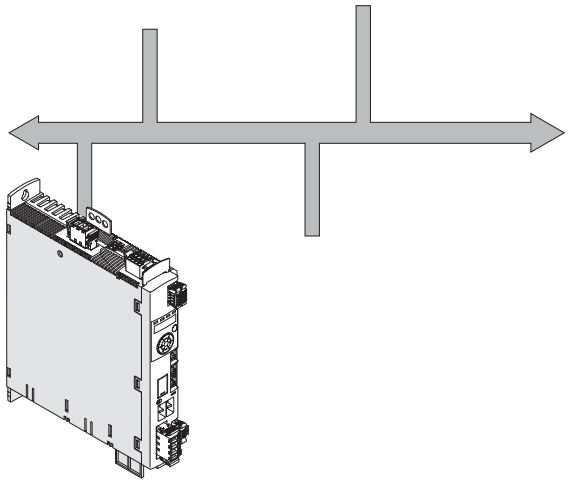
LXM32A CANopen

Fieldbus interface

Fieldbus manual

V1.05, 10.2012



0198441113779, V1.05, 10.2012

www.schneider-electric.com

Important information

This manual is part of the product.

Carefully read this manual and observe all instructions.

Keep this manual for future reference.

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

Carefully read and observe all safety instructions as well as chapter "2 Before you begin - safety information".

Some products are not available in all countries.
Please consult the latest catalog for information on the availability of products.

Subject to technical modifications without notice.

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

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

0198441113779, V1.05, 10.2012

Table of contents

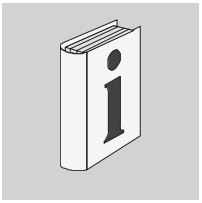
0198441113779, V1.05, 10.2012

0198441113779, V1.05, 10.2012

0198441113779, V1.05, 10.2012

0198441113779, V1.05, 10.2012

About this manual



The information provided in this manual supplements the product manual.

Source manuals The latest versions of the manuals can be downloaded from the Internet at:

http://www.schneider-electric.com

Source CAD data For easier engineering, CAD data (drawings or EPLAN macros) are available for download from the Internet at:

http://www.schneider-electric.com

Corrections and suggestions We always try to further optimize our manuals. We welcome your suggestions and corrections.

Please get in touch with us by e-mail:
techcomm@schneider-electric.com.

Work steps If work steps must be performed consecutively, this sequence of steps is represented as follows:

- ■ Special prerequisites for the following work steps
- ▸ Step 1
- ◃ Specific response to this work step
- ▸ Step 2

If a response to a work step is indicated, this allows you to verify that the work step has been performed correctly.

Unless otherwise stated, the individual steps must be performed in the specified sequence.

Making work easier Information on making work easier is highlighted by this symbol:

Sections highlighted this way provide supplementary information on making work easier.

Parameters In text sections, parameters are shown with the parameter name, for example `_IO_act`. A list of the parameters can be found in the product manual in the chapter Parameters.

SI units SI units are the original values. Converted units are shown in brackets behind the original value; they may be rounded.

Example:
Minimum conductor cross section: 1.5 mm^2 (AWG 14)

Inverted signals Inverted signals are represented by an overline, for example $\overline{\texttt{STO\_A}}$ or $\overline{\texttt{STO\_B}}$.

Glossary Explanations of special technical terms and abbreviations.

Index List of keywords with references to the corresponding page numbers.

0198441113779, V1.05, 10.2012

Further reading

Recommended literature for further reading

CAN users and manufacturers organization

CiA - CAN in Automation
Am Weichselgarten 26
D-91058 Erlangen
http://www.can-cia.org/

CANopen standards

- CiA Standard 301 (DS301)
  CANopen application layer and communication profile
- CiA Standard 402 (DSP402)
  Device profile for drives and motion control
- ISO 11898: Controller Area Network (CAN) for high speed communication
- EN 50325-4: Industrial communications subsystem based on ISO 11898 for controller device interfaces (CANopen)

Literature

Controller Area Network
Konrad Etschberger, Carl Hanser Verlag
ISBN 3-446-19431-2

0198441113779, V1.05, 10.2012

1 Introduction

1

1.1 CAN bus

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

Features of the CAN bus

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

- Multimaster capability

  Each device in the fieldbus can transmit and receive data independently without depending on an "ordering" master functionality.
- Message-oriented communication

  Devices can be integrated into a running network without reconfiguration of the entire system. The address of a new device does not need to be specified on the network.
- Prioritization of messages

  Messages with higher priority are sent first for time-critical applications.
- Residual error probability

  Various security features in the network reduce the probability of undetected incorrect data transmission to less than 10^{-11}.

Transmission technology

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

Network devices

Examples of CAN bus devices are

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

0198441113779, V1.05, 10.2012

1.2 CANopen technology

1.2.1 CANopen description language

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

1.2.2 Communication layers

CANopen uses the CAN bus technology for data communication.

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

- Physical Layer
- Data Link Layer
- Application Layer

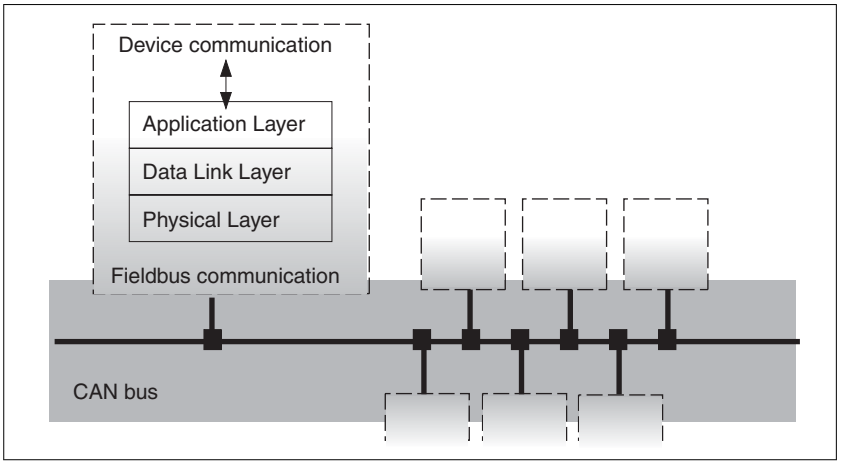


Figure 1: CANopen layer model

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 coding and bit timing.

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

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

0198441113779, V1.05, 10.2012

1.2.3 Objects

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

The product provides corresponding parameters for CANopen object groups 3000_h and 6000_h.
The names of the parameters and the data type of the parameters may be different from the DSP402 definition for object group 6000_h. In this case, enter the data type according to the DS402.
A detailed description of the parameters can be found in the product manual in the Parameters chapter.

Object dictionary

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

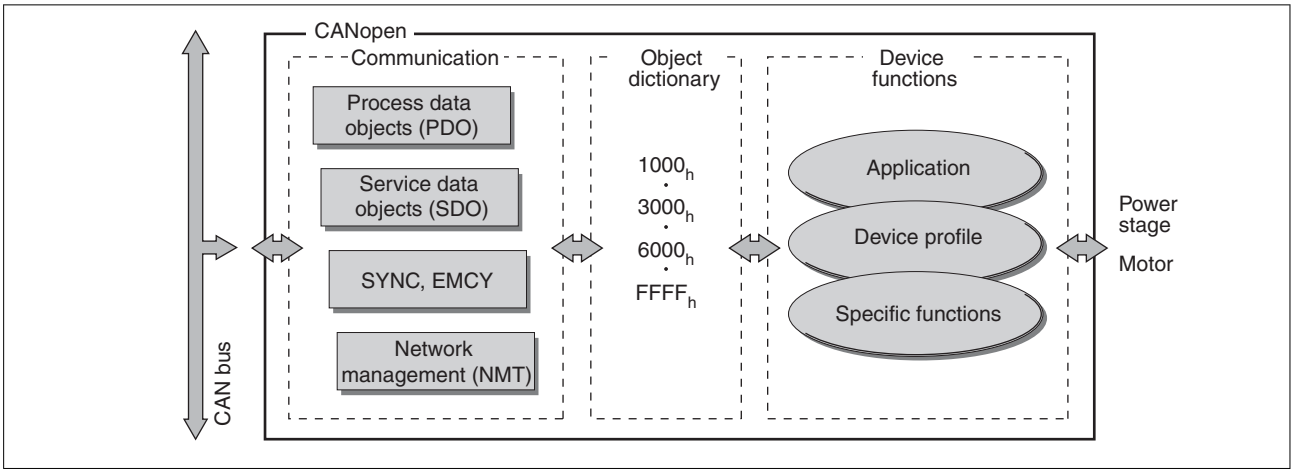


Figure 2: Device model with object dictionary

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.

Index range (hex)	Object groups
1000_h-$2FFF_h$	Communication profile
3000_h-$5FFF_h$	Vendor-specific objects
6000_h-$9FFF_h$	Standardized device profiles
$A000_h$-$FFFF_h$	Reserved

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

0198441113779, V1.05, 10.2012

1.2.4 CANopen profiles

Standardized profiles

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

- DS301 communication profile
- DSP402 device profile

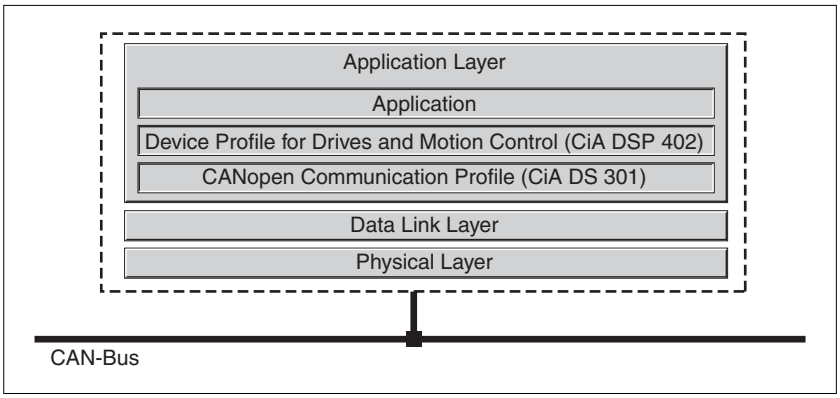


Figure 3: CANopen reference model

DS301 communication profile

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

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

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

0198441113779, V1.05, 10.2012

2 Before you begin - safety information

2

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

2.1 Qualification of personnel

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

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

2.2 Intended use

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

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

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

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

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

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

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

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

0198441113779, V1.05, 10.2012

2.3 Hazard categories

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

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

⚠ DANGER

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

⚠ WARNING

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

⚠ CAUTION

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

NOTICE

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

0198441113779, V1.05, 10.2012

2.4 Basic information

⚠ WARNING

LOSS OF CONTROL

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

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

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

0198441113779, V1.05, 10.2012

2.5 Standards and terminology

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

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

Among others, these standards include:

- IEC 61800: "Adjustable speed electrical power drive systems"
- IEC 61158: "Digital data communications for measurement and control – Fieldbus for use in industrial control systems"
- IEC 61784: "Industrial communication networks – Profiles"
- IEC 61508: "Functional safety of electrical/electronic/programmable electronic safety-related systems"

Also see the glossary at the end of this manual.

0198441113779, V1.05, 10.2012

3 Basics

3

3.1 Communication profile

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

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

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

3.1.1 Object dictionary

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

Index, subindex

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

Example

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

Index	Subindex	Name	Meaning
$607D_h$	00_h	-	Number of data fields
$607D_h$	01_h	minimum position limit	Lower limit switch
$607D_h$	02_h	maximum position limit	Upper limit switch

Table 1: Example of index and subindex entries

Object descriptions in the manual

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

- $1xxx_h$ objects: Communication objects in this chapter.
- $3xxx_h$ objects: Vendor-specific objects required to control the device in chapter "6 Operation".
- $6xxx_h$ objects: Standardized objects of the device profile in chapter "6 Operation".

0198441113779, V1.05, 10.2012

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

3.1.2 Communication objects

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

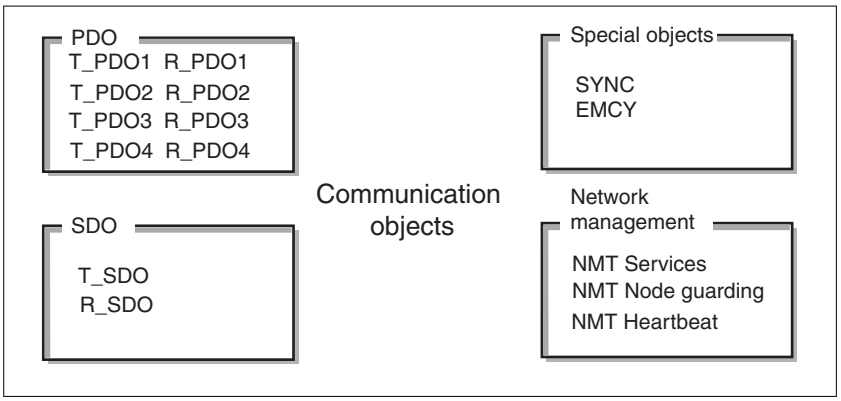


Figure 4: Communication objects; the following applies to the perspective of the network device: T_..: "Transmit", R_..: "Receive"

- 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

0198441113779, V1.05, 10.2012

CAN message

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

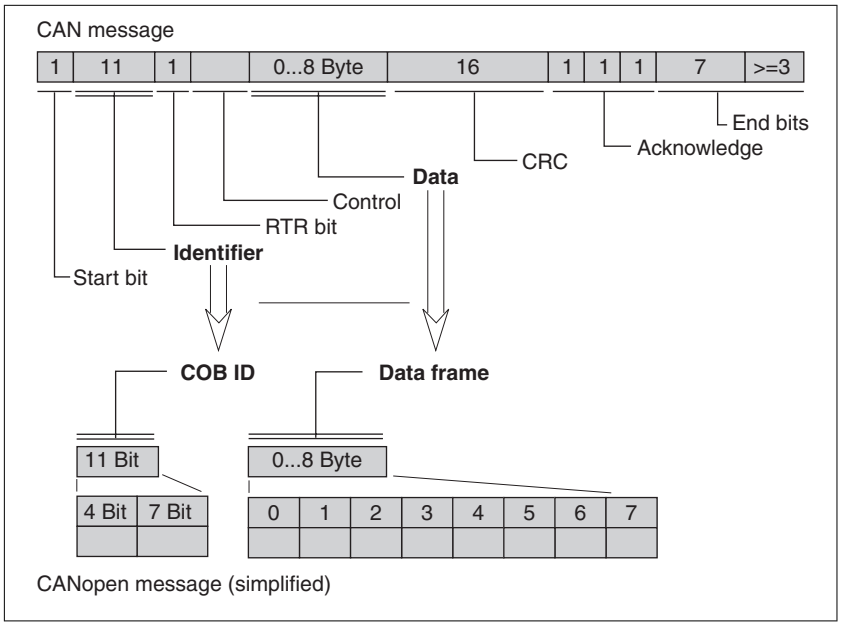


Figure 5: CAN message and simplified representation of CANopen message

CANopen message

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

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

COB ID

The COB ID (**C**ommunication **OB**ject **Id**entifier) has 2 tasks as far as controlling communication objects is concerned:

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

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

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

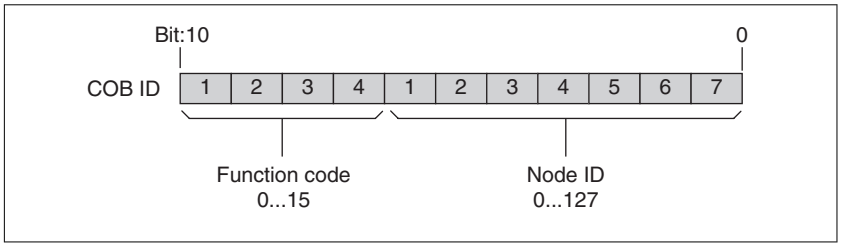


Figure 6: COB ID with function code and node address

0198441113779, V1.05, 10.2012

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_h)	-
SYNC object	0 0 0 1	0 0 0 0 0 0 0	128 (80_h)	1005_h ... 1007_h
EMCY object	0 0 0 1	x x x x x x x	128 (80_h) + node ID	1014_h, 1015_h
T_PDO1 1)	0 0 1 1	x x x x x x x	384 (180_h) + node ID	1800_h
R_PDO1 1)	0 1 0 0	x x x x x x x	512 (200_h) + node ID	1400_h
T_PDO2 1)	0 1 0 1	x x x x x x x	640 (280_h) + node ID	1801_h
R_PDO2 1)	0 1 1 0	x x x x x x x	768 (300_h) + node ID	1401_h
T_PDO3 1)	0 1 1 1	x x x x x x x	896 (380_h) + node ID	1802_h
R_PDO3 1)	1 0 0 0	x x x x x x x	1024 (400_h) + node ID	1402_h
T_PDO4	1 0 0 1	x x x x x x x	1152 (480_h) + node ID	1803_h
R_PDO4	1 0 1 0	x x x x x x x	1280 (500_h) + node ID	1403_h
T_SDO	1 0 1 1	x x x x x x x	1408 (580_h) + node ID	-
R_SDO	1 1 0 0	x x x x x x x	1536 (600_h) + node ID	-
NMT error control	1 1 1 0	x x x x x x x	1792 (700_h) + node ID	-
LMT Services 1)	1 1 1 1	1 1 0 0 1 0 x	2020 ($7E4_h$), 2021 ($7E5_h$)	-
NMT Identify Service 1)	1 1 1 1	1 1 0 0 1 1 0	2022 ($7E6_h$)	-
DBT Services 1)	1 1 1 1	1 1 0 0 x x x	2023 ($7E7_h$), 2024 ($7F8_h$)	-
NMT Services 1)	1 1 1 1	1 1 0 1 0 0 x	2025 ($7E9_h$), 2026 ($7EA_h$)	-

1) Not supported by the device

Table 2: COB IDs of the communication objects

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

Function code

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

Node address

Each network device is configured before it can be operated on the network. The device is assigned a unique 7 bit node address (node ID) between 1 (01_h) and 127 ($7F_h$). The device address "0" is reserved for "broadcast transmissions" which are used to send messages to all 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+node ID = 384 (180_h) + 5 = 389 (185_h).

0198441113779, V1.05, 10.2012

Data frame

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

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

The data frames contain the respective communication objects.

3.1.3 Communication relationships

CANopen uses 3 relationships for communication between network devices:

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

Master-slave relationship

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

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

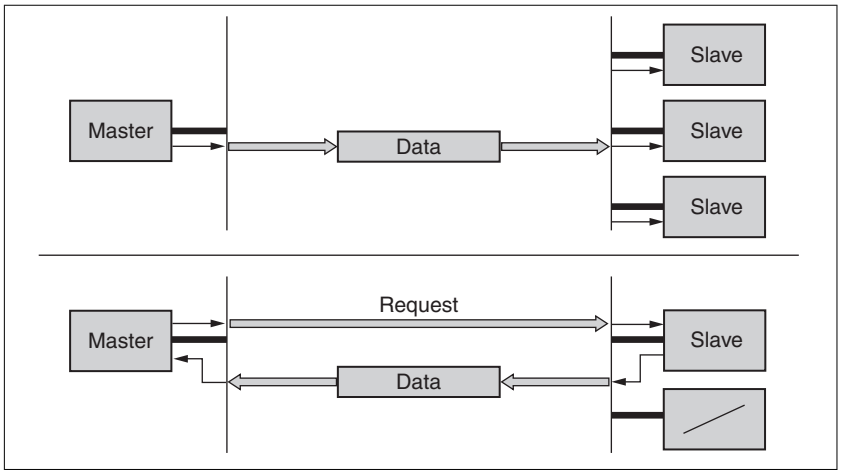


Figure 7: Master - slave relationships

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 all reachable slaves or by no slave.

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

0198441113779, V1.05, 10.2012

Client-server relationship

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

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

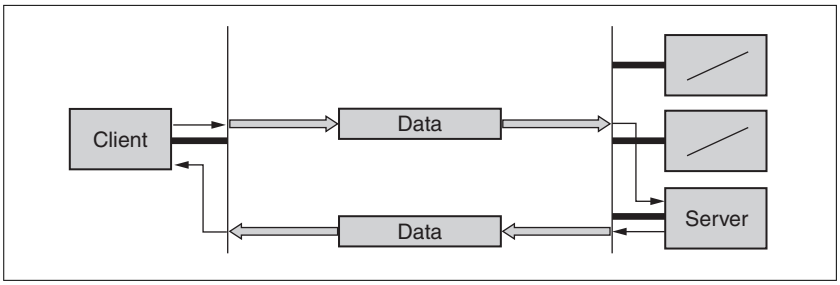


Figure 8: Client-server relationship

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

Producer-consumer relationship

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

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

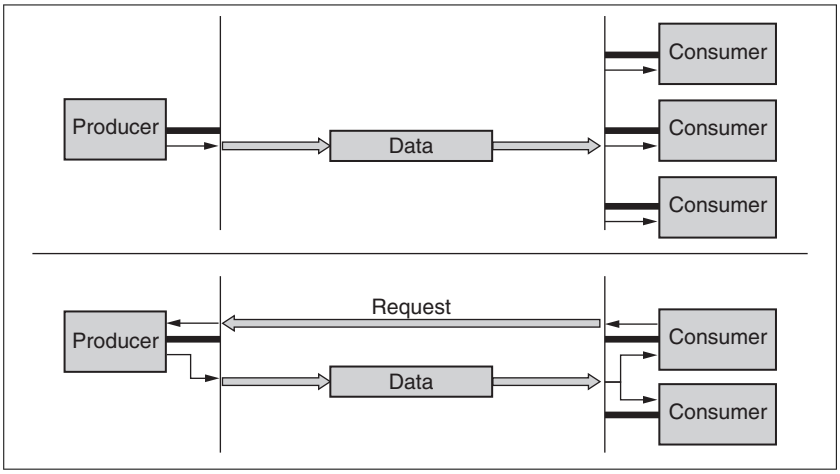


Figure 9: Producer-consumer relationships

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

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

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

0198441113779, V1.05, 10.2012

3.2 Service data communication

3.2.1 Overview

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

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

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

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

3.2.2 SDO data exchange

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

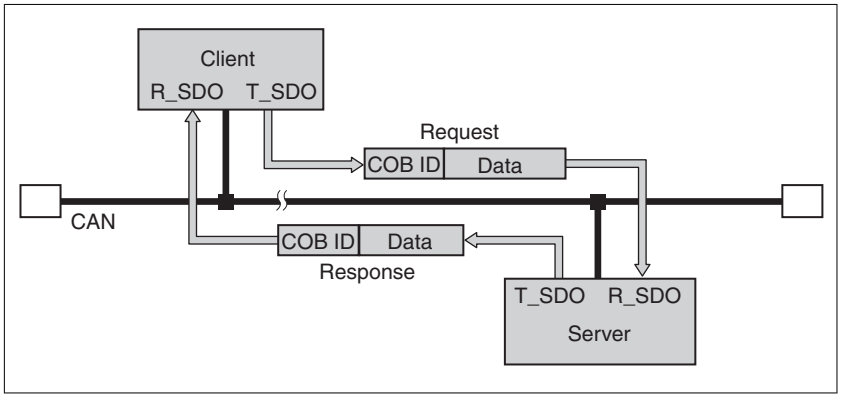


Figure 10: SDO message exchange with request and response

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.

0198441113779, V1.05, 10.2012

3.2.3 SDO message

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

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

Example

The following illustration shows an example of an SDO message.

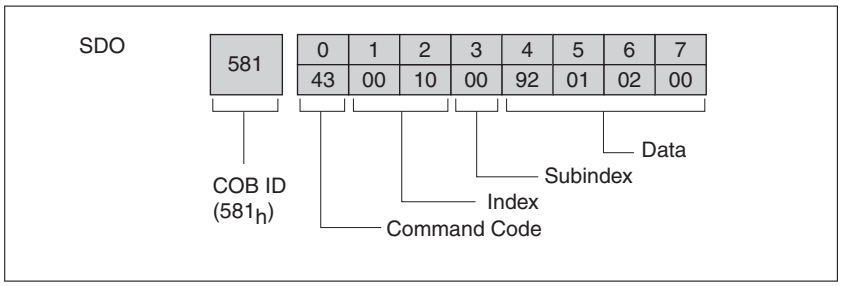


Figure 11: SDO message, example

COB ID and data frame

R_SDO and T_SDO have different COB IDs.
The data frame of an SDO messages consists of:

- Command Code: 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.

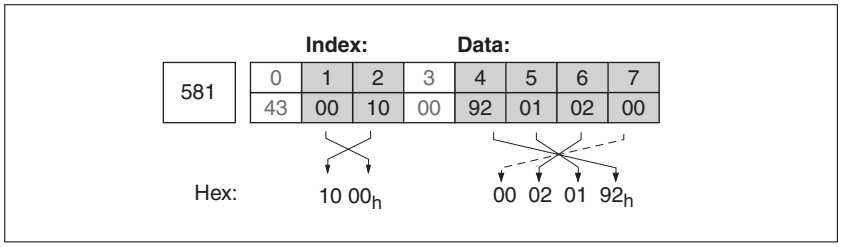


Figure 12: Rearranging numeric values greater than 1 byte

0198441113779, V1.05, 10.2012

3.2.4 Reading and writing data

Writing data

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

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

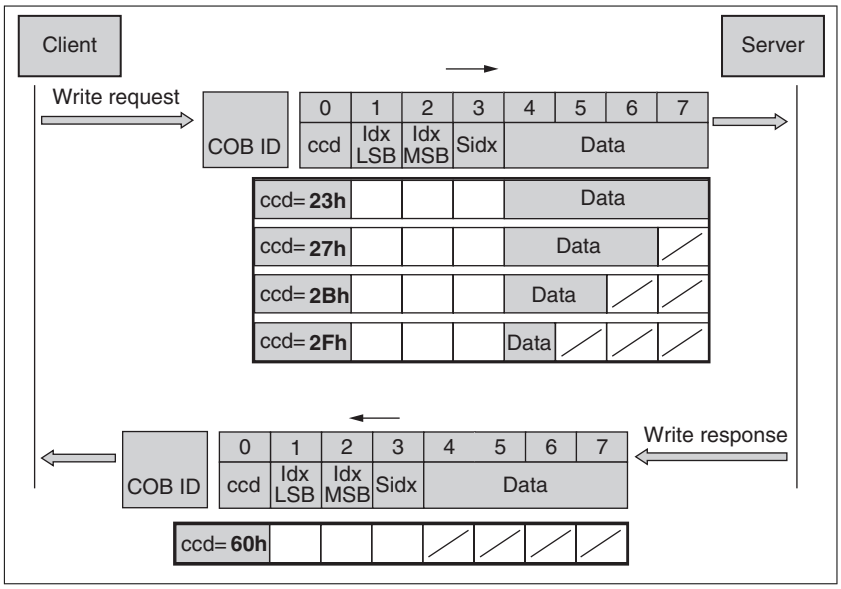


Figure 13: Writing parameter values

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

ccd coding

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

Message type	**Data length used**				
	4 byte	**3 byte**	**2 byte**	**1 byte**	
Write request	23_h	27_h	$2B_h$	$2F_h$	Transmitting parameters
Write response	60_h	60_h	60_h	60_h	Confirmation
Error response	80_h	80_h	80_h	80_h	Error

Table 3: Command code for writing parameter values

0198441113779, V1.05, 10.2012

Reading data

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

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

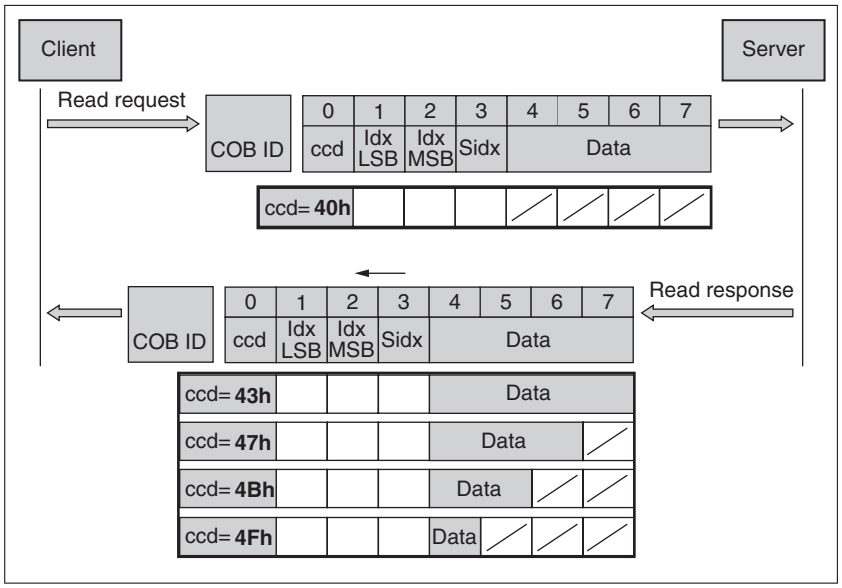


Figure 14: Reading a parameter value

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

ccd coding

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

Message type	**Data length used**				
	4 byte	**3 byte**	**2 byte**	**1 byte**	
Read request	40_h	40_h	40_h	40_h	Request read value
Read response	43_h	47_h	$4B_h$	$4F_h$	Return read value
Error response	80_h	80_h	80_h	80_h	Error

Table 4: Command code for transmitting a read value

Error response

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

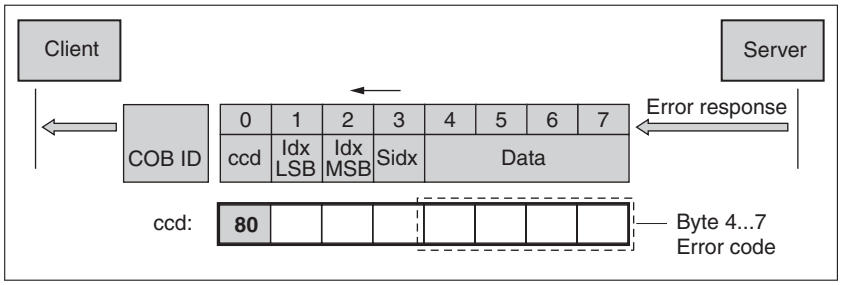


Figure 15: Response with error message (error response)

0198441113779, V1.05, 10.2012

3.2.5 Reading data longer than 4 bytes

If values of more than 4 bytes are to be transmitted with an SDO message, the message must be divided into several 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 whose value it wants to read.

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

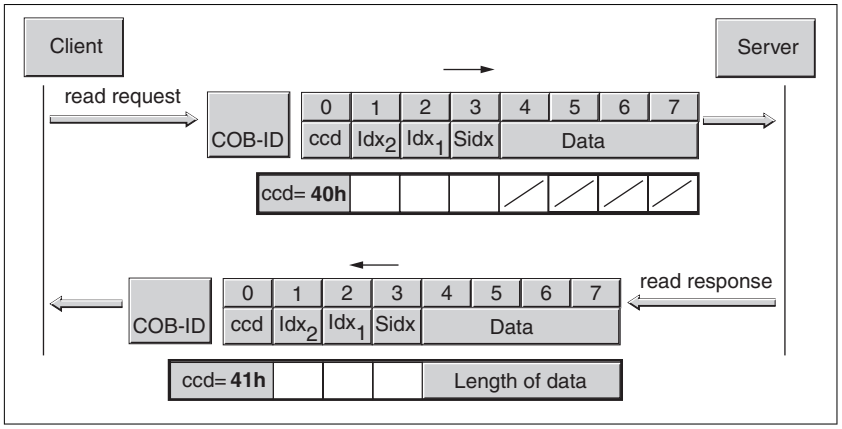


Figure 16: First read request

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 all data is transmitted.

0198441113779, V1.05, 10.2012

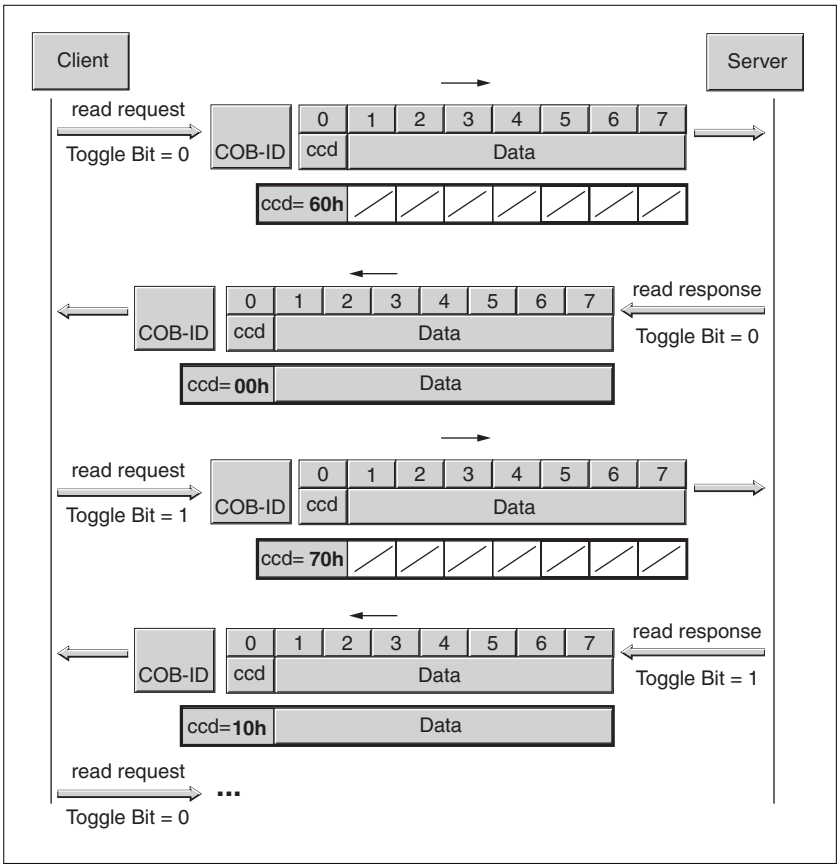


Figure 17: Additional read requests

It is possible to detect whether all data has been transmitted on the basis of the command code of the server. Once all 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.

0198441113779, V1.05, 10.2012

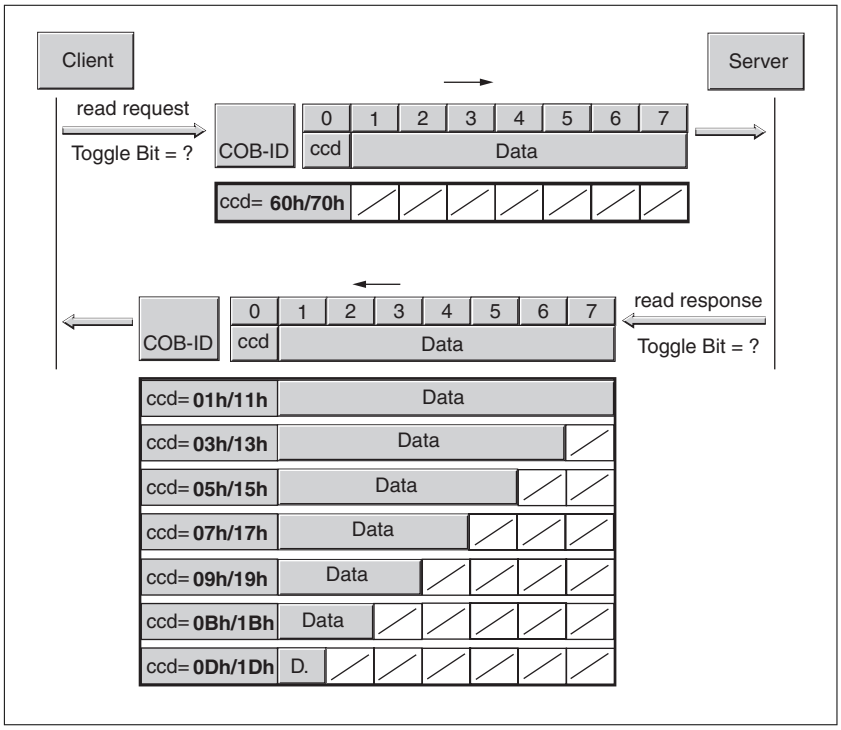


Figure 18: Final read request

3.3 Process data communication

3.3.1 Overview

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

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

The length of a PDO message and the assignment of the data fields are specified by PDO mapping. See chapter "3.3.5 PDO mapping" for additional information.

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

3.3.2 PDO data exchange

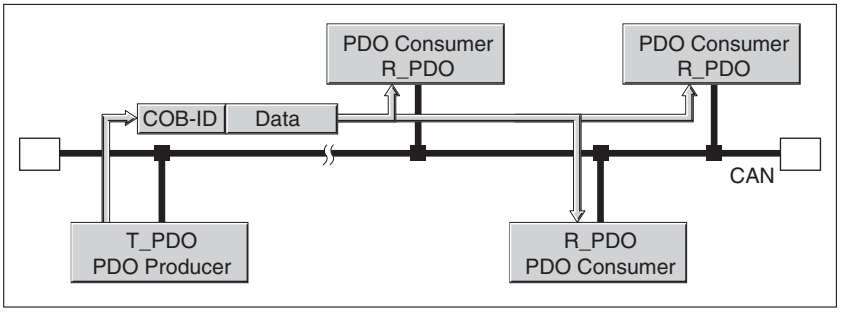


Figure 19: PDO data exchange

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

- Synchronized
- Event-driven, asynchronous

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

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

The transmission type can be specified separately for each PDO with subindex 02_h (transmission type) of the PDO communication parameter. The objects are listed in Table 5.

0198441113779, V1.05, 10.2012

3.3.3 PDO message

T_PDO, R_PDO

One PDO each is available for sending and receiving a PDO message:

- T_PDO to transmit the PDO message (T: Transmit),
- R_PDO to receive PDO messages (R: Receive).

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

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

PDO settings

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

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

Table 5: Communication objects for PDO

Activating PDOs

With the default PDO settings, R_PDO1 and T_PDO1 are activated. The other PDOs must be activated first.

A PDO is activated with bit 31 (valid bit) in subindex 01_h of the respective communication object:

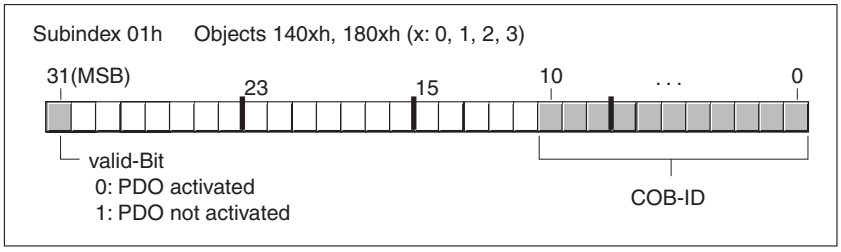


Figure 20: Activating PDOs via subindex 01_h, bit 31

Example

Setting for R_PDO3 in object 1402_h

- Subindex 01_h = 8000 $04xx_h$: R_PDO3 not activated
- Subindex 01_h = 0000 $04xx_h$: R_PDO3 activated.

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

0198441113779, V1.05, 10.2012

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_h.
- The time interval "event timer" cyclically triggers an event message. After the time intervals has elapsed, the device transmits the event-controlled T_PDO. The time is set with subindex 05_h.

Receive PDOs

The objects for R_PDO1, R_PDO2, R_PDO3 and R_PDO4 are preset.

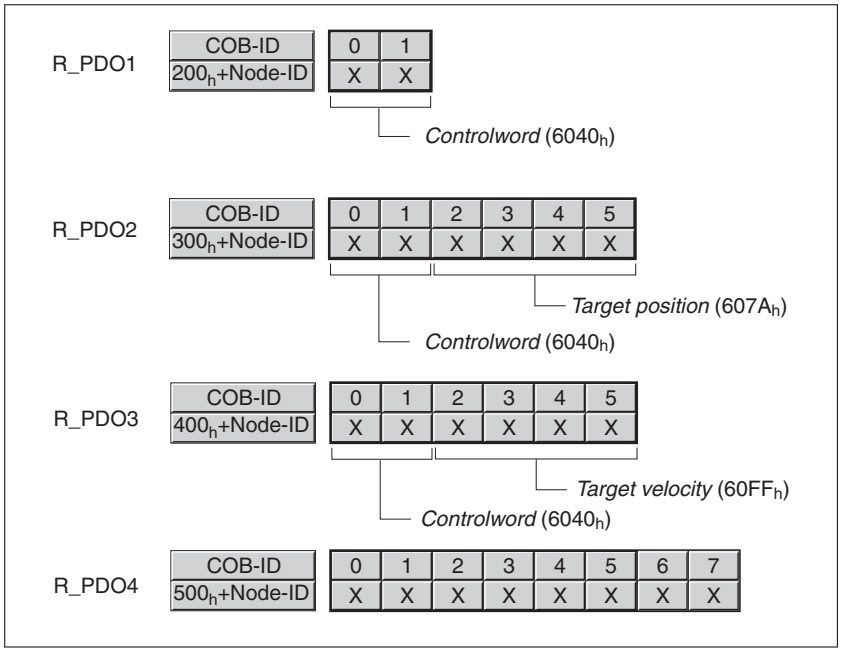


Figure 21: Receive PDOs

R_PDO1

R_PDO1 contains the control word, object `controlword (6040h)`, of the state machine which can be used to set the operating state of the device.

R_PDO1 is evaluated asynchronously, i.e. it is event-driven. R_PDO1 is preset.

R_PDO2

With R_PDO2, the control word and the target position of a motion command, object `target position (607Ah)`, are received for a movement in the operating mode "Profile Position".

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

For details on the SYNC object see chapter "3.4 Synchronization".

R_PDO3

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

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

0198441113779, V1.05, 10.2012

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.

The R_PDOs can be used to map various vendor-specific objects by means of PDO mapping.

Transmit PDOs The objects for T_PDO1, T_PDO2, T_PDO3 and T_PDO4 can be changed by means of PDO mapping.

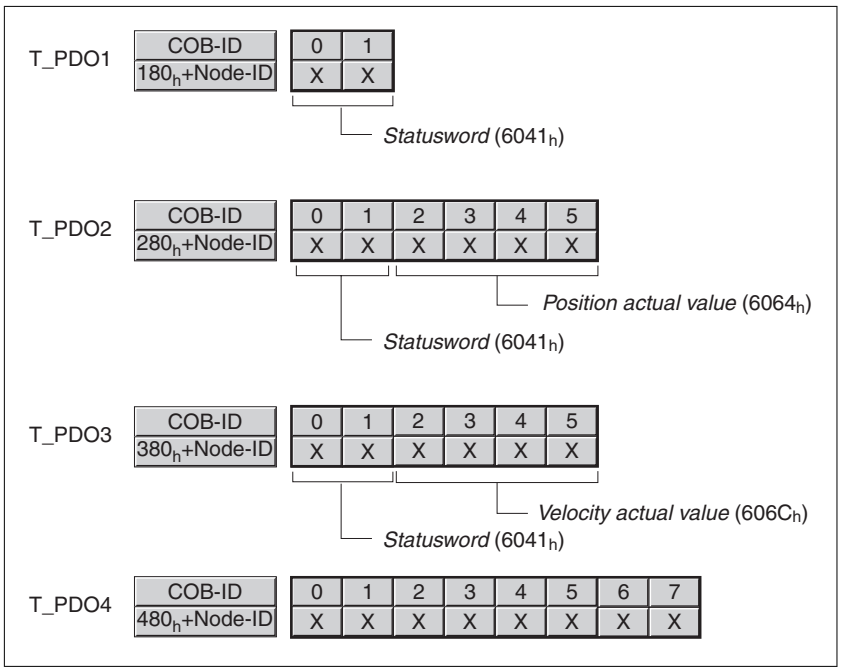


Figure 22: Transmit PDOs

T_PDO1 T_PDO1 contains the status word, object `statusword (6041h)`, of the state machine.

T_PDO1 is transmitted asynchronously and in an event-driven way whenever the status information changes.

T_PDO2 T_PDO2 contains the status word and the actual position of the motor, object `Position actual value (6064h)`, to monitor movements in the operating mode "Profile Position".

T_PDO2 is transmitted after receipt of a SYNC object and in an event-driven way.

T_PDO3 T_PDO3 contains the status word and the actual velocity, object `Velocity actual value (606Ch)`, for monitoring the actual velocity in the operating mode "Profile Velocity".

T_PDO3 is transmitted asynchronously and in an event-driven way whenever the status information changes.

T_PDO4 Vendor-specific object values (for monitoring) are transmitted with T_PDO4. By default, T_PDO4 is empty.

T_PDO4 is transmitted asynchronously and in an event-driven way whenever the data changes.

The T_PDOs can be used to map various vendor-specific objects via PDO mapping.

0198441113779, V1.05, 10.2012

3.3.4 PDO events

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
`CANpdo1Event`	PDO 1 event mask Changes of values in the object trigger an event: Bit 0: First PDO object Bit 1: Second PDO object Bit 2: Third PDO object Bit 3: Fourth PDO object Changed settings become active immediately.	- 0 1 15	UINT16 UINT16 R/W - -	CANopen 3041:B_h Modbus 16662
`CANpdo2Event`	PDO 2 event mask Changes of values in the object trigger an event: Bit 0: First PDO object Bit 1: Second PDO object Bit 2: Third PDO object Bit 3: Fourth PDO object Changed settings become active immediately.	- 0 1 15	UINT16 UINT16 R/W - -	CANopen 3041:C_h Modbus 16664
`CANpdo3Event`	PDO 3 event mask Changes of values in the object trigger an event: Bit 0: First PDO object Bit 1: Second PDO object Bit 2: Third PDO object Bit 3: Fourth PDO object Changed settings become active immediately.	- 0 1 15	UINT16 UINT16 R/W - -	CANopen 3041:D_h Modbus 16666
`CANpdo4Event`	PDO 4 event mask Changes of values in the object trigger an event: Bit 0: First PDO object Bit 1: Second PDO object Bit 2: Third PDO object Bit 3: Fourth PDO object Changed settings become active immediately.	- 0 15 15	UINT16 UINT16 R/W - -	CANopen 3041:E_h Modbus 16668

0198441113779, V1.05, 10.2012

3.3.5 PDO mapping

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

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

The picture below shows the data exchange between PDOs and object dictionary on the basis of two examples of objects in T_PDO4 and R_PDO4 of the PDOs.

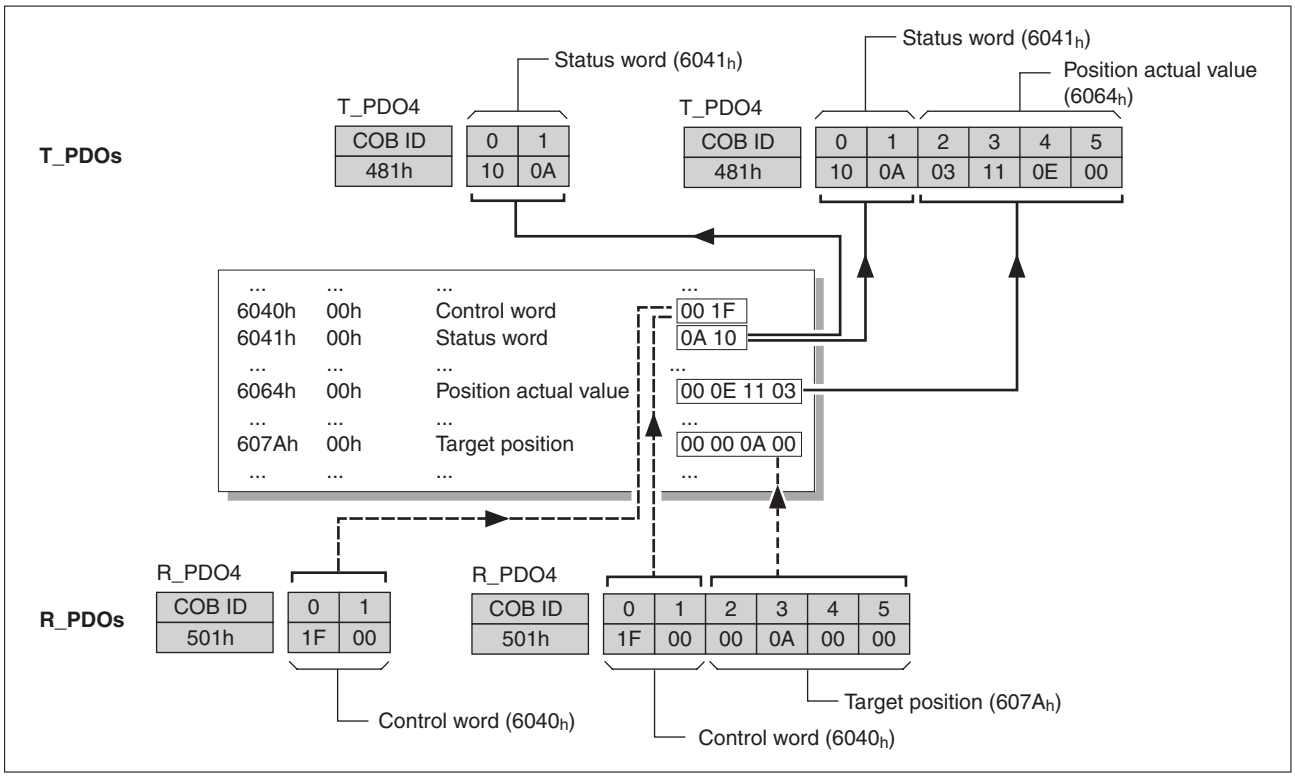


Figure 23: PDO mapping, in this case for a device with node address 1

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_h)	R_PDO1	Dynamic
2nd receive PDO mapping (1601_h)	R_PDO2	Dynamic
3rd receive PDO mapping (1602_h)	R_PDO3	Dynamic
4th receive PDO mapping (1603_h)	R_PDO4	Dynamic
1st transmit PDO mapping ($1A00_h$)	T_PDO1	Dynamic
2nd transmit PDO mapping ($1A01_h$)	T_PDO2	Dynamic
3rd transmit PDO mapping ($1A02_h$)	T_PDO3	Dynamic
4th transmit PDO mapping ($1A03_h$)	T_PDO4	Dynamic

0198441113779, V1.05, 10.2012

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.

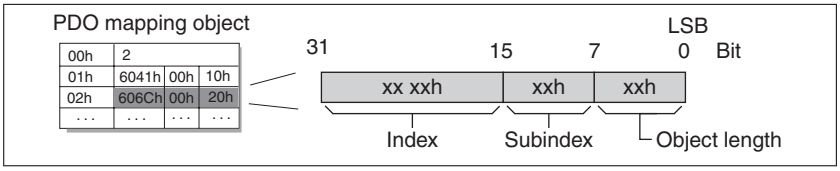


Figure 24: Structure of entries for PDO mapping

Subindex 00_h of the communication object contains the number of valid subindex entries.

Object length	**Bit value**
08_h	8 bits
10_h	16 bits
20_h	32 bits

0198441113779, V1.05, 10.2012

PDO mapping objects

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

Index	Subindex	Object	PDO	Data type	Parameter name	Takes effect
6040h	0	Controlword	R_PDO	UINT16	DCOMcontrol	Immediately
6041h	0	Statusword	T_PDO	UINT16	_DCOMstatus	-
6060h	0	Modes of Operation	R_PDO	INT8	DCOMopmode	Immediately
6061h	0	Modes of Operation Display	T_PDO	INT8	_DCOMopmd_act	-
6063h	0	Position actual Value	T_PDO	INT32	_p_actInt	-
6064h	0	Position actual Value	T_PDO	INT32	_p_act	-
606Ch	0	Velocity actual value	T_PDO	INT32	_v_act	-
6071h	0	Target Torque	R_PDO	INT16	PTtq_target	Immediately
6077h	0	Torque actual Value	T_PDO	INT16	_tq_act	-
607Ah	0	Target position	R_PDO	INT32	PPp_target	Immediately
6081h	0	Profile velocity	R_PDO	UINT32	PPv_target	Next movement
6083h	0	Profile acceleration	R_PDO	UINT32	RAMP_v_acc	Next movement
6084h	0	Profile deceleration	R_PDO	UINT32	RAMP_v_dec	Next movement
6087	0	Torque slope	R_PDO	UNIT32	RAMP_tq_slope	Immediately
60FFh	0	Target velocity	R_PDO	INT32	PVv_target	Immediately

0198441113779, V1.05, 10.2012

3.4 Synchronization

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

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

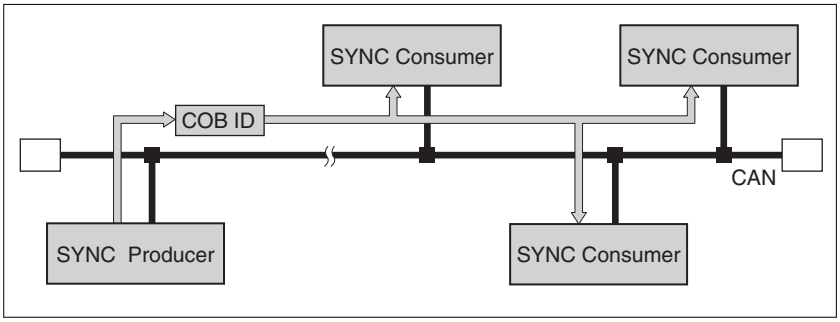


Figure 25: SYNC message

Time values for synchronization

Two time values define the behavior of synchronous data transmission:

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

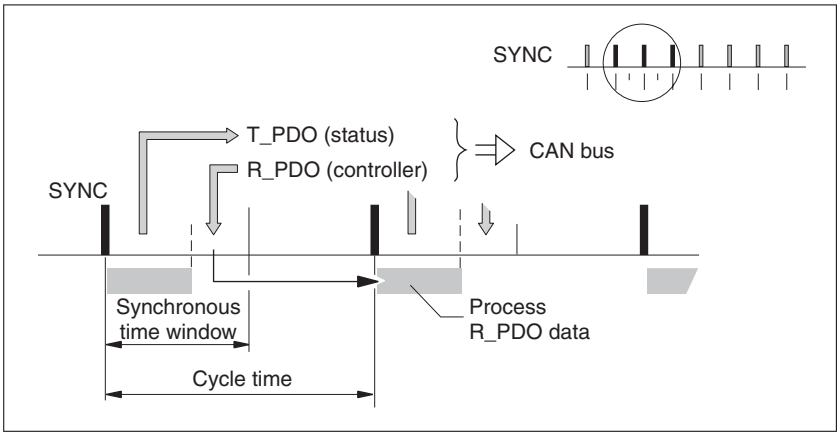


Figure 26: Synchronization times

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.

0198441113779, V1.05, 10.2012

Cyclic ad acyclic data transmission

Synchronous exchange of messages can be cyclic or acyclic.

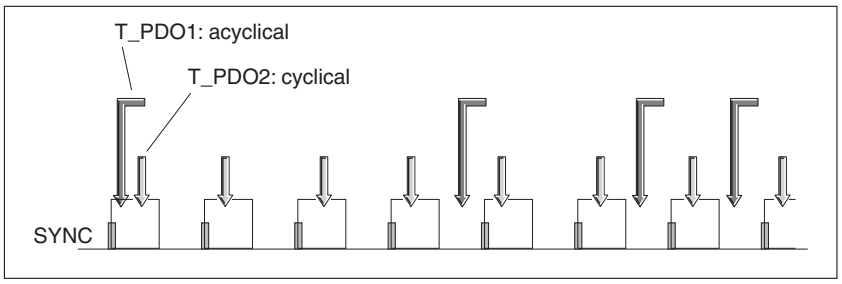


Figure 27: Cyclic and acyclic transmission

In the case of cyclic transmission, PDO messages are exchanged continuously in a specified cycle, for example with each SYNC message.

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

Cyclic or acyclic behavior of a PDO is specified in the subindex `transmission type (02h)` of the corresponding PDO parameter, for example, in the object `1st receive PDO parameter (1400h: 02h)` for R_PDO1.

COB ID, SYNC object

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

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

"Start" PDO

With the default settings of the PDOs, R_PDO1 ... R_PDO4 and T_PDO1 ... T_PDO4 are received and transmitted asynchronously. T_PDO2 ... T_PDO3 are transmitted additionally after the event timer has elapsed. The synchronization allows an operating mode to be started simultaneously on multiple devices so that, for example, the feed of a portal drive with several motors can be synchronized.

0198441113779, V1.05, 10.2012

3.5 Emergency object service

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.

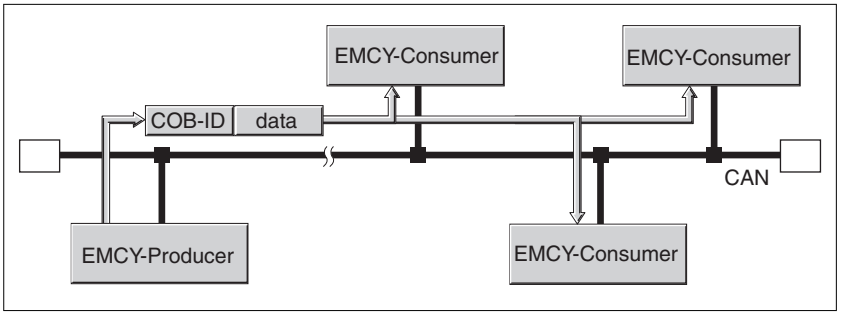


Figure 28: Error message via EMCY objects

Boot-up message

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

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

3.5.1 Error evaluation and handling

EMCY message

If an 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.

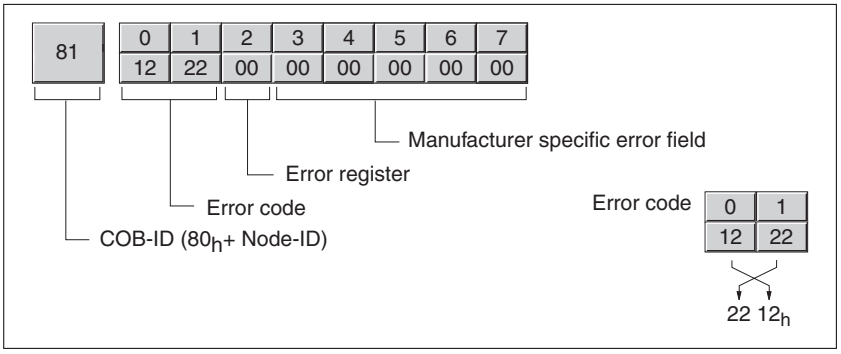


Figure 29: EMCY message

Bytes 0 ... 1: Error code (as per DS301)
The value is also saved in the object `Predefined error field (1003:1h)`.

Byte 2: Error register
The value is also saved in the object `Error register (1001h)`, see "7.4.1 Error register ".

Bytes 3 ... 4: Reserved

Byte 5: PDO: Number of the PDO

Bytes 6 ... 7: Vendor-specific error number
The value is also saved in the object `Error code (603Fh)`.

0198441113779, V1.05, 10.2012

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 = Function code EMCY object (80_h) + node ID

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

Error register and error code

The error register contains bit-coded information on the error. Bit 0 remains set as long as an error is active. The remaining bits identify the error type. The exact cause of error can be determined on the basis of the error code. The error code is transmitted in Intel format as a 2 byte value; the bytes must be reversed for evaluation.

See chapter "7 Diagnostics and troubleshooting" for a list of the error messages and error responses by the device as well as remedies.

Error memory

The device saves the error register in the object `Error register (1001h)` and the last error that occurred in the object `Error code (603Fh)`. The last 10 error messages are stored in the object `FLT_err_num (303C:1h)` in the order in which the errors occurred. `FLT_MemReset (303B:5h)` resets the read pointer of the error memory to the oldest error.

0198441113779, V1.05, 10.2012

3.6 Network management services

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

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

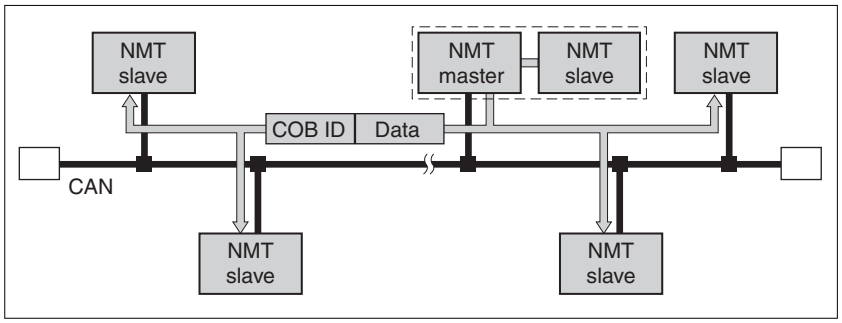


Figure 30: NMT services via the master-slave relationship

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

NMT services

NMT services can be divided into 2 groups:

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

0198441113779, V1.05, 10.2012

3.6.1 NMT services for device control

NMT state machine

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

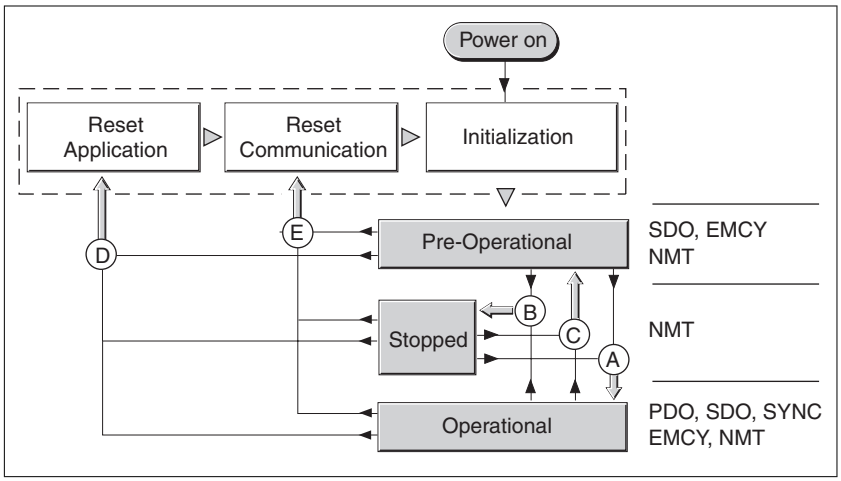


Figure 31: NMT state machine and available communication objects

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

NMT service	Transition	Meaning
Start remote node (Start network node)	A	Transition to operating state "Operational" Start normal operation on the network
Stop remote node (Stop network node)	B	Transition to operating state "Stopped" Stops communication of the network device on the network. If connection monitoring is active, it remains on. If the power stage is 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". 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 switched on (power on), the device loads the saved object data from the non-volatile EEPROM for persistent data to the RAM.

0198441113779, V1.05, 10.2012

NMT message

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

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

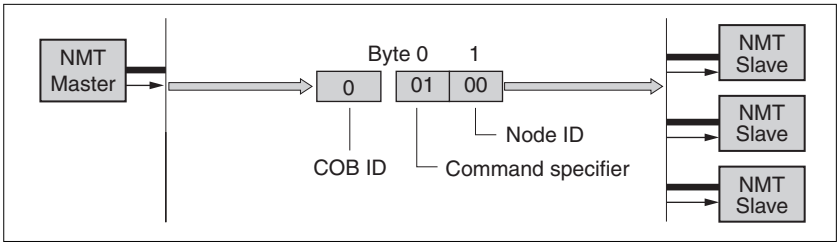


Figure 32: NMT message

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

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

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

0198441113779, V1.05, 10.2012

3.6.2 NMT-service Node Guarding/Life Guarding

COB ID

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

COB ID = function code `NMTerror control (700h) + Node-ID`.

Structure of the NMT message

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

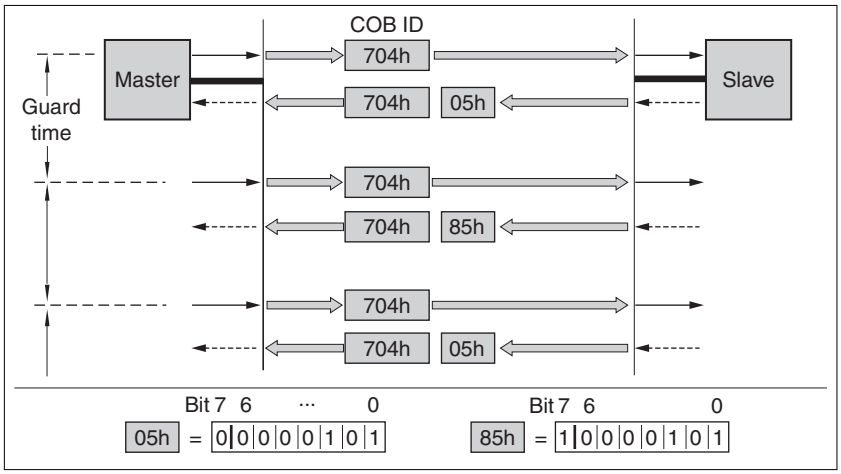


Figure 33: Acknowledgement of the NMT slave

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

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

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

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

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

Configuration

Node Guarding/Life Guarding is configured via:

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

0198441113779, V1.05, 10.2012

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 of a missing response from an NMT slave.

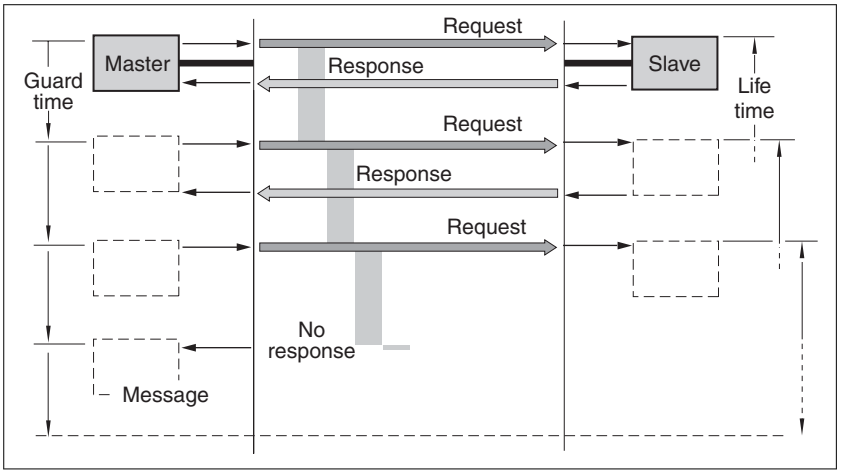


Figure 34: "Node Guarding" and "Life Guarding" with time intervals

0198441113779, V1.05, 10.2012

3.6.3 NMT service Heartbeat

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

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

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

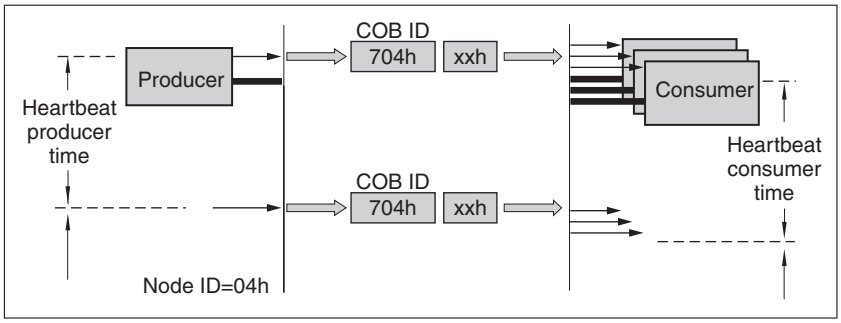


Figure 35: "Heartbeat" monitoring

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

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

Time intervals

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

Start of monitoring

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

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

0198441113779, V1.05, 10.2012

0198441113779, V1.05, 10.2012

4 Installation

4

⚠ WARNING

SIGNAL AND DEVICE INTERFERENCE

Signal interference can cause unexpected responses of the device.

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

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

For information on installation of the device and connecting the device to the fieldbus see the product manual.

0198441113779, V1.05, 10.2012

0198441113779, V1.05, 10.2012

5 Commissioning

5

⚠ WARNING

LOSS OF CONTROL

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

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

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

⚠ WARNING

UNINTENDED OPERATION

- Do not write values to reserved parameters.
- Do not write values to parameters unless you fully understand the function.
- Run initial tests without coupled loads.
- Verify correct word order for fieldbus communication.
- Do not establish a fieldbus connection unless you have fully understood the communication principles.
- Only start the system if there are no persons or obstructions in the hazardous area.

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

Using the library considerably facilitates controlling the device. The library is available for download from the Internet.
http://www.schneider-electric.com

5.1 Commissioning the device

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

▸ Commission the device as per product manual.

0198441113779, V1.05, 10.2012

5.2 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 its own node address, i.e. any given node address may be assigned only once in the network.

Setting address and baud rate

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` (CoAd).
- Set the transmission rate in the parameter `CANbaud` (Cobd) 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 field-bus**
`CANaddress` ConF → CoN- ConF → FSu- CoAd	CANopen address (node number) Changed settings become active the next time the product is switched on.	- 1 - 127	 R/W per. -	
`CANbaud` ConF → CoN- ConF → FSu- Cobd	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 Changed settings become active the next time the product is switched on.	- 50 250 1000	 R/W per. -	

0198441113779, V1.05, 10.2012

6 Operation

6

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

⚠ WARNING

UNINTENDED OPERATION

- Do not write values to reserved parameters.
- Do not write values to parameters unless you fully understand the function.
- Run initial tests without coupled loads.
- Verify correct word order for fieldbus communication.
- Do not establish a fieldbus connection unless you have fully understood the communication principles.
- Only start the system if there are no persons or obstructions in the hazardous area.

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

Using the library considerably facilitates controlling the device. The library is available for download from the Internet.
http://www.schneider-electric.com

0198441113779, V1.05, 10.2012

6.1 Operating states

6.1.1 State diagram

After switching on and when an operating mode is started, the product goes through a number of operating states.

The state diagram (state machine) shows the relationships between the operating states and the state transitions.

The operating states are internally monitored and influenced by monitoring functions

Graphical representation The state diagram is represented as a flow chart.

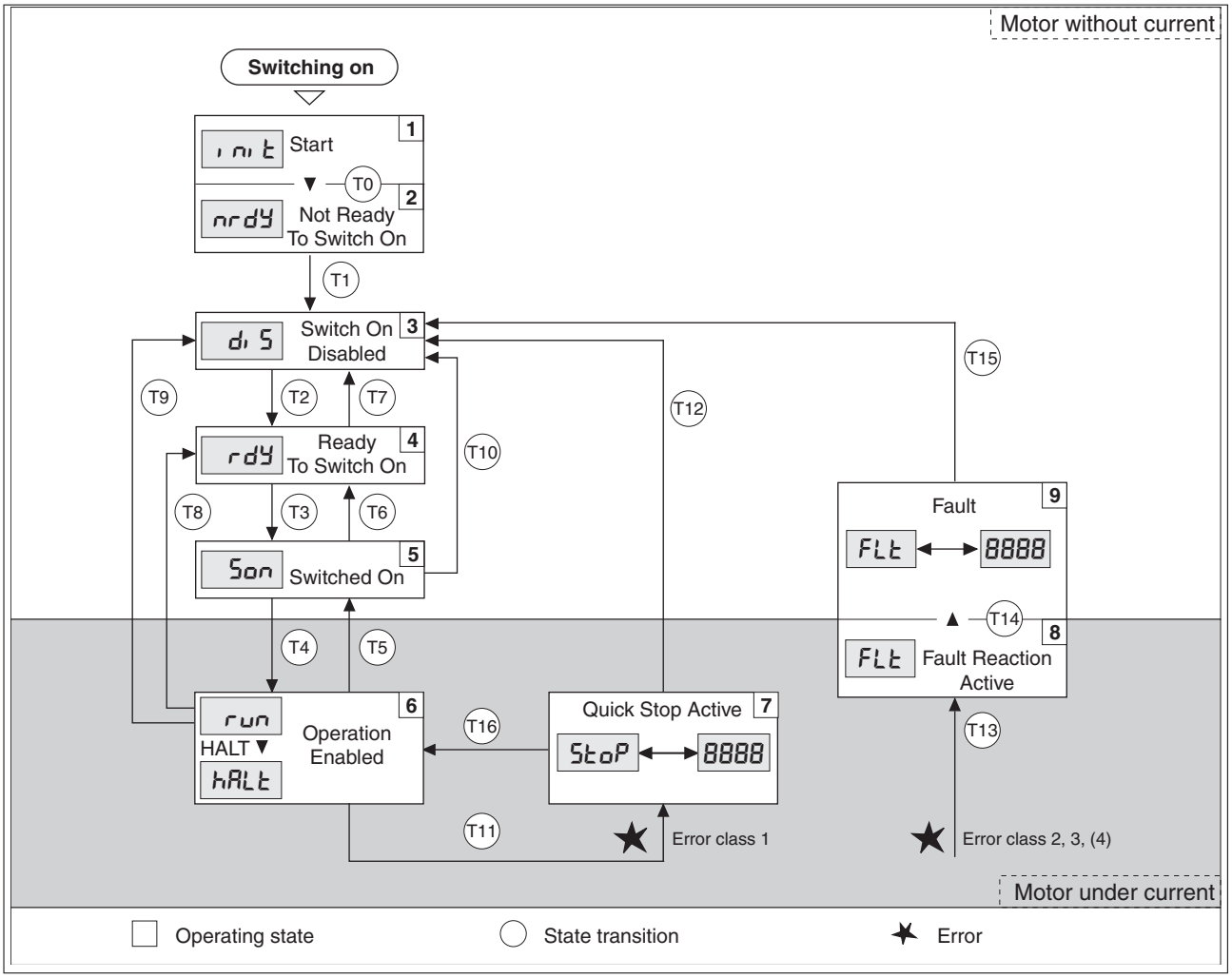


Figure 36: State diagram

0198441113779, V1.05, 10.2012

Operating states

Operating state	Description
1 Start	Electronics are initialized
2 Not Ready To Switch On	The power stage is not ready to switch on
3 Switch On Disabled	Impossible to enable the power stage
4 Ready To Switch On	The power stage is ready to switch on.
5 Switched On	Power stage is switched on
6 Operation Enabled	Power stage is enabled Selected operating mode is active
7 Quick Stop Active	"Quick Stop" is being executed
8 Fault Reaction Active	Error response is active
9 Fault	Error response terminated Power stage is disabled

Error class

The product triggers an error response if an error occurs. Depending upon the severity of the error, the device responds in accordance with one of the following error classes:

Error class	Response	Meaning
0	Warning	A monitoring function has detected a problem. No interruption of the movement.
1	"Quick Stop"	Motor stops with "Quick Stop", the power stage remains enabled.
2	"Quick Stop" with switch-off	Motor stops with "Quick Stop", the power stage is disabled after standstill has been achieved.
3	Fatal error	The power stage is immediately disabled without stopping the motor first.
4	Uncontrolled operation	The power stage is immediately disabled without stopping the motor first. The error can only be reset by switching off the product.

Error response

The state transition T13 (error class 2, 3 or 4) initiates an error response as soon as an internal occurrence signals an error to which the device must react.

Error class	Response
2	Movement is stopped with "Quick Stop" Holding brake is applied Power stage is disabled
3, 4 or Safety function STO	Power stage is immediately disabled

An error can be triggered by a temperature sensor, for example. The product cancels the current movement and triggers an error response. Subsequently, the operating state changes to **9** Fault.

To exit the **9** Fault operating state, the cause of the error must be remedied and a Fault Reset must be executed.

Resetting an error message

A "Fault Reset" resets an error message.

0198441113779, V1.05, 10.2012

*In the event of a "Quick Stop" triggered by an error of class 1 (operating state **7** Quick Stop Active), a "Fault Reset" causes a direct transition to operating state **6** Operation Enabled.*

6.1.2 Indication of the operating state

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

Parameter name HMI menu HMI name	Description	Unit Minimum value Factory setting Maximum value	Data type R/W Persistent Expert	Parameter address via fieldbus
`_DCOMstatus`	DriveCom status word Bit assignments: Bits 0 ... 3: Status bits Bit 4: Voltage enabled Bits 5 ... 6: Status bits Bit 7: Warning Bit 8: HALT request active Bit 9: Remote Bit 10: Target reached Bit 11: Assignment can be set via parameter DS402intLim Bit 12: Operating mode-specific Bit 13: x_err Bit 14: x_end Bit 15: ref_ok	- - - -	UINT16 UINT16 R/- - -	CANopen 6041:0$_h$ Modbus 6916

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

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

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

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

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

0198441113779, V1.05, 10.2012

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

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

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

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

Bit 13 Bit 13 only becomes "1" in the case of an error which needs to be remedied prior to further processing. The device responds corresponding to an error class.

Bit 14 Bit 14 changes to "0" if an operating mode is started. When processing is terminated or interrupted, for example by a "Halt", bit 14 toggles back to "1" once the motor has come to a standstill. The signal change of bit 14 to "1" is suppressed if one process is followed immediately by a new process in a different operating mode.

Bit 15 Bit 15 is "1" if the motor has a valid zero point, for example as a result of a reference movement. A valid zero point remains valid even if the power stage is disabled.

0198441113779, V1.05, 10.2012

6.1.3 Changing the operating state

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

Parameter name HMI menu HMI name	Description	Unit Minimum value Factory setting Maximum value	Data type R/W Persistent Expert	Parameter address via fieldbus
`DCOMcontrol`	DriveCom control word Refer to chapter Operation, Operating States, for bit coding information. Bit 0: Switch on Bit 1: Enable Voltage Bit 2: Quick Stop Bit 3: Enable Operation Bits 4 ... 6: Operating mode specific Bit 7: Fault Reset Bit 8: Halt Bit 9: Change on setpoint Bits 10 ... 15: Reserved (must be 0) Changed settings become active immediately.	- - - -	UINT16 UINT16 R/W - -	CANopen 6040:0_h 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	X	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	X	X	0	X
Quick Stop	T7, T10 T11	**3** Switch On Disabled **7** Quick Stop Active	0	X	0	1	X
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	X	X	X	X

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

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

Bits 9 ... 15 Reserved.

0198441113779, V1.05, 10.2012

6.2 Operating modes

6.2.1 Starting and changing an operating mode

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

Parameter name HMI menu HMI name	Description	Unit Minimum value Factory setting Maximum value	Data type R/W Persistent Expert	Parameter address via field-bus
DCOMopmode	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 Changed settings become active immediately.	- -6 - 7	INT8 INT16 R/W - -	CANopen 6060:0h Modbus 6918

▸ Set the operating mode with the parameter DCOMopmode.

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

Parameter name HMI menu HMI name	Description	Unit Minimum value Factory setting Maximum value	Data type R/W Persistent Expert	Parameter address via field-bus
_DCOMopmd_act	Active operating mode **-6 / Manual Tuning / Autotuning**: Manual Tuning / 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	- -6 - 10	INT8 INT16 R/- - -	CANopen 6061:0h Modbus 6920

0198441113779, V1.05, 10.2012

6.2.2 Operating mode Jog

Starting the operating mode

The operating mode must be set in the parameter `DCOMopmode`. Writing the parameter value activates the operating mode.

The parameter `JOGactivate` starts the movement.

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

Control word

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

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

Status word

Information on the current movement is available via bits 10 and 12 ... 15 in the parameter `DCOMstatus`.

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

Terminating the operating mode

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

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

0198441113779, V1.05, 10.2012

6.2.2.1 Example Node address 1

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$_h$ 0064$_h$
▸ Fast velocity to 250 601 / 23 29 30 05 FA 00 00 00 ◃ 581 / 60 29 30 05 00 00 00 00	3029:5$_h$ 00FA$_h$
▸ NMT Start remote node 0 / 01 00 ◃ T_PDO1 with status word 181 / 31 62	
▸ Enable power stage with R_PDO1 201 / 00 00 201 / 06 00 201 / 0F 00 ◃ T_PDO1 (operating state: **6** Operation Enabled) 181 / 37 42	
▸ Starting the operating mode 601 / 2F 60 60 00 FF 00 00 00 ◃ 581 / 60 60 60 00 00 00 00 00	6060$_h$ FF$_h$
▸ 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$_h$ FF$_h$
▸ Start movement (positive direction, slow) 601 / 23 1B 30 09 01 00 00 00 ◃ 581 / 60 1B 30 09 00 00 00 00 ◃ T_PDO1 with status word 181 / 37 02	301B:9$_h$ 01$_h$
▸ Start movement (positive direction, fast) 601 / 23 1B 30 09 05 00 00 00 ◃ 581 / 60 1B 30 09 00 00 00 00 ◃ T_PDO1 with status word 181 / 37 02	301B:9$_h$ 05$_h$
▸ Terminate movement 601 / 23 1B 30 09 00 00 00 00 ◃ 581 / 60 1B 30 09 00 00 00 00 ◃ T_PDO1 with status word 181 / 37 42	301B:9$_h$ 00$_h$

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

0198441113779, V1.05, 10.2012

6.2.3 Operating mode Profile Torque

Starting the operating mode

The operating mode must be set in the parameter `DCOMopmode`. Writing the parameter value activates the operating mode.

The parameter `PTtq_target` starts the movement.

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

Control word

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

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

Status word

Information on the current movement is available via bits 10 and 12 ... 15 in the parameter `DCOMstatus`.

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

Terminating the operating mode

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

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

0198441113779, V1.05, 10.2012

6.2.3.1 Example Node address 1

Work step COB ID / data	Object Value
▸ NMT Start remote node 0 / 01 00 ◃ T_PDO1 with status word 181 / 31 62	
▸ Enable power stage with R_PDO1 201 / 00 00 201 / 06 00 201 / 0F 00 ◃ T_PDO1 (operating state: **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	6060h 04h
▸ Check operating mode [1)] 601 / 40 61 60 00 00 00 00 00 ◃ Operating mode active 581 / 4F 61 60 00 04 61 01 00	6061h 04h
▸ 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	6071h 64h
▸ Terminate operating mode with "Quick Stop" with R_PDO1 201 / 0B 00 ◃ T_PDO1 with status word 181 / 17 66	
▸ Clear "Quick Stop" with R_PDO1 201 / 0F 00 ◃ T_PDO1 with status word 181 / 37 46	

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

0198441113779, V1.05, 10.2012

6.2.4 Operating mode Profile Velocity

Starting the operating mode

The operating mode must be set in the parameter `DCOMopmode`. Writing the parameter value activates the operating mode.

The parameter `PVv_target` starts the movement.

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

Control word

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

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

Status word

Information on the current movement is available via bits 10 and 12 ... 15 in the parameter `DCOMstatus`.

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

Terminating the operating mode

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

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

0198441113779, V1.05, 10.2012

6.2.4.1 Example Node address 1

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$_h$ 0400 0401$_h$
▸ Activate T_PDO3 601 / 23 02 18 01 81 03 00 04 ◃ 581 / 60 02 18 01 00 00 00 00	1802:1$_h$ 0400 0381$_h$
▸ Set acceleration to 2000 601 / 23 83 60 00 D0 07 00 00 ◃ 581 / 60 83 60 00 00 00 00 00	6083$_h$ 0000 07D0$_h$
▸ NMT Start remote node 0 / 01 00 ◃ T_PDO3 with status word 381 / 31 66 00 00 00 00	
▸ Enable power stage with R_PDO3 401 / 00 00 00 00 00 00 401 / 06 00 00 00 00 00 401 / 0F 00 00 00 00 00 ◃ T_PDO3 (operating state: **6** Operation Enabled) 381 / 37 46 00 00 00 00	
▸ Starting the operating mode 601 / 2F 60 60 00 03 00 00 00 ◃ 581 / 60 60 60 00 00 00 00 00	6060$_h$ 03$_h$
▸ 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$_h$ 03$_h$
▸ R_PDO3: Specification of target velocity 1000 401 / 0F 00 E8 03 00 00 ◃ T_PDO2 with status word and velocity actual value 381 / 37 02 00 00 00 00 ◃ Target velocity reached 381 / 37 06 E8 03 00 00	
▸ Terminate operating mode with "Quick Stop" with R_PDO3 401 / 0B 00 00 00 00 00 ◃ T_PDO3 with status word 381 / 17 66 00 00 00 00	
▸ Clear "Quick Stop" with R_PDO3 401 / 0F 00 00 00 00 00 ◃ T_PDO3 with status word 381 / 37 46 00 00 00 00	

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

0198441113779, V1.05, 10.2012

6.2.5 Operating mode Profile Position

Starting the operating mode

The operating mode must be set in the parameter `DCOMopmode`. Writing the parameter value activates the operating mode.

The movement is started via the control word.

Control word

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

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

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

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

Status word

Information on the current movement is available via bits 10 and 12 ... 15 in the parameter `DCOMstatus`.

Parameter value	Meaning
Bit 10: Target reached	0: Halt = 0: Target position not reached Halt = 1: Motor decelerates 1: Halt = 0: Target position reached Halt = 1: Motor standstill
Bit 12: Target value acknowledge	0: New position possible 1: New target position accepted
Bit 13: x_err	1: Error
Bit 14: x_end	0: Operating mode started 1: Operating mode terminated
Bit 15: ref_ok	1: Zero point is valid

Terminating the operating mode

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

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

0198441113779, V1.05, 10.2012

6.2.5.1 Example Node address 1

Work step COB ID / data	Object Value
▸ Activate R_PDO2 601 / 23 01 14 01 01 03 00 04 ◃ 581 / 60 01 14 01 00 00 00 00	1401:1$_h$ 0400 0301$_h$
▸ Activate T_PDO2 601 / 23 01 18 01 81 02 00 04 ◃ 581 / 60 01 18 01 00 00 00 00	1801:1$_h$ 0400 0281$_h$
▸ Set acceleration to 2000 601 / 23 83 60 00 D0 07 00 00 ◃ 581 / 60 83 60 00 00 00 00 00	6083$_h$ 0000 07D0$_h$
▸ Set deceleration to 4000 601 / 23 84 60 00 A0 0F 00 00 ◃ 581 / 60 84 60 00 00 00 00 00	6084$_h$ 0000 0FA0$_h$
▸ Set target velocity to 4000 601 / 23 81 60 00 A0 0F 00 00 ◃ 581 / 60 81 60 00 00 00 00 00	6081$_h$ 0000 0FA0$_h$
▸ 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$_h$ 01$_h$
▸ 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$_h$ 01$_h$
▸ 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 checked until the device has activated the specified operating mode.

0198441113779, V1.05, 10.2012

6.2.6 Operating mode Interpolated Position

Availability Available as of firmware version ≥V01.08.

Description In the operating mode Interpolated Position, movements are made to cyclically set reference positions.

The monitoring functions Heartbeat and Node Guarding cannot be used in this operating mode.

- Check cyclical reception of PDOs at the PLC in order to detect an interruption of the connection.

The reference positions are transmitted synchronously with each cycle. The cycle time of a cycle can be set from 1 ... 20 ms.

The movement to the reference positions starts with the SYNC signal.

The drive performs an internal fine interpolation with a raster of 250 µs.

The illustration below provides an overview:

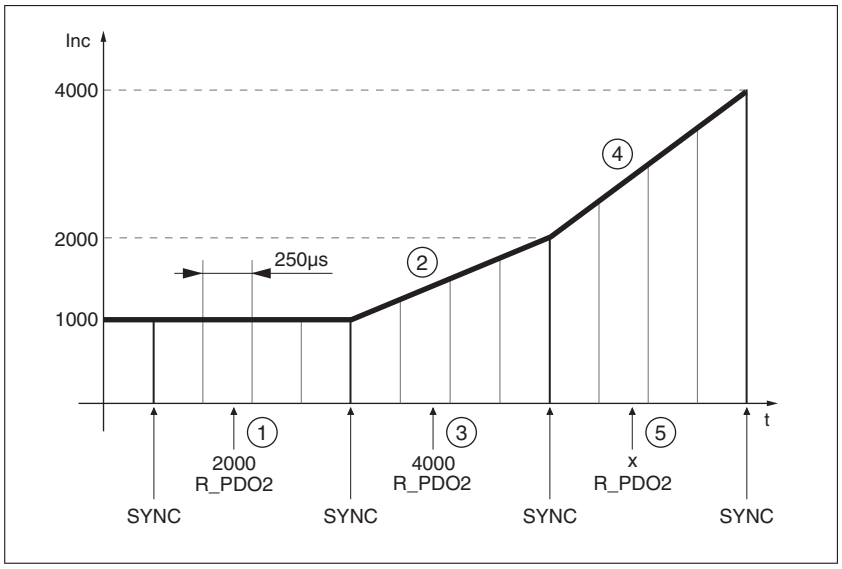


Figure 37: Overview

(1) Transmission of first reference position (example)
(2) Movement to first reference position
(3) Transmission of second reference position (example)
(4) Movement to second reference position
(5) Transmission of next reference position (example)

0198441113779, V1.05, 10.2012

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.

NOTE: In the operating mode Interpolated Position,t eh scaling factor of the user-defined unit usr_p must be set to 1 min^{-1}/131072. Among other things, this scaling factor is written by means of the initialization sequence.

Index	Subindex	Length in bytes	Value	Meaning
1400_h	1_h	4	80000200_h + node id	Deactivate R_PDO1
1800_h	1_h	4	80000180_h + node id	Deactivate T_PDO1
1401_h	1_h	4	00000300_h + node id	Activate R_PDO2
1801_h	1_h	4	00000280_h + node id	Activate T_PDO2
1402_h	1_h	4	80000400_h + node id	Deactivate R_PDO3
1802_h	1_h	4	80000380_h + node id	Deactivate R_PDO3
1403_h	1_h	4	80000500_h + node id	Deactivate R_PDO4
1803_h	1_h	4	80000480_h + node id	Deactivate R_PDO4
1401_h	2	1	1_h	Activate cyclic transmission of R_PDO2
1801_h	2_h	1	1_h	Activate cyclic transmission of T_PDO2
6040_h	0_h	2	0_h	Control word = 0
6040_h	0_h	2	80_h	Perform Fault Reset
1601_h	0_h	1	0_h	Change PDO mapping for R_PDO2
1601_h	1_h	4	60400010_h	Map control word
1601_h	2_h	4	$60C10120_h$	Map reference position for Interpolated Position
1601_h	0_h	1	2_h	Finalize mapping for R_PDO2
$1a01_h$	0_h	1	0_h	Change PDO mapping for T_PDO2
$1a01_h$	1_h	4	60410010_h	Map status word
$1a01_h$	2_h	4	60640020_h	Map Position actual Value
$1a01_h$	0_h	1	2_h	Finalize mapping for T_PDO2
3006_h	7_h	4	20000_h	Position scaling: denominator
3006	8_h	4	1_h	Position scaling: numerator
6060_h	0_h	1	7_h	Select operating mode Interpolated Position
3006	$3D_h$	2	1_h	Must be written for reasons of compatibility
$60C2_h$	1_h	1	2	Cycle time 2 ms (example)
3012_h	6_h	2	$3E8_h$	Velocity feed-forward control 100% CTRL1
3013_h	6_h	2	$3E8_h$	Velocity feed-forward control 100% CTRL2
3006	6_h	2	1_h	Suppress error message for LIMP or LIMN when the power stage is enabled
3022_h	4_h	2	1_h	Tolerance for synchronization mechanism (example)
3022_h	5_h	2	2_h	Activate synchronization mechanism

0198441113779, V1.05, 10.2012

Control word

Parameter value	Meaning
Bit 4: Enable interpolation [1)]	0: Terminate operating mode 1: Start operating mode
Bit 5: Reserved	Not relevant for this operating mode
Bit 6: Reserved	Not relevant for this operating mode
Bit 8: Halt	Stop movement with "Halt"
Bit 9: Change on setpoint	Not relevant for this operating mode

1) 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.

Status word

Information on the current movement is available via bits 10 and 12 ... 15 in the parameter `DCOMstatus`.

Parameter value	Meaning
Bit 10: Target reached	0: Halt = 0: Position not (yet) reached Halt = 1: Motor decelerates 1: Halt = 0: Position reached Halt = 1: Motor standstill
Bit 12: IP mode active	0: Operating mode terminated 1: Operating mode started
Bit 13: x_err	1: Error
Bit 14: x_end	0: Operating mode started 1: Operating mode terminated
Bit 15: ref_ok	1: Zero point is valid

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 an error

0198441113779, V1.05, 10.2012

6.2.6.1 Parameterization

Synchronization mechanism

The synchronization mechanism must be activated for the operating mode Interpolated Position.

The synchronization mechanism is activated via the parameter `SyncMechStart` = 2.

The parameter `SyncMechTol` is used to set a synchronization tolerance. The value of the parameter `SyncMechTol` is internally multiplied by 250 µs. For example, a value of 4 corresponds to a tolerance of 1 ms.

The status of the synchronizations mechanism can be read by means of the parameter `SyncMechStatus`.

- Activate the synchronization mechanism by means of the parameter `SyncMechStart`.

Parameter name HMI menu HMI name	Description	Unit Minimum value Factory setting Maximum value	Data type R/W Persistent Expert	Parameter address via fieldbus
`SyncMechStart`	Activation of synchronization mechanism Value 0: Deactivate synchronization mechanism Value 1: Activate synchronization mechanism (CANmotion). Value 2: Activate synchronization mechanism, standard CANopen mechanism. The cycle time of the synchronization signal is derived from the parameters intTimPerVal and intTimInd. Changed settings become active immediately.	- 0 0 2	UINT16 UINT16 R/W - -	CANopen 3022:5h Modbus 8714
`SyncMechTol`	Synchronization tolerance This parameter is used to increase the synchronization tolerance in the operating mode Interpolated Position. The value is applied when the synchronization mechanism is activated via the parameter SyncMechStart. Changed settings become active immediately. Available with firmware version ≥V01.08.	- 1 1 20	UINT16 UINT16 R/W - -	CANopen 3022:4h Modbus 8712
`SyncMechStatus`	Status of synchronization mechanism Status of synchronization mechanism: Value 1: Synchronization mechanism of drive is inactive. Value 32: Drive is synchronizing with external sync signal. Value 64: Drive is synchronized with external sync signal. Available with firmware version ≥V01.08.	- - - -	UINT16 UINT16 R/- - -	CANopen 3022:6h Modbus 8716

0198441113779, V1.05, 10.2012

Cycle time

The cycle time is set via the parameters `IP_IntTimPerVal` and `IP_IntTimInd`.

The cycle time depends on the following factors:

- Number of drives
- Baud rate
- Time of the minimum data packets per cycle:
  - SYNC
  - R_PDO2, T_PDO2
  - EMCY (This time must be reserved.)
- Optionally the time of the additional data packets per cycle:
  - R_SDO and T_SDO
    The PLC must make sure that the number of requests (R_SDO) and the cycle time match. The response (T_SDO) is transmitted with the next cycle.
  - n_{PDO} - additional R_PDO and T_PDO:
    R_PDO1, T_PDO1, R_PDO3, T_PDO3, R_PDO4 and T_PDO4

The table below shows the typical values for the individual data packets, depending on the baud rate:

Data packets	Size in bytes	1 Mbit	500 kbit	250 kbit
R_PDO2	6	0.114 ms	0.228 ms	0.456 ms
T_PDO2	6	0.114 ms	0.228 ms	0.456 ms
SYNC	0	0.067 ms	0.134 ms	0.268 ms
EMCY	8	0.13 ms	0.26 ms	0.52 ms
R_PDOx	8	0.13 ms	0.26 ms	0.52 ms
T_PDOx	8	0.13 ms	0.26 ms	0.52 ms
R_SDO and T_SDO	16	0.26 ms	0.52 ms	1.040 ms

0198441113779, V1.05, 10.2012

In the case of one drive, the minimum cycle time is calculated as follows: t_{cycle}= SYNC + R_PDO2+ T_PDO2 + EMCY + SDO +n_{PDO}

The following table shows t_{cycle} depending on the baud rate and the number of additional PDOs n_{PDO}, based on one drive:

Number of additional PDOs (n_{PDO})	Minimum cycle time at 1 Mbit	Minimum cycle time at 500 kbit	Minimum cycle time at 250 kbit
0	1 ms	2 ms	3 ms
1	1 ms	2 ms	3 ms
2	1 ms	2 ms	4 ms
3	2 ms	2 ms	4 ms
4	2 ms	3 ms	5 ms
5	2 ms	3 ms	5 ms
6	2 ms	3 ms	6 ms

Cycle time in seconds: `IP_IntTimPerVal` * 10 `IP_IntTimInd`

- Set the desired cycle time with the parameters `IP_IntTimPerVal` and `IP_IntTimInd`.

  Valid cycle times are 1 ... 20 ms in increments of 1 ms.

Parameter name HMI menu HMI name	Description	Unit Minimum value Factory setting Maximum value	Data type R/W Persistent Expert	Parameter address via fieldbus
`IP_IntTimPerVal`	Interpolation time period value Available with firmware version ≥V01.08.	s 0 1 255	UINT8 UINT16 R/W - -	CANopen 60C2:1$_h$ Modbus 7000
`IP_IntTimInd`	Interpolation time index Available with firmware version ≥V01.08.	- -128 -3 63	INT8 INT16 R/W - -	CANopen 60C2:2$_h$ Modbus 7002

Position comparison

The drive cyclically processed the reference position as soon as bit 4 of the control word is set to 1 ne. If the difference between reference position and actual position is too great, this results in a following error. To avoid such an error, the actual position must be read via the parameter `_p_act` before the operating mode is activated or continued. New reference positions must correspond to the actual position in the first cycle.

Parameter name HMI menu HMI name	Description	Unit Minimum value Factory setting Maximum value	Data type R/W Persistent Expert	Parameter address via fieldbus
`_p_act`	Actual position	usr_p - - -	INT32 INT32 R/- - -	CANopen 6064:0$_h$ Modbus 7706

Reference position

The parameter `IPp_target` cyclically transmits a reference value.

0198441113779, V1.05, 10.2012

▶ Set the desired reference value with the parameter `IPp_target`.

Parameter name HMI menu HMI name	Description	Unit Minimum value Factory setting Maximum value	Data type R/W Persistent Expert	Parameter address via field-bus
`IPp_target`	Position reference value for operating mode Interpolated Position Available with firmware version ≥V01.08.	- -2147483648 - 2147483647	INT32 INT32 R/W - -	CANopen 60C1:1$_h$ Modbus 7004

0198441113779, V1.05, 10.2012

6.2.7 Operating mode Homing

Starting the operating mode

The operating mode must be set in the parameter `DCOMopmode`. Writing the parameter value activates the operating mode.

The movement is started via the control word.

The parameter `HMmethod` lets you set the method.

Control word

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

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

Status word

Information on the current movement is available via bits 10 and 12 ... 15 in the parameter `DCOMstatus`.

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

Terminating the operating mode

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

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

0198441113779, V1.05, 10.2012

6.2.7.1 Example Node address 1

Work step COB ID / data	Object Value
▸ Velocity for searching the limit switch to 100 601 / 23 99 60 01 64 00 00 00 ◃ 581 / 60 99 60 01 00 00 00 00	6099:1$_h$ 0000 0064$_h$
▸ 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$_h$ 0000 000A$_h$
▸ NMT Start remote node 0 / 01 00 ◃ T_PDO1 with status word 181 / 31 62	
▸ Enable power stage with R_PDO1 201 / 00 00 201 / 06 00 201 / 0F 00 ◃ T_PDO1 (operating state: **6** Operation Enabled) 181 / 37 42	
▸ Starting the operating mode 601 / 2F 60 60 00 06 00 00 00 ◃ 581 / 60 60 60 00 00 00 00 00	6060$_h$ 06$_h$
▸ 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$_h$ 06$_h$
▸ Select method 17 601 / 2F 98 60 00 11 00 00 00 ◃ 581 / 60 98 60 00 00 00 00 00	6098$_h$ 11$_h$
▸ Start reference movement (Homing operation start) 201 / 1F 00 ◃ T_PDO1 reference movement active 181 / 37 02 ◃ T_PDO1 reference movement terminated 181 / 37 D6	

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

0198441113779, V1.05, 10.2012

7 Diagnostics and troubleshooting

7

7.1 Fieldbus communication error diagnostics

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

Connections for fieldbus operation

If the product cannot be addressed via the fieldbus, first check the connections. Check the following:

- Power connections to the device
- Fieldbus cable and fieldbus wiring

You can also use the commissioning software for troubleshooting.

Baud rate and address

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

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

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

Fieldbus function test

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

- Switch the power supply off and on again.
- Observe the network messages after switching on. After initialization of the bus, the device sends a boot-up message (COB ID 700_h + node ID and 1 data byte with the content 00_h).

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

0198441113779, V1.05, 10.2012

7.2 Fieldbus status LEDs

General The fieldbus status LEDs visualize the status of the fieldbus.

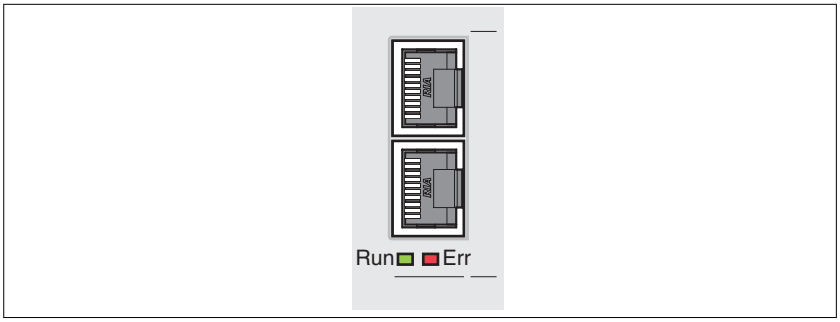


Figure 38: Fieldbus status LEDs

The illustration below shows the fieldbus communication states.

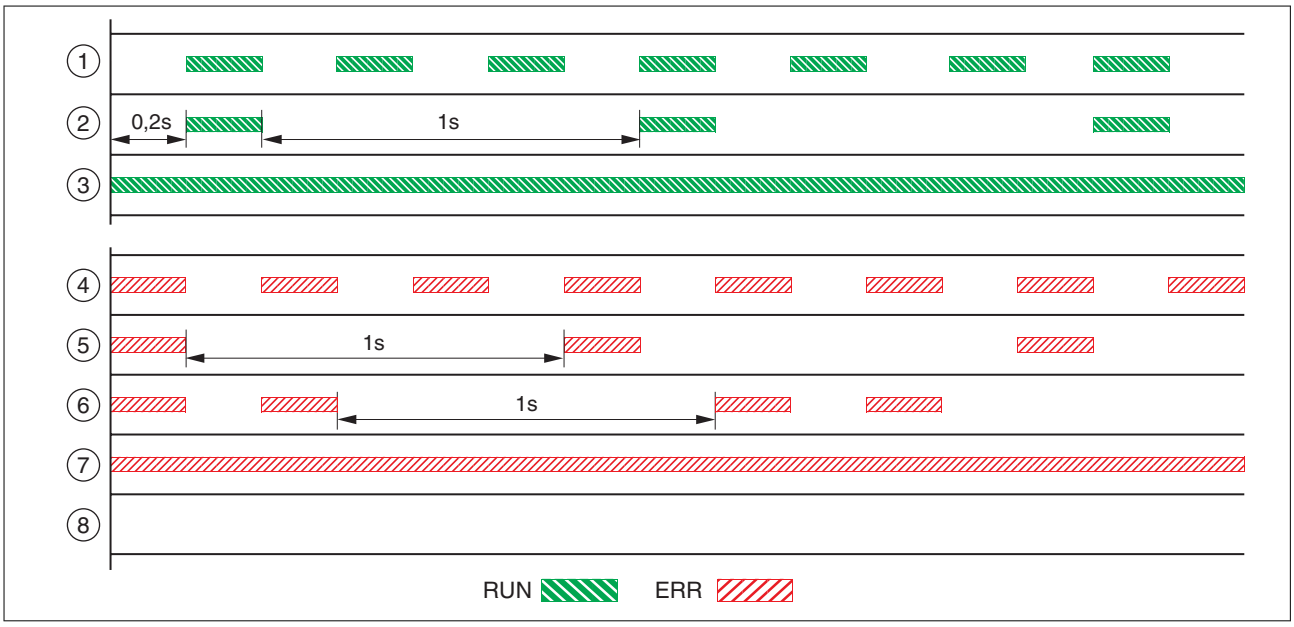


Figure 39: Signals of the CAN bus status LEDs (Run=GN; Err=RD)

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

0198441113779, V1.05, 10.2012

7.3 Error diagnostics via fieldbus

7.3.1 Message objects

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

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

7.3.2 Messages on the device status

Synchronous and asynchronous errors are distinguished in 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. See chapter "7.4.1 Error register " for a list of synchronous errors.

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 (6041h)`. In the case of errors that cause an interruption of the movement, the device transmits an EMCY message.

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

0198441113779, V1.05, 10.2012

7.4 CANopen error messages

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

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

7.4.1 Error register

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

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

7.4.2 Error code table

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

0198441113779, V1.05, 10.2012

7.4.3 SDO error message ABORT

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

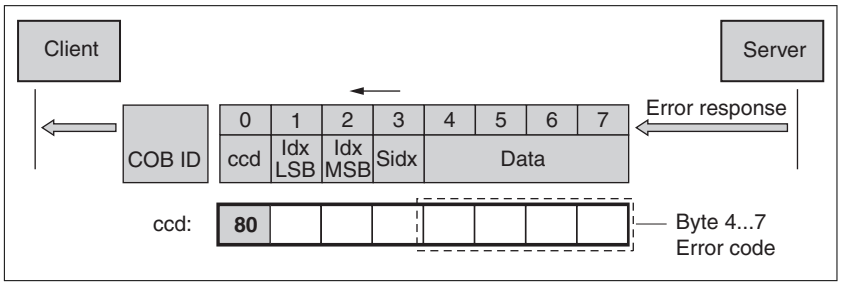


Figure 40: SDO error message as a response to an SDO message

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

0198441113779, V1.05, 10.2012

Error code	Meaning
0503 0000$_h$	Toggle bit not toggled
0504 0000$_h$	Time-out during SDO transfer
0504 0001$_h$	Command specifier CS incorrect or unknown
0504 0005$_h$	No memory available
0601 0000$_h$	Access to object not possible
0601 0001$_h$	No read access, because write-only object (wo)
0601 0002$_h$	No write access, because read object (ro)
0602 0000$_h$	Object does not exist in object dictionary
0604 0041$_h$	Object does not support PDO mapping
0604 0042$_h$	PDO mapping: Number or length of objects exceed the byte length of the PDO
0604 0043$_h$	Parameters are incompatible
0604 0047$_h$	Device detects internal incompatibility
0606 0000$_h$	Hardware error, access denied
0607 0010$_h$	Data type and parameter length do not match
0607 0012$_h$	Data type does not match, parameter too long
0607 0013$_h$	Data type does not match, parameter too short
0609 0011$_h$	Subindex not supported
0609 0030$_h$	Value range of parameter too large (relevant only for write access)
0609 0031$_h$	Parameter values too great
0609 0032$_h$	Parameter values too small
0609 0036$_h$	Upper value is less than lower value
0800 0000$_h$	General error. See parameter `_ManuSdoAbort` at the bottom of this table.
0800 0020$_h$	Data can neither be transmitted to the application nor saved.
0800 0021$_h$	Local control mode, data can neither be transmitted nor saved.
0800 0022$_h$	Data can neither be transmitted nor saved in this device state.
0800 0023$_h$	Object dictionary does not exist or cannot be generated (for example, if data error occurs during generation from file)
0800 0024$_h$	Data not available.

Parameter name HMI menu HMI name	Description	Unit Minimum value Factory setting Maximum value	Data type R/W Persistent Expert	Parameter address via field-bus
`_ManuSdoAbort`	CANopen Manufacturer-specific SDO Abort Code Provides more detailed information on a general SDO Abort Code (0800 0000).	- - - -	UINT16 UINT16 R/- - -	CANopen 3041:A$_h$ Modbus 16660

0198441113779, V1.05, 10.2012

8 Object dictionary

8

8.1 Specifications for the objects

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

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

Object code	Meaning	Coding
VAR	A simple value, for example of the type Integer8, Unsigned32 or Visible String8.	7
ARR (ARRAY)	A data field in which 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 Factory settings when the product is shipped

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

0198441113779, V1.05, 10.2012

8.2 Overview of object group 1000_h

Index	Sub-index	Name	Object code	Data type	Access	PDO	Description	Page
1000_h		Device type	VAR	Unsigned32	RO		Device type and profile	101
1001_h		Error register	VAR	Unsigned8	RO		Error register	101
1003_h		Predefined error field	ARR		RW		Error history, memory for error messages	102
1003_h	00_h	Number of errors	VAR	Unsigned8	RW		Number of error entries	102
1003_h	01_h	Error field	VAR	Unsigned32	RO		Error number	102
1005_h		COB-ID SYNC	VAR	Unsigned32	RW		Identifier of the synchronization object	103
1008_h		Manufacturer device name	VAR	Visible String8	RO		Manufacturer's designation	104
1009_h		Manufacturer hardware version	VAR	Visible String8	RO		Hardware version	104
$100A_h$		Manufacturer software version	VAR	Visible String8	RO		Software version	104
$100C_h$		Guard time	VAR	Unsigned16	RW		Time span for Node Guarding [ms]	105
$100D_h$		Life time factor	VAR	Unsigned8	RW		Repeat factor for the Node Guarding protocol	105
1014_h		COB-ID EMCY	VAR	Unsigned32	RW		Unsigned16	106
1015_h		Inhibit time EMCY	VAR	Unsigned16	RW		Unsigned16	107
1016_h		Consumer Heartbeat Time	ARR	Unsigned32	RW		Unsigned16	107
1016_h	01_h	Consumer Heartbeat Time	VAR	Unsigned32	RW		Time interval and node ID of the "Heartbeat" recipient	107
1017_h		Producer Heartbeat Time	VAR	Unsigned16	RW		Time interval for producer "Heartbeat"	108
1018_h		Identity Object	REC	Identity	RO		Identification object:	109
1018_h	01_h	Vendor ID	VAR	Unsigned32	RO		Vendor ID	109
1018_h	02_h	Product code	VAR	Unsigned32	RO		Product code	109
1018_h	03_h	Revision number	VAR	Unsigned32	RO		Revision number	109
1029_h		Number of elements	ARR	Unsigned8	RO		Number of values for the object	111
1029_h	01_h	Communication error	ARR	Unsigned8	RW		Communication error	111
1200_h		1st server SDO parameter	REC	SDO server param.	RO		First server SDO, settings	111
1200_h	01_h	COB-ID Client -> Server	VAR	Unsigned32	RO		Identifier client -> server	111
1200_h	02_h	COB-ID Server -> Client	VAR	Unsigned32	RO		Identifier server -> client	111
1201_h		2nd server SDO parameter	REC	SDO server param.	RW		Second server SDO, settings	112
1201_h	01_h	COB-ID Client -> Server	VAR	Unsigned32	RW		Identifier client -> server	112
1201_h	02_h	COB-ID Server -> Client	VAR	Unsigned32	RW		Identifier server -> client	112
1201_h	03_h	Node-ID SDO Client	VAR	Unsigned32	RW		Node ID SDO client	112
1400_h		1st receive PDO parameter	REC	PDO comm. param.	RW		First receive PDO (R_PDO1), settings	113

0198441113779, V1.05, 10.2012

Index	Sub-index	Name	Object code	Data type	Access	PDO	Description	Page
1400h	01h	COB-ID R_PDO1	VAR	Unsigned32	RW		Identifier of the R_PDO1	113
1400h	02h	Transmission type R_PDO1	VAR	Unsigned8	RW		Transmission type	113
1401h		2nd receive PDO parameter	REC	PDO comm. param.	RW		Second receive PDO (R_PDO2), settings	115
1401h	01h	COB-ID R_PDO2	VAR	Unsigned32	RW		Identifier of the R_PDO2	115
1401h	02h	Transmission type R_PDO2	VAR	Unsigned8	RW		Transmission type	115
1402h		3rd receive PDO parameter	REC	PDO comm. param.	RW		Third receive PDO (R_PDO3), settings	116
1402h	01h	COB-ID R_PDO3	VAR	Unsigned32	RW		Identifier of the R_PDO3	116
1402h	02h	Transmission type R_PDO3	VAR	Unsigned8	RW		Transmission type	116
1403h		4th receive PDO parameter	REC	PDO comm. param.	RW		Fourth receive PDO (R_PDO4), settings	118
1403h	01h	COB-ID R_PDO4	VAR	Unsigned32	RW		Identifier of the R_PDO4	118
1403h	02h	Transmission type R_PDO4	VAR	Unsigned8	RW		Transmission type	118
1600h		1st receive PDO mapping	REC	PDO mapping	RO		PDO mapping for R_PDO1, settings	118
1600h	01h	1st mapped object R_PDO1	VAR	Unsigned32	RO		First object for the mapping in R_PDO1	118
1601h		2nd receive PDO mapping	REC	PDO mapping	RO		PDO mapping for R_PDO2, settings	120
1601h	01h	1st mapped object R_PDO2	VAR	Unsigned32	RO		First object for the mapping in R_PDO2	120
1601h	02h	2nd mapped object R_PDO2	VAR	Unsigned32	RO		Second object for the mapping in R_PDO2	120
1602h		3rd receive PDO mapping	REC	PDO mapping	RO		PDO mapping for R_PDO3, settings	122
1602h	01h	1st mapped object R_PDO3	VAR	Unsigned32	RO		First object for the mapping in R_PDO3	122
1602h	02h	2nd mapped object R_PDO3	VAR	Unsigned32	RO		Second object for the mapping in R_PDO3	122
1603h		4th receive PDO mapping	REC	PDO mapping	RW		PDO mapping for R_PDO3, settings	123
1603h	01h	1st mapped object R_PDO4	VAR	Unsigned32	RW		First object for the mapping in R_PDO4	123
1603h	02h	2nd mapped object R_PDO4	VAR	Unsigned32	RW		Second object for the mapping in R_PDO4	123
1603h	03h	3rd mapped object R_PDO4	VAR	Unsigned32	RW		Third object for mapping in R_PDO4	123
1800h		1st transmit PDO parameter	REC	PDO comm. param.	RW		First transmit PDO (T_PDO1), settings	124
1800h	01h	COB-ID T_PDO1	VAR	Unsigned32	RW		Identifier of the T_PDO1	124
1800h	02h	Transmission type T_PDO1	VAR	Unsigned8	RW		Transmission type	124
1800h	03h	Inhibit time T_PDO1	VAR	Unsigned16	RW		Inhibit time for locking bus access (1=100µs)	124

0198441113779, V1.05, 10.2012

Index	Sub-index	Name	Object code	Data type	Access	PDO	Description	Page
1800h	04h	Reserved T_PDO1	VAR	Unsigned8	RW		Priority for CAN bus arbitration ([0-7]).	124
1800h	05h	Event timer T_PDO1	VAR	Unsigned16	RW		Time span for event triggering (1=1 ms)	124
1801h		2nd transmit PDO parameter	REC	PDO comm. param.	RW		Second transmit PDO (T_PDO2), settings	126
1801h	01h	COB-ID T_PDO2	VAR	Unsigned32	RW		Identifier of the T_PDO2	126
1801h	02h	Transmission type T_PDO2	VAR	Unsigned8	RW		Transmission type	126
1801h	03h	Inhibit time T_PDO2	VAR	Unsigned16	RW		Inhibit time for locking bus access (1=100µs)	126
1801h	04h	Reserved T_PDO2	VAR	Unsigned8	RW		Reserved	126
1801h	05h	Event timer T_PDO2	VAR	Unsigned16	RW		Time span for event triggering (1=1 ms)	126
1802h		3rd transmit PDO parameter	REC	PDO comm. param.	RW		Third transmit PDO (T_PDO3), settings	128
1802h	01h	COB-ID T_PDO3	VAR	Unsigned32	RW		Identifier of the T_PDO3	128
1802h	02h	Transmission type T_PDO3	VAR	Unsigned8	RW		Transmission type	128
1802h	03h	Inhibit time T_PDO3	VAR	Unsigned16	RW		Inhibit time for locking bus access (1=100µs)	128
1802h	04h	Reserved T_PDO3	VAR	Unsigned8	RW		Reserved	128
1802h	05h	Event timer T_PDO3	VAR	Unsigned16	RW		Time span for event triggering (1=1 ms)	128
1803h		4th transmit PDO parameter	REC	PDO comm. param.	RW		Fourth transmit PDO (T_PDO4), settings	129
1803h	01h	COB-ID T_PDO4	VAR	Unsigned32	RW		Identifier of the T_PDO4	129
1803h	02h	Transmission type T_PDO4	VAR	Unsigned8	RW		Transmission type	129
1803h	03h	Inhibit time T_PDO4	VAR	Unsigned16	RW		Inhibit time for locking bus access (1=100µs)	129
1803h	04h	Reserved T_PDO4	VAR	Unsigned8	RO		Reserved	129
1803h	05h	Event timer T_PDO4	VAR	Unsigned16	RW		Time span for event triggering (1=1 ms)	129
1A00h		1st transmit PDO mapping	REC	PDO mapping	RW		PDO mapping for T_PDO1, settings	131
1A00h	01h	1st mapped object T_PDO1	VAR	Unsigned32	RO		First object for the mapping in T_PDO1	131
1A01h		2nd transmit PDO mapping	REC	PDO mapping	RW		PDO mapping for T_PDO2, settings	132
1A01h	01h	1st mapped object T_PDO2	VAR	Unsigned32	RO		First object for the mapping in T_PDO2	132
1A01h	02h	2nd mapped object T_PDO2	VAR	Unsigned32	RO		Second object for the mapping in T_PDO2	132
1A02h		3rd transmit PDO mapping	REC	PDO mapping	RW		PDO mapping for T_PDO3, settings	134
1A02h	01h	1st mapped object T_PDO3	VAR	Unsigned32	RO		First object for the mapping in T_PDO3	134

0198441113779, V1.05, 10.2012

Index	Sub-index	Name	Object code	Data type	Access	PDO	Description	Page
1A02h	02h	2nd mapped object T_PDO3	VAR	Unsigned32	RO		Second object for the mapping in T_PDO3	134
1A03h		4th transmit PDO mapping	REC	PDO mapping	RW		PDO mapping for T_PDO4, settings	135
1A03h	01h	1st mapped object T_PDO4	VAR	Unsigned32	RW		First object for the mapping in T_PDO4	135
1A03h	02h	2nd mapped object T_PDO4	VAR	Unsigned32	RW		Second object for the mapping in T_PDO4	135
1A03h	03h	3rd mapped object T_PDO4	VAR	Unsigned32	RW		Third object for the mapping in T_PDO4	135
1A03h	04h	4th mapped object T_PDO4	VAR	Unsigned32	RW		Fourth object for the mapping in T_PDO4	135

0198441113779, V1.05, 10.2012

8.3 Assignment object group 3000_h

The product provides corresponding parameters for the CANopen object group 3000_h. A detailed description of the parameters can be found in the product manual in the Parameters chapter.

0198441113779, V1.05, 10.2012

Index	Subindex	Object	PDO	Data type	Parameter name	Takes effect
3001_h	01_h	Firmware program number	-	UINT32	_prgNoDEV	-
3001_h	02_h	Firmware version number	-	UINT16	_prgVerDEV	-
3001_h	04_h	Firmware revision number	-	UINT16	_prgRevDEV	-
3001_h	$0C_h$	Current access channel	-	UINT16	_AccessInfo	-
3001_h	$0E_h$	Locking other access channels	-	UINT16	AccessLock	Immediately
3004_h	07_h	Reset controller parameters	-	UINT16	PAR_CTRLreset	Immediately
3004_h	08_h	Reset user parameters	-	UINT16	PARuserReset	Enabling the power stage
3005_h	05_h	Commutation monitoring	-	UINT16	MON_commutat	Enabling the power stage
3005_h	06_h	Enabling the power stage at PowerOn	-	UINT16	IO_AutoEnable	Enabling the power stage
3005_h	07_h	Additional time delay for releasing the holding brake	-	INT16	BRK_AddT_release	Enabling the power stage
3005_h	08_h	Additional time delay for applying the holding brake	-	INT16	BRK_AddT_apply	Enabling the power stage
3005_h	09_h	Selection of internal or external braking resistor	-	UINT16	RESint_ext	Enabling the power stage
3005_h	$0A_h$	Error response to missing mains phase of three-phase devices	-	UINT16	ErrorResp_Flt_AC	Enabling the power stage
3005_h	$0B_h$	Error response to following error	-	UINT16	ErrorResp_p_dif	Enabling the power stage
3005_h	$0F_h$	Detection and monitoring of mains phases	-	UINT16	MON_MainsVolt	Enabling the power stage
3005_h	10_h	Ground fault monitoring	-	UINT16	MON_GroundFault	Start-up
3005_h	11_h	Max. permissible switch-on time of external braking resistor	-	UINT16	RESext_ton	Enabling the power stage
3005_h	12_h	Nominal power of external braking resistor	-	UINT16	RESext_P	Enabling the power stage
3005_h	13_h	Resistance value of external braking resistor	-	UINT16	RESext_R	Enabling the power stage
3005_h	20_h	Mains reactor	-	UINT16	Mains_reactor	Immediately
3005_h	21_h	Shifting of the encoder working range	-	UINT16	ShiftEncWorkRang	Start-up
3005_h	22_h	Error response to 100% I2t braking resistor	-	UINT16	ErrorResp_I2tRES	Enabling the power stage
3006_h	01_h	Acceleration and deceleration of the motion profile for velocity	R_PDO	UINT16	RAMP_v_sym	Next movement
3006_h	03_h	Monitoring of software limit switches	-	UINT16	MON_SW_Limits	Immediately
3006_h	07_h	Position scaling: Denominator	-	INT32	ScalePOSdenom	Parameters
3006_h	08_h	Position scaling: Numerator	-	INT32	ScalePOSnum	Immediately
3006_h	$0C_h$	Inversion of direction of movement	-	UINT16	InvertDirOfMove	Start-up
3006_h	$0D_h$	Jerk limitation of the motion profile for velocityv	-	UINT16	RAMP_v_jerk	Next movement
3006_h	$0E_h$	Signal evaluation for reference switch	-	UINT16	IOsigREF	Enabling the power stage

0198441113779, V1.05, 10.2012

Index	Subindex	Object	PDO	Data type	Parameter name	Takes effect
3006h	0Fh	Signal evaluation for negative limit switch	-	UINT16	IOsigLIMN	Enabling the power stage
3006h	10h	Signal evaluation for positive limit switch	-	UINT16	IOsigLIMP	Enabling the power stage
3006h	12h	Deceleration ramp for Quick Stop	-	UINT32	RAMPquickstop	Next movement
3006h	16h	Absolute positioning only after homing	-	UINT16	AbsHomeRequest	Immediately
3006h	18h	Quick Stop option code	-	INT16	LIM_QStopReact	Immediately
3006h	19h	Monitoring of position deviation	-	UINT16	MON_p_DiffWin	Immediately
3006h	1Ah	Monitoring of velocity deviation	-	UINT32	MON_v_DiffWin	Immediately
3006h	1Bh	Monitoring of velocity threshold	-	UINT32	MON_v_Threshold	Immediately
3006h	1Ch	Monitoring of current threshold	-	UINT16	MON_I_Threshold	Immediately
3006h	1Dh	Monitoring of time window	-	UINT16	MON_ChkTime	Immediately
3006h	1Eh	Velocity limitation via input	-	UINT32	IO_v_limit	Immediately
3006h	21h	Velocity scaling: Denominator	-	INT32	ScaleVELdenom	Parameters
3006h	22h	Velocity scaling: Numerator	-	INT32	ScaleVELnum	Immediately
3006h	26h	Timeout time for standstill window monitoring	-	UINT16	MON_p_winTout	Immediately
3006h	27h	Current limitation via input	-	UINT16	IO_I_limit	Immediately
3006h	28h	Velocity limit for Zero Clamp	-	UINT32	MON_v_zeroclamp	Immediately
3006h	29h	Maximum load-dependent position deviation (warning)	-	UINT16	MON_p_dif_warn	Immediately
3006h	2Dh	Torque window, permissible deviation	-	UINT16	MON_tq_win	Immediately
3006h	2Eh	Torque window, time	-	UINT16	MON_tq_winTime	Immediately
3006h	30h	Ramp scaling: Denominator	-	INT32	ScaleRAMPdenom	Parameters
3006h	31h	Ramp scaling: Numerator	-	INT32	ScaleRAMPnum	Immediately
3007h	01h	Function Input DI0	-	UINT16	IOfunct_DI0	Start-up
3007h	02h	Function Input DI1	-	UINT16	IOfunct_DI1	Start-up
3007h	03h	Function Input DI2	-	UINT16	IOfunct_DI2	Start-up
3007h	04h	Function Input DI3	-	UINT16	IOfunct_DI3	Start-up
3007h	09h	Function Output DQ0	-	UINT16	IOfunct_DQ0	Start-up
3007h	0Ah	Function Output DQ1	-	UINT16	IOfunct_DQ1	Start-up
3008h	01h	Physical status of the digital inputs and outputs	T_PDO	UINT16	_IO_act	-
3008h	0Fh	Status of digital inputs	T_PDO	UINT16	_IO_DI_act	-
3008h	10h	Status of digital outputs	T_PDO	UINT16	_IO_DQ_act	-
3008h	11h	Setting the digital outputs directly	R_PDO	UINT16	IO_DQ_set	-
3008h	20h	Debounce time of DI0	-	UINT16	DI_0_Debounce	Immediately
3008h	21h	Debounce time of DI1	-	UINT16	DI_1_Debounce	Immediately
3008h	22h	Debounce time of DI2	-	UINT16	DI_2_Debounce	Immediately

0198441113779, V1.05, 10.2012

Index	Subindex	Object	PDO	Data type	Parameter name	Takes effect
3008h	23h	Debounce time of DI3	-	UINT16	DI_3_Debounce	Immediately
3008h	26h	Status of the inputs for the safety function STO	-	UINT16	_IO_STO_act	-
300Ah	01h	Status of the capture inputs	-	UINT16	_CapStatus	-
300Ah	02h	Capture input 1 configuration	-	UINT16	Cap1Config	Immediately
300Ah	04h	Capture input 1 start/stop	-	UINT16	Cap1Activate	Immediately
300Ah	06h	Capture input 1 captured position	T_PDO	UINT32	_Cap1Pos	-
300Ah	08h	Capture input 1 event counter	T_PDO	UINT16	_Cap1Count	-
300Dh	02h	Motor type	-	UINT32	_M_Type	-
300Dh	03h	Motor encoder type	-	UINT16	_M_Sensor	-
300Dh	04h	Maximum permissible motor speed of rotation	-	UINT16	_M_n_max	-
300Dh	05h	Nominal motor speed of rotation	-	UINT16	_M_n_nom	-
300Dh	06h	Maximum motor current	-	UINT16	_M_I_max	-
300Dh	07h	Nominal motor current	-	UINT16	_M_I_nom	-
300Dh	08h	Nominal motor torque	-	UINT16	_M_M_nom	-
300Dh	09h	Maximum motor torque	-	UINT16	_M_M_max	-
300Dh	0Ah	Nominal motor voltage	-	UINT16	_M_U_nom	-
300Dh	0Bh	Motor EMF constant kE	-	UINT32	_M_kE	-
300Dh	0Ch	Moment of inertia of motor	-	UINT32	_M_Jrot	-
300Dh	0Dh	Motor winding resistance	-	UINT16	_M_R_UV	-
300Dh	0Eh	Motor inductance q component	-	UINT16	_M_L_q	-
300Dh	0Fh	Motor inductance d component	-	UINT16	_M_L_d	-
300Dh	10h	Maximum motor temperature	-	INT16	_M_T_max	-
300Dh	11h	Maximum permissible time for maximum motor current	-	UINT16	_M_I2t	-
300Dh	13h	Continuous stall current of motor	-	UINT16	_M_I_0	-
300Dh	14h	Number of pole pairs of motor	-	UINT16	_M_Polepair	-
3010h	01h	Nominal current of power stage	-	UINT16	_PS_I_nom	-
3010h	02h	Maximum current of power stage	-	UINT16	_PS_I_max	-
3010h	03h	Maximum permissible DC bus voltage	-	UINT16	_PS_U_maxDC	-
3010h	04h	Minimum permissible DC bus voltage	-	UINT16	_PS_U_minDC	-
3010h	06h	Temperature warning threshold of power stage	-	INT16	_PS_T_warn	-
3010h	07h	Maximum power stage temperature	-	INT16	_PS_T_max	-
3010h	08h	Resistance value of internal braking resistor	-	UINT16	_RESint_R	-
3010h	09h	Nominal power of internal braking resistor	-	UINT16	_RESint_P	-
3010h	0Ah	DC bus voltage low threshold for Quick Stop	-	UINT16	_PS_U_minStopDC	-

0198441113779, V1.05, 10.2012

Index	Subindex	Object	PDO	Data type	Parameter name	Takes effect
3011_h	01_h	Current controller d component P gain	-	UINT16	_CTRL_KPid	Immediately
3011_h	02_h	Current controller d component integral action time	-	UINT16	_CTRL_TNid	Immediately
3011_h	03_h	Current controller q component P gain	-	UINT16	_CTRL_KPiq	Immediately
3011_h	04_h	Current controller q component integral action time	-	UINT16	_CTRL_TNiq	Immediately
3011_h	05_h	PID velocity controller time constant of D gain	-	UINT16	CTRL_nPIDTime	Immediately
3011_h	06_h	PID velocity controller amplification of D gain	-	UINT16	CTRL_nPIDDPart	Immediately
3011_h	08_h	Filter time constant to smooth velocity of motor	-	UINT16	CTRL_TAUnact	Immediately
3011_h	09_h	Speed of rotation up to which the friction compensation is linear	-	UINT32	CTRL_SpdFric	Immediately
3011_h	$0A_h$	Gain acceleration feed forward	-	UINT16	CTRL_KFAcc	Immediately
3011_h	$0C_h$	Current limitation	-	UINT16	CTRL_I_max	Immediately
3011_h	$0D_h$	Current value for Quick Stop	-	UINT16	LIM_I_maxQSTP	Immediately
3011_h	$0E_h$	Current value for Halt	-	UINT16	LIM_I_maxHalt	Immediately
3011_h	$0F_h$	Maximum field current of field weakness controller	-	UINT16	CTRL_I_max_fw	Enabling the power stage
3011_h	10_h	Velocity limitation	-	UINT32	CTRL_v_max	Immediately
3011_h	14_h	Period of time for parameter switching	-	UINT16	CTRL_ParChgTime	Immediately
3011_h	15_h	Global gain factor (affects parameter set 1)	-	UINT16	CTRL_GlobGain	Immediately
3011_h	16_h	Controller parameter set copying	-	UINT16	CTRL_ParSetCopy	Immediately
3011_h	17_h	Active controller parameter set	-	UINT16	_CTRL_ActParSet	-
3011_h	18_h	Selection of controller parameter set at power u	-	UINT16	CTRL_PwrUpParSet	Immediately
3011_h	19_h	Selection of controller parameter set (non-persistent)	-	UINT16	CTRL_SelParSet	Immediately
3011_h	$1A_h$	Condition for parameter set switching	-	UINT16	CLSET_ParSwiCond	Immediately
3011_h	$1B_h$	Time window for parameter set switching	-	UINT16	CLSET_winTime	Immediately
3011_h	$1C_h$	Position deviation for parameter set switching	-	UINT16	CLSET_p_DiffWin	Immediately
3011_h	$1D_h$	Velocity threshold for parameter set switching	-	UINT32	CLSET_v_Threshol	Immediately
3012_h	01_h	Velocity controller P gain	-	UINT16	CTRL1_KPn	Immediately
3012_h	02_h	Velocity controller integral action time	-	UINT16	CTRL1_TNn	Immediately
3012_h	03_h	Position controller P gain	-	UINT16	CTRL1_KPp	Immediately
3012_h	04_h	Filter time constant of the reference speed value filter	-	UINT16	CTRL1_TAUnref	Immediately

0198441113779, V1.05, 10.2012

Index	Subindex	Object	PDO	Data type	Parameter name	Takes effect
3012h	05h	Filter time constant of the reference current value filter	-	UINT16	CTRL1_TAUiref	Immediately
3012h	06h	Velocity feed-forward	-	UINT16	CTRL1_KFPp	Immediately
3012h	08h	Notch filter 1: Damping	-	UINT16	CTRL1_Nf1damp	Immediately
3012h	09h	Notch filter 1: Frequency	-	UINT16	CTRL1_Nf1freq	Immediately
3012h	0Ah	Notch filter 1: Bandwidth	-	UINT16	CTRL1_Nf1bandw	Immediately
3012h	0Bh	Notch filter 2: Damping	-	UINT16	CTRL1_Nf2damp	Immediately
3012h	0Ch	Notch filter 2: Frequency	-	UINT16	CTRL1_Nf2freq	Immediately
3012h	0Dh	Notch filter 2: Bandwidth	-	UINT16	CTRL1_Nf2bandw	Immediately
3012h	0Eh	Overshoot suppression filter: Damping	-	UINT16	CTRL1_Osupdamp	Immediately
3012h	0Fh	Overshoot suppression filter: Time delay	-	UINT16	CTRL1_Osupdelay	Immediately
3012h	10h	Friction compensation: Gain	-	UINT16	CTRL1_Kfric	Immediately
3013h	01h	Velocity controller P gain	-	UINT16	CTRL2_KPn	Immediately
3013h	02h	Velocity controller integral action time	-	UINT16	CTRL2_TNn	Immediately
3013h	03h	Position controller P gain	-	UINT16	CTRL2_KPp	Immediately
3013h	04h	Filter time constant of the reference speed value filter	-	UINT16	CTRL2_TAUnref	Immediately
3013h	05h	Filter time constant of the reference current value filter	-	UINT16	CTRL2_TAUiref	Immediately
3013h	06h	Velocity feed-forward	-	UINT16	CTRL2_KFPp	Immediately
3013h	08h	Notch filter 1: Damping	-	UINT16	CTRL2_Nf1damp	Immediately
3013h	09h	Notch filter 1: Frequency	-	UINT16	CTRL2_Nf1freq	Immediately
3013h	0Ah	Notch filter 1: Bandwidth	-	UINT16	CTRL2_Nf1bandw	Immediately
3013h	0Bh	Notch filter 2: Damping	-	UINT16	CTRL2_Nf2damp	Immediately
3013h	0Ch	Notch filter 2: Frequency	-	UINT16	CTRL2_Nf2freq	Immediately
3013h	0Dh	Notch filter 2: Bandwidth	-	UINT16	CTRL2_Nf2bandw	Immediately
3013h	0Eh	Overshoot suppression filter: Damping	-	UINT16	CTRL2_Osupdamp	Immediately
3013h	0Fh	Overshoot suppression filter: Time delay	-	UINT16	CTRL2_Osupdelay	Immediately
3013h	10h	Friction compensation: Gain	-	UINT16	CTRL2_Kfric	Immediately
301Bh	06h	Error response to data error (DE bit)	-	INT16	ErrorResp_bit_DE	-
301Bh	07h	Error response to mode error (ME bit)	-	INT16	ErrorResp_bit_ME	-
301Bh	09h	Activation of operating mode Jog	R_PDO	UINT16	JOGactivate	Immediately
301Bh	13h	DS402 state machine: State transition from 3 to 4	-	UINT16	DS402compatib	Immediately
301Bh	16h	Position for Position Setting	-	INT32	HMp_setP	Immediately
301Bh	19h	Error code for synchronous errors (ME bit)	-	UINT16	_ModeError	-
301Bh	1Bh	Error code for synchronous errors (DE bit)	-	UINT16	_DataError	-

0198441113779, V1.05, 10.2012

Index	Subindex	Object	PDO	Data type	Parameter name	Takes effect
301B$_h$	1C$_h$	Additional error information of a ModeError (ME bit)	-	UINT16	_ModeErrorInfo	-
301B$_h$	1D$_h$	Additional error information of a DataError (DE bit)	-	UINT16	_DataErrorInfo	-
301B$_h$	1E$_h$	DS402 statusword: Selection for internal limit bit	-	UINT16	DS402intLim	Immediately
301C$_h$	04$_h$	Action word	T_PDO	UINT16	_actionStatus	-
301C$_h$	06$_h$	Modbus address of parameter with invalid value	-	UINT16	_InvalidParam	-
301C$_h$	07$_h$	Current status of monitoring signals	-	UINT32	_SigActive	-
301C$_h$	08$_h$	Saved status of monitoring signals	-	UINT32	_SigLatched	-
301C$_h$	09$_h$	Number of last warning (error class 0)	-	UINT16	_LastWarning	-
301C$_h$	0A$_h$	Operating hours counter	-	UINT32	_OpHours	-
301C$_h$	0B$_h$	Active warnings, bit-coded	-	UINT32	_WarnActive	-
301C$_h$	0C$_h$	Saved warnings, bit-coded	-	UINT32	_WarnLatched	-
301C$_h$	0D$_h$	Current output power	-	INT16	_Power_act	-
301C$_h$	0E$_h$	Mean output power	-	INT16	_Power_mean	-
301C$_h$	0F$_h$	Voltage at DC bus	-	UINT16	_UDC_act	-
301C$_h$	10$_h$	Current power stage temperature	-	INT16	_PS_T_current	-
301C$_h$	11$_h$	Current motor temperature	-	INT16	_M_T_current	-
301C$_h$	12$_h$	Current device temperature	-	INT16	_DEV_T_current	-
301C$_h$	13$_h$	Current overload of braking resistor (I2t)	-	INT16	_RES_overload	-
301C$_h$	14$_h$	Current load of braking resistor	-	INT16	_RES_load	-
301C$_h$	15$_h$	Maximum value of overload of braking resistor	-	INT16	_RES_maxoverload	-
301C$_h$	16$_h$	Current overload of power stage (I2t)	-	INT16	_PS_overload_I2t	-
301C$_h$	17$_h$	Current load of power stage	-	INT16	_PS_load	-
301C$_h$	18$_h$	Maximum value of overload of power stage	-	INT16	_PS_maxoverload	-
301C$_h$	19$_h$	Current overload of motor (I2t)	-	INT16	_M_overload	-
301C$_h$	1A$_h$	Current load of motor	-	INT16	_M_load	-
301C$_h$	1B$_h$	Maximum value of overload of motor	-	INT16	_M_maxoverload	-
301C$_h$	1E$_h$	Maximum possible value for operating mode Profile Torque	-	INT16	_PT_max_val	-
301C$_h$	1F$_h$	Aditional info of last error	-	UINT16	_LastError_Qual	-
301C$_h$	22$_h$	Current overload of power stage (chip temperature)	-	INT16	_PS_overload_cte	-
301C$_h$	23$_h$	Current overload of power stage (power squared)	-	INT16	_PS_overload_psq	-
301C$_h$	24$_h$	Current overload of power stage	-	INT16	_PS_overload	-

0198441113779, V1.05, 10.2012

Index	Subindex	Object	PDO	Data type	Parameter name	Takes effect
301C$_h$	27$_h$	Current limitation of the system	-	UINT16	_Imax_system	-
301C$_h$	28$_h$	Actual current limit	-	UINT16	_Imax_act	-
301C$_h$	29$_h$	Actual velocity limit	-	UINT32	_Vmax_act	-
301E$_h$	01$_h$	Actual motor current (q component, generating torque)	-	INT16	_Iq_act_rms	-
301E$_h$	02$_h$	Actual motor current (d component, field weakening)	-	INT16	_Id_act_rms	-
301E$_h$	03$_h$	Total motor current	T_PDO	INT16	_I_act	-
301E$_h$	04$_h$	Reference motor voltage q component	-	INT16	_Uq_ref	-
301E$_h$	05$_h$	Reference motor voltage d component	-	INT16	_Ud_ref	-
301E$_h$	06$_h$	Total motor voltage (vector sum d components and q components)	-	INT16	_Udq_ref	-
301E$_h$	07$_h$	Reference speed of rotation of velocity controller	-	INT16	_n_ref	-
301E$_h$	08$_h$	Actual speed of rotation of motor	-	INT16	_n_act	-
301E$_h$	09$_h$	Reference position in internal units	-	INT32	_p_refInt	-
301E$_h$	0C$_h$	Reference position	-	INT32	_p_ref	-
301E$_h$	0E$_h$	Absolute position with reference to one motor revolution	-	UINT32	_p_absmodulo	-
301E$_h$	0F$_h$	Motor position with reference to encoder range	-	UINT32	_p_absENC	-
301E$_h$	10$_h$	Reference motor current (q component, generating torque)	-	INT16	_Iq_ref_rms	-
301E$_h$	11$_h$	Reference motor current (d component, field weakening)	-	INT16	_Id_ref_rms	-
301E$_h$	13$_h$	Degree of utilization of DC bus voltage	-	INT16	_VoltUtil	-
301E$_h$	1B$_h$	Maximum value of the load-dependent position deviation	-	UINT32	_p_dif_load_peak	Immediately
301E$_h$	1C$_h$	Current load-dependent position deviation between reference and actual position	-	INT32	_p_dif_load	-
301E$_h$	1F$_h$	Reference velocity of velocity controller	-	INT32	_v_ref	-
301F$_h$	01$_h$	Target position of profile generator	-	INT32	_RAMP_p_target	-
301F$_h$	02$_h$	Actual position of profile generator	T_PDO	INT32	_RAMP_p_act	-
301F$_h$	05$_h$	Target velocity of profile generator	-	INT32	_RAMP_v_target	-
301F$_h$	07$_h$	Velocity of reference value for profile generator	-	INT32	_pref_v	-
301F$_h$	09$_h$	Acceleration of reference value for profile generator	-	INT32	_pref_acc	-
301F$_h$	0A$_h$	Maximum possible value of position scaling	-	INT32	_ScalePOSmaxusr	-

0198441113779, V1.05, 10.2012

Index	Subindex	Object	PDO	Data type	Parameter name	Takes effect
301F$_h$	0B$_h$	Maximum possible value of velocity scaling	-	INT32	_ScaleVELmaxusr	-
301F$_h$	0C$_h$	Maximum possible value of ramp scaling	-	INT32	_ScaleRAMPmaxusr	-
3028$_h$	06$_h$	Maximum distance for search for switching point	-	INT32	HMoutdis	Next movement
3028$_h$	07$_h$	Distance from switching point	-	INT32	HMdis	Next movement
3028$_h$	0A$_h$	Preferred homing method	-	INT16	HMprefmethod	Immediately
3028$_h$	0B$_h$	Position at reference point	-	INT32	HMp_home	Next movement
3028$_h$	0C$_h$	Distance from switching point to index pulse	-	INT32	_HMdisREFtoIDX	-
3028$_h$	0D$_h$	Maximum search distance after overtravel of switch	-	INT32	HMsrchdis	Next movement
3029$_h$	03$_h$	Selection of jog method	-	UINT16	JOGmethod	Immediately
3029$_h$	04$_h$	Velocity for slow movement	-	UINT32	JOGv_slow	Immediately
3029$_h$	05$_h$	Velocity for fast movement	-	UINT32	JOGv_fast	Immediately
3029$_h$	07$_h$	Distance for step movement	-	INT32	JOGstep	Next movement
3029$_h$	08$_h$	Wait time for step movement	-	UINT16	JOGtime	Next movement
302F$_h$	01$_h$	Autotuning start	-	UINT16	AT_start	Immediately
302F$_h$	02$_h$	Autotuning status	-	UINT16	_AT_state	-
302F$_h$	03$_h$	Movement range for Autotuning	-	UINT32	AT_dis	Next movement
302F$_h$	04$_h$	Direction of movement for Autotuning	-	UINT16	AT_dir	Next movement
302F$_h$	06$_h$	Jump of speed of rotation for Autotuning	-	UINT32	AT_n_ref	Next movement
302F$_h$	07$_h$	Friction torque of the system	-	UINT16	_AT_M_friction	-
302F$_h$	08$_h$	Constant load torque	-	INT16	_AT_M_load	-
302F$_h$	09$_h$	Waiting time between Autotuning steps	-	UINT16	AT_wait	Next movement
302F$_h$	0B$_h$	Progress of Autotuning	-	UINT16	_AT_progress	-
302F$_h$	0C$_h$	Moment of inertia of the complete system	-	UINT16	_AT_J	Parameters
302F$_h$	0E$_h$	Type of coupling of the system	-	UINT16	AT_mechanical	Next movement
303A$_h$	01$_h$	Lock HMI	-	UINT16	HMIlocked	Immediately
303A$_h$	02$_h$	HMI display when motor moves	-	UINT16	HMIDispPara	Immediately
303B$_h$	02$_h$	Number of power on cycles	-	UINT32	_ERR_powerOn	-
303B$_h$	04$_h$	Clear error memory	-	UINT16	ERR_clear	Immediately
303B$_h$	05$_h$	Reset error memory read pointer	-	UINT16	ERR_reset	Immediately
303B$_h$	06$_h$	First number for the signal output function Selected Error	-	UINT16	MON_IO_SelErr1	Immediately
303B$_h$	07$_h$	Second number for the signal output function Selected Error	-	UINT16	MON_IO_SelErr2	Immediately
303B$_h$	08$_h$	First number for the signal output function Selected Warning	-	UINT16	MON_IO_SelWar1	Immediately

0198441113779, V1.05, 10.2012

Index	Subindex	Object	PDO	Data type	Parameter name	Takes effect
303B$_h$	09$_h$	Second number for the signal output function Selected Warning	-	UINT16	MON_IO_SelWar2	Immediately
303C$_h$	01$_h$	Error number	-	UINT16	_ERR_number	-
303C$_h$	02$_h$	Error class	-	UINT16	_ERR_class	-
303C$_h$	03$_h$	Error time	-	UINT32	_ERR_time	-
303C$_h$	04$_h$	Error additional information	-	UINT16	_ERR_qual	-
303C$_h$	05$_h$	Number of cycles of enabling the power stage at error time	-	UINT16	_ERR_enable_cycl	-
303C$_h$	06$_h$	Time between enabling of power stage and occurrence of the error	-	UINT16	_ERR_enable_time	-
303C$_h$	07$_h$	DC bus voltage at error time	-	UINT16	_ERR_DCbus	-
303C$_h$	08$_h$	Motor velocity at error time	-	INT32	_ERR_motor_v	-
303C$_h$	09$_h$	Motor current at error time	-	UINT16	_ERR_motor_I	-
303C$_h$	0A$_h$	Temperature of power stage at error time	-	INT16	_ERR_temp_ps	-
303C$_h$	0B$_h$	Temperature of device at error time	-	INT16	_ERR_temp_dev	-
3041$_h$	06$_h$	CANopen diagnosis word	-	UINT16	_CanDiag	-
3041$_h$	0A$_h$	CANopen Manufacturer-specific SDO Abort Code	-	UINT16	_ManuSdoAbort	-
3041$_h$	0B$_h$	PDO 1 event mask	-	UINT16	CANpdo1Event	Immediately
3041$_h$	0C$_h$	PDO 2 event mask	-	UINT16	CANpdo2Event	Immediately
3041$_h$	0D$_h$	PDO 3 event mask	-	UINT16	CANpdo3Event	Immediately
3041$_h$	0E$_h$	PDO 4 event mask	-	UINT16	CANpdo4Event	Immediately

0198441113779, V1.05, 10.2012

8.4 Assignment object group 6000_h

The product provides corresponding parameters for the CANopen object group 6000_h. A detailed description of the parameters can be found in the product manual in the Parameters chapter.

0198441113779, V1.05, 10.2012

Index	Subindex	Object	PDO	Data type	Parameter name	Takes effect
603F$_h$	00$_h$	Error Code	-	UINT16	_LastError	-
6040$_h$	00$_h$	Controlword	R_PDO	UINT16	DCOMcontrol	Immediately
6041$_h$	00$_h$	Statusword	T_PDO	UINT16	_DCOMstatus	-
605D$_h$	00$_h$	Halt option code	-	INT16	LIM_HaltReaction	Immediately
6060$_h$	00$_h$	Modes of Operation	R_PDO	INT8	DCOMopmode	Immediately
6061$_h$	00$_h$	Modes of Operation Display	T_PDO	INT8	_DCOMopmd_act	-
6063$_h$	00$_h$	Position actual Value	T_PDO	INT32	_p_actInt	-
6064$_h$	00$_h$	Position actual Value	T_PDO	INT32	_p_act	-
6065$_h$	00$_h$	Following error window	-	UINT32	MON_p_dif_load	Immediately
6067$_h$	00$_h$	Position Window	-	UINT32	MON_p_win	Immediately
6068$_h$	00$_h$	Position window time	-	UINT16	MON_p_winTime	Immediately
606B$_h$	00$_h$	Velocity demand value	-	INT32	_RAMP_v_act	-
606C$_h$	00$_h$	Velocity actual value	T_PDO	INT32	_v_act	-
606D$_h$	00$_h$	Velocity Window	-	UINT16	MON_v_win	Immediately
606E$_h$	00$_h$	Velocity Window Time	-	UINT16	MON_v_winTime	Immediately
6071$_h$	00$_h$	Target Torque	R_PDO	INT16	PTtq_target	Immediately
6077$_h$	00$_h$	Torque actual Value	T_PDO	INT16	_tq_act	-
607A$_h$	00$_h$	Target position	R_PDO	INT32	PPp_target	Immediately
607D$_h$	01$_h$	negative Software position limit	-	INT32	MON_swLimN	Enabling the power stage
607D$_h$	02$_h$	positive Software position limit	-	INT32	MON_swLimP	Enabling the power stage
607F$_h$	00$_h$	Maximal profile velocity	-	UINT32	RAMP_v_max	Next movement
6081$_h$	00$_h$	Profile velocity	R_PDO	UINT32	PPv_target	Next movement
6083$_h$	00$_h$	Profile acceleration	R_PDO	UINT32	RAMP_v_acc	Next movement
6084$_h$	00$_h$	Profile deceleration	R_PDO	UINT32	RAMP_v_dec	Next movement
6086$_h$	00$_h$	Motion profile type	-	UINT16	ProfileType	
6087$_h$	00$_h$	Torque slope	R_PDO	UNIT32	RAMP_tq_slope	Immediately
6098$_h$	00$_h$	Homing method	-	INT8	HMmethod	Immediately
6099$_h$	01$_h$	Homing speed search	-	UINT32	HMv	Next movement
6099$_h$	02$_h$	Homing speed moving away	-	UINT32	HMv_out	Next movement
60C1$_h$	01$_h$	Interpolation data record	-	UINT8	-	-
60C2$_h$	01$_h$	Interpolation Time Period value	-	UINT32	cycletime	-
60C2$_h$	02$_h$	Interpolation Time Period index	-	INT8	timedimension	-
60C4$_h$	01$_h$	Interpolation Position Buffer Size	-	INT8	-	-
60C4$_h$	02$_h$	Interpolation Position Actual Buffer Size	-	INT8	-	-
60C4$_h$	03$_h$	Interpolation Position Buffer Organisation	-	INT8	-	-
60C4$_h$	03$_h$	Interpolation Position Buffer Position	-	INT8	-	-
60C4$_h$	05$_h$	Interpolation Position Size of Data Record	-	INT8	-	-

0198441113779, V1.05, 10.2012

Index	Subindex	Object	PDO	Data type	Parameter name	Takes effect
60C4h	06h	Interpolation Position Buffer Clear	-	INT8	-	-
60F2h	00h	Position Option Code	-	UINT16	PPoption	Next movement
60F4h	00h	Following error actual value	-	INT32	_p_dif	-
60FFh	00h	Target velocity	R_PDO	INT32	PVv_target	Immediately
6502h	00h	Supported Drive Modes	-	UINT32	_SuppDriveModes	-

0198441113779, V1.05, 10.2012

8.5 Details of object group 1000h

8.5.1 1000_h Device type

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

Object description

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

Value description

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

Bit coding, subindex 00_h

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

8.5.2 1001_h Error register

The object specifies the error of the device. The detailed cause of error can be determined with the object `Predefined error field (1003h)` and - for reasons of compatibility with devices with other fieldbus profiles - with the object `Error code (603Fh)`.

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

Object description

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

0198441113779, V1.05, 10.2012

Value description

Subindex	00_h, error register
Meaning	Error register
Access	RO
PDO mapping	–
Value range	–
Default value	–
Can be saved	–

Bit coding, subindex 00_h

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

8.5.3 1003_h Predefined error field

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

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

Object description

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

Value description

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

0198441113779, V1.05, 10.2012

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

Bit coding, subindex 00_h ... 05_h

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

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

8.5.4 1005_h 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_h
Object name	COB ID SYNC
Object code	VAR
Data type	Unsigned32

Value description

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

Bit coding, subindex 00_h

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_h	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	$7F_h$	Node address, bit 6 ... 0 of COB ID

0198441113779, V1.05, 10.2012

8.5.5 1008_h Manufacturer device name

The object specifies the device name of the manufacturer.

Object description

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

Value description

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

The following objects contain additional information on the device:-
Objects 6404_h, 6410_h: Motor data

8.5.6 1009_h Manufacturer hardware version

The object specifies the version of the device hardware.

Object description

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

Value description

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

8.5.7 $100A_h$ Manufacturer software version

The object specifies the version of the device software.

0198441113779, V1.05, 10.2012

Object description

Index	$100A_h$
Object name	Manufacturer software version
Object code	VAR
Data type	Visible String8

Value description

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

8.5.8 $100C_h$ 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`$_h$`)`.

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

Object description

Index	$100C_h$
Object name	Guard Time
Object code	VAR
Data type	Unsigned16

Value description

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

8.5.9 $100D_h$ 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

0198441113779, V1.05, 10.2012

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 (100Ch)`.

Object description

Index	$100D_h$
Object name	Life Time Factor
Object code	VAR
Data type	Unsigned8

Value description

Subindex	00_h, 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	–

8.5.10 1014_h COB ID Emergency object message

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

Object description

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

Value description

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

0198441113779, V1.05, 10.2012

Bit coding, subindex 00_h

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_h	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"

8.5.11 1015_h 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_h
Object name	Inhibit time EMCY
Object code	VAR
Data type	Unsigned16

Value description

Subindex	00_h, 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	–

8.5.12 1016_h Consumer Heartbeat Time

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

Object description

Index	1016_h
Object name	Consumer Heartbeat Time
Object code	ARRAY
Data type	Unsigned32

0198441113779, V1.05, 10.2012

Value description

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

Subindex	01_h, 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 coding subindex 01_h ... 03_h

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

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

8.5.13 1017_h Producer Heartbeat Time

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

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

Object description

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

0198441113779, V1.05, 10.2012

Value description

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

8.5.14 1018_h Identity Object

The object provides information on the product.

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

Object description

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

Value description

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

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

0198441113779, V1.05, 10.2012

Subindex	02_h, product code
Meaning	Product code
Access	RO
PDO mapping	–
Value range	–
Default value	–
Can be saved	–

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

0198441113779, V1.05, 10.2012

8.5.15 1029_h error behavior

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

Object description

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

Value description

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

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

Settings, subindex 01_h

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

8.5.16 1200_h 1st server SDO parameter

The object contains the settings for the first server SDO.

Object description

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

0198441113779, V1.05, 10.2012

Value description

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

Subindex	01_h, 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_h, COB ID server -> client
Meaning	Identifier server -> client
Access	RO
PDO mapping	–
Value range	0...4294967295
Default value	1408 + node ID
Can be saved	–

8.5.17 1201_h 2nd server SDO parameter

The object contains the settings for the second server SDO.

Object description

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

Value description

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

0198441113779, V1.05, 10.2012

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

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

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

8.5.18 1400_h 1st receive PDO parameter

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

Object description

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

Value description

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

0198441113779, V1.05, 10.2012

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

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

Bit assignment subindex 01_h

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_h	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

Bit 31 A R_PDO can only be used if bit 31="0".

Bit coding, subindex 02_h The type of control for evaluating R_PDO data is specified via subindex 02_h. The values 241 ... 251 are reserved.

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

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

0198441113779, V1.05, 10.2012

- 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 (1600h)` . The following default assignment is used for R_PDO1:

- Bytes 0 ... 1: Control word `controlword (6040h)`.

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

8.5.19 1401_h 2nd receive PDO parameter

The object contains settings for the second receive PDO R_PDO2.

Object description

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

Value description

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

0198441113779, V1.05, 10.2012

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

Subindex	02_h, 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_h).

Settings

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

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

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

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

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

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

8.5.20 1402_h 3rd receive PDO parameter

The object contains settings for the third receive PDO R_PDO3.

Object description

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

0198441113779, V1.05, 10.2012

Value description

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

Subindex	01_h, COB ID used by PDO
Meaning	Identifier of the R_PDO3
Access	RW
PDO mapping	–
Value range	0 ... 4294967295
Default value	8000 0400_h + node ID
Can be saved	–

Subindex	02_h, 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 (1400h)`.

Settings

R_PDO3 is processed synchronously, acyclically and in an event-driven way and must be activated with bit 31=1 in subindex 01_h before it can be used.

The byte assignment of the R_PDO3 is specified via PDO mapping with the object `3rd Receive PDO mapping (1602h)`. The following default assignment is set for the operating mode "Profile Velocity":

- Bytes 0 ... 1: Control word `controlword (6040h)`
- Bytes 2 ... 5: Reference velocity of motion command `Target velocity (60FFh)`

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

0198441113779, V1.05, 10.2012

8.5.21 1403_h 4th receive PDO parameter

The object stores settings for the fourth receive PDO R_PDO4.

Object description

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

Value description

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

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

Subindex	02_h, 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 (1400h)`.

PDO settings

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

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

8.5.22 1600_h 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_h, the number of mapped objects is read.

0198441113779, V1.05, 10.2012

Object description

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

Value description

Subindex	00_h, 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_h, CMD: Control word
Meaning	First object for mapping
Access	RW
PDO mapping	–
Value range	0...4294967295
Default value	6040 0010_h
Can be saved	–

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

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

0198441113779, V1.05, 10.2012

Subindex	04_h
Meaning	Fourth object for mapping
Access	RW
PDO mapping	–
Value range	0...4294967295
Default value	–
Can be saved	–

Bit coding starting at subindex 01_h

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

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

Settings

The following default assignment is used:

- Subindex 01_h: `controlword (6040h)`

8.5.23 1601_h 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_h, the number of mapped objects is read.

Object description

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

Value description

Subindex	00_h, 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	–

0198441113779, V1.05, 10.2012

Subindex	01_h, 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_h
Can be saved	–

Subindex	02_h, 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_h
Can be saved	–

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

Subindex	04_h
Meaning	Fourth object for mapping
Access	RW
PDO mapping	–
Value range	0...4294967295
Default value	–
Can be saved	–

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

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

- Subindex 01_h: `controlword (6040h)`
- Subindex 02_h: `target position (607Ah)`

0198441113779, V1.05, 10.2012

8.5.24 1602_h 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_h, the number of mapped objects is read.

Object description

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

Value description

Subindex	00_h, 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_h, 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_h
Can be saved	–

Subindex	02_h, 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_h
Can be saved	–

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

0198441113779, V1.05, 10.2012

Subindex	04_h
Meaning	Fourth object for mapping
Access	RW
PDO mapping	–
Value range	0...4294967295
Default value	–
Can be saved	–

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

Settings

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

- Subindex 01_h: `controlword (6040h)`
- Subindex 02_h: `target velocity (60FFh)`

8.5.25 1603_h 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_h, the number of mapped objects is read.

Object description

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

Value description

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

0198441113779, V1.05, 10.2012

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

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

Subindex	04_h
Meaning	Fourth object for mapping
Access	RW
PDO mapping	–
Value range	0...4294967295
Default value	–
Can be saved	–

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

Settings The PDO assignment for R_PDO4 can be modified.

8.5.26 1800_h 1st transmit PDO parameter

The object contains settings for the first transmit PDO T_PDO1.

Object description

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

0198441113779, V1.05, 10.2012

Value description

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

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

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

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

0198441113779, V1.05, 10.2012

Subindex	05_h, 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 (1400h)`.

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 (1A00h)`. The following default assignment is used:

- Bytes 0 ... 1: Status word `statusword (6041h)`.

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

8.5.27 1801_h 2nd transmit PDO parameter

The object contains settings for the second transmit PDO T_PDO2.

Object description

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

Value description

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

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

0198441113779, V1.05, 10.2012

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

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

Subindex	05_h, 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 (1400h)`.

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 (1A01h)`. The following default assignment is set for the operating mode "Profile Position":

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

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

0198441113779, V1.05, 10.2012

8.5.28 1802_h 3rd transmit PDO parameter

The object contains settings for the third transmit PDO T_PDO3.

Object description

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

Value description

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

Subindex	01_h, COB ID used by PDO
Meaning	Identifier of the T_PDO3
Access	RW
PDO mapping	–
Value range	0...4294967295
Default value	C000 0380_h + node ID
Can be saved	–

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

Subindex	03_h, 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	–

0198441113779, V1.05, 10.2012

Subindex	04_h, reserved
Meaning	Reserved
Access	–
PDO mapping	–
Value range	0...255
Default value	–
Can be saved	–

Subindex	05_h, 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 (1400h)`.

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 (1A02h)`. The following default assignment is set for the operating mode "Profile Position":

- Bytes 0 ... 1: Status word `statusword (6041h)`
- Bytes 2 ... 5: Actual velocity `velocity actual value (606Ch)`.

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

8.5.29 1803_h 4th transmit PDO parameter

The object contains settings for the fourth transmit PDO T_PDO4.

Object description

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

0198441113779, V1.05, 10.2012

Value description

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

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

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

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

0198441113779, V1.05, 10.2012

Subindex	05_h, 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 (1400h)`.

Settings

R_PDO4 is transmitted asynchronously and in an event-driven way.

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

8.5.30 $1A00_h$ 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_h, the number of mapped objects is read.

Object description

Index	$1A00_h$
Object name	1st transmit PDO mapping
Object code	RECORD
Data type	PDO mapping

Value description

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

0198441113779, V1.05, 10.2012

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

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

Subindex	04_h
Meaning	Fourth object for mapping
Access	RW
PDO mapping	–
Value range	0...4294967295
Default value	–
Can be saved	–

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

Settings

The following default assignment is used:

- Subindex 01_h: `statusword (6041h)`

8.5.31 1A01h 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_h, the number of mapped objects is read.

Object description

Index	$1A01_h$
Object name	2nd transmit PDO mapping
Object code	RECORD
Data type	PDO mapping

0198441113779, V1.05, 10.2012

Value description

Subindex	00_h, 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_h, 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_h
Can be saved	–

Subindex	02_h, 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_h
Can be saved	–

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

Subindex	04_h
Meaning	Fourth object for mapping
Access	RW
PDO mapping	–
Value range	0...4294967295
Default value	–
Can be saved	–

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

0198441113779, V1.05, 10.2012

Settings

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

- Subindex 01_h: `statusword (6041h)`
- Subindex 02_h: `position actual value (6064h)`

8.5.32 $1A02_h$ 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_h, the number of mapped objects is read.

Object description

Index	$1A02_h$
Object name	3rd transmit PDO mapping
Object code	RECORD
Data type	PDO mapping

Value description

Subindex	00_h, 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_h, 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_h
Can be saved	–

Subindex	02_h, 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_h
Can be saved	–

0198441113779, V1.05, 10.2012

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

Subindex	04_h
Meaning	Fourth object for mapping
Access	RW
PDO mapping	–
Value range	0...4294967295
Default value	–
Can be saved	–

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

Settings

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

- Subindex 01_h: `statusword (6041h)`
- Subindex 02_h: `velocity actual value (606Ch)`

8.5.33 $1A03_h$ 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_h, the number of mapped objects is read.

Object description

Index	$1A03_h$
Object name	4th transmit PDO mapping
Object code	RECORD
Data type	PDO mapping

Value description

Subindex	00_h, number of elements
Meaning	Number of values for the object
Access	RW
PDO mapping	–
Value range	0 ... 4
Default value	0
Can be saved	–

0198441113779, V1.05, 10.2012

Subindex	01_h
Meaning	First object for mapping
Access	RW
PDO mapping	–
Value range	0...4294967295
Default value	–
Can be saved	–

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

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

Subindex	04_h
Meaning	Fourth object for mapping
Access	RW
PDO mapping	–
Value range	0...4294967295
Default value	–
Can be saved	–

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

Settings The PDO assignment for T_PDO4 can be modified.

0198441113779, V1.05, 10.2012

9 Accessories and spare parts

9

9.1 CANopen cable with connectors

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

0198441113779, V1.05, 10.2012

9.2 CANopen connectors, distributors, terminating resistors

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

9.3 CANopen cables

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

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

0198441113779, V1.05, 10.2012

10 Glossary

10

10.1 Units and conversion tables

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

Example: conversion of 5 meters [m] to yards [yd]
5 m / 0.9144 = 5.468 yd

10.1.1 Length

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

10.1.2 Mass

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

10.1.3 Force

	lb	oz	p	N
lb	-	* 16	* 453.55358	* 4.448222
oz	/ 16	-	* 28.349524	* 0.27801
p	/ 453.55358	/ 28.349524	-	* $9.807*10^{-3}$
N	/ 4.448222	/ 0.27801	/ $9.807*10^{-3}$	-

10.1.4 Power

	HP	W
HP	-	* 746
W	/ 746	-

0198441113779, V1.05, 10.2012

10.1.5 Rotation

	min^{-1} (RPM)	**rad/s**	**deg./s**
min^{-1} (RPM)	-	* π / 30	* 6
rad/s	* 30 / π	-	* 57.295
deg./s	/ 6	/ 57.295	-

10.1.6 Torque

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

10.1.7 Moment of inertia

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

10.1.8 Temperature

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

10.1.9 Conductor cross section

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

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

0198441113779, V1.05, 10.2012

10.2 Terms and Abbreviations

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

AC	Alternating current
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
COB	**C**ommunication **OB**ject, transport unit in a CAN network.
COB ID	**C**ommunication **OB**ject **ID**entifier; uniquely identifies each communication object in a CAN network
CiA	**C**AN **i**n **A**utomation, CAN interest group, standardization group for CAN and CANopen.
DC	Direct current
DOM	**D**ate **o**f **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
DS301	Standardizes the CANopen communication profile
DSP402	Standardizes the CANopen device profile for drives
DriveCom	Specification of the DSP402 state machine was created in accordance with the DriveCom specification.
E	Encoder
EDS	(**E**lectronic **D**ata **S**heet); contains the specific properties of a product.
EMC	Electromagnetic compatibility
EMCY object	Emergency Object
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.
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	Discrepancy between a detected (computed, measured or signaled) value or condition and the specified or theoretically correct value or condition.
Error class	Classification of errors into groups. The different error classes allow for specific responses to errors, for example by severity.
Factory setting	Factory settings when the product is shipped
Fatal error	In the case of fatal error, the product is no longer able to control the motor so that the power stage must be immediately disabled.

0198441113779, V1.05, 10.2012

Fault	Fault is a state that can be caused by an error. Further information can be found in the pertinent standards such as IEC 61800-7, ODVA Common Industrial Protocol (CIP).
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.
HMI	Human Machine Interface
Heartbeat	Used for unconfirmed connection acknowledgement messages from network devices.
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.
Life guarding	For monitoring the connection of an NMT master
Limit switch	Switches that signal overtravel of the permissible range of travel.
Mapping	Assignment of object dictionary entries to PDOs
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 ID	Node address assigned to a device on the network.
Node guarding	Monitoring of the connection to the slave at an interface for cyclic data traffic.
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).
PDO	Process Data Object
Parameter	Device data and values that can be read and set (to a certain extent) by the user.
Persistent	Indicates whether the value of the parameter remains in the memory after the device is switched off.
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.
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_PDO	Receive PDO
SDO	Service Data Object
SYNC object	Synchronization object
T_PDO	Transmit PDO
Warning	If the term is used outside the context of safety instructions, a warning alerts to a potential problem that was detected by a monitoring function. A warning does not cause a transition of the operating state.

0198441113779, V1.05, 10.2012

11 Index

11

0198441113779, V1.05, 10.2012

D

E

0198441113779, V1.05, 10.2012

0198441113779, V1.05, 10.2012

N

O

P

0198441113779, V1.05, 10.2012

0198441113779, V1.05, 10.2012

U

V

W

0198441113779, V1.05, 10.2012