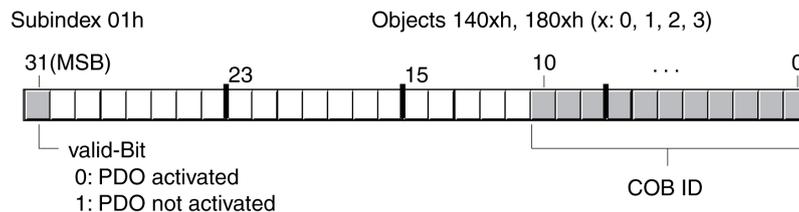
PDO	Object
Settings for R_PDO1	1st receive PDO parameter (1400 hex)
Settings for R_PDO2	2nd receive PDO parameter (1401 hex)
Settings for R_PDO3	3rd receive PDO parameter (1402 hex)
Settings for R_PDO4	4th receive PDO parameter (1403 hex)
Settings for T_PDO1	1st transmit PDO parameter (1800 hex)
Settings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

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 hex of the respective communication object.

Activating PDOs via subindex 01 hex, bit 31:



Example

Setting for R_PDO3 in object 1402 hex:

- Subindex 01 hex = 8000 04xx hex: R_PDO3 not activated
- Subindex 01 hex = 0000 04xx hex: 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.

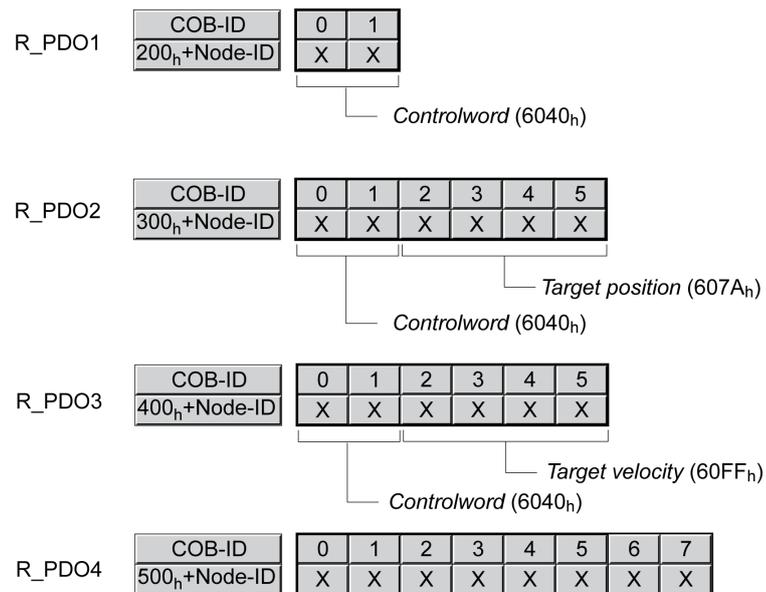
- The time interval "inhibit time" can be used to reduce the CAN bus load, which can be the result of continuous transmission of T_PDOs. If an inhibit time not equal to zero is entered, a transmitted PDO will only be re-transmitted after the inhibit time has elapsed. The time is set with subindex 03 hex.
- 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 hex.

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 hex), 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 hex).

R_PDO2 is evaluated asynchronously, i.e. it is event-driven. R_PDO2 is preset.

For details on the SYNC object see Synchronization, page 31.

R_PDO3

R_PDO3 contains the control word and the target velocity, object *Target velocity* (60FF hex), for the operating mode "Profile Velocity".

R_PDO3 is evaluated asynchronously, i.e. it is event-driven. R_PDO3 is preset.

R_PDO4

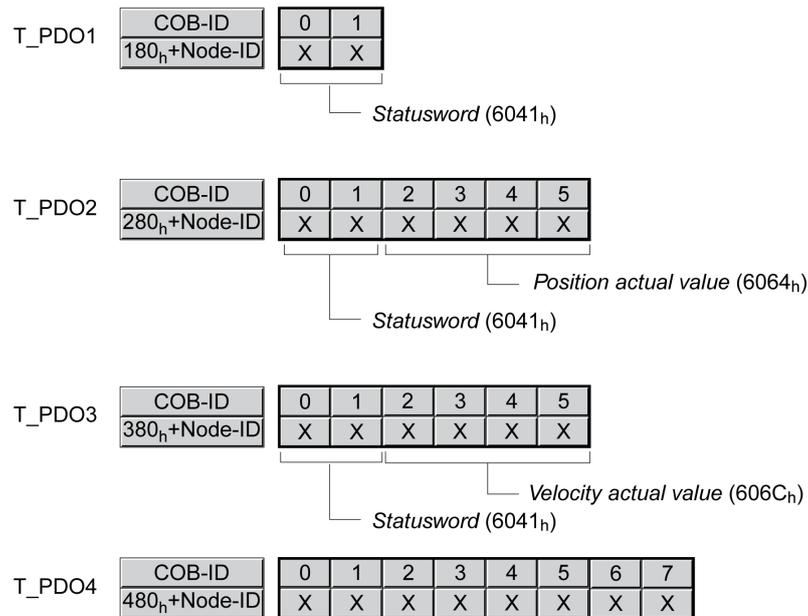
R_PDO4 is used to transmit vendor-specific object values. By default, R_PDO4 is empty.

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 (6041 hex)*, 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 (6064 hex)*, 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 (606C hex)*, 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
<i>CANpdo1Event</i>	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 Modified settings become active immediately.	- 0 1 15	UINT16 R/W - -	CANopen 3041:B _h Modbus 16662
<i>CANpdo2Event</i>	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 Modified settings become active immediately.	- 0 1 15	UINT16 R/W - -	CANopen 3041:C _h Modbus 16664
<i>CANpdo3Event</i>	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 Modified settings become active immediately.	- 0 1 15	UINT16 R/W - -	CANopen 3041:D _h Modbus 16666
<i>CANpdo4Event</i>	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 Modified settings become active immediately.	- 0 15 15	UINT16 R/W - -	CANopen 3041:E _h Modbus 16668

PDO Mapping

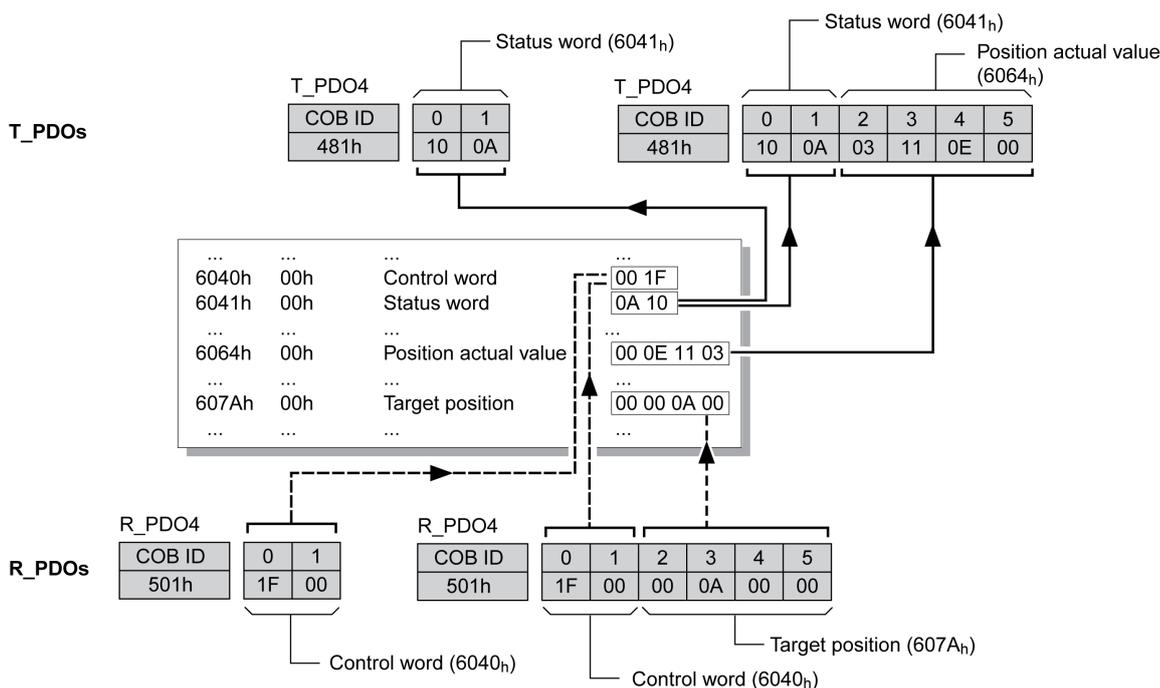
Overview

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.

Section Assignment Object Group 3000 hex, page 67 and Assignment Object Group 6000 hex, page 77 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.

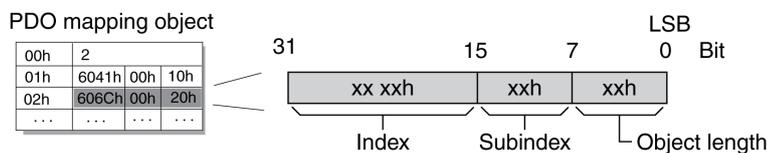
The settings for PDO mapping are defined in an assigned communication object for each PDO.

Object	PDO mapping for	Type
1st receive PDO mapping (1600 hex)	R_PDO1	Dynamic
2nd receive PDO mapping (1601 hex)	R_PDO2	Dynamic
3rd receive PDO mapping (1602 hex)	R_PDO3	Dynamic
4th receive PDO mapping (1603 hex)	R_PDO4	Dynamic
1st transmit PDO mapping (1A00 hex)	T_PDO1	Dynamic
2nd transmit PDO mapping (1A01 hex)	T_PDO2	Dynamic
3rd transmit PDO mapping (1A02 hex)	T_PDO3	Dynamic
4th transmit PDO mapping (1A03 hex)	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 hex of the communication object contains the number of valid subindex entries.

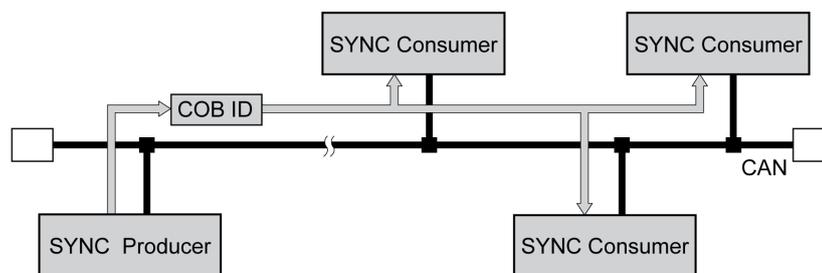
Object length	Bit value
08 hex	8 bits
10 hex	16 bits
20 hex	32 bits

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.

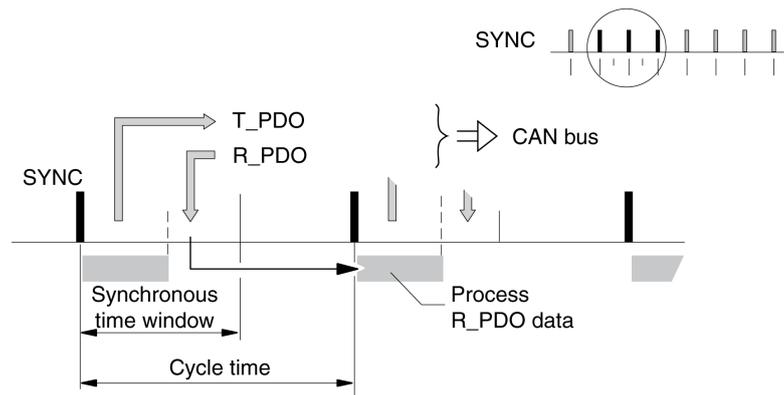


Time Values for Synchronization

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 (1006 hex)*.
- 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 (1007 hex)*.

Synchronization times:



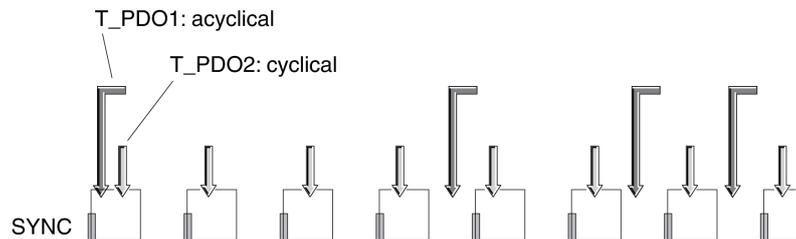
Synchronous Data Transmission

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* (02 hex) of the corresponding PDO parameter, for example, in the object *1st receive PDO parameter* (1400 hex:02 hex) 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 hex) by default. The value can be modified after initialization of the network with the object *COB-ID SYNC Message* (1005 hex).

"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

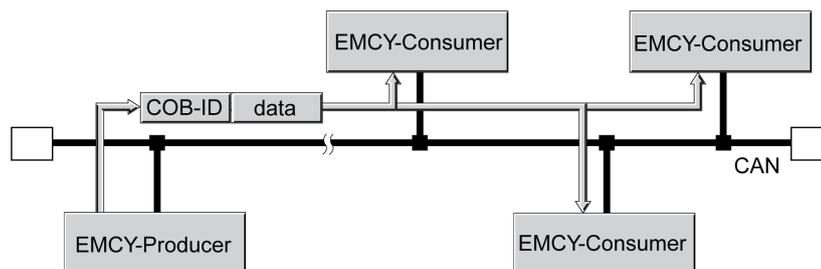
allows an operating mode to be started simultaneously on multiple devices so that, for example, the feed of a portal drive with several motors can be synchronized.

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:

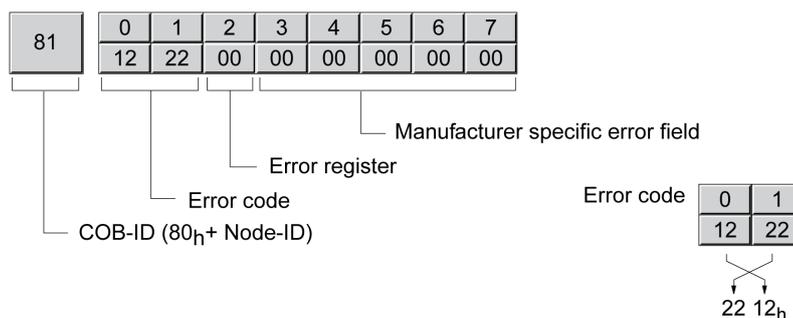


Boot-Up Message

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* (1003:1 hex).

Byte 2: Error register

The value is also saved in the object *Error register* (1001 hex).

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* (603F hex).

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 hex) + node ID

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

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

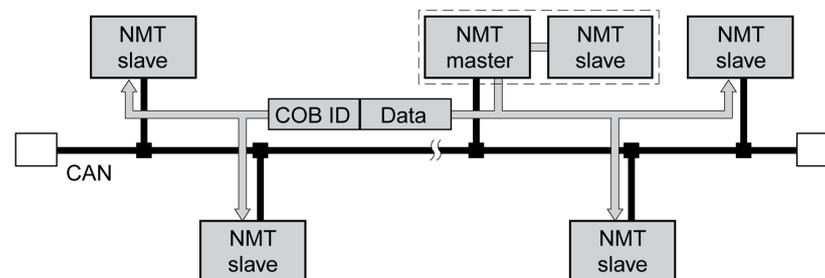
The device saves the error register in the object *Error register* (1001 hex) and the last error that was detected in the object *Error code* (603F hex).

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

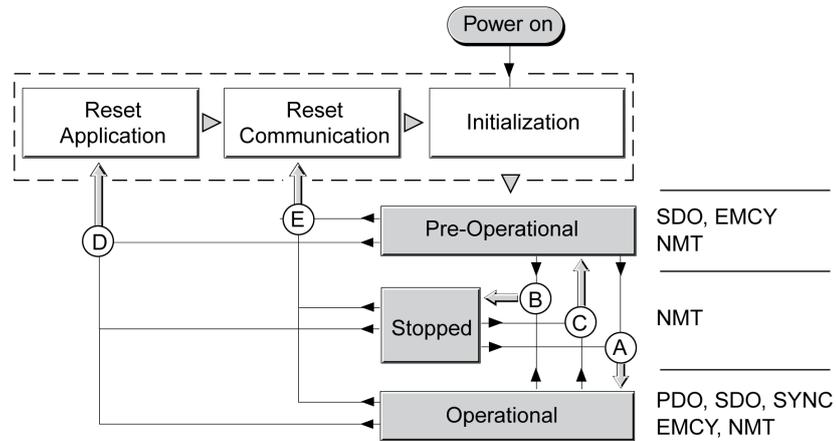
NMT services can be divided into 2 groups:

- 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. NOTE: 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". NOTE: 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.

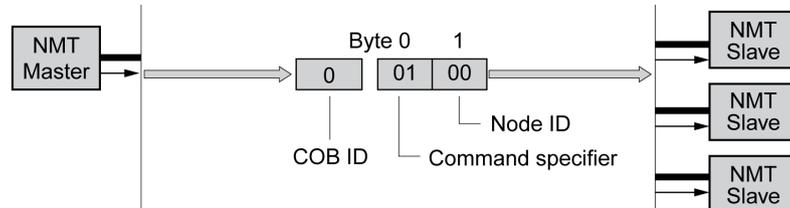
Persistent Data Memory

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 hex)	Start remote node	A
2 (02 hex)	Stop remote node	B
128 (80 hex)	Enter Pre-Operational	C
129 (81 hex)	Reset node	D
130 (82 hex)	Reset communication	E

The second byte addresses the recipient of an NMT message with a node address between 1 and 127 (7F hex). A message with node address "0" is broadcast to the reachable NMT slaves.

NMT Service Node Guarding/Life Guarding

COB ID

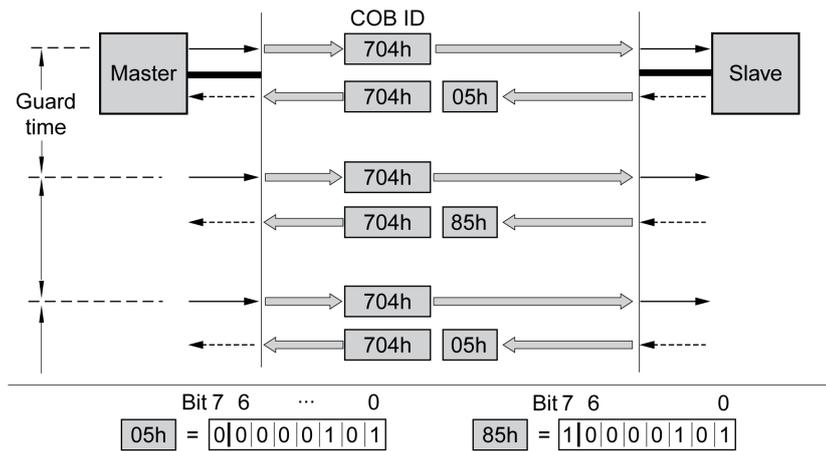
The communication object *NMT error control* (700 hex+Node-ID) is used for connection monitoring. The COB ID for each NMT slave is determined on the basis of the node address:

$$\text{COB ID} = \text{function code } NMT \text{ error control (700 hex)} + \text{Node-ID}.$$

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 hex): "Stopped"
- 5 (05 hex): "Operational"
- 127 (7F hex): "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 hex)
- Life time factor (100D hex)

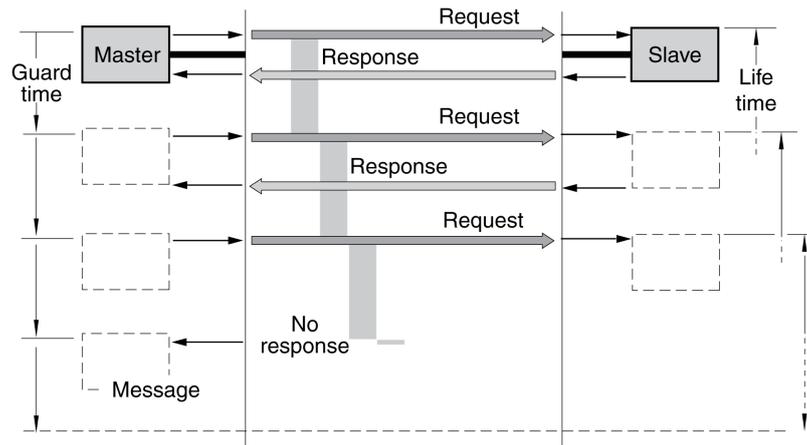
Connection Error

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:



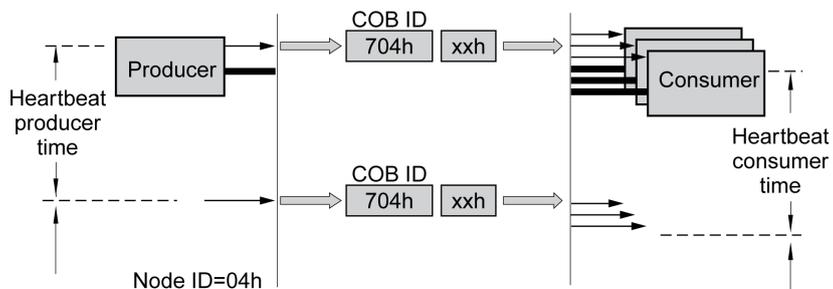
NMT Service Heartbeat

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 (1017 hex)*. One or several consumers can receive this message. *Producer heartbeat time (1017 hex) = 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 (1016 hex)*, it generates an error message (Heartbeat event). *Consumer heartbeat time (1016 hex) = 0* deactivates monitoring by a consumer.



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

- 0 (00 hex): "Boot-Up"
- 4 (04 hex): "Stopped"
- 5 (05 hex): "Operational"
- 127 (7F hex): "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.

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”
www.se.com/en/download/document/Lexium_DTM_Library/
- Fieldbus converter for the commissioning software for connection via the commissioning interface
- Lexium 32A Drive User Guide and this user guide, LXM32A CANopen Interface 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* (CAN address).
- Set the transmission rate in the parameter *CANbaud* (CAN baud) 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
<i>CANaddress</i> CAN F → CAN - CAN address	CANopen address (node number). Modified settings become active the next time the product is powered on.	- 1 - 127	UINT16 R/W per. -	-
<i>CANbaud</i> CAN F → CAN - CAN baud	CANopen baud rate. 50 kBaud / 50 : 50 kBaud 125 kBaud / 125 : 125 kBaud 250 kBaud / 250 : 250 kBaud 500 kBaud / 500 : 500 kBaud 1 MBaud / 1000 : 1 MBaud Modified settings become active the next time the product is powered on.	- 50 250 1000	UINT16 R/W per. -	-

Operating States and Operating Modes

Operating States

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	Description	Unit	Data type	Parameter address via fieldbus
HMI menu		Minimum value	R/W	
HMI name		Factory setting	Persistent	
		Maximum value	Expert	
<i>_DCOMstatus</i>	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: x_err Bit 14: x_end Bit 15: ref_ok	- - - -	UINT16 R/- - -	CANopen 6041:0h 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	N/A	0	0	0	0
3 Switch On Disabled	1	N/A	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

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
8 Fault Reaction Active	0	N/A	1	1	1	1
9 Fault	0	N/A	1	0	0	0

Bit 4

Bit 4=1 indicates that 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 sections about 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 sections about 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 response corresponds 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	Description	Unit	Data type	Parameter address via fieldbus
HMI menu		Minimum value	R/W	
HMI name		Factory setting	Persistent	
		Maximum value	Expert	
<i>DCOMcontrol</i>	DriveCom control word. See 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) Modified settings become active immediately.	- - - -	UINT16 R/W - -	CANopen 6040:0h 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	4 Ready To Switch On	0	N/A	1	1	0
Switch On	T3	5 Switched On	0	0	1	1	1
Disable Voltage	T7, T9, T10, T12	3 Switch On Disabled	0	N/A	N/A	0	N/A
Quick Stop	T7, T10 T11	3 Switch On Disabled 7 Quick Stop Active	0	N/A	0	1	N/A
Disable Operation	T5	5 Switched On	0	0	1	1	1
Enable Operation	T4, T16	6 Operation Enabled	0	1	1	1	1
Fault Reset	T15	3 Switch On Disabled	0->1	N/A	N/A	N/A	N/A

For details on the state transitions, refer to the user guide of the drive.

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 section.

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 section.

Bits 10 ... 15

Reserved.

Operating Modes

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
<i>DCOMopmode</i>	Operating mode. -6 / Manual Tuning / Autotuning: Manual Tuning or Autotuning -1 / Jog: Jog 0 / Reserved: Reserved 1 / Profile Position: Profile Position 3 / Profile Velocity: Profile Velocity 4 / Profile Torque: Profile Torque 6 / Homing: Homing 7 / Interpolated Position: Interpolated Position 8 / Cyclic Synchronous Position: Cyclic Synchronous Position 9 / Cyclic Synchronous Velocity: Cyclic Synchronous Velocity 10 / Cyclic Synchronous Torque: Cyclic Synchronous Torque Modified settings become active immediately. * Datatype for CANopen: INT8	- -6 - 7	INT16* R/W - -	CANopen 6060:0h Modbus 6918

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

Parameter name	Description	Unit	Data type	Parameter address via fieldbus
HMI menu		Minimum value	R/W	
HMI name		Factory setting	Persistent	
		Maximum value	Expert	
<i>_DCOMopmd_act</i>	<p>Active operating mode.</p> <p>-6 / Manual Tuning / Autotuning: Manual Tuning / Autotuning</p> <p>-1 / Jog: Jog</p> <p>0 / Reserved: Reserved</p> <p>1 / Profile Position: Profile Position</p> <p>3 / Profile Velocity: Profile Velocity</p> <p>4 / Profile Torque: Profile Torque</p> <p>6 / Homing: Homing</p> <p>7 / Interpolated Position: Interpolated Position</p> <p>8 / Cyclic Synchronous Position: Cyclic Synchronous Position</p> <p>9 / Cyclic Synchronous Velocity: Cyclic Synchronous Velocity</p> <p>10 / Cyclic Synchronous Torque: Cyclic Synchronous Torque</p> <p>* Datatype for CANopen: INT8</p>	<p>-</p> <p>-6</p> <p>0</p> <p>10</p>	<p>INT16*</p> <p>R/-</p> <p>-</p> <p>-</p>	<p>CANopen 6061:0h</p> <p>Modbus 6920</p>

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

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 [Changing the Operating State](#), page 44.

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 Indication of the Operating State, page 42.

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	Description	Unit	Data type	Parameter address via fieldbus
HMI menu		Minimum value	R/W	
HMI name		Factory setting	Persistent	
		Maximum value	Expert	
<i>PTtq_target</i>	Target torque. 100.0 % correspond to the continuous stall torque <i>_M_M_0</i> . In increments of 0.1 %. Modified settings become active immediately.	% -3000.0 0.0 3000.0	INT16 R/W - -	CANopen 6071:0 _n 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 Changing the Operating State, page 44.

Status Word

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

For the common bits of the Status Word see Indication of the Operating State, page 42.

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	Description	Unit	Data type	Parameter address via fieldbus
HMI menu		Minimum value	R/W	
HMI name		Factory setting	Persistent	
		Maximum value	Expert	
<i>PVv_target</i>	Target velocity. The target velocity is limited to the setting in CTRL_v_max and RAMP_v_max. Modified settings become active immediately.	usr_v - 0 -	INT32 R/W - -	CANopen 60FF:0h 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 [Changing the Operating State](#), page 44.

Status Word

Parameter <i>DCOMstatus</i>	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 [Indication of the Operating State](#), page 42.

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
<i>PPp_target</i>	Target position for operating mode Profile Position. Minimum/maximum values depend on: - Scaling factor - Software limit switches (if they are activated) Modified settings become active immediately.	usr_p - - -	INT32 R/W - -	CANopen 607A:0h Modbus 6940
<i>PVv_target</i>	Target velocity. The target velocity is limited to the setting in <i>CTRL_v_max</i> and <i>RAMP_v_max</i> . Modified settings become active immediately.	usr_v - 0 -	INT32 R/W - -	CANopen 60FF:0h 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.
N/A	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 [Changing the Operating State](#), page 44.

Status Word

Parameter <i>DCOMstatus</i>	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 Indication of the Operating State, page 42.

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 hex	1 hex	4	80000200 hex + node id	Deactivate R_PDO1
1800 hex	1 hex	4	80000180 hex + node id	Deactivate T_PDO1
1401 hex	1 hex	4	00000300 hex + node id	Activate R_PDO2
1801 hex	1 hex	4	00000280 hex + node id	Activate T_PDO2
1402 hex	1 hex	4	80000400 hex + node id	Deactivate R_PDO3
1802 hex	1 hex	4	80000380 hex + node id	Deactivate T_PDO3
1403 hex	1 hex	4	80000500 hex + node id	Deactivate R_PDO4
1803 hex	1 hex	4	80000480 hex + node id	Deactivate T_PDO4
1401 hex	2 hex	1	1 hex	Activate cyclic transmission of R_PDO2
1801 hex	2 hex	1	1 hex	Activate cyclic transmission of T_PDO2
6040 hex	0 hex	2	0 hex	Control word = 0
6040 hex	0 hex	2	80 hex	Perform Fault Reset
1601 hex	0 hex	1	0 hex	Change PDO mapping for R_PDO2
1601 hex	1 hex	4	60400010 hex	Map control word
1601 hex	2 hex	4	60C10120 hex	Map reference position for Interpolated Position
1601 hex	0 hex	1	2 hex	Finalize mapping for R_PDO2
1A01 hex	0 hex	1	0 hex	Change PDO mapping for T_PDO2
1A01 hex	1 hex	4	60410010 hex	Map status word
1A01 hex	2 hex	4	60640020 hex	Map Position actual Value

Index	Subindex	Length in bytes	Value	Meaning
1A01 hex	0 hex	1	2 hex	Finalize mapping for T_PDO2
3006 hex	7 hex	4	20000 hex	Position scaling: denominator
3006 hex	8 hex	4	1 hex	Position scaling: numerator
6060 hex	0 hex	1	7 hex	Select operating mode Interpolated Position
3006 hex	3D hex	2	1 hex	Must be written for reasons of compatibility
60C2 hex	1 hex	1	2 hex	Cycle time 2 ms (example)
3012 hex	6 hex	2	3E8 hex	Velocity feed-forward control 100% CTRL1
3013 hex	6 hex	2	3E8 hex	Velocity feed-forward control 100% CTRL2
3006 hex	6 hex	2	1 hex	Suppress error message for LIMP or LIMN when the power stage is enabled
3022 hex	4 hex	2	1 hex	Tolerance for synchronization mechanism (example)
3022 hex	5 hex	2	2 hex	Activate synchronization mechanism

Control Word

Parameter <i>DCOMcontrol</i>	Meaning
Bit 4	0: Terminate operating mode 1: Start operating mode NOTE: 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 [Changing the Operating State](#), page 44.

Status Word

Parameter <i>DCOMstatus</i>	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 [Indication of the Operating State](#), page 42.

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
<i>HMmethod</i>	<p>Homing method.</p> <p>1: LIMN with index pulse</p> <p>2: LIMP with index pulse</p> <p>7: REF+ with index pulse, inv., outside</p> <p>8: REF+ with index pulse, inv., inside</p> <p>9: REF+ with index pulse, not inv., inside</p> <p>10: REF+ with index pulse, not inv., outside</p> <p>11: REF- with index pulse, inv., outside</p> <p>12: REF- with index pulse, inv., inside</p> <p>13: REF- with index pulse, not inv., inside</p> <p>14: REF- with index pulse, not inv., outside</p> <p>17: LIMN</p> <p>18: LIMP</p> <p>23: REF+, inv., outside</p> <p>24: REF+, inv., inside</p> <p>25: REF+, not inv., inside</p> <p>26: REF+, not inv., outside</p> <p>27: REF-, inv., outside</p> <p>28: REF-, inv., inside</p> <p>29: REF-, not inv., inside</p> <p>30: REF-, not inv., outside</p> <p>33: Index pulse negative direction</p> <p>34: Index pulse positive direction</p> <p>35: Position setting</p> <p>Abbreviations:</p> <p>REF+: Search movement in positive direction</p> <p>REF-: Search movement in negative direction</p> <p>inv.: Invert direction in switch</p> <p>not inv.: Direction not inverted in switch</p> <p>outside: Index pulse / distance outside switch</p> <p>inside: Index pulse / distance inside switch</p> <p>Modified settings become active immediately.</p> <p>* Datatype for CANopen: INT8</p>	- 1 18 35	INT16* R/W - -	CANopen 6098:0 _h Modbus 6936

Control Word

Parameter <i>DCOMcontrol</i>	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 Changing the Operating State, page 44.

Status Word

Parameter <i>DCOMstatus</i>	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 Indication of the Operating State, page 42.

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 hex 0064 hex
Fast velocity to 250 →601 / 23 29 30 05 FA 00 00 00 ←581 / 60 29 30 05 00 00 00 00	3029:5 hex 00FA hex
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	6060 hex

Work step	Object		
COB ID / data	Value		
→601 / 2F 60 60 00 FF 00 00 00 ←581 / 60 60 60 00 00 00 00 00	FF hex		
Check operating mode ⁽¹⁾ →601 / 40 61 60 00 00 00 00 00 Operating mode active ←581 / 4F 61 60 00 FF 61 01 00	6061 hex FF hex		
Start movement (positive direction, slow) →601 / 2B 1B 30 09 01 00 00 00 ←581 / 60 1B 30 09 00 00 00 00 T_PDO1 with status word ←181 / 37 02	301B:9 hex 01 hex		
Start movement (positive direction, fast) →601 / 2B 1B 30 09 05 00 00 00 ←581 / 60 1B 30 09 00 00 00 00 T_PDO1 with status word ←181 / 37 02	301B:9 hex 05 hex		
Terminate movement →601 / 2B 1B 30 09 00 00 00 00 ←581 / 60 1B 30 09 00 00 00 00 T_PDO1 with status word ←181 / 37 42	301B:9 hex 00 hex		
<table border="1"> <tr> <td>(1)</td> <td>The operating mode must be verified to have taken effect until the drive has activated the specified operating mode.</td> </tr> </table>		(1)	The operating mode must be verified to have taken effect until the drive has activated the specified operating mode.
(1)	The operating mode must be verified to have taken effect until the drive has activated the specified operating mode.		

Operating Mode Profile Torque

Work step	Object
COB ID / data	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 hex 04 hex
Check operating mode ⁽¹⁾	6061 hex

Work step COB ID / data	Object Value		
→601 / 40 61 60 00 00 00 00 00 Operating mode active ←581 / 4F 61 60 00 04 61 01 00	04 hex		
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 hex 64 hex		
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			
<table border="1"> <tr> <td>(1)</td> <td>The operating mode must be verified to have taken effect until the drive has activated the specified operating mode.</td> </tr> </table>		(1)	The operating mode must be verified to have taken effect until the drive has activated the specified operating mode.
(1)	The operating mode must be verified to have taken effect until the drive 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 hex 0400 0401 hex
Activate T_PDO3 →601 / 23 02 18 01 81 03 00 04 ←581 / 60 02 18 01 00 00 00 00	1802:1 hex 0400 0381 hex
Set acceleration to 2000 →601 / 23 83 60 00 D0 07 00 00 ←581 / 60 83 60 00 00 00 00 00	6083 hex 0000 07D0 hex
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	

Work step	Object
COB ID / data	Value
Starting the operating mode →601 / 2F 60 60 00 03 00 00 00 ←581 / 60 60 60 00 00 00 00 00	6060 hex 03 hex
Check operating mode ⁽¹⁾ →601 / 40 61 60 00 00 00 00 00 Operating mode active ←581 / 4F 61 60 00 03 61 01 00	6061 hex 03 hex
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 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 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 00	
(1)	The operating mode must be verified to have taken effect until the drive has activated the specified operating mode.

Operating Mode Profile Position

Work step	Object
COB ID / data	Value
Activate R_PDO2 →601 / 23 01 14 01 01 03 00 04 ←581 / 60 01 14 01 00 00 00 00	1401:1 hex 0400 0301 hex
Activate T_PDO2 →601 / 23 01 18 01 81 02 00 04 ←581 / 60 01 18 01 00 00 00 00	1801:1 hex 0400 0281 hex
Set acceleration to 2000 →601 / 23 83 60 00 D0 07 00 00 ←581 / 60 83 60 00 00 00 00 00	6083 hex 0000 07D0 hex
Set deceleration to 4000 →601 / 23 84 60 00 A0 0F 00 00 ←581 / 60 84 60 00 00 00 00 00	6084 hex 0000 0FA0 hex
Set target velocity to 4000 →601 / 23 81 60 00 A0 0F 00 00 ←581 / 60 81 60 00 00 00 00 00	6081 hex 0000 0FA0 hex

Work step COB ID / data	Object Value
NMT Start remote node → 0 / 01 00 T_PDO2 with status word ← 281 / 31 66 00 00 00 00	
Enable power stage with R_PDO2 → 301 / 00 00 00 00 00 00 → 301 / 06 00 00 00 00 00 → 301 / 0F 00 00 00 00 00 T_PDO2 (operating state: 6 Operation Enabled) ← 281 / 37 42 00 00 00 00	
Starting the operating mode → 601 / 2F 60 60 00 01 00 00 00 ← 581 / 60 60 60 00 00 00 00 00	6060 hex 01 hex
Check operating mode ⁽¹⁾ → 601 / 40 61 60 00 00 00 00 00 Operating mode active ← 581 / 4F 61 60 00 01 61 01 00	6061 hex 01 hex
R_PDO2: Start relative movement with NewSetpoint=1 → 301 / 5F 00 30 75 00 00 T_PDO2 with status word and position actual value ← 281 / 37 12 00 00 00 00 Target position reached ← 281 / 37 56 30 75 00 00	
R_PDO2: NewSetpoint=0 → 301 / 4F 00 30 75 00 00	
(1)	The operating mode must be verified to have taken effect until the drive has activated the specified operating mode.

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 hex 0000 0064 hex
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 hex 0000 000A hex
NMT Start remote node → 0 / 01 00 T_PDO1 with status word ← 181 / 31 62	
Enable power stage with R_PDO1	

Work step COB ID / data	Object Value
→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 hex 06 hex
Check operating mode ⁽¹⁾ →601 / 40 61 60 00 00 00 00 00 Operating mode active ←581 / 4F 61 60 00 06 61 01 00	6061 hex 06 hex
Select method 17 →601 / 2F 98 60 00 11 00 00 00 ←581 / 60 98 60 00 00 00 00 00	6098 hex 11 hex
Start reference movement (Homing operation start) →201 / 1F 00 T_PDO1 reference movement active ←181 / 37 02 T_PDO1 reference movement terminated ←181 / 37 D6	
(1)	The operating mode must be verified to have taken effect until the drive has activated the specified operating mode.

Diagnostics and Troubleshooting

Fieldbus Communication Error Diagnostics

Verifying 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](#), page 40.

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:

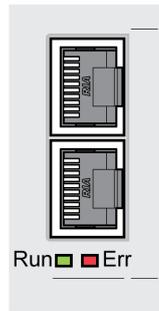
- Power cycle the drive.
- Observe the network messages. After initialization of the bus, the device sends a boot-up message (COB ID 700 hex + node ID and 1 data byte with the content 00 hex).

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

Fieldbus Status LEDs

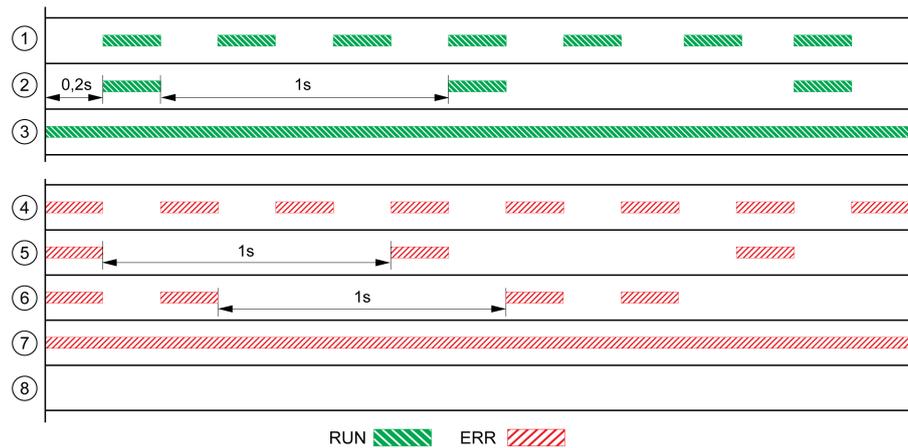
Overview

The fieldbus status LEDs represent 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 hex)
Operating states, see *Operating States*, page 42
- Object *EMCY* (80 hex + Node-ID)
Error message from a device with error and error code, see *Emergency Object Service*, page 33
- Object *Error register* (1001 hex)
- Object *Error code* (603F hex)
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 hex). 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* (1001 hex) and *Error code* (603F hex). For information on the object *EMCY* see *Emergency Object Service*, page 33.

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

Error Register

The object *Error register* (1001 hex) 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

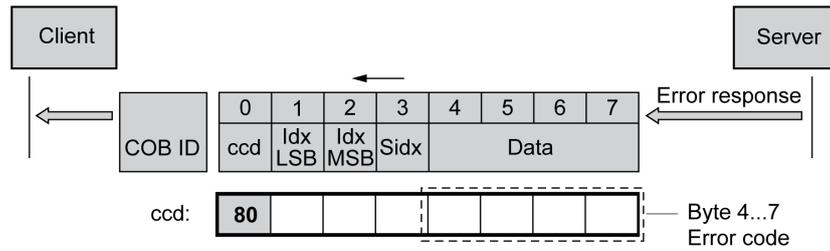
Error Code Table

The error code is evaluated with the object *error code* (603F hex), 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 drive 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 hex	Toggle bit not toggled
0504 0000 hex	Time-out during SDO transfer
0504 0001 hex	Command specifier CS incorrect or indeterminable
0504 0005 hex	No memory available
0601 0000 hex	Access to object not possible
0601 0001 hex	No read access, because write-only object (wo)
0601 0002 hex	No write access, because read object (ro)
0602 0000 hex	Object does not exist in object dictionary
0604 0041 hex	Object does not support PDO mapping
0604 0042 hex	PDO mapping: Number or length of objects exceed the byte length of the PDO
0604 0043 hex	Parameters are incompatible
0604 0047 hex	Device detects internal incompatibility
0606 0000 hex	Hardware error, access denied
0607 0010 hex	Data type and parameter length do not match
0607 0012 hex	Data type does not match, parameter too long
0607 0013 hex	Data type does not match, parameter too short
0609 0011 hex	Subindex not supported
0609 0030 hex	Value range of parameter too large (relevant only for write access)
0609 0031 hex	Parameter values above maximum value
0609 0032 hex	Parameter values below minimum value
0609 0036 hex	Upper value is less than lower value
0800 0000 hex	General error. See parameter <i>_ManuSdoAbort</i> after this table. This parameter contains the drive specific error code.
0800 0020 hex	Data can neither be transmitted to the application nor saved.
0800 0021 hex	Local control mode, data can neither be transmitted nor saved.
0800 0022 hex	Data can neither be transmitted nor saved in this device state.
0800 0023 hex	Object dictionary does not exist or cannot be generated (for example, if data error occurs during generation from file)
0800 0024 hex	Data not available.

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

Object Dictionary

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 hex

Overview

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

Index (hex)	Sub-index (hex)	Name	Object code	Data type	Access	Description
1400	02	Transmission type R_PDO1	VAR	Unsigned8	RW	Transmission type
1401	-	2nd receive PDO parameter	REC	PDO comm. param.	RW	Second receive PDO (R_PDO2), settings
1401	01	COB-ID R_PDO2	VAR	Unsigned32	RW	Identifier of the R_PDO2
1401	02	Transmission type R_PDO2	VAR	Unsigned8	RW	Transmission type
1402	-	3rd receive PDO parameter	REC	PDO comm. param.	RW	Third receive PDO (R_PDO3), settings
1402	01	COB-ID R_PDO3	VAR	Unsigned32	RW	Identifier of the R_PDO3
1402	02	Transmission type R_PDO3	VAR	Unsigned8	RW	Transmission type
1403	-	4th receive PDO parameter	REC	PDO comm. param.	RW	Fourth receive PDO (R_PDO4), settings
1403	01	COB-ID R_PDO4	VAR	Unsigned32	RW	Identifier of the R_PDO4
1403	02	Transmission type R_PDO4	VAR	Unsigned8	RW	Transmission type
1600	-	1st receive PDO mapping	REC	PDO mapping	RO	PDO mapping for R_PDO1, settings
1600	01	1st mapped object R_PDO1	VAR	Unsigned32	RO	First object for the mapping in R_PDO1
1601	-	2nd receive PDO mapping	REC	PDO mapping	RO	PDO mapping for R_PDO2, settings
1601	01	1st mapped object R_PDO2	VAR	Unsigned32	RO	First object for the mapping in R_PDO2
1601	02	2nd mapped object R_PDO2	VAR	Unsigned32	RO	Second object for the mapping in R_PDO2
1602	-	3rd receive PDO mapping	REC	PDO mapping	RO	PDO mapping for R_PDO3, settings
1602	01	1st mapped object R_PDO3	VAR	Unsigned32	RO	First object for the mapping in R_PDO3
1602	02	2nd mapped object R_PDO3	VAR	Unsigned32	RO	Second object for the mapping in R_PDO3
1603	-	4th receive PDO mapping	REC	PDO mapping	RW	PDO mapping for R_PDO3, settings
1603	01	1st mapped object R_PDO4	VAR	Unsigned32	RW	First object for the mapping in R_PDO4
1603	02	2nd mapped object R_PDO4	VAR	Unsigned32	RW	Second object for the mapping in R_PDO4
1603	03	3rd mapped object R_PDO4	VAR	Unsigned32	RW	Third object for mapping in R_PDO4
1800	-	1st transmit PDO parameter	REC	PDO comm. param.	RW	First transmit PDO (T_PDO1), settings
1800	01	COB-ID T_PDO1	VAR	Unsigned32	RW	Identifier of the T_PDO1
1800	02	Transmission type T_PDO1	VAR	Unsigned8	RW	Transmission type
1800	03	Inhibit time T_PDO1	VAR	Unsigned16	RW	Inhibit time for locking bus access (1=100µs)
1800	04	Reserved T_PDO1	VAR	Unsigned8	RW	Priority for CAN bus arbitration ([0-7]).
1800	05	Event timer T_PDO1	VAR	Unsigned16	RW	Time span for event triggering (1=1 ms)
1801	-	2nd transmit PDO parameter	REC	PDO comm. param.	RW	Second transmit PDO (T_PDO2), settings
1801	01	COB-ID T_PDO2	VAR	Unsigned32	RW	Identifier of the T_PDO2
1801	02	Transmission type T_PDO2	VAR	Unsigned8	RW	Transmission type
1801	03	Inhibit time T_PDO2	VAR	Unsigned16	RW	Inhibit time for locking bus access (1=100µs)
1801	04	Reserved T_PDO2	VAR	Unsigned8	RW	Reserved
1801	05	Event timer T_PDO2	VAR	Unsigned16	RW	Time span for event triggering (1=1 ms)
1802	-	3rd transmit PDO parameter	REC	PDO comm. param.	RW	Third transmit PDO (T_PDO3), settings
1802	01	COB-ID T_PDO3	VAR	Unsigned32	RW	Identifier of the T_PDO3
1802	02	Transmission type T_PDO3	VAR	Unsigned8	RW	Transmission type

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

Assignment Object Group 3000 hex

Overview

The product provides corresponding parameters for the CANopen object group 3000 hex.

Address (hex)	Object	PDO	Data type	Parameter name
3001:1	Firmware number of device	-	UINT32	<i>_prgNoDEV</i>
3001:2	Firmware version of device	-	UINT16	<i>_prgVerDEV</i>
3001:4	Firmware revision of device	-	UINT16	<i>_prgRevDEV</i>
3001:C	Access channel information	T_PDO	UINT16	<i>_AccessInfo</i>
3001:E	Locking other access channels	-	UINT16	<i>AccessLock</i>
3001:43	User-specific data 1	-	UINT32	<i>UsrAppDataMem1</i>
3001:44	User-specific data 2	-	UINT32	<i>UsrAppDataMem2</i>
3002:12	Hardware version of control board	T_PDO	UINT16	<i>_hwVersCPU</i>
3002:14	Hardware version of power stage	T_PDO	UINT16	<i>_hwVersPS</i>
3004:1	Save parameter values to the nonvolatile memory	-	UINT16	<i>PAReeprSave</i>

Address (hex)	Object	PDO	Data type	Parameter name
3004:7	Reset control loop parameters	-	UINT16	<i>PAR_CTRLreset</i>
3004:8	Reset user parameters	-	UINT16	<i>PARuserReset</i>
3004:14	Recalculation of parameters with user-defined units	-	UINT16	<i>PAR_ScalingStart</i>
3004:15	Status of recalculation of the parameters with user-defined units	T_PDO	UINT16	<i>_PAR_ScalingState</i>
3004:16	Additional information on error detected during recalculation	T_PDO	UINT32	<i>_PAR_ScalingError</i>
3004:1D	Configuration modification monitoring	-	UINT16	<i>MON_ConfModification</i>
3005:4	Enabling the power stage as set via IO_AutoEnable even after error	-	UINT16	<i>IO_AutoEnaConfig</i>
3005:5	Commutation monitoring	-	UINT16	<i>MON_commutat</i>
3005:6	Enabling the power stage at PowerOn	-	UINT16	<i>IO_AutoEnable</i>
3005:7	Additional time delay for releasing the holding brake	-	INT16	<i>BRK_AddT_release</i>
3005:8	Additional time delay for applying the holding brake	-	INT16	<i>BRK_AddT_apply</i>
3005:9	Selection of type of braking resistor	-	UINT16	<i>RESint_ext</i>
3005:A	Error response to missing mains phase	-	UINT16	<i>ErrorResp_Flt_AC</i>
3005:B	Error response to excessively high load-dependent position deviation	-	UINT16	<i>ErrorResp_p_dif</i>
3005:F	Detection and monitoring of mains phases	-	UINT16	<i>MON_MainsVolt</i>
3005:10	Ground monitoring	-	UINT16	<i>MON_GroundFault</i>
3005:11	Maximum permissible switch-on time of external braking resistor	-	UINT16	<i>RESext_ton</i>
3005:12	Nominal power of external braking resistor	-	UINT16	<i>RESext_P</i>
3005:13	Resistance value of external braking resistor	-	UINT16	<i>RESext_R</i>
3005:16	Adjustment of absolute position of encoder 1	-	INT32	<i>ENC1_adjustment</i>
3005:20	Mains reactor	-	UINT16	<i>Mains_reactor</i>
3005:21	Shifting of the encoder working range	-	UINT16	<i>ShiftEncWorkRang</i>
3005:22	Error response to 100% I2t braking resistor	-	UINT16	<i>ErrorResp_I2tRES</i>
3005:23	Simulation of absolute position at power cycling	-	UINT16	<i>SimAbsolutePos</i>
3005:26	DC bus compatibility LXM32 and ATV32	-	UINT16	<i>DCbus_compat</i>
3005:34	Additional 'Fault Reset' for the signal input function 'Enable'	-	UINT16	<i>IO_FaultResOnEnalnp</i>
3005:3A	Error response to detected error with quasi absolute position	-	UINT16	<i>ErrorResp_QuasiAbs</i>
3005:3C	Error response to excessively high load-dependent velocity deviation	-	UINT16	<i>ErrorResp_v_dif</i>
3005:3D	DC bus overvoltage monitoring threshold	-	UINT16	<i>MON_DCbusVdcThresh</i>
3005:3E	Value for increased threshold for commutation monitoring	-	INT16	<i>CommutCntCred</i>
3006:1	Acceleration and deceleration of the motion profile for velocity	R_PDO	UINT16	<i>RAMP_v_sym</i>
3006:2	Acceleration and deceleration for the Drive Profile Lexium	-	UINT32	<i>RAMPaccdec</i>
3006:3	Activation of software limit switches	-	UINT16	<i>MON_SW_Limits</i>
3006:6	Response to active limit switch during enabling of power stage	-	UINT16	<i>IOsigRespOfPS</i>
3006:7	Position scaling: Denominator	-	INT32	<i>ScalePOSdenom</i>
3006:8	Position scaling: Numerator	-	INT32	<i>ScalePOSnum</i>
3006:C	Inversion of direction of movement	-	UINT16	<i>InvertDirOfMove</i>
3006:D	Jerk limitation of the motion profile for velocity	-	UINT16	<i>RAMP_v_jerk</i>
3006:E	Signal evaluation for reference switch	-	UINT16	<i>IOsigREF</i>
3006:F	Signal evaluation for negative limit switch	-	UINT16	<i>IOsigLIMN</i>
3006:10	Signal evaluation for positive limit switch	-	UINT16	<i>IOsigLIMP</i>
3006:12	Deceleration ramp for Quick Stop	-	UINT32	<i>RAMPquickstop</i>

Address (hex)	Object	PDO	Data type	Parameter name
3006:16	Absolute positioning only after homing	-	UINT16	<i>AbsHomeRequest</i>
3006:18	Quick Stop option code	-	INT16	<i>LIM_QStopReact</i>
3006:19	Monitoring of position deviation	-	UINT16	<i>MON_p_DiffWin</i>
3006:1A	Monitoring of velocity deviation	-	UINT32	<i>MON_v_DiffWin</i>
3006:1B	Monitoring of velocity threshold	R_PDO	UINT32	<i>MON_v_Threshold</i>
3006:1C	Monitoring of current threshold	R_PDO	UINT16	<i>MON_I_Threshold</i>
3006:1D	Monitoring of time window	-	UINT16	<i>MON_ChkTime</i>
3006:1E	Velocity limitation via input	-	UINT32	<i>IO_v_limit</i>
3006:21	Velocity scaling: Denominator	-	INT32	<i>ScaleVELdenom</i>
3006:22	Velocity scaling: Numerator	-	INT32	<i>ScaleVELnum</i>
3006:26	Timeout time for standstill window monitoring	-	UINT16	<i>MON_p_winTout</i>
3006:27	Current limitation via input	-	UINT16	<i>IO_I_limit</i>
3006:28	Velocity limit for Zero Clamp	-	UINT32	<i>MON_v_zeroclamp</i>
3006:29	Advisory limit of the load-dependent position deviation (error class 0)	-	UINT16	<i>MON_p_dif_warn</i>
3006:2B	Activation of the motion profile for velocity	-	UINT16	<i>RAMP_v_enable</i>
3006:2C	Activation of the motion profile for torque	-	UINT16	<i>RAMP_tq_enable</i>
3006:2D	Torque window, permissible deviation	-	UINT16	<i>MON_tq_win</i>
3006:2E	Torque window, time	-	UINT16	<i>MON_tq_winTime</i>
3006:30	Ramp scaling: Denominator	-	INT32	<i>ScaleRAMPdenom</i>
3006:31	Ramp scaling: Numerator	-	INT32	<i>ScaleRAMPnum</i>
3006:38	Activation of Modulo function	-	UINT16	<i>MOD_Enable</i>
3006:39	Minimum position of modulo range	-	INT32	<i>MOD_Min</i>
3006:3A	Maximum position of modulo range	-	INT32	<i>MOD_Max</i>
3006:3B	Direction of absolute movement with Modulo	-	UINT16	<i>MOD_AbsDirection</i>
3006:3C	Multiple ranges for absolute movement with Modulo	-	UINT16	<i>MOD_AbsMultiRng</i>
3006:3E	Maximum load-dependent position deviation	-	INT32	<i>MON_p_dif_load_usr</i>
3006:3F	Monitoring of position deviation	-	INT32	<i>MON_p_DiffWin_usr</i>
3006:40	Standstill window, permissible control deviation	-	INT32	<i>MON_p_win_usr</i>
3006:41	Processing mode of backlash compensation	-	UINT16	<i>BLSH_Mode</i>
3006:42	Position value for backlash compensation	-	INT32	<i>BLSH_Position</i>
3006:44	Processing time for backlash compensation	-	UINT16	<i>BLSH_Time</i>
3006:47	Behavior when position limit is reached	-	UINT16	<i>MON_SWLimMode</i>
3006:48	Maximum load-dependent velocity deviation for operating states 5, 7 and 8	-	UINT32	<i>MON_VelDiffOpSt578</i>
3006:49	Response to write command (operating state is not Operation Enabled)	-	UINT16	<i>ResWriComNotOpEn</i>
3006:4B	Maximum load-dependent velocity deviation	-	UINT32	<i>MON_VelDiff</i>
3006:4C	Time window for maximum load-dependent velocity deviation	-	UINT16	<i>MON_VelDiff_Time</i>
3007:1	Function Input DI0	-	UINT16	<i>IOfunct_DI0</i>
3007:2	Function Input DI1	-	UINT16	<i>IOfunct_DI1</i>
3007:3	Function Input DI2	-	UINT16	<i>IOfunct_DI2</i>
3007:4	Function Input DI3	-	UINT16	<i>IOfunct_DI3</i>
3007:9	Function Output DQ0	-	UINT16	<i>IOfunct_DQ0</i>

Address (hex)	Object	PDO	Data type	Parameter name
3007:A	Function Output DQ1	-	UINT16	<i>IOfunct_DQ1</i>
3008:1	Physical status of the digital inputs and outputs	T_PDO	UINT16	<i>_IO_act</i>
3008:A	Manual operation of the holding brake	-	UINT16	<i>BRK_release</i>
3008:F	Status of digital inputs	T_PDO	UINT16	<i>_IO_DI_act</i>
3008:10	Status of digital outputs	T_PDO	UINT16	<i>_IO_DQ_act</i>
3008:11	Setting the digital outputs directly	R_PDO	UINT16	<i>IO_DQ_set</i>
3008:20	Debounce time of DI0	-	UINT16	<i>DI_0_Debounce</i>
3008:21	Debounce time of DI1	-	UINT16	<i>DI_1_Debounce</i>
3008:22	Debounce time of DI2	-	UINT16	<i>DI_2_Debounce</i>
3008:23	Debounce time of DI3	-	UINT16	<i>DI_3_Debounce</i>
3008:26	Status of the inputs for the safety-related function STO	T_PDO	UINT16	<i>_IO_STO_act</i>
3008:27	Signal evaluation for signal input function Velocity Limitation	-	UINT16	<i>IOsigVelLim</i>
3008:28	Signal evaluation for signal input function Current Limitation	-	UINT16	<i>IOsigCurrLim</i>
300A:1	Status of the capture inputs	T_PDO	UINT16	<i>_CapStatus</i>
300A:2	Capture input 1 configuration	-	UINT16	<i>Cap1Config</i>
300A:3	Capture input 2 configuration	-	UINT16	<i>Cap2Config</i>
300A:4	Capture input 1 start/stop	-	UINT16	<i>Cap1Activate</i>
300A:5	Capture input 2 start/stop	-	UINT16	<i>Cap2Activate</i>
300A:6	Capture input 1 captured position (one-time)	T_PDO	INT32	<i>_Cap1Pos</i>
300A:7	Capture input 2 captured position (one-time)	T_PDO	INT32	<i>_Cap2Pos</i>
300A:8	Capture input 1 event counter (one-time)	T_PDO	UINT16	<i>_Cap1Count</i>
300A:9	Capture input 2 event counter (one-time)	T_PDO	UINT16	<i>_Cap2Count</i>
300A:A	Capture input 1 encoder source	-	UINT16	<i>Cap1Source</i>
300A:B	Capture input 2 encoder source	-	UINT16	<i>Cap2Source</i>
300A:17	Capture input 1 event counter (continuous)	T_PDO	UINT16	<i>_Cap1CountCons</i>
300A:18	Capture input 1 captured position (continuous)	T_PDO	INT32	<i>_Cap1PosCons</i>
300A:19	Capture input 2 event counter (continuous)	T_PDO	UINT16	<i>_Cap2CountCons</i>
300A:1A	Capture input 2 captured position (continuous)	T_PDO	INT32	<i>_Cap2PosCons</i>
300A:2B	Capture input 1 event counter at rising edges (DS402)	T_PDO	UINT16	<i>_Cap1CntRise</i>
300A:2C	Capture input 1 event counter at falling edges (DS402)	T_PDO	UINT16	<i>_Cap1CntFall</i>
300A:2D	Capture input 2 event counter at rising edges (DS402)	T_PDO	UINT16	<i>_Cap2CntRise</i>
300A:2E	Capture input 2 event counter at falling edges (DS402)	T_PDO	UINT16	<i>_Cap2CntFall</i>
300A:2F	Capture inputs 1 and 2 summary of event counters (DS402)	T_PDO	UINT16	<i>_CapEventCounters</i>
300B:1	Status of the position register channels	T_PDO	UINT16	<i>_PosRegStatus</i>
300B:2	Start/stop of position register channel 1	R_PDO	UINT16	<i>PosReg1Start</i>
300B:3	Start/stop of position register channel 2	R_PDO	UINT16	<i>PosReg2Start</i>
300B:4	Selection of comparison criterion for position register channel 1	-	UINT16	<i>PosReg1Mode</i>
300B:5	Selection of comparison criterion for position register channel 2	-	UINT16	<i>PosReg2Mode</i>
300B:6	Selection of source for position register channel 1	-	UINT16	<i>PosReg1Source</i>
300B:7	Selection of source for position register channel 2	-	UINT16	<i>PosReg2Source</i>
300B:8	Comparison value A for position register channel 1	R_PDO	INT32	<i>PosReg1ValueA</i>

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

Address (hex)	Object	PDO	Data type	Parameter name
3010:4	Minimum permissible DC bus voltage	T_PDO	UINT16	<i>_PS_U_minDC</i>
3010:6	Advisory temperature limit of power stage (error class 0)	T_PDO	INT16	<i>_PS_T_warn</i>
3010:7	Maximum temperature of power stage	T_PDO	INT16	<i>_PS_T_max</i>
3010:8	Resistance value of internal braking resistor	T_PDO	UINT16	<i>_RESint_R</i>
3010:9	Nominal power of internal braking resistor	T_PDO	UINT16	<i>_RESint_P</i>
3010:A	DC bus voltage low threshold for Quick Stop	T_PDO	UINT16	<i>_PS_U_minStopDC</i>
3011:1	Current controller d component P gain	-	UINT16	<i>_CTRL_KPid</i>
3011:2	Current controller d component integral action time	-	UINT16	<i>_CTRL_TNid</i>
3011:3	Current controller q component P gain	-	UINT16	<i>_CTRL_KPiq</i>
3011:4	Current controller q component integral action time	-	UINT16	<i>_CTRL_TNiq</i>
3011:5	PID velocity controller: Time constant of D term smoothing filter	-	UINT16	<i>CTRL_vPIDDTime</i>
3011:6	PID velocity controller: D gain	-	UINT16	<i>CTRL_vPIDDPart</i>
3011:8	Filter time constant to smooth velocity of motor	-	UINT16	<i>CTRL_TAUact</i>
3011:9	Speed of rotation up to which the friction compensation is linear	-	UINT32	<i>CTRL_SpdFric</i>
3011:A	Acceleration feed-forward control	-	UINT16	<i>CTRL_KFAcc</i>
3011:C	Current limitation	R_PDO	UINT16	<i>CTRL_I_max</i>
3011:D	Current for Quick Stop	-	UINT16	<i>LIM_I_maxQSTP</i>
3011:E	Current for Halt	-	UINT16	<i>LIM_I_maxHalt</i>
3011:F	Maximum current for field weakening (d component)	-	UINT16	<i>CTRL_I_max_fw</i>
3011:10	Velocity limitation	R_PDO	UINT32	<i>CTRL_v_max</i>
3011:14	Period of time for control loop parameter set switching	-	UINT16	<i>CTRL_ParChgTime</i>
3011:15	Global gain factor (affects control loop parameter set 1)	-	UINT16	<i>CTRL_GlobGain</i>
3011:16	Control loop parameter set copying	-	UINT16	<i>CTRL_ParSetCopy</i>
3011:17	Active control loop parameter set	T_PDO	UINT16	<i>_CTRL_ActParSet</i>
3011:18	Selection of control loop parameter set at power up	-	UINT16	<i>CTRL_PwrUpParSet</i>
3011:19	Selection of control loop parameter set	-	UINT16	<i>CTRL_SelParSet</i>
3011:1A	Condition for parameter set switching	-	UINT16	<i>CLSET_ParSwiCond</i>
3011:1B	Time window for parameter set switching	-	UINT16	<i>CLSET_winTime</i>
3011:1C	Position deviation for control loop parameter set switching	-	UINT16	<i>CLSET_p_DiffWin</i>
3011:1D	Velocity threshold for control loop parameter set switching	-	UINT32	<i>CLSET_v_Threshold</i>
3011:22	Activation of velocity observer	-	UINT16	<i>CTRL_VelObsActiv</i>
3011:23	Dynamics of velocity observer	-	UINT16	<i>CTRL_VelObsDyn</i>
3011:24	Inertia value for velocity observer	-	UINT32	<i>CTRL_VelObsInert</i>
3011:25	Position deviation for control loop parameter set switching	-	INT32	<i>CLSET_p_DiffWin_usr</i>
3011:26	Smoothing factor for current controller	-	UINT16	<i>CTRL_SmoothCurr</i>
3012:1	Velocity controller P gain	-	UINT16	<i>CTRL1_KPn</i>
3012:2	Velocity controller integral action time	-	UINT16	<i>CTRL1_TNn</i>
3012:3	Position controller P gain	-	UINT16	<i>CTRL1_KPp</i>
3012:4	Filter time constant of the reference velocity value filter	-	UINT16	<i>CTRL1_TAUref</i>
3012:5	Filter time constant of the reference current value filter	-	UINT16	<i>CTRL1_TAUiref</i>
3012:6	Velocity feed-forward control	-	UINT16	<i>CTRL1_KFPp</i>

Address (hex)	Object	PDO	Data type	Parameter name
3012:8	Notch filter 1: Damping	-	UINT16	<i>CTRL1_Nf1damp</i>
3012:9	Notch filter 1: Frequency	-	UINT16	<i>CTRL1_Nf1freq</i>
3012:A	Notch filter 1: Bandwidth	-	UINT16	<i>CTRL1_Nf1bandw</i>
3012:B	Notch filter 2: Damping	-	UINT16	<i>CTRL1_Nf2damp</i>
3012:C	Notch filter 2: Frequency	-	UINT16	<i>CTRL1_Nf2freq</i>
3012:D	Notch filter 2: Bandwidth	-	UINT16	<i>CTRL1_Nf2bandw</i>
3012:E	Overshoot suppression filter: Damping	-	UINT16	<i>CTRL1_Osupdamp</i>
3012:F	Overshoot suppression filter: Time delay	-	UINT16	<i>CTRL1_Osupdelay</i>
3012:10	Friction compensation: Gain	-	UINT16	<i>CTRL1_Kfric</i>
3013:1	Velocity controller P gain	-	UINT16	<i>CTRL2_KPn</i>
3013:2	Velocity controller integral action time	-	UINT16	<i>CTRL2_TNn</i>
3013:3	Position controller P gain	-	UINT16	<i>CTRL2_KPp</i>
3013:4	Filter time constant of the reference velocity value filter	-	UINT16	<i>CTRL2_TAUref</i>
3013:5	Filter time constant of the reference current value filter	-	UINT16	<i>CTRL2_TAUiref</i>
3013:6	Velocity feed-forward control	-	UINT16	<i>CTRL2_KFPp</i>
3013:8	Notch filter 1: Damping	-	UINT16	<i>CTRL2_Nf1damp</i>
3013:9	Notch filter 1: Frequency	-	UINT16	<i>CTRL2_Nf1freq</i>
3013:A	Notch filter 1: Bandwidth	-	UINT16	<i>CTRL2_Nf1bandw</i>
3013:B	Notch filter 2: Damping	-	UINT16	<i>CTRL2_Nf2damp</i>
3013:C	Notch filter 2: Frequency	-	UINT16	<i>CTRL2_Nf2freq</i>
3013:D	Notch filter 2: Bandwidth	-	UINT16	<i>CTRL2_Nf2bandw</i>
3013:E	Overshoot suppression filter: Damping	-	UINT16	<i>CTRL2_Osupdamp</i>
3013:F	Overshoot suppression filter: Time delay	-	UINT16	<i>CTRL2_Osupdelay</i>
3013:10	Friction compensation: Gain	-	UINT16	<i>CTRL2_Kfric</i>
3016:3	Modbus baud rate	-	UINT32	<i>MBbaud</i>
3016:4	Modbus address	-	UINT16	<i>MBaddress</i>
301B:5	Bit shift for RefA16 for Drive Profile Lexium	T_PDO	UINT16	<i>_DPL_BitShiftRefA16</i>
301B:6	Error response to detected data error (DE bit)	-	INT16	<i>ErrorResp_bit_DE</i>
301B:7	Error response to detected mode error (ME bit)	-	INT16	<i>ErrorResp_bit_ME</i>
301B:8	Activation of Drive Profile Lexium	-	UINT16	<i>DPL_Activate</i>
301B:9	Activation of operating mode Jog	R_PDO	UINT16	<i>JOGactivate</i>
301B:13	DS402 state machine: State transition from 3 to 4	-	UINT16	<i>DS402compatib</i>
301B:16	Position for Position Setting	-	INT32	<i>HMp_setP</i>
301B:19	Error code for detected synchronous errors (ME bit)	T_PDO	UINT16	<i>_ModeError</i>
301B:1B	Error code for detected synchronous errors (DE bit)	T_PDO	UINT16	<i>_DataError</i>
301B:1C	Additional error information of a detected ModeError (ME bit)	T_PDO	UINT16	<i>_ModeErrorInfo</i>
301B:1D	Additional error information of a detected DataError (DE bit)	T_PDO	UINT16	<i>_DataErrorInfo</i>
301B:1E	DS402 status word: Setting for bit 11 (internal limit)	-	UINT16	<i>DS402intLim</i>
301B:1F	Drive Profile Lexium dmControl	R_PDO	UINT16	<i>DPL_dmControl</i>
301B:21	Drive Profile Lexium RefB32	R_PDO	INT32	<i>DPL_RefB32</i>
301B:22	Drive Profile Lexium RefA16	R_PDO	INT16	<i>DPL_RefA16</i>
301B:25	Drive Profile Lexium driveStat	T_PDO	UINT16	<i>_DPL_driveStat</i>

Address (hex)	Object	PDO	Data type	Parameter name
301B:26	Drive Profile Lexium mfStat	T_PDO	UINT16	<i>_DPL_mfStat</i>
301B:27	Drive Profile Lexium motionStat	T_PDO	UINT16	<i>_DPL_motionStat</i>
301B:28	Drive Profile Lexium driveInput	T_PDO	UINT16	<i>_DPL_driveInput</i>
301B:35	Setting for bit 9 of <i>_DPL_motionStat</i> and <i>_actionStatus</i>	-	UINT16	<i>DPL_intLim</i>
301C:4	Action word	T_PDO	UINT16	<i>_actionStatus</i>
301C:6	Modbus address of parameter with invalid value	T_PDO	UINT16	<i>_InvalidParam</i>
301C:7	Status of monitoring signals	T_PDO	UINT32	<i>_SigActive</i>
301C:8	Saved status of monitoring signals	T_PDO	UINT32	<i>_SigLatched</i>
301C:9	Code of most recent error of error class 0	T_PDO	UINT16	<i>_LastWarning</i>
301C:A	Operating hours counter	T_PDO	UINT32	<i>_OpHours</i>
301C:B	Active errors of error class 0, bit-coded	T_PDO	UINT32	<i>_WarnActive</i>
301C:C	Saved errors of error class 0, bit-coded	T_PDO	UINT32	<i>_WarnLatched</i>
301C:D	Output power	T_PDO	INT32	<i>_Power_act</i>
301C:E	Mean output power	T_PDO	UINT16	<i>_Power_mean</i>
301C:F	Voltage at DC bus	T_PDO	UINT16	<i>_UDC_act</i>
301C:10	Temperature of power stage	T_PDO	INT16	<i>_PS_T_current</i>
301C:11	Temperature of motor	T_PDO	INT16	<i>_M_T_current</i>
301C:12	Temperature of device	T_PDO	INT16	<i>_DEV_T_current</i>
301C:13	Overload of braking resistor (I2t)	T_PDO	INT16	<i>_RES_overload</i>
301C:14	Load of braking resistor	T_PDO	INT16	<i>_RES_load</i>
301C:15	Maximum value of overload of braking resistor	T_PDO	INT16	<i>_RES_maxoverload</i>
301C:16	Overload of power stage (I2t)	T_PDO	INT16	<i>_PS_overload_I2t</i>
301C:17	Load of power stage	T_PDO	INT16	<i>_PS_load</i>
301C:18	Maximum value of overload of power stage	T_PDO	INT16	<i>_PS_maxoverload</i>
301C:19	Overload of motor (I2t)	T_PDO	INT16	<i>_M_overload</i>
301C:1A	Load of motor	T_PDO	INT16	<i>_M_load</i>
301C:1B	Maximum value of overload of motor	T_PDO	INT16	<i>_M_maxoverload</i>
301C:1E	Maximum possible value for operating mode Profile Torque	T_PDO	INT16	<i>_PT_max_val</i>
301C:1F	Additional info on most recent error	T_PDO	UINT16	<i>_LastError_Qual</i>
301C:22	Overload of power stage (chip temperature)	T_PDO	INT16	<i>_PS_overload_cte</i>
301C:23	Overload of power stage (power squared)	T_PDO	INT16	<i>_PS_overload_psq</i>
301C:24	Overload of power stage	T_PDO	INT16	<i>_PS_overload</i>
301C:26	Conditions for transition to operating state Ready To Switch On	T_PDO	UINT16	<i>_Cond_State4</i>
301C:27	Current limitation of the system	T_PDO	UINT16	<i>_Imax_system</i>
301C:28	Currently effective current limitation	T_PDO	UINT16	<i>_Imax_act</i>
301C:29	Currently effective velocity limitation	T_PDO	UINT32	<i>_Vmax_act</i>
301C:2B	Voltage of cosine signal of encoder	-	INT16	<i>_M_Enc_Cosine</i>
301C:2C	Voltage of sine signal of encoder	-	INT16	<i>_M_Enc_Sine</i>
301E:1	Actual motor current (q component, generating torque)	T_PDO	INT16	<i>_Iq_act_rms</i>
301E:2	Actual motor current (d component, field weakening)	T_PDO	INT16	<i>_Id_act_rms</i>
301E:3	Total motor current	T_PDO	INT16	<i>_I_act</i>

Address (hex)	Object	PDO	Data type	Parameter name
301E:4	Reference motor voltage q component	T_PDO	INT16	<i>_Uq_ref</i>
301E:5	Reference motor voltage d component	T_PDO	INT16	<i>_Ud_ref</i>
301E:6	Total motor voltage (vector sum d components and q components)	T_PDO	INT16	<i>_Udq_ref</i>
301E:7	Reference speed of rotation	T_PDO	INT16	<i>_n_ref</i>
301E:8	Actual speed of rotation	T_PDO	INT16	<i>_n_act</i>
301E:9	Reference position in internal units	T_PDO	INT32	<i>_p_ref_int</i>
301E:C	Reference position	T_PDO	INT32	<i>_p_ref</i>
301E:E	Absolute position with reference to internal resolution in internal units	T_PDO	UINT32	<i>_p_absmodulo</i>
301E:F	Absolute position with reference to the encoder range	T_PDO	UINT32	<i>_p_absENC</i>
301E:10	Reference motor current (q component, generating torque)	T_PDO	INT16	<i>_Iq_ref_rms</i>
301E:11	Reference motor current (d component, field weakening)	T_PDO	INT16	<i>_Id_ref_rms</i>
301E:13	Degree of utilization of DC bus voltage	T_PDO	INT16	<i>_VoltUtil</i>
301E:14	Position deviation including dynamic position deviation	T_PDO	INT32	<i>_p_dif_usr</i>
301E:15	Maximum value of the load-dependent position deviation	-	INT32	<i>_p_dif_load_peak_usr</i>
301E:16	Load-dependent position deviation between reference and actual positions	T_PDO	INT32	<i>_p_dif_load_usr</i>
301E:1B	Maximum value of the load-dependent position deviation	-	UINT32	<i>_p_dif_load_peak</i>
301E:1C	Load-dependent position deviation between reference and actual positions	T_PDO	INT32	<i>_p_dif_load</i>
301E:1F	Reference velocity	T_PDO	INT32	<i>_v_ref</i>
301E:26	Actual position of encoder 1 in internal units	T_PDO	INT32	<i>_p_act_ENC1_int</i>
301E:27	Actual position of encoder 1	T_PDO	INT32	<i>_p_act_ENC1</i>
301E:28	Actual speed of rotation of encoder 1	T_PDO	INT16	<i>_n_act_ENC1</i>
301E:29	Actual velocity of encoder 1	T_PDO	INT32	<i>_v_act_ENC1</i>
301E:2C	Load-dependent velocity deviation	T_PDO	INT32	<i>_v_dif_usr</i>
301F:1	Target position of profile generator	T_PDO	INT32	<i>_RAMP_p_target</i>
301F:2	Actual position of profile generator	T_PDO	INT32	<i>_RAMP_p_act</i>
301F:5	Target velocity of profile generator	T_PDO	INT32	<i>_RAMP_v_target</i>
301F:7	Velocity of reference value for velocity feed-forward control	T_PDO	INT32	<i>_pref_v</i>
301F:9	Acceleration of reference value for acceleration feed-forward control	T_PDO	INT32	<i>_pref_acc</i>
301F:A	Maximum user-defined value for positions	T_PDO	INT32	<i>_ScalePOSmax</i>
301F:B	Maximum user-defined value for velocity	T_PDO	INT32	<i>_ScaleVELmax</i>
301F:C	Maximum user-defined value for acceleration and deceleration	T_PDO	INT32	<i>_ScaleRAMPmax</i>
3022:4	Synchronization tolerance	-	UINT16	<i>SyncMechTol</i>
3022:5	Activation of synchronization mechanism	-	UINT16	<i>SyncMechStart</i>
3022:6	Status of synchronization mechanism	T_PDO	UINT16	<i>SyncMechStatus</i>
3023:7	Absolute movement beyond movement range	-	UINT16	<i>PP_ModeRangeLim</i>
3023:9	Change to operating mode Profile Position during movements	-	UINT16	<i>PP_OpmChgType</i>
3023:C	Activation of relative movement after capture	-	UINT16	<i>RMAC_Activate</i>
3023:D	Target position of relative movement after capture	-	INT32	<i>RMAC_Position</i>
3023:E	Velocity of relative movement after capture	-	UINT32	<i>RMAC_Velocity</i>
3023:F	Response if target position is overtraveled	-	UINT16	<i>RMAC_Response</i>

Address (hex)	Object	PDO	Data type	Parameter name
3023:10	Edge of capture signal for relative movement after capture	-	UINT16	<i>RMAC_Edge</i>
3023:11	Status of relative movement after capture	T_PDO	UINT16	<i>_RMAC_Status</i>
3023:12	Detailed status of relative movement after capture (RMAC)	T_PDO	UINT16	<i>_RMAC_DetailStatus</i>
3028:6	Maximum distance for search for switching point	-	INT32	<i>HMoutdis</i>
3028:7	Distance from switching point	-	INT32	<i>HMdis</i>
3028:A	Preferred homing method	-	INT16	<i>HMprefmethod</i>
3028:B	Position at reference point	R_PDO	INT32	<i>HMp_home</i>
3028:C	Distance from switching point to index pulse	T_PDO	INT32	<i>_HMdisREFtoIDX</i>
3028:D	Maximum search distance after overtravel of switch	-	INT32	<i>HMsrchdis</i>
3028:F	Distance from switching point to index pulse	T_PDO	INT32	<i>_HMdisREFtoIDX_usr</i>
3029:3	Selection of jog method	R_PDO	UINT16	<i>JOGmethod</i>
3029:4	Velocity for slow movement	R_PDO	UINT32	<i>JOGv_slow</i>
3029:5	Velocity for fast movement	R_PDO	UINT32	<i>JOGv_fast</i>
3029:7	Distance for step movement	-	INT32	<i>JOGstep</i>
3029:8	Wait time for step movement	-	UINT16	<i>JOGtime</i>
302E:3	Maximum permissible distance	-	UINT16	<i>MT_dismax</i>
302E:A	Maximum permissible distance	-	INT32	<i>MT_dismax_usr</i>
302F:1	Autotuning start	-	UINT16	<i>AT_start</i>
302F:2	Autotuning status	T_PDO	UINT16	<i>_AT_state</i>
302F:3	Movement range for Autotuning	-	UINT32	<i>AT_dis</i>
302F:4	Direction of movement for Autotuning	-	UINT16	<i>AT_dir</i>
302F:6	Velocity jump for Autotuning	-	UINT32	<i>AT_n_ref</i>
302F:7	Friction torque of the system	T_PDO	UINT16	<i>_AT_M_friction</i>
302F:8	Constant load torque	T_PDO	INT16	<i>_AT_M_load</i>
302F:9	Waiting time between Autotuning steps	-	UINT16	<i>AT_wait</i>
302F:B	Progress of Autotuning	T_PDO	UINT16	<i>_AT_progress</i>
302F:C	Moment of inertia of the system	T_PDO	UINT16	<i>_AT_J</i>
302F:E	Type of coupling of the system	-	UINT16	<i>AT_mechanical</i>
302F:12	Movement range for Autotuning	-	INT32	<i>AT_dis_usr</i>
302F:13	Velocity jump for Autotuning	-	INT32	<i>AT_v_ref</i>
303A:1	Lock HMI	-	UINT16	<i>HMIlocked</i>
303A:2	HMI display when motor moves	-	UINT16	<i>HMIDispPara</i>
303B:2	Number of power on cycles	T_PDO	UINT32	<i>_ERR_powerOn</i>
303B:4	Clear error memory	-	UINT16	<i>ERR_clear</i>
303B:5	Reset error memory read pointer	-	UINT16	<i>ERR_reset</i>
303B:6	Signal output function Selected Error (error classes 1 to 4): First error code	-	UINT16	<i>MON_IO_SelErr1</i>
303B:7	Signal output function Selected Error (error classes 1 to 4): Second error code	-	UINT16	<i>MON_IO_SelErr2</i>
303B:8	Signal output function Selected Warning (error class 0): First error code	-	UINT16	<i>MON_IO_SelWar1</i>
303B:9	Signal output function Selected Warning (error class 0): Second error code	-	UINT16	<i>MON_IO_SelWar2</i>
303C:1	Error code	-	UINT16	<i>_ERR_number</i>

Address (hex)	Object	PDO	Data type	Parameter name
303C:2	Error class	-	UINT16	<i>_ERR_class</i>
303C:3	Time of detection of error	-	UINT32	<i>_ERR_time</i>
303C:4	Additional information on detected error	-	UINT16	<i>_ERR_qual</i>
303C:5	Number of cycles of enabling the power stage at error time	-	UINT16	<i>_ERR_enable_cycl</i>
303C:6	Time between enabling of power stage and detection of the error	-	UINT16	<i>_ERR_enable_time</i>
303C:7	DC bus voltage at the time the error was detected	-	UINT16	<i>_ERR_DCbus</i>
303C:8	Motor velocity at the time the error was detected	-	INT32	<i>_ERR_motor_v</i>
303C:9	Motor current at the time the error was detected	-	UINT16	<i>_ERR_motor_I</i>
303C:A	Temperature of power stage at the time the error was detected	-	INT16	<i>_ERR_temp_ps</i>
303C:B	Temperature of device at the time the error was detected	-	INT16	<i>_ERR_temp_dev</i>
303F:5D	Value of the SinCos amplitude	-	UINT16	<i>_ENC_AmplVal</i>
303F:5E	Mean value of the SinCos amplitude	-	UINT16	<i>_ENC_AmplMean</i>
303F:5F	Minimum value of the SinCos amplitude	-	UINT16	<i>_ENC_AmplMin</i>
303F:60	Maximum value of the SinCos amplitude	-	UINT16	<i>_ENC_AmplMax</i>
303F:61	Activation of monitoring of SinCos amplitude	-	UINT16	<i>MON_ENC_Ampl</i>
303F:62	Actual value of commutation monitoring counter	-	INT16	<i>_CommutCntAct</i>
303F:63	Maximum value the commutation monitoring counter has reached	-	INT16	<i>CommutCntMax</i>
303F:68	Motor overload and overtemperature monitoring	-	UINT16	<i>MON_MotOvLoadOvTemp</i>
3040:43	Last error code of fieldbus parameter services	-	UINT16	<i>_ErrNumFbParSvc</i>
3041:6	CANopen diagnostics word	-	UINT16	<i>_CanDiag</i>
3041:A	CANopen manufacturer-specific SDO abort code	-	UINT16	<i>_ManuSdoAbort</i>
3041:B	PDO 1 event mask	-	UINT16	<i>CANpdo1Event</i>
3041:C	PDO 2 event mask	-	UINT16	<i>CANpdo2Event</i>
3041:D	PDO 3 event mask	-	UINT16	<i>CANpdo3Event</i>
3041:E	PDO 4 event mask	-	UINT16	<i>CANpdo4Event</i>
3041:11	CANopen error response to detected Heartbeat or Life Guarding error	-	UINT16	<i>ErrResp_HeartB_LifeG</i>

Assignment Object Group 6000 hex

Overview

The product provides corresponding parameters for the CANopen object group 6000 hex.

Address (hex)	Object	PDO	Data type	Parameter name
603F:0	Detected error causing a stop (error classes 1 to 4)	T_PDO	UINT16	<i>_LastError</i>
6040:0	DriveCom control word	R_PDO	UINT16	<i>DCOMcontrol</i>
6041:0	DriveCom status word	T_PDO	UINT16	<i>_DCOMstatus</i>
605B:0	Behavior for disabling the power stage during movement	-	INT16	<i>DSM_ShutDownOption</i>
605D:0	Halt option code	-	INT16	<i>LIM_HaltReaction</i>
6060:0	Operating mode	R_PDO	INT8	<i>DCOMopmode</i>
6061:0	Active operating mode	T_PDO	INT8	<i>_DCOMopmd_act</i>

Address (hex)	Object	PDO	Data type	Parameter name
6063:0	Actual position in internal units	T_PDO	INT32	<i>_p_act_int</i>
6064:0	Actual position	T_PDO	INT32	<i>_p_act</i>
6065:0	Maximum load-dependent position deviation	R_PDO	UINT32	<i>MON_p_dif_load</i>
6067:0	Standstill window, permissible control deviation	R_PDO	UINT32	<i>MON_p_win</i>
6068:0	Standstill window, time	-	UINT16	<i>MON_p_winTime</i>
606B:0	Actual velocity of profile generator	T_PDO	INT32	<i>_RAMP_v_act</i>
606C:0	Actual velocity	T_PDO	INT32	<i>_v_act</i>
606D:0	Velocity window, permissible deviation	-	UINT16	<i>MON_v_win</i>
606E:0	Velocity window, time	-	UINT16	<i>MON_v_winTime</i>
6071:0	Target torque	R_PDO	INT16	<i>PTtq_target</i>
6077:0	Actual torque	T_PDO	INT16	<i>_tq_act</i>
607A:0	Target position for operating mode Profile Position	R_PDO	INT32	<i>PPp_target</i>
607D:1	Negative position limit for software limit switch	-	INT32	<i>MON_swLimN</i>
607D:2	Positive position limit for software limit switch	-	INT32	<i>MON_swLimP</i>
607F:0	Maximum velocity of the motion profile for velocity	-	UINT32	<i>RAMP_v_max</i>
6081:0	Target velocity for operating mode Profile Position	R_PDO	UINT32	<i>PPv_target</i>
6083:0	Acceleration of the motion profile for velocity	R_PDO	UINT32	<i>RAMP_v_acc</i>
6084:0	Deceleration of the motion profile for velocity	R_PDO	UINT32	<i>RAMP_v_dec</i>
6087:0	Slope setting of the motion profile for torque	R_PDO	UINT32	<i>RAMP_tq_slope</i>
6098:0	Homing method	R_PDO	INT8	<i>HMmethod</i>
6099:1	Target velocity for searching the switch	-	UINT32	<i>HMv</i>
6099:2	Target velocity for moving away from switch	-	UINT32	<i>HMv_out</i>
60B8:0	Touch Probe function (DS402)	R_PDO	UINT16	<i>TouchProbeFct</i>
60B9:0	Touch Probe status (DS402)	T_PDO	UINT16	<i>_TouchProbeStat</i>
60BA:0	Capture input 1 captured position at rising edge (DS402)	T_PDO	INT32	<i>_Cap1PosRisEdge</i>
60BB:0	Capture input 1 captured position at falling edge (DS402)	T_PDO	INT32	<i>_Cap1PosFallEdge</i>
60BC:0	Capture input 2 captured position at rising edge (DS402)	T_PDO	INT32	<i>_Cap2PosRisEdge</i>
60BD:0	Capture input 2 captured position at falling edge (DS402)	T_PDO	INT32	<i>_Cap2PosFallEdge</i>
60C1:1	Position reference value for operating mode Interpolated Position	R_PDO	INT32	<i>IPp_target</i>
60C2:1	Interpolation time period value	-	UINT8	<i>IP_IntTimPerVal</i>
60C2:2	Interpolation time index	-	INT8	<i>IP_IntTimInd</i>
60F2:0	Options for operating mode Profile Position	-	UINT16	<i>PPoption</i>
60F4:0	Position deviation including dynamic position deviation	T_PDO	INT32	<i>_p_dif</i>
60FF:0	Target velocity	R_PDO	INT32	<i>PVv_target</i>
6502:0	Supported operating modes as per DSP402	T_PDO	UINT32	<i>_SuppDriveModes</i>

Details of Object Group 1000 hex

1000 hex Device Type

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

Object description

Index	1000 hex
Object name	Device type
Object code	VAR
Data type	Unsigned32

Value description

Subindex	00 hex, device type
Meaning	Device type and device profile
Access	RO
PDO mapping	–
Value range	–
Default value	0042 0192 hex
Can be saved	–

Bit assignment, subindex 00h

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

1001 hex 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 hex)* and - for reasons of compatibility with devices with other fieldbus profiles - with the object *Error code (603F hex)*.

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

Object description

Index	1001 hex
Object name	Error register
Object code	VAR
Data type	Unsigned8

Value description

Subindex	00 hex, 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

bit	Access	Value	Meaning
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 hex Predefined Error Field

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

- The subindex 00 hex entry contains the number of saved error messages.
- The most recent error message is stored at subindex 01 hex, older messages are moved to higher subindex entries.
- Writing '0' to subindex 00 hex resets the error list.

Object description

Index	1003 hex
Object name	Predefined error field
Object code	ARRAY
Data type	Unsigned32

Value description

Subindex	00 hex, number of errors
Meaning	Number of error entries
Access	RW
PDO mapping	–
Value range	0 ... 1
Default value	1
Can be saved	–

Subindex	01 hex, error field
Meaning	Error number
Access	RO
PDO mapping	–
Value range	–
Default value	0
Can be saved	–

Bit assignment, subindex 00 hex ... 05 hex

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

Bits 16 ... 31: Error code 1000 hex: Vendor-specific error number.

1005 hex 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 hex
Object name	COB ID SYNC
Object code	VAR
Data type	Unsigned32

Value description

Subindex	00 hex, COB ID SYNC
Meaning	Identifier of the synchronization object
Access	RW
PDO mapping	–
Value range	0 ... 4294967295
Default value	8000 0080 hex
Can be saved	–

Bit assignment, subindex 00 hex

bit	Access	Value	Meaning
31	RO	0 _b	1: Device can receive SYNC messages (SYNC consumer)
30	RO	1 _b	1: Device can send SYNC messages (SYNC producer)
29	RO	0 _b	0: 11 bit identifier (CAN 3.0A) 1: 29 bit identifier (CAN 3.0B)
28-11	RO	0000 hex	Only relevant if bit 29=1 is not used by the device.
10-7	RW	0001 _b	Function code, bits 10 ... 7 of COB ID
6-0	RO	7F hex	Node address, bit 6 ... 0 of COB ID

1008 hex Manufacturer Device Name

The object specifies the device name of the manufacturer.

Object description

Index	1008 hex
Object name	Manufacturer device name
Object code	VAR
Data type	Visible String8

Value description

Subindex	00 hex, manufacturer device name
Meaning	Manufacturer's designation
Access	RO
PDO mapping	–
Value range	–
Default value	–
Can be saved	–

1009 hex Manufacturer Hardware Version

The object specifies the version of the device hardware.

Object description

Index	1009 hex
Object name	Manufacturer hardware version
Object code	VAR
Data type	Visible String8

Value description

Subindex	00 hex, manufacturer hardware version
Meaning	Hardware version
Access	RO
PDO mapping	–
Value range	–
Default value	–
Can be saved	–

100A hex Manufacturer Software Version

The object specifies the version of the device software.

Object description

Index	100A hex
Object name	Manufacturer software version
Object code	VAR
Data type	Visible String8

Value description

Subindex	00 hex, manufacturer software version
Meaning	Software version
Access	RO
PDO mapping	–
Value range	–
Default value	–
Can be saved	–

100C hex Guard Time

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

The time span for connection monitoring of an NMT master results from the time span "Guard Time" multiplied by the factor "Life Time", object *Life time factor* (100D hex).

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

Object description

Index	100C hex
Object name	Guard Time
Object code	VAR
Data type	Unsigned16

Value description

Subindex	00 hex, Guard Time
Meaning	Time span for Node Guarding [ms]
Access	RW
PDO mapping	–
Value range	0 ... 65535
Default value	0
Can be saved	–

100D hex 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 hex)*.

Object description

Index	100D hex
Object name	Life Time Factor
Object code	VAR
Data type	Unsigned8

Value description

Subindex	00 hex, 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 hex COB ID Emergency Object Message

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

Object description

Index	1014 hex
Object name	COB ID EMCY
Object code	VAR
Data type	Unsigned32

Value description

Subindex	00 hex, COB ID EMCY
Meaning	Identifier of the emergency object
Access	RW
PDO mapping	–
Value range	0 ... 4294967295
Default value	4000 0080 hex + node ID
Can be saved	–

Bit assignment, subindex 00h

bit	Access	Value	Meaning
31, 30	RO	0 _b	Reserved
29	RO	0 _b	0: 11 bit identifier (CAN 3.0A) 1: 29 bit identifier (CAN 3.0B)
28-11	RO	0000 hex	Only relevant if bit 29=1 is not used by the device.
10-7	RW	0001 _b	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 hex 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 hex
Object name	Inhibit time EMCY
Object code	VAR
Data type	Unsigned16

Value description

Subindex	00 hex, 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 hex 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 hex
Object name	Consumer Heartbeat Time
Object code	ARRAY
Data type	Unsigned32

Value description

Subindex	00 hex, number of elements
Meaning	Number of values for the object
Access	RO
PDO mapping	–
Value range	–
Default value	3
Can be saved	–

Subindex	01 hex, 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 hex ... 03 hex

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

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

1017 hex Producer Heartbeat Time

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

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

Object description

Index	1017 hex
Object name	Producer Heartbeat Time

Object code	VAR
Data type	Unsigned16

Value description

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

1018 hex Identity Object

The object provides information on the product.

- Subindex 01 hex (vendor ID) contains the manufacturer ID
- Subindex 02 hex (product ID) contains the manufacturer-specific product code
- Subindex 03 hex (revision number) identifies special CANopen properties for the device

Object description

Index	1018 hex
Object name	Identity Object
Object code	RECORD
Data type	Identity

Value description

Subindex	00 hex, number of elements
Meaning	Number of values for the object
Access	RO
PDO mapping	–
Value range	–
Default value	3
Can be saved	–

Subindex	01 hex, vendor ID
Meaning	Vendor ID
Access	RO
PDO mapping	–
Value range	–
Default value	0800 005A hex
Can be saved	–

Subindex	02 hex, product code
Meaning	Product code
Access	RO
PDO mapping	–

Value range	–
Default value	–
Can be saved	–

Subindex	03 hex, revision number
Meaning	Revision number
Access	RO
PDO mapping	–
Value range	–
Default value	–
Can be saved	–

1029 hex Error Behavior

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

Object description

Index	1029 hex
Object name	Error behavior
Object code	ARRAY
Data type	Unsigned8

Value description

Subindex	00 hex, number of elements
Meaning	Number of values for the object
Access	RO
PDO mapping	–
Value range	–
Default value	1
Can be saved	–

Subindex	01 hex, Communication Error
Meaning	Communication errors
Access	RW
PDO mapping	–
Value range	0 ... 2
Default value	0
Can be saved	–

Settings, subindex 01 hex

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

1200 hex 1st Server SDO Parameter

The object contains the settings for the first server SDO.

Object description

Index	1200 hex
Object name	1st server SDO parameter
Object code	RECORD
Data type	SDO server parameter

Value description

Subindex	00 hex, number of elements
Meaning	Number of values for the object
Access	RO
PDO mapping	–
Value range	–
Default value	2
Can be saved	–

Subindex	01 hex, 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 hex, 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 hex 2nd Server SDO Parameter

The object contains the settings for the second server SDO.

Object description

Index	1201 hex
Object name	2nd server SDO parameter
Object code	RECORD
Data type	SDO server parameter

Value description

Subindex	00 hex, number of elements
Meaning	Number of values for the object

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

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

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

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

1400 hex 1st Receive PDO Parameter

The object contains the settings for the first receive PDO R_PDO1.

Object description

Index	1400 hex
Object name	1st receive PDO parameter
Object code	RECORD
Data type	PDO Communication Parameter

Value description

Subindex	00 hex, number of entries
Meaning	Number of values for the object
Access	RO
PDO mapping	–
Value range	–

Default value	2
Can be saved	–

Subindex	01 hex, COB ID used by PDO
Meaning	Identifier of the R_PDO1
Access	RW
PDO mapping	–
Value range	0 ... 4294967295
Default value	0200 hex + node ID
Can be saved	–

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

Bit assignment, subindex 01 hex

bit	Access	Value	Meaning
31	RW	0 _b	0: PDO is active 1: PDO is inactive
30	RO	0 _b	0: RTR (see below) is possible 1: RTR not permitted
29	RO	0 _b	0: 11-bit identifier (CAN 3.0A) 1: 29 bit identifier (CAN 3.0B)
28-11	RO	0000 hex	Only relevant if bit 29=1 is not used by the device.
10-7	RW	0100 _b	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 hex

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

The type of control for evaluating R_PDO data is specified via subindex 02 hex. 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 hex)*. The following default assignment is used for R_PDO1:

- Bytes 0 ... 1: Control word *controlword (6040 hex)*.

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

1401 hex 2nd Receive PDO Parameter

The object contains settings for the second receive PDO R_PDO2.

Object description

Index	1401 hex
Object name	2nd receive PDO parameter
Object code	RECORD
Data type	PDO Communication Parameter

Value description

Subindex	00 hex, largest subindex supported
Meaning	Largest subindex supported
Access	RO
PDO mapping	–
Value range	–
Default value	2
Can be saved	–

Subindex	01 hex, COB ID R_PDO2
Meaning	Identifier of the R_PDO2
Access	RW
PDO mapping	–
Value range	0 ... 4294967295
Default value	8000 0300 hex + node ID
Can be saved	–

Subindex	02 hex, 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 hex).

Settings:

R_PDO2 is processed synchronously, acyclically and in an event-driven way and must be activated with bit 31=1 in subindex 01 hex before it can be used.

The byte assignment of R_PDO2 is specified via PDO mapping with the object 2nd Receive PDO mapping (1601 hex). The following default assignment is set for the operating mode "Profile Position":

- Bytes 0 ... 1: Control word *controlword* (6040 hex)
- Bytes 2 ... 5: Target position of the motion command *target position* (607A hex)

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 hex 3rd Receive PDO Parameter

The object contains settings for the third receive PDO R_PDO3.

Object description

Index	1402 hex
Object name	3rd receive PDO parameter
Object code	RECORD
Data type	PDO Communication Parameter

Value description

Subindex	00 hex, largest subindex supported
Meaning	Largest subindex supported
Access	RO
PDO mapping	–
Value range	–
Default value	2
Can be saved	–

Subindex	01 hex, COB ID used by PDO
Meaning	Identifier of the R_PDO3
Access	RW

PDO mapping	–
Value range	0 ... 4294967295
Default value	8000 0400 hex + node ID
Can be saved	–

Subindex	02 hex, 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 hex)*.

Settings:

R_PDO3 is processed synchronously, acyclically and in an event-driven way and must be activated with bit 31=1 in subindex 01 hex 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 hex)*. The following default assignment is set for the operating mode "Profile Velocity":

- Bytes 0 ... 1: Control word *controlword (6040 hex)*
- Bytes 2 ... 5: Reference velocity of motion command *Target velocity (60FF hex)*

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 hex 4th Receive PDO Parameter

The object stores settings for the fourth receive PDO R_PDO4.

Object description

Index	1403 hex
Object name	4th receive PDO parameter
Object code	RECORD
Data type	PDO Communication Parameter

Value description

Subindex	00 hex, largest subindex supported
Meaning	Largest subindex supported
Access	RO
PDO mapping	–
Value range	–

Default value	2
Can be saved	–

Subindex	01 hex, COB ID used by PDO
Meaning	Identifier of the R_PDO4
Access	RW
PDO mapping	–
Value range	0 ... 4294967295
Default value	8000 0500 hex + node ID
Can be saved	–

Subindex	02 hex, 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 hex)*.

Settings:

R_PDO4 is processed asynchronously and in an event-driven way and must be activated with bit 31=1 in subindex 01 hex before it can be used.

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

1600 hex 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 hex, the number of mapped objects is read.

Object description

Index	1600 hex
Object name	1st receive PDO mapping
Object code	RECORD
Data type	PDO mapping

Value description

Subindex	00 hex, 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 hex, CMD: Control word
Meaning	First object for mapping

Access	RW
PDO mapping	–
Value range	0 ... 4294967295
Default value	6040 0010 hex
Can be saved	–

Subindex	02 hex
Meaning	Second object for mapping
Access	RW
PDO mapping	–
Value range	0 ... 4294967295
Default value	–
Can be saved	–

Subindex	03 hex
Meaning	Third object for mapping
Access	RW
PDO mapping	–
Value range	0 ... 4294967295
Default value	–
Can be saved	–

Subindex	04 hex
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 hex 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 hex: *controlword (6040 hex)*

1601 hex 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 hex, the number of mapped objects is read.

Object description

Index	1601 hex
Object name	2nd receive PDO mapping
Object code	RECORD
Data type	PDO mapping

Value description

Subindex	00 hex, 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 hex, 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 hex
Can be saved	–

Subindex	02 hex, 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 hex
Can be saved	–

Subindex	03 hex
Meaning	Third object for mapping
Access	RW
PDO mapping	–
Value range	0 ... 4294967295
Default value	–
Can be saved	–

Subindex	04 hex
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 hex)*.

Settings:

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

- Subindex 01 hex: *controlword (6040 hex)*
- Subindex 02 hex: *target position (607A hex)*

1602 hex 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 hex, the number of mapped objects is read.

Object description

Index	1602 hex
Object name	3rd receive PDO mapping
Object code	RECORD
Data type	PDO mapping

Value description

Subindex	00 hex, 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 hex, 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 hex
Can be saved	–

Subindex	02 hex, 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 hex
Can be saved	–

Subindex	03 hex
Meaning	Third object for mapping
Access	RW

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

Subindex	04 hex
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 hex)*.

Settings:

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

- Subindex 01 hex: *controlword (6040 hex)*
- Subindex 02 hex: *target velocity (60FF hex)*

1603 hex 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 hex, the number of mapped objects is read.

Object description

Index	1603 hex
Object name	4th receive PDO mapping
Object code	RECORD
Data type	PDO mapping

Value description

Subindex	00 hex, 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 hex
Meaning	First object for mapping
Access	RW
PDO mapping	–
Value range	0 ... 4294967295
Default value	–
Can be saved	–

Subindex	02 hex
Meaning	Second object for mapping
Access	RW
PDO mapping	–
Value range	0 ... 4294967295
Default value	–
Can be saved	–

Subindex	03 hex
Meaning	Third object for mapping
Access	RW
PDO mapping	–
Value range	0 ... 4294967295
Default value	–
Can be saved	–

Subindex	04 hex
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 hex)*.

Settings:

The PDO assignment for R_PDO4 can be modified.

1800 hex 1st Transmit PDO Parameter

The object contains settings for the first transmit PDO T_PDO1.

Object description

Index	1800 hex
Object name	1st transmit PDO parameter
Object code	RECORD
Data type	PDO Communication Parameter

Value description

Subindex	00 hex, number of entries
Meaning	Number of values for the object
Access	RO
PDO mapping	–
Value range	–
Default value	5
Can be saved	–

Subindex	01 hex, COB ID used by PDO
Meaning	Identifier of the T_PDO1
Access	RW
PDO mapping	–
Value range	0 ... 4294967295
Default value	4000 0180 hex + node ID
Can be saved	–

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

Subindex	03 hex, 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 hex, reserved
Meaning	Reserved
Access	–
PDO mapping	–
Value range	0 ... 255
Default value	–
Can be saved	–

Subindex	05 hex, 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 hex)*.

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 hex)*. The following default assignment is used:

- Bytes 0 ... 1: Status word *statusword (6041 hex)*.

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

1801 hex 2nd Transmit PDO Parameter

The object contains settings for the second transmit PDO T_PDO2.

Object description

Index	1801 hex
Object name	2nd transmit PDO parameter
Object code	RECORD
Data type	PDO Communication Parameter

Value description

Subindex	00 hex, largest subindex supported
Meaning	Largest subindex supported
Access	RO
PDO mapping	–
Value range	–
Default value	5
Can be saved	–

Subindex	01 hex, COB ID used by PDO
Meaning	Identifier of the T_PDO2
Access	RW
PDO mapping	–
Value range	0 ... 4294967295
Default value	C000 0280 hex + node ID
Can be saved	–

Subindex	02 hex, transmission type
Meaning	Transmission type
Access	RW
PDO mapping	–
Value range	0 ... 255
Default value	255
Can be saved	–

Subindex	03 hex, 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 hex, reserved
Meaning	Reserved
Access	–

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

Subindex	05 hex, 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 hex)*.

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 hex)*. The following default assignment is set for the operating mode "Profile Position":

- Bytes 0 ... 1: Status word *statusword (6041 hex)*
- Bytes 2 ... 5: Actual position *position actual value (6064 hex)*.

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

1802 hex 3rd Transmit PDO Parameter

The object contains settings for the third transmit PDO T_PDO3.

Object description

Index	1802 hex
Object name	3rd transmit PDO parameter
Object code	RECORD
Data type	PDO Communication Parameter

Value description

Subindex	00 hex, largest subindex supported
Meaning	Largest subindex supported
Access	RO
PDO mapping	–
Value range	–
Default value	5
Can be saved	–

Subindex	01 hex, COB ID used by PDO
Meaning	Identifier of the T_PDO3
Access	RW
PDO mapping	–
Value range	0 ... 4294967295

Default value	C000 0380 hex + node ID
Can be saved	–

Subindex	02 hex, transmission type
Meaning	Transmission type
Access	RW
PDO mapping	–
Value range	0 ... 255
Default value	255
Can be saved	–

Subindex	03 hex, 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 hex, reserved
Meaning	Reserved
Access	–
PDO mapping	–
Value range	0 ... 255
Default value	–
Can be saved	–

Subindex	05 hex, 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 hex)*.

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 hex)*. The following default assignment is set for the operating mode "Profile Position":

- Bytes 0 ... 1: Status word *statusword (6041 hex)*
- Bytes 2 ... 5: Actual velocity *velocity actual value (606C hex)*.

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

1803 hex 4th Transmit PDO Parameter

The object contains settings for the fourth transmit PDO T_PDO4.

Object description

Index	1803 hex
Object name	4th transmit PDO parameter
Object code	RECORD
Data type	PDO Communication Parameter

Value description

Subindex	00 hex, largest subindex supported
Meaning	Largest subindex supported
Access	RO
PDO mapping	–
Value range	–
Default value	5
Can be saved	–

Subindex	01 hex, COB ID used by PDO
Meaning	Identifier of the T_PDO4
Access	RW
PDO mapping	–
Value range	0 ... 4294967295
Default value	C000 0480 hex + node ID
Can be saved	–

Subindex	02 hex, transmission type
Meaning	Transmission type
Access	RO
PDO mapping	–
Value range	0 ... 255
Default value	254
Can be saved	–

Subindex	03 hex, 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 hex, reserved
Meaning	Reserved
Access	–
PDO mapping	–
Value range	0 ... 255
Default value	–
Can be saved	–

Subindex	05 hex, 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 hex)*.

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 hex 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 hex, the number of mapped objects is read.

Object description

Index	1A00 hex
Object name	1st transmit PDO mapping
Object code	RECORD
Data type	PDO mapping

Value description

Subindex	00 hex, 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 hex, ETA: status word
Meaning	First object for mapping
Access	RW
PDO mapping	–
Value range	0 ... 4294967295
Default value	6041 0010 hex
Can be saved	–

Subindex	02 hex
Meaning	Second object for mapping
Access	RW
PDO mapping	–
Value range	0 ... 4294967295

Default value	–
Can be saved	–

Subindex	03 hex
Meaning	Third object for mapping
Access	RW
PDO mapping	–
Value range	0 ... 4294967295
Default value	–
Can be saved	–

Subindex	04 hex
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 hex)*.

Settings:

The following default assignment is used:

- Subindex 01 hex: *statusword (6041 hex)*

1A01 hex 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 hex, the number of mapped objects is read.

Object description

Index	1A01 hex
Object name	2nd transmit PDO mapping
Object code	RECORD
Data type	PDO mapping

Value description

Subindex	00 hex, 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 hex, 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 hex
Can be saved	–

Subindex	02 hex, 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 hex
Can be saved	–

Subindex	03 hex
Meaning	Third object for mapping
Access	RW
PDO mapping	–
Value range	0 ... 4294967295
Default value	–
Can be saved	–

Subindex	04 hex
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 hex)*.

Settings:

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

- Subindex 01 hex: *statusword (6041 hex)*
- Subindex 02 hex: *position actual value (6064 hex)*

1A02 hex 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 hex, the number of mapped objects is read.

Object description

Index	1A02 hex
Object name	3rd transmit PDO mapping
Object code	RECORD
Data type	PDO mapping

Value description

Subindex	00 hex, 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 hex, 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 hex
Can be saved	–

Subindex	02 hex, 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 hex
Can be saved	–

Subindex	03 hex
Meaning	Third object for mapping
Access	RW
PDO mapping	–
Value range	0 ... 4294967295
Default value	–
Can be saved	–

Subindex	04 hex
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 hex)*.

Settings:

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

- Subindex 01 hex: *statusword (6041 hex)*

- Subindex 02 hex: *velocity actual value (606C hex)*

1A03 hex 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 hex, the number of mapped objects is read.

Object description

Index	1A03 hex
Object name	4th transmit PDO mapping
Object code	RECORD
Data type	PDO mapping

Value description

Subindex	00 hex, 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 hex
Meaning	First object for mapping
Access	RW
PDO mapping	–
Value range	0 ... 4294967295
Default value	–
Can be saved	–

Subindex	02 hex
Meaning	Second object for mapping
Access	RW
PDO mapping	–
Value range	0 ... 4294967295
Default value	–
Can be saved	–

Subindex	03 hex
Meaning	Third object for mapping
Access	RW
PDO mapping	–
Value range	0 ... 4294967295
Default value	–
Can be saved	–

Subindex	04 hex
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 hex)*.

Settings:

The PDO assignment for T_PDO4 can be modified.

Glossary

C

CAN:

(**C**ontroller **A**rea **N**etwork), standardized open fieldbus as per ISO 11898, allows drives and other devices from different manufacturers to communicate.

CANopen:

Device- and manufacturer-independent description language for communication via the CAN bus

CiA:

CAN in **A**utomation, CAN interest group, standardization group for CAN and CANopen.

COB ID:

Communication **O**bject **I**Dentifier; uniquely identifies each communication object in a CAN network

COB:

Communication **O**bject, transport unit in a CAN network.

D

DOM:

Date of **m**anufacturing: 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

DriveCom:

Specification of the DSP402 state machine was created in accordance with the DriveCom specification.

DS301:

Standardizes the CANopen communication profile

DSP402:

Standardizes the CANopen device profile for drives

E

EDS:

(**E**lectronic **D**ata **S**heet); contains the specific properties of a product.

Electronic gear:

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.

EMC:

Electromagnetic compatibility

EMCY object:

Emergency Object

Encoder :

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.

Error class :

Classification of errors into groups. The different error classes allow for specific responses to errors, for example by severity.

Error:

Discrepancy between a detected (computed, measured or signaled) value or condition and the specified or theoretically correct value or condition.

F**Factory setting:**

Factory settings when the product is shipped

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.

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).

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

Parameter :

Device data and values that can be read and set (to a certain extent) by the user.

PDO:

Process Data Object

Persistent:

Indicates whether the value of the parameter remains in the memory after power to the drive has been removed.

Power stage:

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

Quick Stop:

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

R

R_PDO:

Receive PDO

S

SDO:

Service Data Object

SYNC object:

Synchronization object

T

T_PDO:

Transmit PDO

Index

B

boot-up message 33

C

CAN message 15
 CANopen message 16
 client-server relationship 18
 COB ID 16

D

data frame 17

E

EMCY message 33
 EMCY object 15
 emergency object service 33
 error memory 34
 error register 61
 error register and error code 34

F

function code 17

H

heartbeat 34

I

intended use 6

L

life guarding 34

M

master-slave relationship 18
 message objects 60

N

NMT 15
 node address 17
 node guarding 34

O

operating states 42

P

parameter *_DCOMopmd_act* 46
 parameter *_DCOMstatus* 42
 parameter *_ManuSdoAbort* 63

parameter *CANaddress* 41
 parameter *CANbaud* 41
 parameter *CANpdo1Event* 29
 parameter *CANpdo2Event* 29
 parameter *CANpdo3Event* 29
 parameter *CANpdo4Event* 29
 parameter *DCOMcontrol* 44
 parameter *DCOMopmode* 45
 parameter *HMmethod* 52
 parameter *JOGactivate* 46
 parameter *PPp_target* 49
 parameter *PTq_target* 47
 parameter *PVv_target* 48–49
 PDO 15
 PDO mapping 30
 producer-consumer relationship 19

Q

qualification of personnel 5

S

SDO 15
 SYNC object 15
 synchronization 31

Schneider Electric
35 rue Joseph Monier
92500 Rueil Malmaison
France

+ 33 (0) 1 41 29 70 00

www.se.com

As standards, specifications, and design change from time to time,
please ask for confirmation of the information given in this publication.

© 2021 – Schneider Electric. All rights reserved.

0198441113779.06