

# Communication / Function Manual

Motion Control

MCBL 300x CO  
MCDC 300x CO  
3564...B CO  
32xx...BX4 CO  
22xx...BX4 COD

**CAN**open®

## Imprint

---

Version:  
2nd edition, 9-11-2018

Copyright  
by Dr. Fritz Faulhaber GmbH & Co. KG  
Daimlerstr. 23 / 25 · 71101 Schönaich

All rights reserved, including those to the translation.  
No part of this description may be duplicated, reproduced,  
stored in an information system or processed or  
transferred in any other form without prior express written  
permission of Dr. Fritz Faulhaber GmbH & Co. KG.

This document has been prepared with care.  
Dr. Fritz Faulhaber GmbH & Co. KG cannot accept any  
liability for any errors in this document or for the  
consequences of such errors. Equally, no liability can be  
accepted for direct or consequential damages resulting  
from improper use of the equipment.

The relevant regulations regarding safety engineering  
and interference suppression as well as the requirements  
specified in this document are to be noted and followed  
when using the software.

Subject to change without notice.

The respective current version of this technical manual is  
available on FAULHABER's internet site:  
[www.faulhaber.com](http://www.faulhaber.com)

# Content

<b>1</b>	<b>About this document</b>	<b>6</b>
1.1	Validity of this document	6
1.2	Associated documents	6
1.3	Using this document	7
1.4	List of abbreviations	7
1.5	Symbols and markers	7
<b>2</b>	<b>Quick start</b>	<b>8</b>
2.1	Start with unconfigured controller	8
2.2	Setting node number and baud rate	9
2.3	Operation using FAULHABER Motion Manager	10
2.4	Operation using own host application	13
2.4.1	Activate CANopen nodes	13
2.4.2	Configure the drives	13
2.4.3	Operation in one of the CANopen CiA 402 drive profiles	14
<b>3</b>	<b>CANopen protocol description</b>	<b>15</b>
3.1	Introduction	15
3.2	Communication services	16
3.3	Identifier distribution	17
3.4	PDO (Process Data Object)	18
3.4.1	PDO configuration	19
3.4.2	PDO mapping in the standard configuration (status as delivered)	20
3.4.3	Dealing with mapping errors	21
3.5	SDO (Service Data Object)	22
3.5.1	Expedited transfer	22
3.5.2	SDO error description	24
3.6	Emergency object (error message)	25
3.7	SYNC object	27
3.7.1	Triggering synchronous PDOs	27
3.8	NMT (Network Management)	28
3.8.1	Boot up	30
3.8.2	Monitoring functions	30
3.8.2.1	Node guarding	30
3.8.2.2	Heartbeat	32
3.8.3	Settings for the monitoring functions	33
3.9	Entries in the object dictionary	33
3.10	Error handling	34
3.10.1	CAN error	34
3.10.2	Equipment faults	34

# Content

<b>4</b>	<b>Functional description .....</b>	<b>36</b>
4.1	Drive data .....	36
4.2	Device Control .....	37
4.2.1	State machine of the drive .....	37
4.2.2	Controlword .....	40
4.2.2.1	Example: Enable Operation .....	41
4.2.2.2	Example: Resetting the fault state .....	42
4.2.3	Quick Stop .....	42
4.2.4	Statusword .....	43
4.2.5	Selecting the operating mode .....	45
4.3	Factor Group .....	46
4.4	Profile Position Mode .....	49
4.4.1	Overview of operating mode .....	50
4.4.2	Notification of the higher level control .....	50
4.4.3	Basic settings .....	50
4.4.4	Query current values / Position Control Function .....	51
4.4.5	Additional settings.....	52
4.4.5.1	Incremental encoder as position sensor .....	52
4.4.5.2	Ramp generator .....	52
4.4.5.3	Velocity controller / current limitation .....	52
4.4.5.4	Motion control commands.....	53
4.5	Homing Mode .....	55
4.5.1	Homing methods .....	55
4.5.2	Homing reference run .....	62
4.6	Profile Velocity Mode .....	63
4.6.1	Overview of operating mode .....	63
4.6.2	Notification of the higher level control .....	63
4.6.3	Basic settings .....	64
4.6.4	Actual velocity value .....	64
4.6.5	Additional settings.....	64
4.6.5.1	Movement limits .....	64
4.6.5.2	Ramp generator .....	64
4.6.5.3	Current limitation .....	64
4.6.6	Motion control commands .....	65
4.6.7	Complex motion profiles .....	65
4.7	Cyclic Synchronous Position Mode .....	65
4.7.1	Overview of operating mode .....	65
4.7.2	Inputs and outputs.....	66
4.7.3	Statusword and Controlword .....	66
4.7.4	Settings .....	67
4.8	Inputs/outputs .....	68
4.8.1	Limit switch connections and switching level .....	68
4.8.1.1	Digital input configuration .....	68
4.8.2	Special functions of the fault pin.....	70
4.8.2.1	Fault pin as error output .....	70
4.8.2.2	Fault pin as digital output.....	71
4.8.2.3	Fault pin as reference input .....	71
4.8.2.4	Fault pin as "Position reached" output.....	71
4.8.3	Query the input states .....	71

# Content

4.9	Query of the device state .....	73
4.10	Technical information .....	74
4.10.1	Ramp generator .....	74
4.10.1.1	Basic settings .....	74
4.10.1.2	Ramp generator in Profile Velocity Mode .....	75
4.10.1.3	Ramp generator in Profile Position Mode .....	76
4.10.2	Sinus commutation .....	77
4.10.3	Current controller and I <sup>2</sup> t current limitation .....	77
4.10.4	Overtemperature protection.....	79
4.10.5	Under-voltage monitoring .....	79
4.10.6	Overvoltage regulation .....	79
4.10.7	Setting the controller parameters .....	79
4.10.7.1	Default behaviour.....	80
4.10.7.2	Available controller parameters .....	80
4.10.7.3	Possible procedure.....	81
<b>5</b>	<b>Commissioning .....</b>	<b>82</b>
5.1	Communication settings .....	82
5.1.1	Setting via the CAN network .....	82
5.1.1.1	Setting the node number.....	83
5.1.1.2	Setting the Baud rate .....	83
5.1.2	Reading the node number .....	83
5.2	Basic settings .....	84
5.3	Configuration using the Motion Manager .....	85
5.3.1	Setting the connection .....	86
5.3.2	Selecting the motor .....	87
5.3.3	Configuring the drive .....	87
5.3.3.1	Performing the basic settings .....	88
5.3.3.2	Factor Group .....	89
5.3.3.3	Setting the controller parameters .....	89
5.3.3.4	Error handling.....	90
5.3.3.5	Setting inputs/outputs and homing .....	91
5.3.4	Managing the data set .....	92
5.3.5	Diagnosis .....	92
5.3.5.1	Status display .....	92
5.3.5.2	Trace function .....	92
<b>6</b>	<b>Parameter description .....</b>	<b>93</b>
6.1	Communication objects acc. to CiA 301 .....	93
6.2	Manufacturer-specific objects .....	101
6.3	Objects of the drive profile acc. to CiA 402 .....	107

## About this document

# 1 About this document

## 1.1 Validity of this document

This document describes:

- Quick start:
  - Initial commissioning and operation of the device with CANopen interface
- Communication:
  - Communication with the drive via CANopen
  - Basic services provided by the Communication structure
  - Methods for accessing the parameters
  - Drive from the viewpoint of the communication system
- Function:
  - Principle of the device controller
  - Commissioning and configuring the device
  - Operating modes and functions

This document is intended for following persons:

- Users who are commissioning a motor on the FAULHABER Motion Controller for the first time
- Software developers and project engineers with experience of interfaces
- Technicians and engineers in the application of controlled electrical drives and industrial communications systems

All data in this document relate to the standard versions of the drives. Changes relating to customer-specific versions can be found in the according data sheet.

## 1.2 Associated documents

For certain actions during commissioning and operation of FAULHABER products additional information from the following manuals is useful:

Manual	Description
Motion Manager 6	Operating instructions for FAULHABER Motion Manager PC software
Technical manual	Instructions for installation and use of the FAULHABER Motion Controller

These manuals can be downloaded in pdf format from the web page [www.faulhaber.com/manuals](http://www.faulhaber.com/manuals).

## About this document

### 1.3 Using this document

- ▶ Read the document carefully before undertaking configuration.
- ▶ Retain the document throughout the entire working life of the product.
- ▶ Keep the document accessible to the operating personnel at all times.
- ▶ Pass the document on to any subsequent owner or user of the product.

### 1.4 List of abbreviations

Abbreviation	Meaning
EEPROM	Electrically Erasable Programmable Read-Only Memory
MOSFET	Metal-Oxide Semiconductor Field-Effect Transistor
PWM	Pulse Width Modulation
PLC	Programmable Logic Controller
TTL	Transistor Transistor Logic

### 1.5 Symbols and markers



**CAUTION!**  
Hazards to persons. Disregard may lead to minor injuries.

- ▶ Measures for avoidance



**NOTICE!**  
Risk of damage.

- ▶ Measures for avoidance



Instructions for understanding or optimising the operational procedures

- ✓ Pre-requirement for a requested action
- 1. First step for a requested action
  - ↪ Result of a step
- 2. Second step of a requested action
  - ↪ Result of an action
- ▶ Request for a single-step action

## 2 Quick start

To facilitate introduction, this chapter highlights the initial steps for commissioning and operation of FAULHABER Motion Controllers with CANopen interface. Additionally, the detailed documentation must be read and taken into account, particularly chap. 5.3.3.1, p. 88.

### 2.1 Start with unconfigured controller

In the unconfigured state, node number 255 is set as the default in the Motion Control systems and automatic detection of the baud rate is active.

1. Set node number and baud rate by means of LSS.

The correct node number and baud rate is set via the LSS service according to CiA 305. The FAULHABER Motion Manager or another CANopen configuration tool can be used for this.

2. Set up communication to the drive node.

↳ The drive node appears with the correct name in the Motion Manager's tree view.

3. Set motor and controller data via the motor selection wizard.

↳ This preconfigures the controllers for the set motor and the corresponding load.

4. Set application parameters using the configuration dialog.

- Use the configuration dialog **Drive Functions** to adjust at least the basic settings such as operating mode, range limits, etc. to your application.
- Optimise the Hall sensor signals of external BL controllers.

5. Operate the drive via the Tool **Controller Tuning**.

If the motor is operated for the first time, use the Tool **Controller Tuning** for the desired operating mode. Here the controller amplifications can be adjusted precisely to the application.


6. Perform further settings.

If necessary, use the configuration dialog to make further application-specific settings. Alternatively, the drive can be started up directly at the own control.



### 2.2 Setting node number and baud rate


The standard units are delivered without valid node address (node ID = 255) and with automatic baud rate detection (AutoBaud) set.

 In order to set the baud rate and node address, the unit must first be connected via CAN to an appropriate configuration tool, which supports the LSS protocol (layer setting services and protocol) according to CiA DSP305.

FAULHABER Motion Manager installed on a PC with supported CAN interface can be used for this. The LSS compatible configuration tool can be used to set the node address and baud rate using the following methods (see chap. 5.1, p. 82):

- In Global mode: if only one drive is connected
- In Selective mode via the serial number: If a drive is to be configured in the network


Procedure, if the FAULHABER Motion Manager is to be used:

1. Connect the drive unit to a voltage source (24 V).  
For details of connection cable assignment see technical manual.
2. Connect drive unit to the CAN interface of the PC and switch on or connect PC to the CAN network.
3. Start FAULHABER Motion Manager.
4. Establish the connection via the CAN interface.
5. Select menu **Configuration - Connection parameters...**
6. Select LSS mode:
  - If only one LSS node is connected and you do not want to enter any further data, globally configure individual drive (LSS Switch Mode Global).
  - If a node is to be configured in the network, selectively configure specified node (LSS Switch Mode Selective). If the node has not yet been found in the Node Explorer, enter the serial number of the drive node to be configured here, otherwise the data fields are already correctly preset.
7. Select the required transfer rate or **Auto** and set the required node number.
8. Press **Send** button.
  -  The settings are transferred and are permanently stored in the controller. The Motion Manager then calls up the Scan function again and the node should now be displayed with the correct node number in the Node Explorer. After switching off and on again, the drive will operate with the set configuration.

### 2.3 Operation using FAULHABER Motion Manager

The FAULHABER Motion Manager provides easy access to the CANopen state machines using menu entries, which can be opened either with the Node Explorer's context menu (right-click) or with the **Commands** menu. The required node must have been activated beforehand by double clicking in the Node Explorer. The current states are always displayed in the status line at the bottom edge of the Motion Manager window.

For further information on the state machine of a CANopen node refer to chap. 3, p. 15.

 For simplified use, the Motion Manager also provides special commands for the CO variants. Those can be entered directly in the command input line in the terminal window or selected from the **Commands** menu. After sending the command, a command interpreter is activated which converts the command into a corresponding CAN telegram.


#### 1. Configuring drive functions:

In Motion Manager the motor selection wizard is available for selecting the motor and the corresponding basic parameters.

Other settings, e. g. for the function of the fault output, can be made under the **Configuration – Drive functions** menu item, where a convenient dialogue is provided (see chap. 5.3, p. 85). The configuration dialogue is also available for direct access in the wizard bar of the Motion Manager.

#### NOTICE!

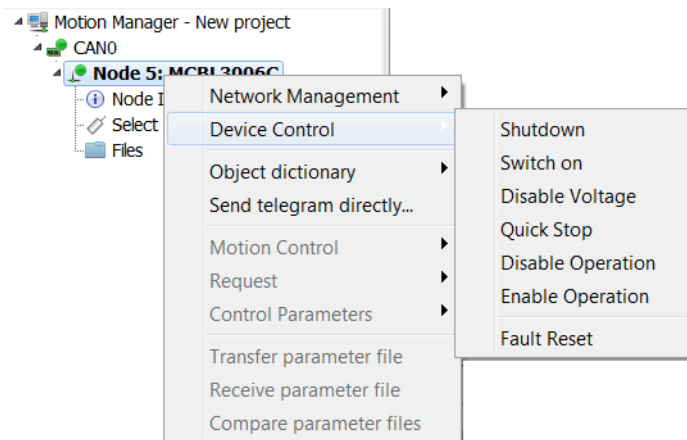
**Damage to the controller and/or drive by incorrect values in the Motion Controller's settings**

- ▶ Check basic settings (see chap. 5.3.3.1, p. 88).
- 2. Select the **Network Management - Start Remote Node** item in the context menu of the Node Explorer or in the **Commands** menu.
  -  The node is in the *Operational* state. PDO communication is now available.
- 3. Activate drive via the CiA 402 state machine:

A drive in accordance with CiA 402 must be activated according to a defined sequence of steps. The necessary commands are directly available in the context menu of the Node Explorer or via the **Commands** menu.

Alternatively, the commands can also be executed directly by entering the corresponding Motion Manager commands in the command input field of the terminal window.

## Quick start



- Prepare the drive for switching on via the **Device Control - Shutdown** menu item or the `SHUTDOWN` command.
- Switch on the output stage via the **Device Control - Switch On** menu item or the `SWITCHON` command.
- Enable operation via the **Device Control - Enable Operation** menu item or the `ENOP` command.

Alternatively, the **Switch on output stage** button can also be pressed to perform these steps all at once.

#### 4. Drive the motor in a speed-controlled manner (e.g., at 100 rpm):

- **Set Profile Velocity Mode:**  
 Select the **Movement Control - Set Operating Mode 0x6060 (OPMOD)** item via the context menu of the Node Explorer or via the **Commands** menu and set to *Profile Velocity Mode (3)*  
 - or -  
 enter the `OPMOD3` command in the command input field of the terminal window.
- **Set the target speed (Target speed) to the value 100:**  
 Select the **Movement Control - Set Speed 0x60FF (V)** item via the context menu of the Node Explorer or via the **Commands** menu and enter the desired value in the input field  
 - or -  
 enter the `V100` command in the command input field of the terminal window.
- **Stop the motor:**  
 Set the target speed to the value 0 (`V0`)  
 - or -  
 select the **Device Control - Disable Operation** menu item or execute the `DIOP` command.

## Quick start

---

5. Move the motor (e.g., relatively by 10 000 increments):
  - **Set Profile Position Mode:**

Select the **Movement Control - Set Operating Mode 0x6060 (OPMOD)** item via the context menu of the Node Explorer or via the **Commands** menu and set to *Profile Position Mode (1)*

- or -

enter the `OPMOD1` command in the command input field of the terminal window.
  - **Set target position (*Target Position*) to the value 10 000:**

Select the **Movement Control - Set Target Position 0x607A (SPOS)** item via the context menu of the Node Explorer or via the **Commands** menu and enter the desired value in the input field

- or -

enter the `SPOS10000` command in the command input field of the terminal window.
  - **Move towards target position:**
    - Select the **Device Control - Enable Operation** item via the context menu of the Node Explorer or via the **Commands** menu
    - or -
    - enter the `ENOP` command in the command input field of the terminal window.
    - Select the **Movement Control - Start Relative Positioning (MR)** item via the context menu of the Node Explorer or via the **Commands** menu
    - or -
    - enter the `MR` command in the command input field of the terminal window.
  - **6. Switch off output stage:**

Select the **Switch off output stage** button in the toolbar

- or -

press the F5 key

- or -

select the **Device Control - Disable Voltage** menu item or execute the `DISABLE` command.

## Quick start

### 2.4 Operation using own host application

#### 2.4.1 Activate CANopen nodes

- ▶ Use the `Start Remote Node` broadcast command with CAN ID 0 to start either an individual node or the whole network and to set it to the *Operational* state.

11-bit identifier	2 bytes user data	
ID 0x000	01	00

The first data byte contains the start command `Start Remote Node`, the second data byte contains the node address or 0 for the whole network.

All functions can be proceeded after the node has been started. The drive can now be activated and operated using the device control functions according to CiA DSP402.


The identifiers of the individual objects are preset according to the *Predefined Connection Set* and depend on the node number.

Object	CAN-ID	Description
TxPDO1	0x180 + Node-ID	Receive drive data (e. g. status values)
RxPDO1	0x200 + Node-ID	Send data to the drives (e. g. control commands)
TxPDO2	0x280 + Node-ID	Receive drive data (e. g. status values)
RxPDO2	0x300 + Node-ID	Send data to the drives (e. g. control commands)
TxPDO3	0x380 + Node-ID	Receive drive data (e. g. status values)
RxPDO3	0x400 + Node-ID	Send data to the drives (e. g. control commands)
TxPDO4	0x480 + Node-ID	Receive drive data (e. g. status values)
RxPDO4	0x500 + Node-ID	Send data to the drives (e. g. control commands)
TxSDO1	0x580 + Node-ID	Read entry of the object dictionary
RxSDO1	0x600 + Node-ID	Write entry of the object dictionary

On delivery, the drives are in operating mode *Modes of Operation* = 1 after switch on (*Profile Position Mode*). The drive is controlled by using the device control state machine, which is operated by the Controlword (object 0x6040 or RxPDO) and queried using the Statusword (object 0x6041 or TxPDO).

#### 2.4.2 Configure the drives

- ▶ Configure the drive by means of SDO transfer using the objects of the object dictionary.

 Use of the FAULHABER Motion Manager is recommended for the basic settings (see chap. 5.3.3.1, p. 88).

### 2.4.3 Operation in one of the CANopen CiA 402 drive profiles

A drive in accordance with CiA402 must be activated according to a defined sequence of steps (see chap. 4.2, p. 37). Write access to the Controlword is possible using the object dictionary at address 0x6040 or using RxPDO:

1. Execute the Shutdown command (Controlword = 0x00 06).
2. Execute the Switch On command (Controlword = 0x00 07).
  - ↪ The drive is in the *Switched On* state. Operation must then be released to enable drive commands to be executed.
3. Execute the Enable Operation command (Controlword = 0x00 0F):
  - ↪ The drive is in the *Operation Enabled* state, in which it can be operated using the corresponding objects of the set control mode (see chap. 4.2, p. 37 and chap. 4.3, p. 46).
4. Drive the motor in a speed-controlled manner (e.g., at  $500\text{min}^{-1}$ ):
  - Modes of Operation (object 0x6060): Set 3 (*Profile Velocity Mode*) via SDO access.
  - Target Velocity (object 0x60FF): 500
5. Stop the motor:
  - Set the target speed (Target Velocity) to the value 0 (object 0x60FF).
  - or
  - Controlword = 0x00 07 (Disable Operation)
6. Move the motor (e.g., relatively by 10 000 increments):
  - Modes of Operation (object 0x6060): Set 1 (*Profile Position Mode*) via SDO access.
  - Target Position (object 0x607A): 10 000
  - Controlword = 0x00 7F (New Set-Point, Change Set Immediately, Rel)

# CANopen protocol description

## 3 CANopen protocol description

### 3.1 Introduction

#### CANopen

CANopen is a standard software protocol. A CAN hardware environment is required for communication using CANopen. Up to 127 nodes can be addressed within a CANopen network. The maximum transmission speed is 1 MBit/s.

#### CAN standardisation

The CiA defines the following aspects in CiA 301:

- communications structure
- Control and monitoring functions

CANopen device profiles have been defined for a wide range of device classes, such as:

- CiA 402 for drives
- CiA 401 for input and output devices

#### Structure of a CANopen telegram

A CANopen telegram has an 11-bit identifier and can contain up to 8 bytes of user data.

*Tab. 1: Schematic structure of a CANopen telegram*

11-bit identifier	up to 8 bytes user data								
11-bit	8-bit	8-bit	8-bit	8-bit	8-bit	8-bit	8-bit	8-bit	8-bit

# CANopen protocol description

## 3.2 Communication services

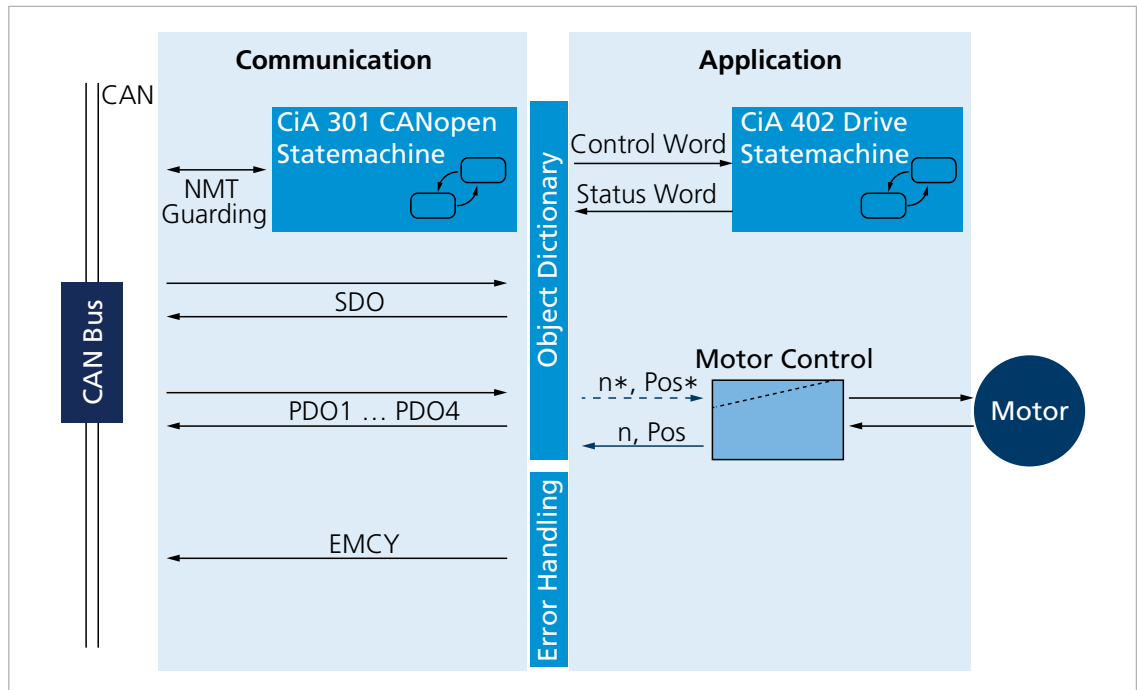


Fig. 1: Communication services of the Motion Controller

The communication part contains communication services as specified in CiA 301.

Tab. 2: Communication services to CiA 301

Communication services	Description
NMT (Network Management)	Activates nodes and monitors the current status of a node (see chap. 3.8, p. 28).
SDO (Service Data Object)	The CANopen master uses the SDO to access parameters within a node. Each SDO access reads or writes exactly one parameter. An SDO can only address one node in a network (see chap. 3.5, p. 22).
PDO (Process Data Object)	The PDO is used to access real-time data. A PDO can use a CAN message to access multiple drive parameters concurrently. The parameters sent or received in a PDO can be freely configured (see chap. 3.4, p. 18).
SYNC object	SYNC objects are used to synchronise different applications on the CAN-BUS (see chap. 3.7, p. 27).
EMCY (Emergency Object)	An emergency message is used to inform the CANopen master about errors. A CAN message conveys the error code asynchronously so that the status of the CANopen slave need not be interrogated after an error (see chap. 3.6, p. 25).

### Communications profile

FAULHABER Motion Controllers support the CANopen communications profile to CiA 301 V4:

- 4 transmission PDOs
- 4 receipt PDOs
- 1 server SDO
- Emergency object
- NMT with node guarding and heartbeat
- SYNC object



## CANopen protocol description

**i** The data assignment of the PDOs is pre-set to the “PDO set for servo drive” as specified in CiA 402 V3, but can be changed by the user (dynamic PDO mapping).

### 3.3 Identifier distribution

The Communication Object Identifier (COB-ID) consists of a 7-bit node address (Node-ID) and a 4-bit function code.

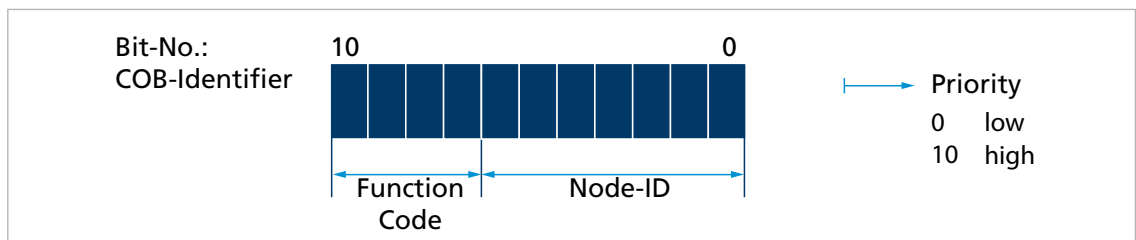


Fig. 2: Identifier distribution

The Predefined-Connection-Set defines the standard identifier for the most important objects.

Tab. 3: Standard identifier

Object	Function code (binary)	Resulting COB-ID	Object index for communication setting
NMT	0000	0	–
SYNC	0001	128 (80 h)	1005 h
EMERGENCY	0001	129 (81h) to 255 (FFh)	1014 h
PDO1 (tx)	0011	385 (181h) to 511 (1FFh)	1800 h
PDO1 (rx)	0100	513 (201h) to 639 (27Fh)	1400 h
PDO2 (tx)	0101	641 (281h) to 767 (2FFh)	1801 h
PDO2 (rx)	0110	769 (301h) to 895 (37Fh)	1401 h
PDO3 (tx)	0111	897 (381h) to 1023 (3FFh)	1802 h
PDO3 (rx)	1000	1025 (401h) to 1151 (47Fh)	1402 h
PDO4 (tx)	1001	1153 (481h) to 1279 (4FFh)	1803 h
PDO4 (rx)	1010	1281 (501h) to 1407 (57Fh)	1403 h
SDO (tx)	1011	1409 (581h) to 1535 (5FFh)	1200 h
SDO (rx)	1100	1537 (601h) to 1663 (67Fh)	1200 h
NMT error control	1110	1793 (701h) to 1919 (77Fh)	–

The COB-IDs of the PDOs, the SYNC objects and the emergency objects can be changed via the communication parameters in the object dictionary. The COB-ID of the SDO telegram cannot be changed and is always derived from the node number.

## CANopen protocol description

**i** As delivered the system is configured with the node number 1. The COB-IDs are pre-set correspondingly:

- RxPDO: 201h, 301h, 401h and 501h
- TxPDO: 181h, 281h, 381h and 481h
- EMCY: 81 h
- RxSDO: 581 h
- TxSDO: 601 h

**i** If the node number is changed using the LSS protocol, the COB-IDs of the PDOs and the Emergency Object remain unchanged. If the COB-IDs of the PDOs and of the emergency object are to change together with the node number, the node number must first be set to 255 (undefined) and then set to the desired number.

### 3.4 PDO (Process Data Object)

PDOs are CAN messages with up to 8 bytes user data. PDOs contain process data for controlling and monitoring the behaviour of the device. The drive makes the distinction between receipt PDOs and transmission PDOs.

- Receipt PDOs (RxPDO): are received by a drive and typically contain control data
- Transmission PDOs (TxPDO): are sent by a drive and typically contain monitoring data

PDOs are evaluated or transmitted only when the device is in the NMT *Operational* state (see chap. 3.8, p. 28).

The transmission of PDOs can be triggered in various different ways. The behaviour can be set for each PDO via the transmission type parameter of the communication parameters in the object dictionary:

*Tab. 4: Types of PDO transmissions*

Transmission Type	Description
Event-driven	Event-driven RxPDOs are processed immediately on receipt. Event-driven TxPDOs are sent when the statusword of the device is contained and has been changed.
Remote request (RTR)	Data are sent in response to a request message.
Synchronised	Data are sent after receipt of a SYNC object (see chap. 3.7, p. 27).

## CANopen protocol description

### 3.4.1 PDO configuration

- A maximum of 4 parameters can be mapped in one PDO.
- The data assignment of PDOs can be changed via the objects 0x1600 to 0x1603 and 0x1A00 to 0x1A03. The mapping procedure necessary for this is described in CiA 301. A suitable tool (such as FAULHABER Motion Manager or System Manager for the PLC controller used) is necessary for the mapping procedure.
- The objects 0x1400 to 0x1403 and 0x1800 to 0x1803 can be used to change the Transmission Type and COB-ID of the PDOs.
- The Transmission Type parameters can be used to change the behaviour of a PDO:

Tab. 5: Transmission type of a PDO

Transmission Type	Meaning
255	asynchronous (event-driven) Only affects TxPDOs that contain Statusword (0x6041). If its value changes, this PDO is sent. For all other PDOs (i.e., those that do not contain the Statusword), this transfer type has no effect. Sending can, however, be forced via the Event Timer (see below).
253	asynchronous, on request (RTR)
1 to 240	synchronous, cyclical A PDO is sent after every SYNC object (see chap. 3.7, p. 27). The value is then equal to the number of SYNC objects that must be received before the PDO is sent again (1 = PDO is sent for every SYNC object)
0	synchronous, acyclical A PDO is sent or executed once after a SYNC object when the contents of the PDO have changed (see chap. 3.7, p. 27).

#### Event Timer for TxPDOs

If transfer type 255 is set, the setting of the Event Timer (subindex 5) of the PDO is also taken into account. The Event Timer specifies the maximum time span for sending TxPDOs. If the time specified in the Event Timer has elapsed since the last transmission of the TxPDO, the PDO is sent automatically. This ensure that the master constantly receives information from objects, even if they are not mapped together with the Statusword.

The value of the Event Timer is specified in milliseconds. The time span that can be set is 5-65 000 ms.

**i** FAULHABER recommends always setting the Event Timer >100 ms. The CAN bus will otherwise be overloaded with messages and the performance of the complete system no longer ensured. In the event of reduced performance, other important messages (e.g., Emergency messages) are no longer sent.

- Keep the bus load as low as possible to ensure that events that cause many CAN telegrams do not result in malfunctions.

For TxPDOs to be sent taking into account the Event Timer, the following preconditions must be met:

- The drive is now in the *Operational* NMT state
- The value of the Event Timer is  $\geq 5$  ms
- The TxPDO is valid
- The transfer type is 255 (asynchronous)

The Event Timer always start simultaneously. There is no comparable function for RxPDOs.

## CANopen protocol description

### 3.4.2 PDO mapping in the standard configuration (status as delivered)

#### RxPDO1: Controlword

11-bit identifier	2 bytes user data	
0x200 (512d) + node ID	LB	HB

The RxPDO1 contains the 16-bit Controlword to CiA DSP402. The Controlword controls the state machine of the drive unit and points to the object index 0x6040 in the object dictionary. The bit distribution is described in chap. 4.2, p. 37.

#### TxPDO1: Statusword

11-bit identifier	2 bytes user data	
0x180 (384d) + node ID	LB	HB

The TxPDO1 contains the 16-bit Statusword to CiA 402. The Statusword indicates the status of the drive unit and points to the object index 0x6041 in the object dictionary. The bit distribution is described in chap. 4.2, p. 37.

#### RxPDO2: Controlword, target position (PP)

11-bit identifier	6 bytes user data					
0x300 (768d) + node ID	LB	HB	LLB	LHB	HLB	HHB

The RxPDO2 contains the 16-bit Controlword and the 32-bit value of the target position (object 0x607A) for the Profile Position mode (PP).

#### TxPDO2: Statusword, Position Actual Value

11-bit identifier	6 bytes user data					
0x280 (640d) + node ID	LB	HB	LLB	LHB	HLB	HHB

The TxPDO2 contains the 16-bit Statusword and the 32-bit value of the actual position (object 0x6064).

#### RxPDO3: Controlword, target velocity (PV)

11-bit identifier	6 bytes user data					
0x400 (1024d) + node ID	LB	HB	LLB	LHB	HLB	HHB

The RxPDO3 contains the 16-bit Controlword and the 32-bit value of the target velocity (object 0x60FF) for the Profile Velocity mode (PV).

#### TxPDO3: Statusword, Velocity Actual Value

11-bit identifier	6 bytes user data					
0x380 (896d) + node ID	LB	HB	LLB	LHB	HLB	HHB

The TxPDO3 contains the 16-bit Statusword and the 32-bit value of the actual speed (object 0x606C).

## CANopen protocol description

### RxPDO4: Controlword, target torque

11-bit identifier	6 bytes user data					
0x400 (1024d) + node ID	LB	HB	LLB	LHB	HLB	HHB

The RxPDO4 contains the 16-bit Controlword and in Profile Position mode (PP) the 32-bit value of the target position in internal units (Object 0x257A).

### TxPDO4: Statusword, torque actual value

11-bit identifier	6 bytes user data					
0x380 (896d) + node ID	LB	HB	LLB	LHB	HLB	HHB

The TxPDO4 contains the 32-bit value of the actual position (object 0x6064) and the 32-bit value of the actual speed (object 0x606C).


### 3.4.3 Dealing with mapping errors

If the mapping procedure specified in CiA 301 is not complied with, one of the following SDO errors will be returned:

*Tab. 6: SDO errors in response the incorrect mapping procedure*

SDO error	Meaning	Cause
0x06090030	General value range error	The mapping parameter lies outside that specified in the mapping procedure.
0x06020000	Object not present in the object dictionary	The value for the number of mapped objects is greater than the number of valid entries in the respective subindexes for the mapping parameter objects.

If the number of mapped objects is 0, the PDO will be flagged internally as invalid and will not be operated.

 Other mapping errors are described in the SDO error table (see chap. 3.5.2, p. 24).

## CANopen protocol description

### 3.5 SDO (Service Data Object)

The SDO reads and describes parameters in the OV (object dictionary). The SDO accesses the object dictionary via the 16-bit index and the 8-bit subindex. At the request of the client (PC, PLC (programmable logic controller)) the Motion Controller makes data available (upload) or receives data from the client (download).

Tab. 7: General structuring of the SDO user data

Byte0	Byte 1 to 2	Byte 3	Byte 4 to 7
Command specifier	16-bit index	8-bit subindex	4-byte parameter data

Tab. 8: Distribution of the SDO types of transmission

Type of transmission	Number of bytes	Purpose
Expedited transfer	maximum 4 bytes	Read and write individual numeric parameters
Segmented transfer	more than 4 bytes	Read text parameters (such as device name, firmware version) and transmit data blocks (such as the trace buffer)

Only the expedited transfer is described in this document. The segmented transfer is described in CiA 301.

#### 3.5.1 Expedited transfer

SDO messages are always size 8 bytes.

##### Read OD entries (Client-to-Server, Upload-Request)

11-bit identifier	8 bytes user data							
0x600 (1536d) + node ID	0x40	Index LB	Index HB	Subindex	0	0	0	0

##### Server-to-Client, Upload-Response

11-bit identifier	8 bytes user data							
0x580 (1408d) + node ID	CS(0x4x)	Index LB	Index HB	Subindex	LLB (D0)	LHB (D1)	HLB (D2)	HHB (D3)

The command specifier CS(0x4x) specifies the number of valid data bytes in D0 to D3 and the transfer code. The command specifier is coded as follows:

- CS = 0x4F, 1 data byte in D0
- CS = 0x4B, 2 data bytes in D0 to D1
- CS = 0x47, 3 data bytes in D0 to D2
- CS = 0x43, 4 data bytes in D0 to D3

## CANopen protocol description

### Write OD entries (Client-to-Server, Download-Request)

11-bit identifier	8 bytes user data							
0x600 (1536d) + node ID	CS(0x2x)	Index LB	Index HB	Subindex	LLB (D0)	LHB (D1)	HLB (D2)	HHB (D3)

The command specifier CS(0x2x) specifies the number of valid data bytes in D0 to D3 and the transfer code. The command specifier is coded as follows:

- CS = 0x2F, 1 data byte in D0
- CS = 0x2B, 2 data bytes in D0 to D1
- CS = 0x27, 3 data bytes in D0 to D2
- CS = 0x23, 4 data bytes in D0 to D3
- CS = 0x22, no specification of the number of data bytes

### Server-to-Client, Download-Response

11-bit identifier	8 bytes user data							
0x580 (1407d) + node ID	0x60	Index LB	Index HB	Subindex	0	0	0	0

### Abort in the event of SDO errors

#### SDO-abort Client-to-Server

11-bit identifier	8 bytes user data							
0x600 (1536d) + node ID	0x80	Index LB	Index HB	Subindex	ERROR 0	ERROR 1	ERROR 2	ERROR 3

#### SDO-abort Server-to-Client

11-bit identifier	8 bytes user data							
0x580 (1536d) + node ID	0x80	Index LB	Index HB	Subindex	ERROR 0	ERROR 1	ERROR 2	ERROR 3

## CANopen protocol description

### 3.5.2 SDO error description

If the SDO protocol on a page cannot be processed further, an SDO-Abort telegram is sent (see chap. 3.5.1, p. 22). The error types are coded as follows:

- Error0: Additional error code HB
- Error1: Additional error code LB
- Error2: Error code
- Error3: Error class

Error class	Error code	Additional code	Description
0x05	0x03	0x0000	The toggle bit is not changed
0x05	0x04	0x0001	SDO command specifier invalid or unknown
0x06	0x01	0x0000	Access to this object is not supported
0x06	0x01	0x0001	Attempt to read a write-only parameter
0x06	0x01	0x0002	Attempt to write to a read-only parameter
0x06	0x02	0x0000	Object not present in the object dictionary
0x06	0x04	0x0041	Object cannot be mapped in a PDO
0x06	0x04	0x0042	Number and/or length of the mapped objects exceed the PDO length
0x06	0x04	0x0043	General parameter incompatibility
0x06	0x04	0x0047	General internal incompatibility error in the device
0x06	0x07	0x0010	Data type or parameter length do not match or are unknown
0x06	0x07	0x0012	Data types do not match, parameter length too long
0x06	0x07	0x0013	Data types do not match, parameter length too short
0x06	0x09	0x0011	Subindex not present
0x06	0x09	0x0030	General value range error
0x06	0x09	0x0031	Value range error: Parameter value too large
0x06	0x09	0x0032	Value range error: Parameter value too small
0x06	0x09	0x0036	Value range error: Maximum value greater than minimum value
0x08	0x00	0x0000	General SDO error
0x08	0x00	0x0020	Cannot be accessed
0x08	0x00	0x0022	Cannot be accessed at current device status



## CANopen protocol description

### 3.6 Emergency object (error message)

The emergency object informs other bus participants of errors asynchronously without requiring interrogation. The emergency object is always size 8 bytes:

11-bit identifier		8 bytes user data						
0x80 (128d) + node ID	Error0(LB)	Error1(HB)	Error-Reg	FE0 (LB)	FE1 (HB)	0	0	0

Assignment of user data:

- Error0(LB)/Error1(HB): 16-bit error code
- Error-Reg: Error register (contents of object 0x1001, see chap. 6.1, p. 93)
- FE0(LB)/FE1(HB): 16-bit FAULHABER error register (contents of object 0x2320, see Tab. 12)
- Bytes 5 to 7: unused (0)

The error register identifies the error type. The individual error types are bit-coded and are assigned to the respective error codes. The object 0x1001 allows interrogation of the last value of the error register.

Tab. 9 lists all the errors that have been reported by emergency messages, providing the respective error is included in the Emergency Mask (0x2321.01) for the FAULHABER error register (Tab. 13).

Tab. 9: Emergency Error Codes

Error Code	Error	Error Mask 0x2321	Error Register Bit
0x0000	No error	–	–
0x1000	Generic error	–	0
0x2000	Current	–	–
0x2300	Current, device output side		
0x2310	Continuous over current	0x0001	1
0x3000	Voltage	–	–
0x3200	Voltage inside the device	–	–
0x3210	Overvoltage	0x0004	2
0x4000	Temperature		
0x4300	Drive temperature	–	–
0x4310	Overtemperature	0x0008	3
0x5000	Device hardware	–	–
0x5500	Data storage	–	–
0x5530	Flash memory error	0x0010	5
0x6000	Device software	–	–
0x6100	Internal software	0x1000	5
0x8000	Monitoring	–	–
0x8100	Communication	–	–
0x8110	CAN Overrun (objects lost)	0x0080	4
0x8120	CAN in error passive mode	0x0040	4

## CANopen protocol description

Error Code	Error	Error Mask 0x2321	Error Register Bit
0x8130	Life guard or heartbeat error	0x0100	4
0x8140	Recovered from bus off	0x0200	4
0x8200	Protocol error	–	–
0x8210	PDO not processed due to length error	0x4000	4
0x8220	PDO length exceeded	0x2000	4
0x8400	Velocity speed controller (deviation)	0x0002	5
0x8600	Positioning controller	–	–
0x8611	Following error (deviation)	0x0002	5
0xFF00	Device specific	–	–
0xFF01	Conversion overflow	0x0800	0

### Example:

An emergency telegram with 8 data bytes is sent (see Tab. 10), if the following conditions apply:

- In Error Mask 0x2321 of the FAULHABER error register, bit 1 is set under subindex 1.
- The continuous current limit value set by means of object 0x2333 was exceeded by longer than the error delay time set by means of object 0x2322.

Tab. 10: Example of user data assignment to an emergency message

8 bytes user data							
0x10	0x23	0x01	0x00	0x00	0x00	0x00	0x00

### Deviation error

In Profile Velocity Mode, emergency error 0x8611 and in Profile Position Mode emergency error 0x8400 is sent if the maximum permissible speed deviation set by means of object 0x2322.02 was exceeded. The error is reset if the DSP402 state machine switches or new positioning is started.

## CANopen protocol description

### 3.7 SYNC object

The SYNC object is a message without any user data. The SYNC object is used to trigger synchronous PDOs and at the same time to start processes on various items of equipment.

The identifier of the SYNC objects is set in the object dictionary under the index 0x1005 (by default 0x80).

11-bit identifier	0 bytes of user data
-------------------	----------------------

0x80	no user data
------	--------------

**i** In order that a SYNC object triggers a PDO, the transmission type of the PDO to be triggered must be set accordingly (see Tab. 5).

#### 3.7.1 Triggering synchronous PDOs

**Synchronous RxPDO:** The command transmitted with the PDO is not executed until a SYNC object is received. The transmission types 1 to 240 of an RxPDO are identical to transmission type 0.

**Synchronous TxPDO:** The PDO with the current data is not sent until a SYNC object is received.

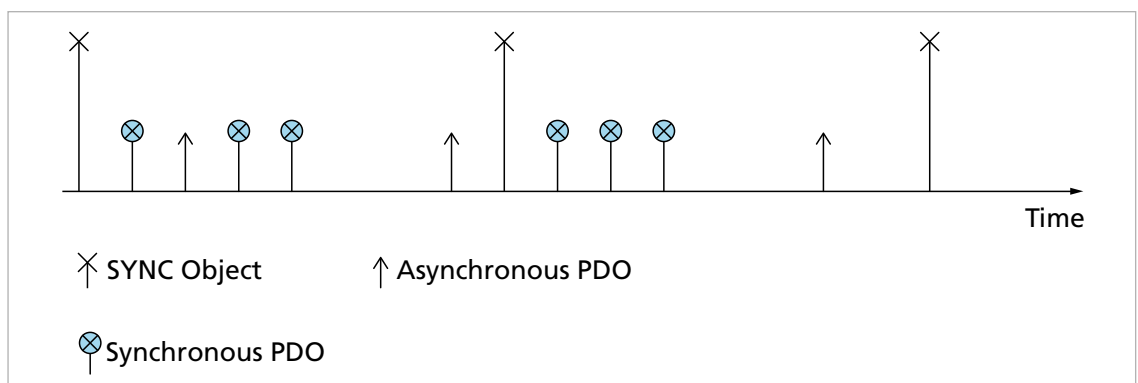


Fig. 3: TxPDO with SYNC chart

**i** The nodes can also be grouped by transmission types 1-240.

## CANopen protocol description

### 3.8 NMT (Network Management)

The network management object governs the CiA 301 state machine of the CANopen device and monitors the network nodes.

After switching on and initialising, the Motion Controller is automatically set to the *Pre-Operational* state. In the *Pre-Operational* state the device can communicate only with NMT messages and via SDOs.

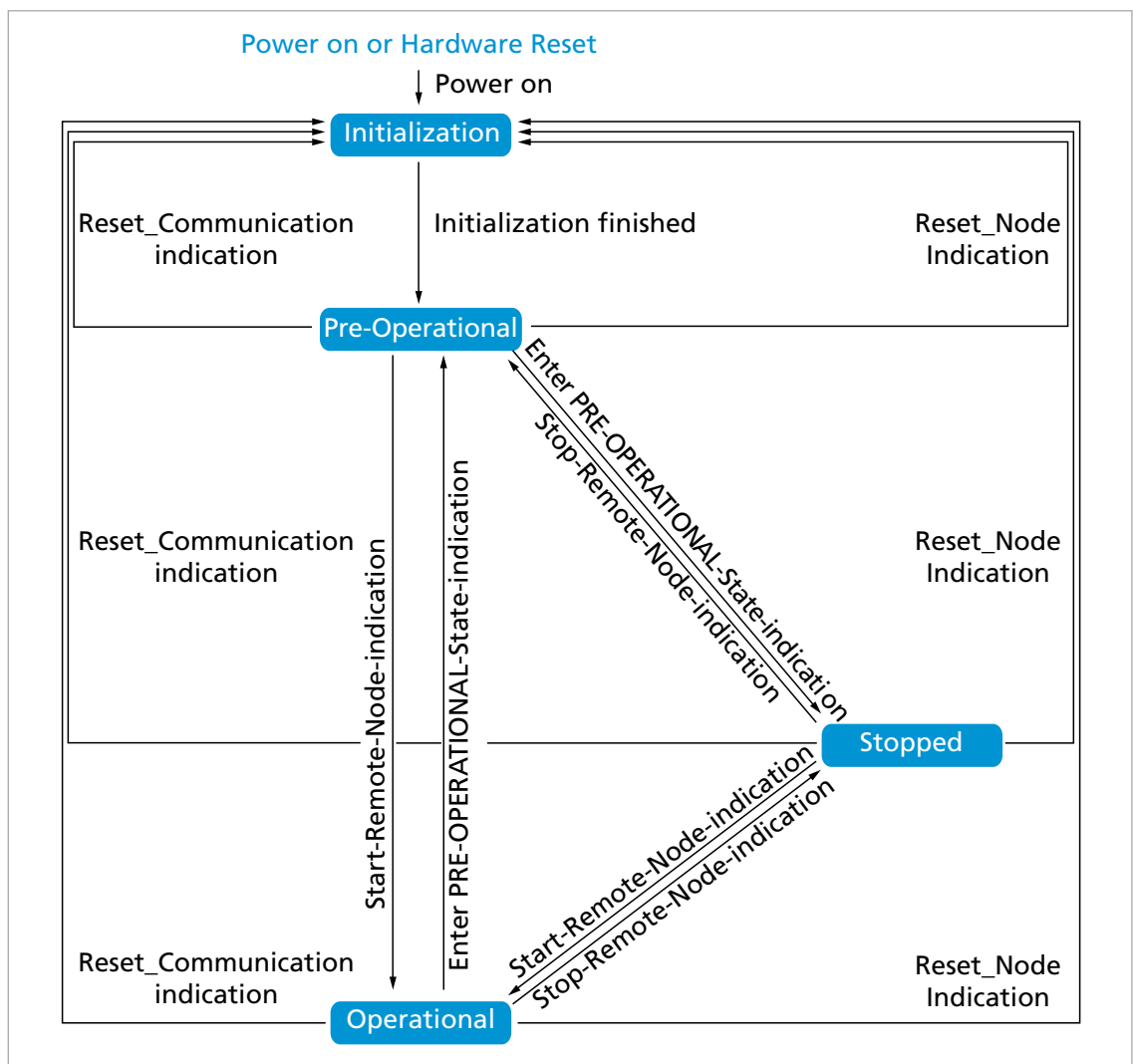


Fig. 4: CiA 301 state machine

Tab. 11: NMT changes of status

Status transition	CS	Meaning
Power on	–	The initialisation state is achieved automatically on switching on.
Initialisation finished	–	After initialisation the device is automatically in the Pre-Operational state, and it sends a boot-up message.
Start remote node indication	0x01 (1d)	This starts the device and enables transmission of PDOs.
Enter Pre-operational state indication	0x80 (128d)	Stops the transmission of PDOs, SDOs are still active.
Stop remote node indication	0x02 (2d)	The drive is set to the stopped status, SDO and PDO are switched off.

## CANopen protocol description

Status transition	CS	Meaning
Reset node indication	0x81 (129d)	Performs a reset. All objects are reset to Power-On standards.
Reset communication indication	0x82 (130d)	Performs a reset of the communications functions.

**i** FAULHABER Motion Controllers are equipped with a standard configuration for all objects. Once commissioning is complete the application-specific settings can be saved directly in the device. In most cases no further parametrisation is necessary at the system start.

### Starting a CANopen node

Start remote node:

11-bit identifier	2 bytes user data	
0x000	0x01	Node ID

An entire network can also be started with a CAN message:

Start all remote nodes:

11-bit identifier	2 bytes user data	
0x000	0x01	0x00

After the node or the entire network is started the device is in the *Operational* state. The device can now be operated using PDOs.

In the *Stopped* state the device is in an error state and can no longer be operated using PDOs. Under these circumstances, communication with the device is available only by NMT messages.

An NMT message always consists of 2 bytes on the identifier 0x000.

### NMT message

11-bit identifier	2 bytes user data	
0x000	CS	Node ID

Assignment of user data:

- CS: Command specifier (see Tab. 11)
- Node ID: Node address (0 = all nodes)

**i** In the event of a serious communications error the Motion Controller switches by default to the *Pre-Operational* NMT status. Different behaviour can be set using the object 0x1029.


## CANopen protocol description

### 3.8.1 Boot up

Immediately after the initialisation phase the Motion Controller sends a boot-up message. A boot-up message signals the end of the initialisation phase of a module after it has been switched on. A boot-up message is a CAN message with one data byte (byte 0 = 0x00) on the identifier of the node guarding message (0x700 + node ID).

11-bit identifier	1 byte of user data
0x700 (1792d) + node ID	0x00

### 3.8.2 Monitoring functions

 Only one monitoring function; node guarding or heartbeat, can be used at one time.

#### 3.8.2.1 Node guarding

The node guarding object interrogates the instantaneous state of the device. To do this, the master sets a remote frame with a request for the guarding identifier of the node to be monitored. The node to be monitored responds with the guarding message which contains the current status of the node and a toggle bit.

## CANopen protocol description

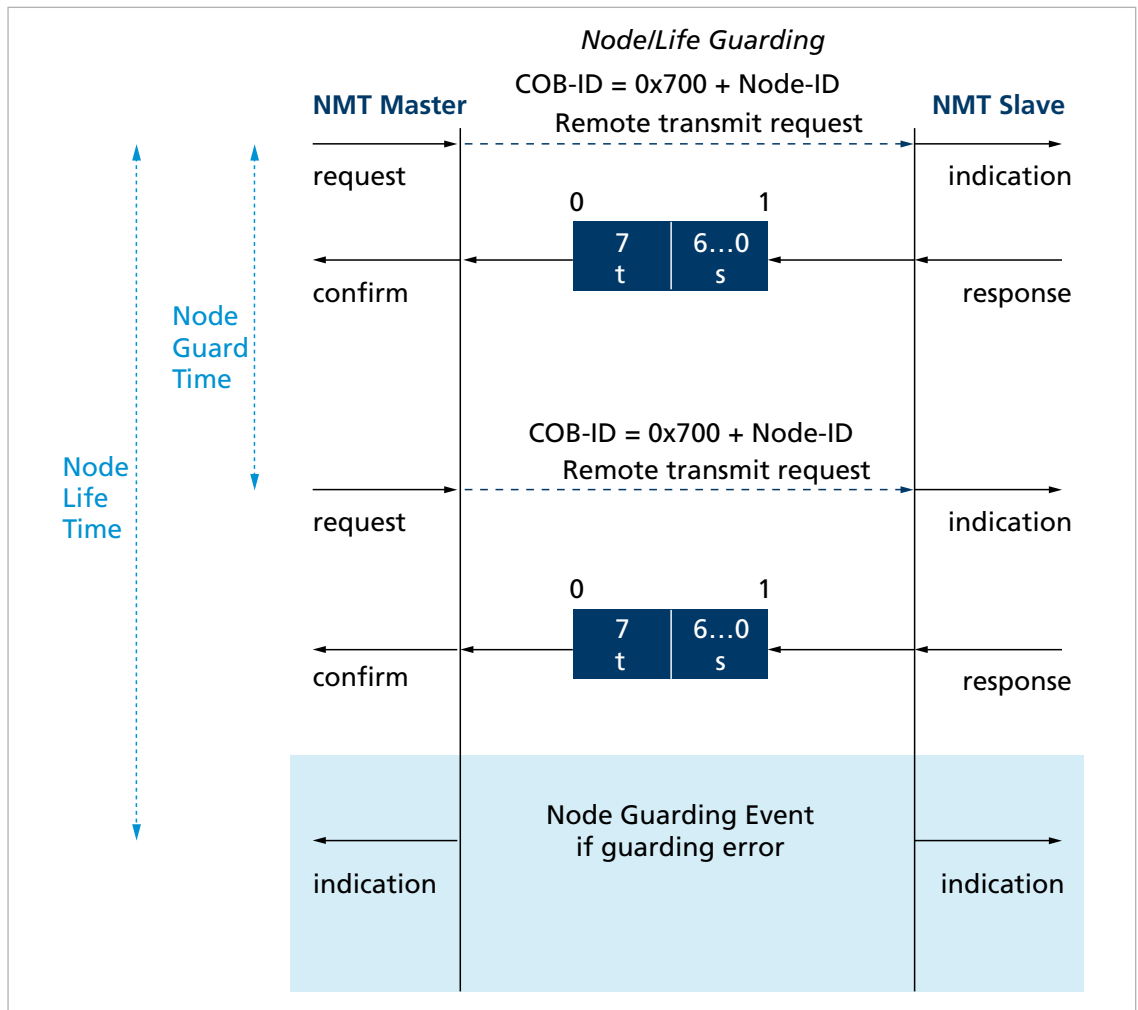


Fig. 5: Diagram of the node guarding protocol

**t: Toggle bit**

Initially 0, changes its value at each guarding telegram

**s: Status**

$s = 0x04$  (4d): Stopped

$s = 0x05$  (5d): Operational

$s = 0x7F$  (127d): Pre-operational

If a node life time  $> 0$  is set (objects 0x100C and 0x100D) and no node guarding request is made by the master within the set node life time, a node guarding error is set. The response to a node guarding error is set by the FAULHABER error register (object 0x2321) (see Tab. 14). The default is to send the emergency message 0x8130.

## CANopen protocol description

### 3.8.2.2 Heartbeat

The Motion Controller can be set to act both as the heartbeat producer and also as the heartbeat consumer.

- **Heartbeat producer:** On a cyclical basis the Motion Controller sends out a message which is received by one or more heartbeat consumers in the network.
- **Heartbeat consumer:** If within the heartbeat consumer time no heartbeat message is received from the heartbeat producers that are being monitored, the Motion Controller responds with the behaviour specified in the FAULHABER error register (see Tab. 12).

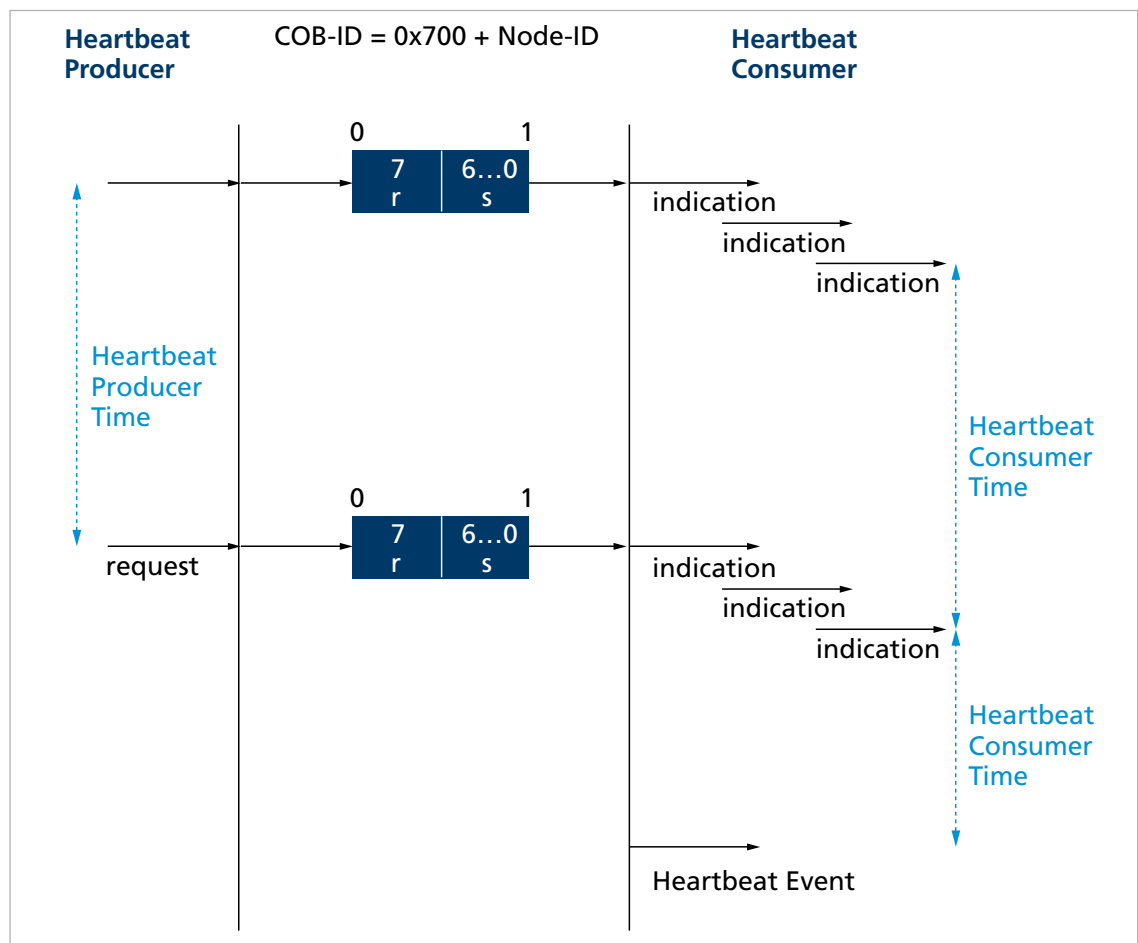


Fig. 6: Chart of the heartbeat protocol

**r: Reserved**

Always 0

**s: Status**

*s = 0x00 (0d): Boot up*

*s = 0x04 (4d): Stopped*

*s = 0x05 (5d): Operational*

*s = 0x7F (127d): Pre-operational*



## CANopen protocol description

### 3.8.3 Settings for the monitoring functions

- Only one of the two monitoring functions (Node Guarding, Heartbeat) can be activated at one time.
- If the Producer Heartbeat Time is  $> 0$  (object 0x1017) the Motion Controller operates as a Heartbeat Producer. At intervals of the Producer Heartbeat Time the Motion Controller sends a Heartbeat message. The Node Guarding Time is set to 0 (see chap. 3.8.2.1, p. 30).
- If the Heartbeat is activated, the boot-up message after the switch-on ranks as the first Heartbeat message. Further heartbeats follow at the interval for the Producer Heartbeat Time.
- If in addition to the Producer Heartbeat Time a Heartbeat Consumer Time  $> 0$  is set (object 0x1016.01), the Motion Controller operates as a heartbeat consumer. The settings for the Heartbeat Producer are inoperative. The node ID of the master to be monitored and the Heartbeat Consumer Time are entered in the object 0x1016.
- The Heartbeat Consumer Time must always be longer than the Producer Heartbeat Time of the master.
- If within the set Heartbeat Consumer Time the Motion Controller receives no Heartbeat message from the master, a heartbeat event is triggered. The response to a Heartbeat event is determined by the FAULHABER error register Error Mask (object 0x2321) (see Tab. 12). The default is to send the emergency message 0x8130.
- If whilst the Heartbeat Producer is activated an attempt is made to set a Node Guarding time, the SDO error 0x08000020 (no access available) is sent.

### 3.9 Entries in the object dictionary

The object dictionary manages the configuration parameters. The object dictionary is divided into three areas. Each object can be referenced by its index and subindex (SDO protocol).

- Communication parameters (index 0x1000 to 0x1FFF) contains communications objects to CiA 301, see chap. 6.1, p. 93)
- Manufacturer-specific area (index 0x2000 to 0x5FFF) contains manufacturer-specific objects, see chap. 6.2, p. 101)
- The standardised device profiles area (0x6000 to 0x9FFF) contains objects supported by the Motion Controller (see the documentation of the drive functions)

## CANopen protocol description

### 3.10 Error handling

#### 3.10.1 CAN error

##### CAN overrun (object lost)

If messages are lost, the controller sends the emergency message 0x8110. Bit 4 (communication error) is set in the error register and Bit 7 (CAN overrun) is set in the FAULHABER error register. The emergency message is sent out after a delay. Issuing of the emergency message (0x000) does not retract the error. The respective bits in the error register and in the FAULHABER error register are not cleared down.

##### CAN in error passive mode

If the CAN module of the drive is set to the *Error-Passive* state, the emergency message 0x8120 is sent. Bit 4 (communication error) is set in the error register and Bit 6 (CAN in error passive mode) is set in the FAULHABER error register. The emergency message (0x000) is sent and the error retracted once the drive is restored to the *Error-Active* state.

##### Recovered from Bus Off

If the CAN module of the drive receives a valid message whilst set to the *Bus Off* state, the emergency message 0x8140 is sent. The emergency message reports that the *Bus Off* state has been exited. Bit 4 (communication error) is set in the error register and Bit 9 (Recovered from Bus-Off) is set in the FAULHABER error register. This does not retract the error. The respective bits in the error register and in the FAULHABER error register are not cleared down.

**i** "CAN-Overrun" and "Recovered from bus off" are serious communications errors. The respective bits in the error register and in the FAULHABER error register can be cleared down only by restarting the Motion Controller. Other serious communications errors are:

- Node guarding timeouts
- Heartbeat timeouts

#### 3.10.2 Equipment faults

Tab. 12: FAULHABER Fehlerregister (0x2320)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x2320	0x00	Fault Register	U16	ro	–	FAULHABER error register

The FAULHABER error register contains the most recent errors in bit-coded form. The errors can be masked by selection of the desired types of error via the Error Mask (0x2321) object.

Tab. 13: Error coding

Error bit	Error message	Description
0x0001	Continuous Over Current	Set continuous current limiting exceeded
0x0002	Deviation	Set maximum permissible speed deviation exceeded
0x0004	Over Voltage	Overvoltage detected
0x0008	Over Temperature	Maximum coil or MOSFET temperature exceeded
0x0010	Flash Memory Error	Memory error
0x0040	CAN In Error Passive Mode	CAN controller in error passive mode
0x0080	CAN Overrun (objects lost)	Overrun of the CAN input buffer

## CANopen protocol description


Error bit	Error message	Description
0x0100	Life Guard Or Heartbeat Error	CAN monitoring error
0x0200	Recovered From Bus Off	Exit CAN bus error "Bus off"
0x0800	Conversion Overflow	Computing overflow
0x1000	Internal Software	Internal software error
0x2000	PDO Length Exceeded	PDO length too long, but is processed
0x4000	PDO not processed due to length error	PDO length too short, cannot be processed

All of these errors correspond to an Emergency Error Code. (siehe chap. 3.6, p. 25).

The error mask describes the handling of internal errors depending on the error coding (siehe Tab. 13).

Tab. 14: Error Mask (0x2321)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x2321	0x00	Number of Entries	U8	ro	3	Number of object entries
	0x01	Emergency Mask	U16	rw	0xFFFF	Errors for which an error message is sent
	0x02	Fault Mask	U16	rw	0	Errors for which the state machine of the drive switches into <i>Fault Reaction Active</i> state
	0x03	Error Mask	U16	rw	0x00FF	Errors for which the error output pin is set

 On setting the fault mask (subindex 2) the corresponding bits are also copied into the emergency mask (subindex 1).

### Examples:

- When the fault mask (subindex 2) of object 0x2321 is set to 0x0001 the drive is switched off due to overcurrent are set to an error state.
- When the subindex 3 of object 0x2321 is set to 0, the error output (fault pin) indicates no error. When the subindex 3 of object 0x2321 is set to 0xFFFF, the error output (fault pin) indicates all errors.

### Error Handling (0x2322)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x2322	0x00	Number of Entries	U8	ro	2	Number of object entries
	0x01	Error Delay	U16	rw	200	Error delay time in 1/100 s
	0x02	Deviation	U16	rw	30 000	Permissible speed deviation in rpm

### Explanations

Subindex	Name	Explanation
0x01	Error Delay	Error delay time, which specifies how long one of the following errors has to be queued before it is reported: <ul style="list-style-type: none"> <li>■ Continuous Over Current</li> <li>■ Deviation</li> <li>■ Over Voltage</li> </ul>
0x02	Deviation	Largest, in terms of the amount, permissible deviation of the actual velocity from the target velocity. If this value is exceeded it is reported after the error delay time has expired.

## Functional description

### 4 Functional description

#### 4.1 Drive data

Fundamental properties of the drive system are stored in the Motor Data (0x2350) and Encoder Data (0x2351) objects.

##### Motor data

For the motor monitoring models the following parameters are required:

- Velocity constant
- Connection resistance
- Pole number for brushless motors
- Thermal time constant

These values are already set for integrated units. These values are suitably preassigned for external controls by selecting a motor type in the Motion Manager's Motor Wizard.

##### Motor Data (0x2350)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x2350	0x00	Number of Entries	U8	ro	5	Number of object entries
	0x01	Speed Constant $K_N$	U16	rw	<sup>a)</sup>	Speed constant
	0x02	Terminal Resistance $R_M$	U16	rw	<sup>a)</sup>	Connection resistance
	0x03	Pole Number	U16	rw	$2/4$ <sup>a)</sup>	Number of poles on BL motors (not MCDC)
	0x05	Thermal Time Constant $TW1$	U16	rw	<sup>a)</sup>	Thermal time constant 1

*a) Dependent on the delivery configuration of the Motion Controller*

##### Encoder Data (0x2351)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x2351	0x00	Number of Entries	U8	ro	3	Number of object entries
	0x01	Sensor Type	U8	rw	0	<ul style="list-style-type: none"> <li>▪ 0: Analogue Hall (int. encoder)</li> <li>▪ 1: Incremental encoder (ext.)</li> <li>▪ 10: Incremental + Hall</li> <li>▪ 104: AES-4096 absolute encoder (not MCDC)</li> </ul>
	0x02	Resolution External Encoder	U32	rw	2 048	4-edge resolution of an externally connected incremental encoder
	0x03	Resolution Internal Encoder	U32	ro	3 000	Resolution of internal Hall sensor encoder (not MCDC)

## Functional description

---

- **Sensor Type:**

The following combinations are supported as position encoder systems for brushless motors:

- Analogue Hall sensors (3 000 increments/revolution, fixed)
- Analogue Hall sensors + incremental encoder (resolution depends on the incremental encoder)
- AES encoder (e.g. AES-4096)

An incremental encoder with selectable resolution is supported as the position encoder for DC motors.

- **Resolution External Encoder:**

If using an external incremental encoder its resolution must be given for 4 edge evaluation (4 times the pulse rate).

- **Resolution Internal Encoder:**

If using the analog Hall sensors of the brushless motors as position encoders, a fixed 3 000 pulses per revolution are supplied.

MCDC only uses an external encoder, therefore the sensor type changeover is not available here. In the case of AES controllers the resolution is defined by the sensor type, an external encoder cannot be used here.

## 4.2 Device Control

FAULHABER Motion Control Systems support Device Control according to CiA 402 and the operating modes Profile Position Mode, Profile Velocity Mode, Cyclic Synchronous Position Mode and Homing Mode.

### 4.2.1 State machine of the drive

During the switch-on and switch-off process, the FAULHABER Motion Controller passes through a state machine with several steps. The sequence corresponds to the process defined in the CiA 402 for CANopen drive.

The transitions are controlled by the Controlword (0x6040) of the drive.

The drive behaviour is represented by a state machine. The Controlword controls the transitions, the Statusword shows the states.

## Functional description

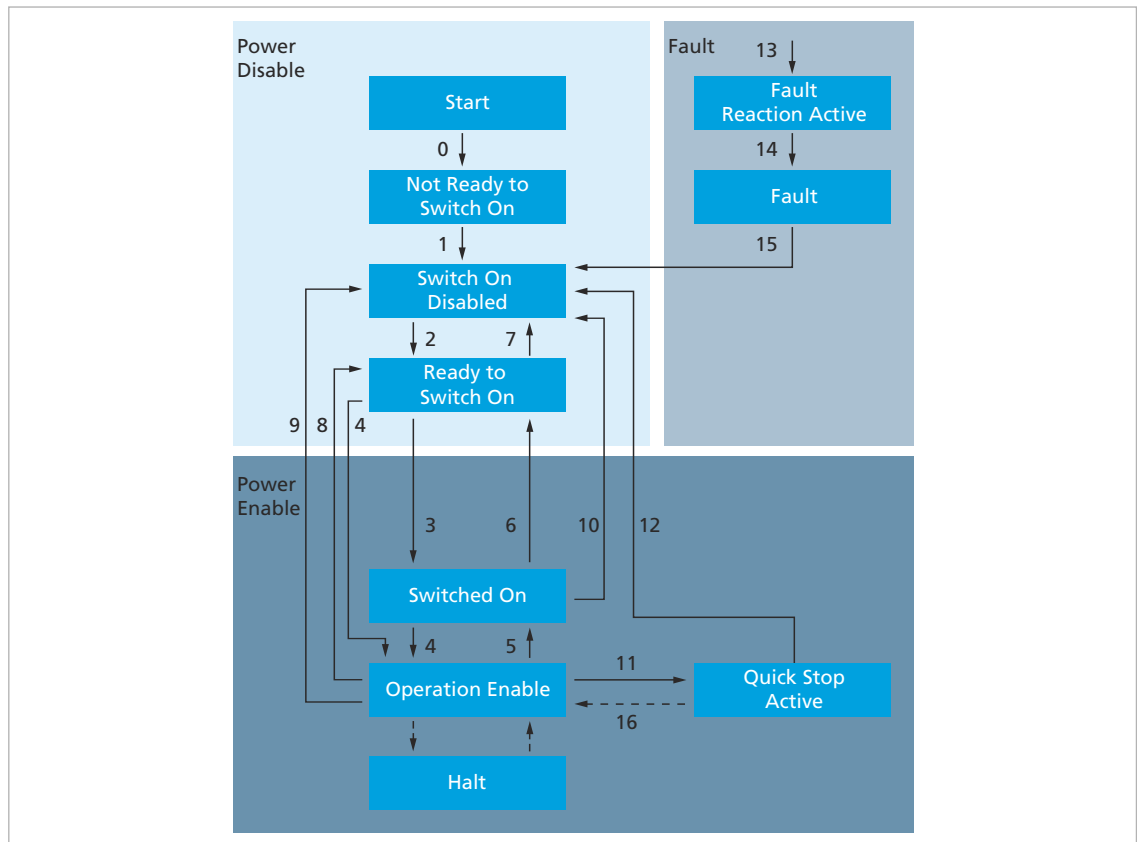


Fig. 7: State machine of the drive

Tab. 15: Command overview

Command	Transitions
Shut Down	2, 6, 8
Switch On	3
Disable Voltage	7, 9, 10, 12
Quick Stop	7, 10, 11
Disable Operation	5
Enable Operation	4, 16
Fault Reset	15

- The *Not Ready to Switch On* state is passed through automatically. The Motion Controller can be configured via the object 0x2503 so that the offsets for the current measurement are automatically readjusted.
- After it has been switched on, the drive is in the *Switch On Disabled* state. The status LED starts to flash green.
- The **Shutdown** command brings the drive into the *Ready to Switch On* state. The option code in the object 0x605B can be used to specify whether the drive should first be brought to a controlled stop.
- The **Switch On** command switches the Motion Controller into the *Switched On* state. The *Switched On* state can be passed through automatically if in the *Ready to Switch On* state the **Enable Operation** command is given directly.

## Functional description

---

- The **Enable Operation** command brings the drive into the *Operation Enabled* state. The transition is performed only if the power supply is within the permissible range.
- The output stage is enabled in the *Operation Enabled* state. The status LED lights up continuously green. The behaviour of the controller depends on the set operating mode.
- The **Disable Operation** command returns the drive to the *Switched On* state. All movement commands outstanding at this stage are cancelled. The Option Code in the object 0x605C can be used to specify whether the drive should first be brought to a controlled stop.
- The **Disable Voltage** command switches the output stage off directly. The motor is not braked. The drive is then in the *Switch On Disabled* state.
- The **Quick Stop** command changes the drive from the *Operation Enabled* state to the *Quick Stop Active* state and slows down the drive with the quick stop ramp. Any outstanding movement commands are discarded when the *Quick Stop Active* state is entered. The state can only be left using the **Disable Voltage** command. The drive can then be put into operation again in the normal manner using the state machine.
- The halt bit in the Controlword allows a drive to be stopped during the course of a movement. The current and following movement jobs are not discarded but merely suspended as long as the halt bit is set. The movement jobs are resumed as soon as the halt bit is unset.
- In response to detection of an error the drive can switch from any state into the fault state. First, the motor is stopped if it is still running. The output stage is then switched off. *Fault Mask* 0x2321.02 can be used to specify the error types for which this is to happen. The user must confirm the error using the *Fault Reset* bit in the controlword. Only then can the user switch on the drive again.

## Functional description

### 4.2.2 Controlword

The commands for performing a change of state are defined by combinations of bits 0–3 in the Controlword. The Controlword is located in the object dictionary under index 0x6040.

#### Controlword

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x6040	0x00	Controlword	U16	rw	–	Drive control

Tab. 16: Overview of the bits of the Controlword and combination permutations of bits 0-3

Bit	Function	Commands for the device control state machine							
		Shut Down	Switch On	Disable Voltage	Quick Stop	Disable Operation	Enable Operation	Fault Reset	
0	Switch On	0	1	X	X	1	1	X	
1	Enable Voltage	1	1	0	1	1	1	X	
2	Quick Stop	1	1	X	0	1	1	X	
3	Enable Operation	X	0	X	X	0	1	X	
4	New Set-Point / Homing Operation Start								
5	Change Set Immediately								
6	Abs/Rel								
7	Fault Reset							0 → 1	
8	Halt								
9	not used								
10	not used								
11	not used								
12	not used								
13	not used								
14	not used								
15	not used								

1 = bit set

0 = bit set

0 → 1 = rising flank, change from 0 to 1

X = bit not used for this command (status irrelevant)



## Functional description

Tab. 17: Meaning of the bits in the Controlword

Bit	Function	Description
0	Switch On	0: No voltage present 1: Power supply being activated
1	Enable Voltage	0: Drive switched off 1: Drive ready
2	Quick Stop	0: Quick Stop enabled 1: Quick Stop disabled
3	Enable Operation	0: Operation disabled 1: Operation enabled
4	New Set-Point	0: No new target position specified 1: New target position specified
5	Change Set Immediately	Not used. New positioning jobs are always started immediately.
6	Abs/Rel	0: Target position is an absolute value 1: Target position is a relative value
7	Fault Reset	0 → 1: Fault Reset
8	Halt	0: Movement can be performed 1: Stop drive

### 4.2.2.1 Example: Enable Operation

Sequence of steps of the transition to bring a drive into the *Operation Enabled* state.

- ✓ The drive is in the *Switch On Disabled* state.
- 1. Enter the Shut Down command (Controlword = 0x00 06).
  - ↪ The drive switches into the state *Ready to Switch On*.
- 2. Enter the Switch On command (Controlword = 0x00 07).
  - ↪ The drive switches into the *Switched On* state.
- 3. Enter the Enable Operation command (Controlword = 0x00 0F).
  - ↪ The drive is now in the *Operation Enabled* state. In this state the set operating mode can be used, using the respective objects.

## Functional description

---

### 4.2.2.2 Example: Resetting the fault state

Sequence of steps of the transition to bring a drive out of the fault state.

1. Enter the Fault Reset command (Controlword = 0x00 08).


↪ The drive switches into the state *Switch On Disabled*.

2. Enter the Shut Down command (Controlword = 0x00 06).

↪ The drive switches into the state *Ready to Switch On*.

3. Enter the Enable Operation command (Controlword = 0x00 0F).

↪ The drive is now in the *Operation Enabled* state. In this state the set operating mode can be used, using the respective objects.

 The current state of the state machine of the drive (see Fig. 7) can be read from bits 0 to 6 of the Statusword.

Only transitions defined in the current states can be performed. Therefore before a change of state, the evaluation of the Statusword must be checked in order to determine the state of the drive.

### 4.2.3 Quick Stop

The drive is decelerated with the deceleration ramp given under Quick Stop Deceleration (0x6085). It then maintains its current position in Profile Position Mode.

## Functional description

### 4.2.4 Statusword

The current state of the drive is represented in bits 0–6 of the Statusword. The Statusword is located in the object dictionary under index 0x6041.

#### Statusword

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x6041	0x00	Statusword	U 16	ro	–	Status display

Tab. 18: Overview of the bits of the statusword and combination possibilities of bits 0-6

Bit	Function	State of the device control state machine							
		Not Ready to Switch On	Switch On Disabled	Ready to Switch On	Switched on	Operation Enabled	Quick Stop Active	Fault Reaction Active	Fault
0	Ready to Switch On	0	0	1	1	1	1	1	0
1	Switched On	0	0	0	1	1	1	1	0
2	Operation Enabled	0	0	0	0	1	1	1	0
3	Fault	0	0	0	0	0	0	1	1
4	Voltage Enabled	X	X	X	X	X	X	X	X
5	Quick Stop	X	X	1	1	1	0	X	X
6	Switch On Disabled	0	1	0	0	0	0	0	0
7	Warning								
8	0								
9	Remote								
10	Target Reached								
11	Internal Limit Active								
12	Set-Point Acknowledge/Speed/Homing Attained								
13	Deviation Error								
14	Not used								
15	Not used								

1 = bit set

0 = bit set

X = bit not used for this command (state irrelevant)

## Functional description

Tab. 19: Meaning of the bits in the Statusword

Bit	Function	Description
0	Ready to Switch On	0: Not ready to switch on 1: Ready to switch on
1	Switched On	0: No voltage present 1: Drive is in the <i>Switched On</i> state
2	Operation Enabled	0: Operation disabled 1: Operation enabled
3	Fault	0: No error present 1: Error present
4	Voltage Enabled	0: Power supply disabled 1: Power supply enabled
5	Quick Stop	0: Quick stop disabled 1: Quick stop enabled
6	Switch On Disabled	0: Switch on enabled 1: Switch on disabled
7	Warning	Not used
8	0	Not used
9	Remote	Not used
10	Target Reached	0: Target position or target speed not yet reached 1: Target Position or Target Velocity reached Set if the drive has reached its target position in Profile Position Mode or has reached its target speed in Profile Velocity Mode. Specification of a new set value deletes the bit.
11	Internal Limit Active	Not used
12	Set-Point Acknowledge <sup>a)</sup>  Speed  Homing Attained	0: New target position not yet adopted (Profile Position Mode) 1: New target position adopted  0: Speed not equal to 0 (Profile Velocity Mode) 1: Speed equal to 0  0: Homing position not yet detected 1: Homing position detected
13	Deviation Error	0: No error 1: Error

a) *is set after receiving a new position command (Controlword with New Set Point). Is reset, when New Set Point has been reset in Controlword (handshake for position command). In the Profile Velocity Mode, the bit at speed is set to 0 and at speed is reset as not equal to 0.*

**i** In the *Fault reaction active* state, the drive is stopped with the brake ramp set in the object 0x6084. In the *Fault* state, an attempt is then made to keep the speed at zero.

## Functional description

### 4.2.5 Selecting the operating mode

The Modes of Operation parameter is used to select the active drive profile. The Modes of Operation Display entry can be used to read back the current mode of operation.

**i** The active operating mode (0x6061) is not always the same as the configured operating mode (0x6060). The PP and PV operating modes are not started until the first set-point value is entered after the choice of operating mode. The Homing operating mode is not started until the rising flank of bit 4 in the controlword after the operating mode has been set via 0x6060.00 = 6.

#### Modes of Operation (0x6060)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x6060	0x00	Modes of Operation	I8	rw	1	Select the operating mode 0: controller not activated 1: Profile Position (position control) 3: Profile Velocity (speed control) 6: Homing (reference run) 8: Cyclic Synchronous Position (position control)

#### Modes of Operation Display (0x6061)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x6061	0x00	Modes of Operation Display	I8	ro	1	Display of the selected operating mode

The meaning of the return values corresponds to the values of the object 0x6060.

## Functional description

### 4.3 Factor Group

The objects of the Factor Group are used to convert internal position values into user-defined units. Internal position values are entered in increments and are dependent on the resolution of the encoder used. User-defined units depend on the encoder resolution and on attached linear reduction.

Current position in user-defined units:

$$\text{Position Actual Value} = \frac{\text{Position Actual Internal Value} \cdot \text{Feed Constant}}{\text{Position Encoder Resolution} \cdot \text{Gear Ratio}}$$

Gearhead ratio:

$$\text{Gear Ratio} = \frac{\text{Motor Revolutions}}{\text{Shaft Revolutions}}$$

Feed units per revolutions at the motor and at the output:

$$\text{Feed Constant} = \frac{\text{Feed}}{\text{Shaft Revolutions}}$$

$$\text{Feed Constant} = \frac{\text{Feed}}{\text{Shaft Movement}}$$

Encoder resolution:

$$\text{Position Encoder Resolution} = \frac{\text{Encoder Increments}}{\text{Motor Revolutions}}$$

Fig. 8 gives the conversion from user units into internal units using the parameters of the Position Factor (0x6093) object. Position Factor only shows an intermediate value, which is calculated from the parameters of the Position Encoder Resolution (0x608F), Gear Ratio (0x6091) and Feed Constant (0x6092) objects. Position Encoder Resolution shows an intermediate value which, depending on the sensor type selected via the Encoder Data (0x2351) object, contains the respective encoder resolution.

## Functional description

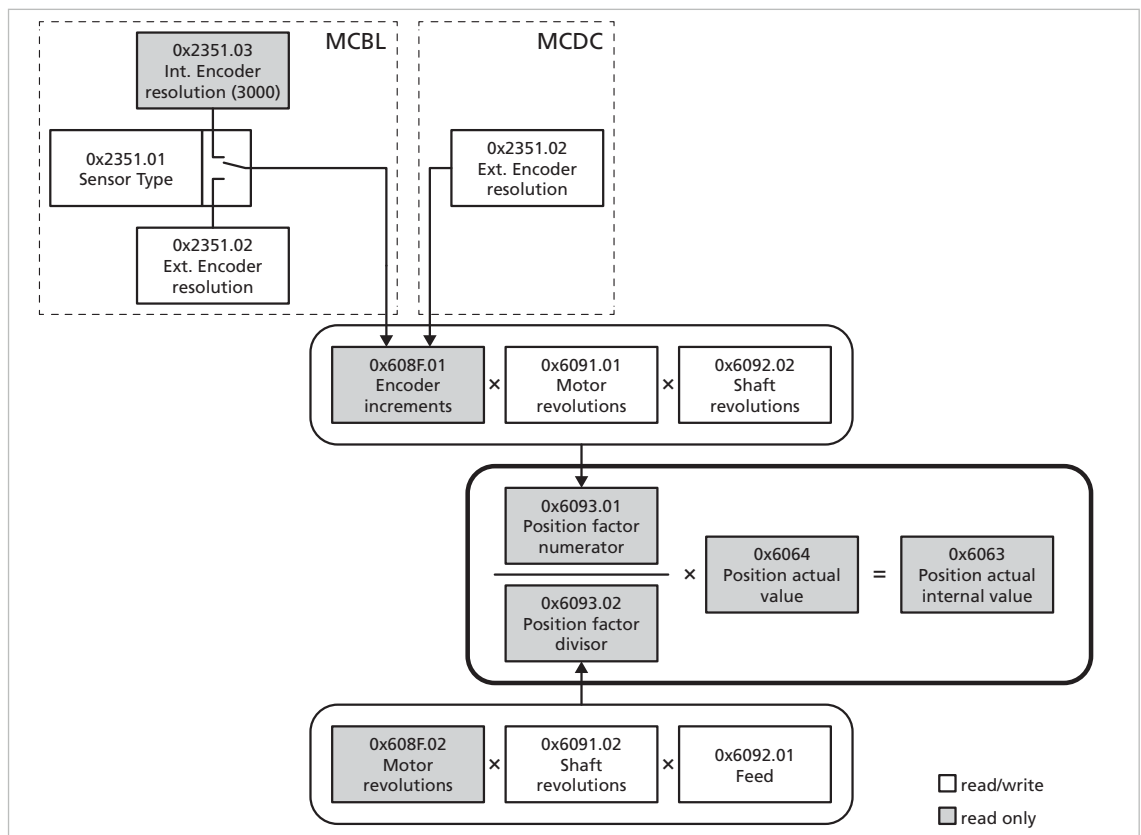


Fig. 8: Conversion from user units into internal units

When setting the conversion factors ensure the following:

- The resulting resolution must be large enough. The maximum resolution with user units is obtained with Gear Ratio = 1 and Feed Constant = Encoder Resolution.
- The counter of the Position Factor must always be smaller than  $2 \times 10^9$ .

As the numerator and the denominator of the actual Position Factor are calculated from the individual conversion factors, a number overflow can occur during the transmission, which is signalled by the SDO error 0x06040047. This error can also occur on switching over the sensor type or on changing the encoder resolution.

If an overflow occurs during subsequent calculations, the emergency telegram 0xFF01 is sent and bit 11 (Conversion overflow) is set in the FAULHABER error register. If there are no longer any conversion errors after correcting the factors, the error is deleted and the emergency telegram 0x0000 is sent.

**i** The Motion Controller basically manages its position parameters in internal units. These are only converted with the help of the Position Factor on writing or reading out.

- Set the Factor Group before the very first parameterisation and if possible do no longer change it.

## Functional description

### Position Encoder Resolution (0x608F)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x608F	0x00	Number of Entries	U8	ro	2	Number of object entries
	0x01	Encoder Increments	U32	ro	3 000/ 2 048 <sup>a)</sup>	Encoder increments
	0x02	Motor Revolutions	U32	ro	1	Number of motor revolutions with the pulse number named in subindex 0x01

a) BL controller/MCDC

The value of the encoder resolution is loaded from the settings in the Encoder Data (0x2351) object and cannot be changed here.

### Gear Ratio (0x6091)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x6091	0x00	Number of Entries	U8	ro	2	Number of object entries
	0x01	Motor Revolutions	U32	rw	1	Number of motor revolutions
	0x02	Shaft Revolutions	U32	rw	1	Number of revolutions of the output shaft

### Feed Constant (0x6092)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x6092	0x00	Number of Entries	U8	ro	2	Number of object entries
	0x01	Feed	U32	rw	3 000/ 2 048 <sup>a)</sup>	Feed in user units
	0x02	Shaft Revolutions	U32	rw	1	Number of revolutions of the output shaft

a) BL controller/MCDC

On delivery the user units are set according to the standard encoder resolution, i.e. positioning takes place in internal units:

- Brushless controllers: 3 000 increments per revolution
- MCDC: 2 048 increments per revolution
- MCBL-AES: 4 096 increments per revolution

**i** If a different encoder resolution is set while keeping the same maximum internal resolution, the Feed Constant must be set to the same resolution. Otherwise the preset user units are maintained.

### Example: user units in angular degrees

Set the Feed Constant to 360 per revolution in order to obtain a resolution of 1°.

### Position Factor (0x6093)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x6093	0x00	Number of Entries	U8	ro	2	Number of object entries
	0x01	Numerator	U32	ro	1	Numerator of the position factor
	0x02	Divisor	U32	ro	1	Denominator (divisor) of the position factor

The position factor is calculated from the Position Encoder Resolution (0x608F), Gear Ratio (0x6091) and Feed Constant (0x6092) and cannot be changed here.



## Functional description

### Polarity (0x607E)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x607E	0x00	Polarity	U8	rw	0	Bit-coded

This object can be used to globally change the direction of rotation, i.e. the default and actual values for position and speed are multiplied by  $-1$ :

- Bit 7 = 1: negative direction of rotation in positioning mode
- Bit 6 = 1: negative direction of rotation in speed mode

**i** Changing the Position Factor and the Polarity affects the set values for Position Range Limit (0x607B), Software Position Limit (0x607D), Position Window (0x6067) and Homing Offset (0x607C), which change accordingly.

- Check these values again and if necessary adjust them.
- In the case of negative Polarity note that the sign of the limits also changes and therefore the Minimum Limit becomes larger than the Maximum Limit.

Functions that read in the complete object dictionary (e.g., the FAULHABER Motion Manager function **Load CANopen configuration file**) should be executed twice to ensure that Position Factor, Minimum Limit and Maximum Limit are correctly interpreted.

## 4.4 Profile Position Mode

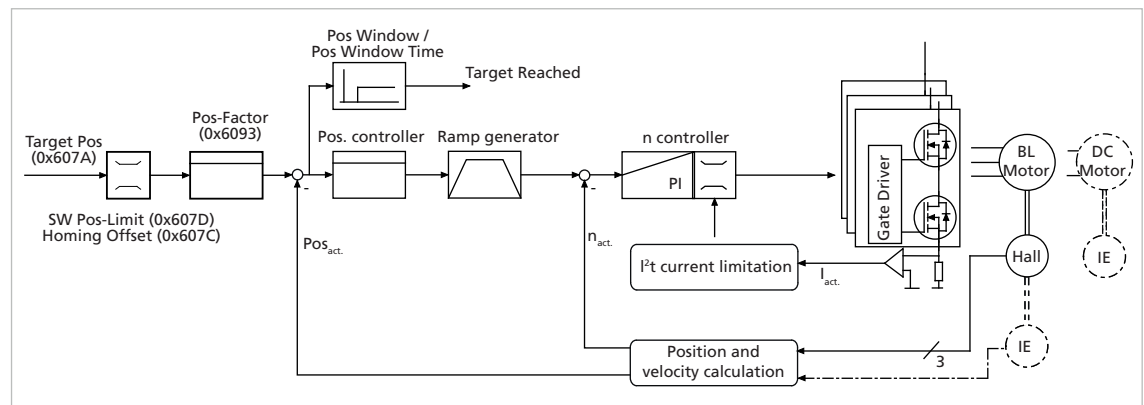


Fig. 9: Controller structure for position control in Profile Position Mode

## Functional description

### 4.4.1 Overview of operating mode

In Profile Position Mode (PP), the drive positions itself to the passed target position.

Prerequisites for operating the drive in PP Mode:

- The Profile Position Mode operating mode is set in parameter Modes of Operation (0x6060).
- The drive is in the *Operation Enabled* NMT state.
- Speed controllers and position controllers are set correctly.

After switching on, a reference run must be generally performed via the Homing Mode, in order to reset the position value to zero at the homing limit switch (see chap. 4.5, p. 55).

A position set value is specified via the Target Position (0x607A) object. The positioning process is started by a change from 0 to 1 of bit 4 (New Set Point) in the Controlword. Bit 6 in the Controlword can be used to additionally specify whether the set value is to be interpreted in absolute or relative terms.

In addition to the set value, the Software Position Limit (0x607D) object can be used to specify range limits for the movement range. These range limits are activated by default, but can be deactivated using the General Settings (0x2338) object.

The set maximum values for acceleration, braking ramp and speed are additionally taken into account.

### 4.4.2 Notification of the higher level control

Attainment of the target position is signalled by bit 10 (Target Reached) in the Statusword of the drive. If the transmission type for the particular PDO is set to 255, the PDO is transmitted asynchronously, triggered by the change in state.

### 4.4.3 Basic settings

The Position Control Parameter Set (0x2332) object can be used to set the proportional amplification and a differential term for the position controller.

Positioning range limits can be defined relative to the reference position using the Software Position Limit (0x607D) object.

The Position Window (0x6067) object can be used to define a window around the target position.

The target position is signalled as being reached using bit 10 (Target Reached) in the Statusword, if the actual position stays within the position window for at least the time set in the Position Window Time (0x6068) object.

#### Software Position Limit (0x607D)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x607D	0x00	Number of Entries	U8	ro	2	Number of object entries
	0x01	Min Position Limit	I32	rw	$-1.8 \cdot 10^9$	Lower limit of the position range
	0x02	Max Position Limit	I32	rw	$1.8 \cdot 10^9$	Upper limit of the position range

The positioning range limits are specified in the units defined by the user and are converted in the internal display using the objects of the Factor Group.

## Functional description

### Position Control Parameter Set (0x2332)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x2332	0x00	Number of Entries	U8	ro	2	Number of object entries
	0x01	Proportional Term PP	U16	rw	a)	Proportional gain of position controller
	0x02	Derivative Term PD	U16	rw	a)	Differential term of position controller

a) Dependent on the configuration of the Motion Controller

### Position Window (0x6067)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x6067	0x00	Position Window	U32	rw	32	Corridor around the target position in user-defined units

Symmetrical area around the target position, which is used for the "Target Reached" message. It is specified in user-defined units, according to the given Position Factor.

### Position Window Time (0x6068)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x6068	0x00	Position Window Time	U16	rw	48	Minimum residence time within the corridor in PP operating mode, until the target position is reported as achieved.

If the drive stays within the range of the Position Window for at least the time set here in milliseconds, bit 10 is set in the Statusword (Target Reached).

#### 4.4.4 Query current values / Position Control Function

The last target position can be read back to index 0x6062 in user-defined units via the Position Demand Value (0x6062) object.

The current position can be read back in internal units using the Position Actual Internal Value (0x6063) object and in user-defined units using the Position Actual Value (0x6064) object (see chap. 6.3, p. 107).

## Functional description

### 4.4.5 Additional settings

#### 4.4.5.1 Incremental encoder as position sensor

By default, the position for BL motors is evaluated using the analog Hall sensors with a resolution of 3 000 increments per revolution.

Alternatively, it is also possible to work with an incremental encoder as position encoder in Profile Position Mode for BL motors. The sensor type and the resolution of the external encoder are set via the 0x2351 object. In the case of DC motors the position is always recorded via an incremental encoder.

#### 4.4.5.2 Ramp generator

The output of the position controller is additionally limited by a ramp generator to the permissible acceleration and deceleration values and the maximum speed.

A trapezoidal profile with linear speed ramps only is supported.

##### Profile Velocity (0x6081) and Max Profile Velocity (0x607F)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x6081	0x00	Profile Velocity	U32	rw	a)	Maximum speed in $\text{min}^{-1}$
0x607F	0x00	Max Profile Velocity	U32	rw	b)	Maximum permitted speed [ $\text{min}^{-1}$ ]

a) *Dependent on the delivery configuration of the Motion Controller*

b) *Dependent on the delivery configuration of the Motion Controller*

Maximum velocity and maximum allowed velocity during positioning.

##### Profile Acceleration (0x6083) and Profile Deceleration (0x6084)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x6083	0x00	Profile Acceleration	U32	rw	30 000	Maximum acceleration [ $1/\text{s}^2$ ]
0x6084	0x00	Profile Deceleration	U32	rw	30 000	Maximum braking rate [ $1/\text{s}^2$ ]

##### Quick Stop Deceleration (0x6085)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x6085	0x00	Quick Stop Deceleration	U32	rw	30 000	Braking ramp value at Quick stop [ $1/\text{s}^2$ ]

#### 4.4.5.3 Velocity controller / current limitation

The controller parameters of the secondary velocity controller can also be adjusted (object 0x2331). In addition the drive can be protected against overload via the peak and continuous current limiting values (object 0x2333) (see chap. 4.6, p. 63).

## Functional description

### 4.4.5.4 Motion control commands

A position set-point is specified using the Target Position (0x607A) object. The positioning process is started by a change from 0 to 1 of bit 4 (New Set-Point) in the Controlword.

Bit 6 in the Controlword can be used to additionally specify whether the set-point should be interpreted as being absolute or relative.

#### Target Position (0x607A)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x607A	0x00	Target Position	I32	rw	–	Position set value in user-defined units

The target position is specified in the units defined by the user and is converted in the internal display using the objects of the Factor Group.

Adoption of a new target position is acknowledged by the drive via the Statusword with set bit 12 (Set-Point Acknowledge). The drive signals that the target position has been reached via the Statusword with set bit 10 (Target Reached). Target Reached remains set until new positioning is started or the output stage is switched off.

If a new setpoint value is specified during positioning (New Set-Point), this is accepted immediately and the drive moves to the new target position. In this way, motion profiles can be run through continuously without having to decelerate the drive to velocity 0 in between times..

Before renewed execution of a positioning operation bit 4 in the Controlword must always be reset, which is acknowledged by the drive with reset bit 12 in the Statusword.

A Statusword with reset bit 12 (Set-Point Acknowledge = 0) signals the readiness to accept a new positioning job.

#### Positioning sequence

- ✓ NMT state *Operational*
- ✓ Drive state *Operation Enabled*
- ✓ Modes of Operation (0x6060) set to Profile Position Mode (1)
  1. Set Target Position (0x607A) to the required value for the target position.
  2. In the Controlword, set bit 4 (New Set-Point) to 1 and set bit 6 (Abs/Rel) depending on whether absolute or relative positioning is required.
    - ↗ The drive responds with bit 12 (Set-Point Acknowledge) set in the Statusword and starts the positioning.
  3. Reset Controlword bit 4 again.
    - ↗ Bit 12 in the Statusword is reset.

As soon as bit 12 in the Statusword has been reset, new positioning can be started by changing the status of bit 4 (New Set-Point) in the Controlword from 0 to 1. Positioning jobs are always executed immediately (Change Set Immediately).

The drive signals that the target position has been reached via the Statusword with set bit 10 (Target Reached).

## Functional description

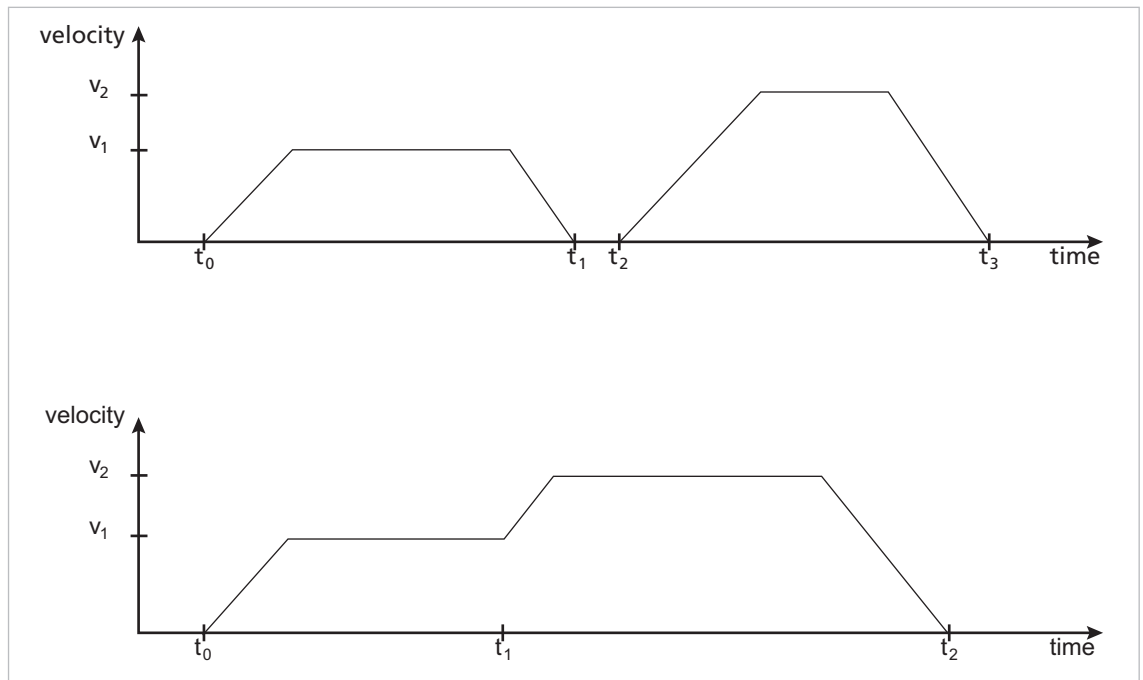


Fig. 10: Positioning sequence

**i** In the case of relative positioning the new target position is added to the last target position.

## Functional description

### 4.5 Homing Mode

After switching on a homing sequence must be performed to reset the position value at the homing limit switch. Object 0x2310 can be used to set which inputs are to be used as homing limit switches (siehe chap. 4.8.1, p. 68).

#### Homing Offset (0x607C)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x607C	0x00	Min Range Limit	l32	rw	0	Offset of the zero position relative to the position of the reference switch

#### Homing Method (0x6098)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x6098	0x00	Homing Method	l8	rw	20	Homing method

#### 4.5.1 Homing methods

The following homing methods defined in DSP402 are supported:

- 1 to 14 (homing with index pulse, if available)
- 17 bis 30 (homing without index pulse)
- 33, 34 (homing at index pulse, if available)
- 35 (homing at the current position)

**i** Limit switches limit the movement range (Negative/Positive Limit Switch), but at the same time can also be used as reference switches for the zero position. A homing switch is a separate reference switch for the zero position.

#### Methods 1 and 17

Homing to the lower limit switch (negative limit switch):

- If the limit switch is inactive, the drive moves first in the direction of the lower limit switch, until its positive flank is detected. Once the limit switch is active, the drive moves upwards away from the limit switch until the negative flank is detected. Under method 1, additionally the drive moves further to the next index impulse at which the home position is set.

## Functional description

### Methods 2 and 18

Homing to the upper limit switch (positive limit switch):

- If the limit switch is inactive, the drive moves first in the direction of the upper limit switch, until its positive flank is detected. Once the limit switch is active, the drive moves downwards away from the limit switch until the negative flank is detected. Under method 2, additionally the drive moves further to the next index impulse at which the home position is set.

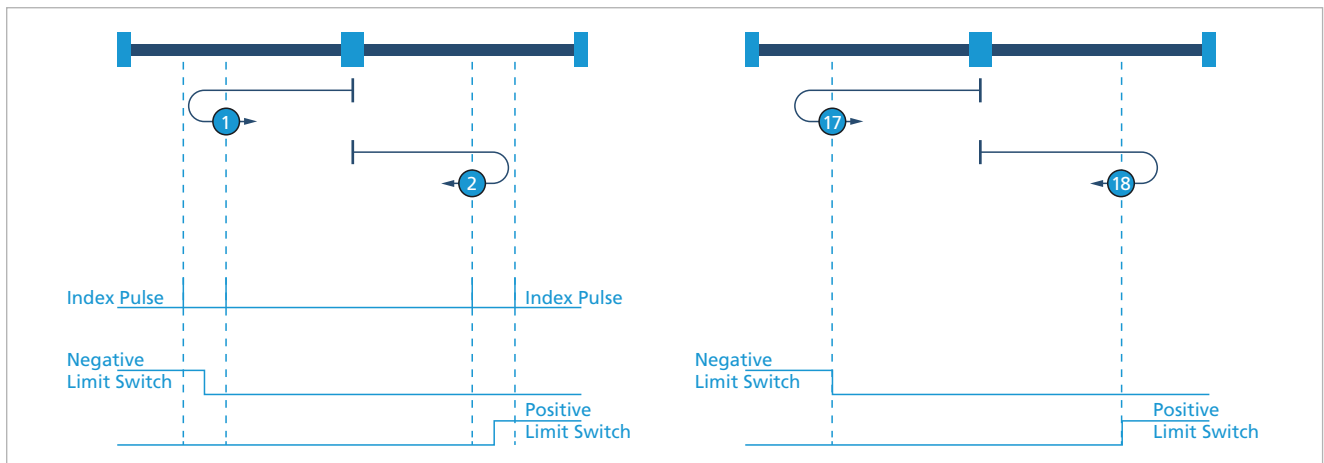


Fig. 11: Homing methods 1, 2, 17 and 18

### Methods 3, 4 and 19, 20

Homing to a positive homing switch

Depending on the status of the homing switch, the drive moves in one direction until the falling (5, 21) or rising (6, 22) flank occurs. In the direction of the upper limit switch there is only one rising flank of the homing switch.

The homing position is at the point where the state of the home switch changes.

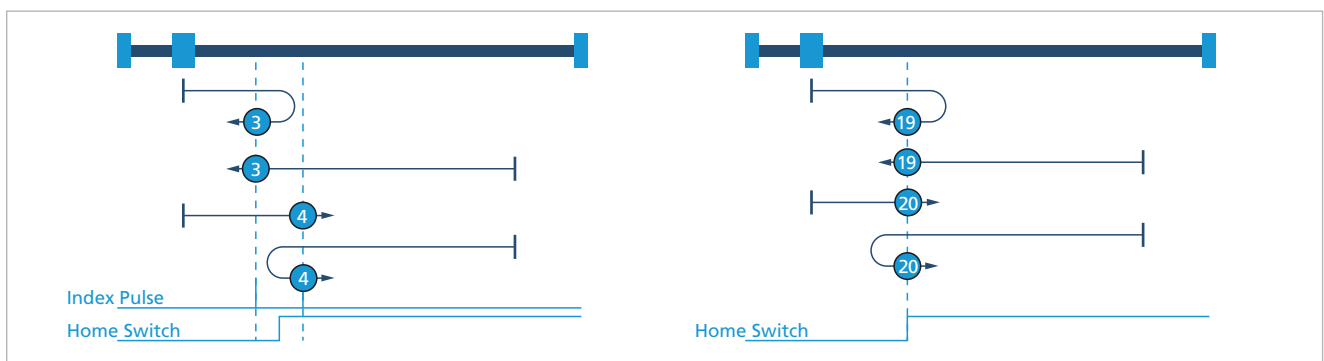


Fig. 12: Homing methods 3, 4, 19 and 20



## Functional description

### Methods 5, 6 and 21, 22

Homing to a negative homing switch (negative home switch).

The initial direction of movement depends on the state of the home switch. The homing position is at the point where the state of the home switch changes. If during a reference run the direction of movement has to be reversed, this is always at the point where the state of the homing switch changes.

The homing position is at the point where the state of the home switch changes.

Under the methods 21 and 22 the homing position is set at a flank. Index impulses are not detected.

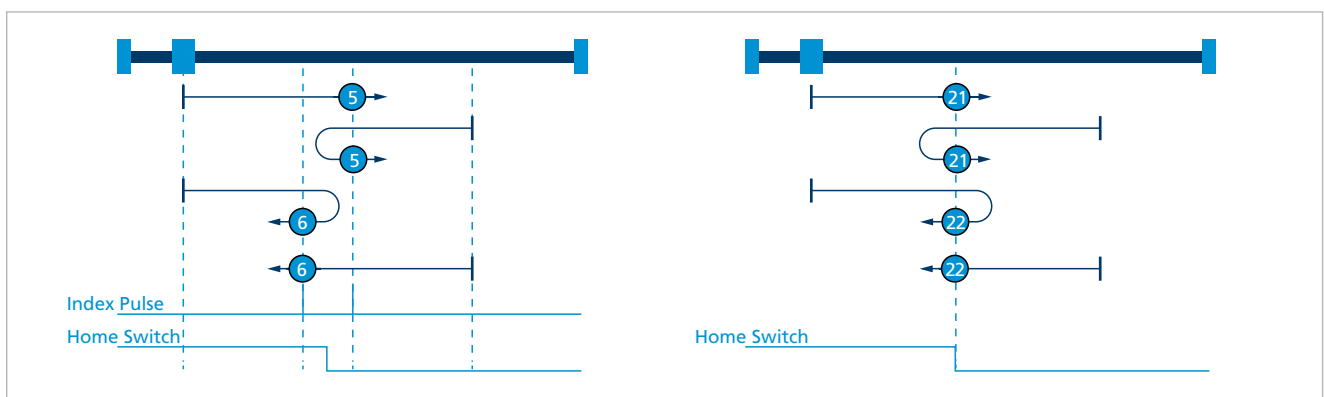


Fig. 13: Homing methods 5, 6, 21 and 22

### Methods 7 to 14 and 23 to 30

Homing at the homing switch (home switch)

These methods use a homing switch which is active only in a particular range of movement. In this case the drive must respond differently to the two flanks of the home switch. Under methods 7 to 14, after detection of the flank the drive moves further to the index impulse, at which point the homing position is then set. Under the methods 23 to 30 the homing position is set at a flank. Index impulses are not detected.

#### ■ Methods 7 and 23:

Homing at the falling flank at the bottom. Starts in a positive direction if the switch is inactive.

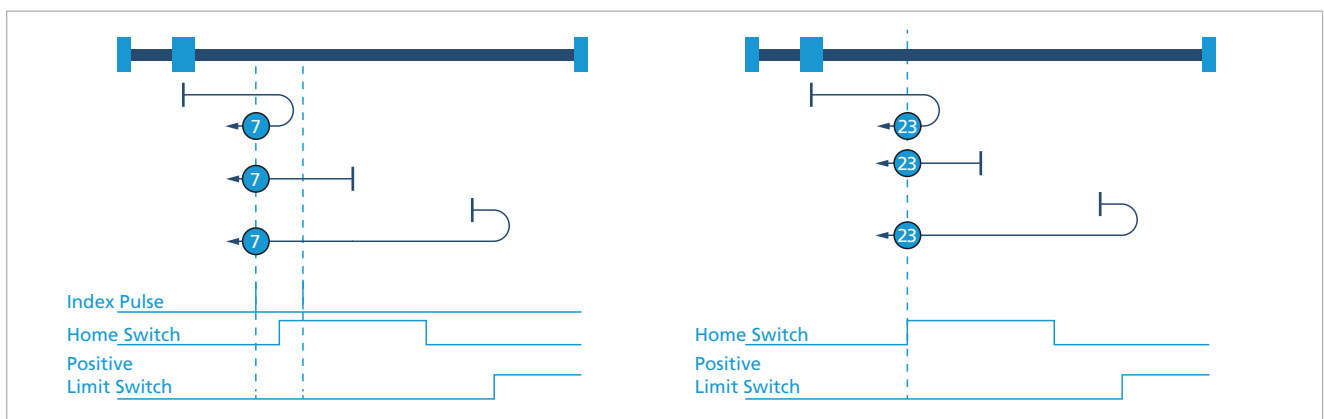


Fig. 14: Homing methods 7 and 23

## Functional description

- Methods 8 and 24:

Homing at the rising flank at the bottom. Starts in a positive direction if the switch is inactive.

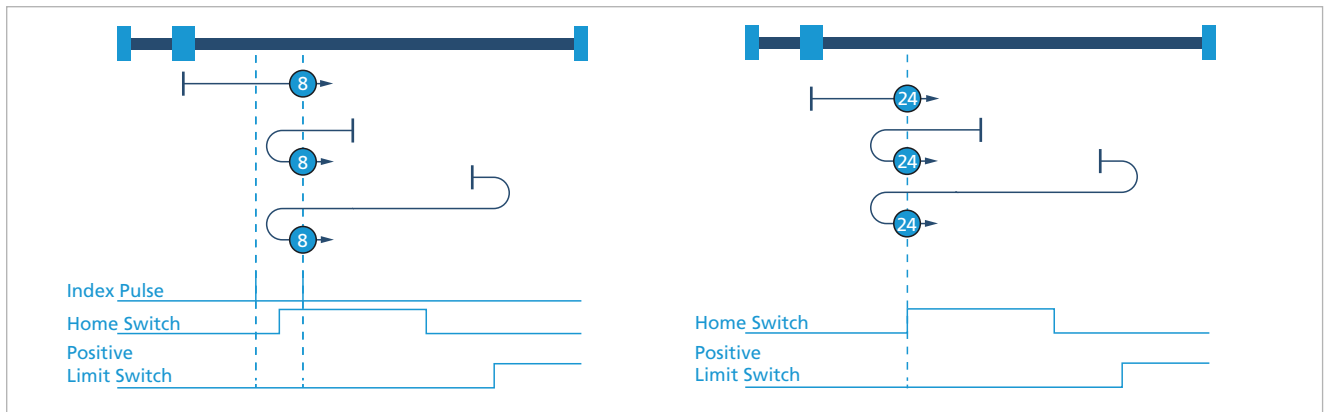


Fig. 15: Homing methods 8 and 24

- Methods 9 and 25:

Homing at the rising flank at the top. Always starts in the positive direction.

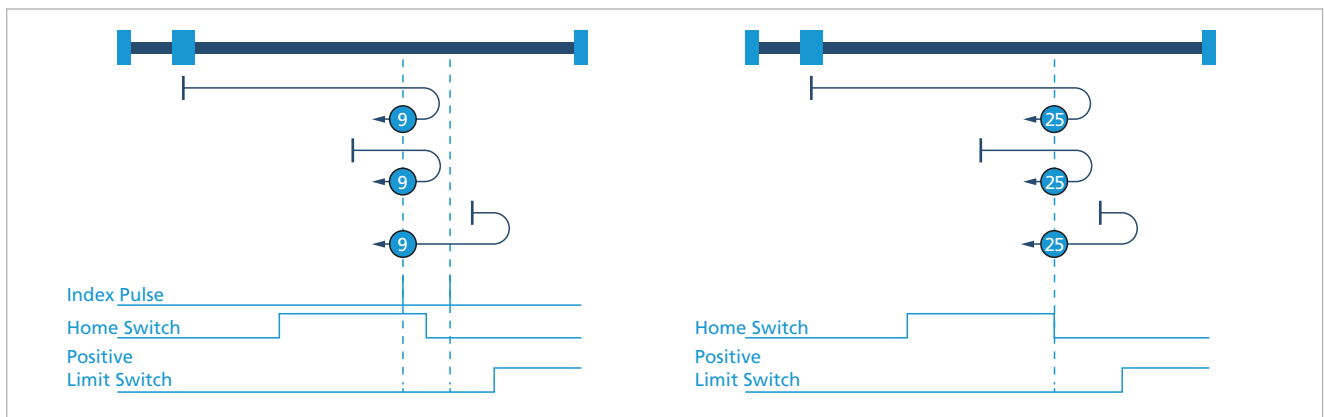


Fig. 16: Homing methods 9 and 25

## Functional description

- **Methods 10 and 26:**  
Homing at the falling flank at the top. Always starts in the positive direction.

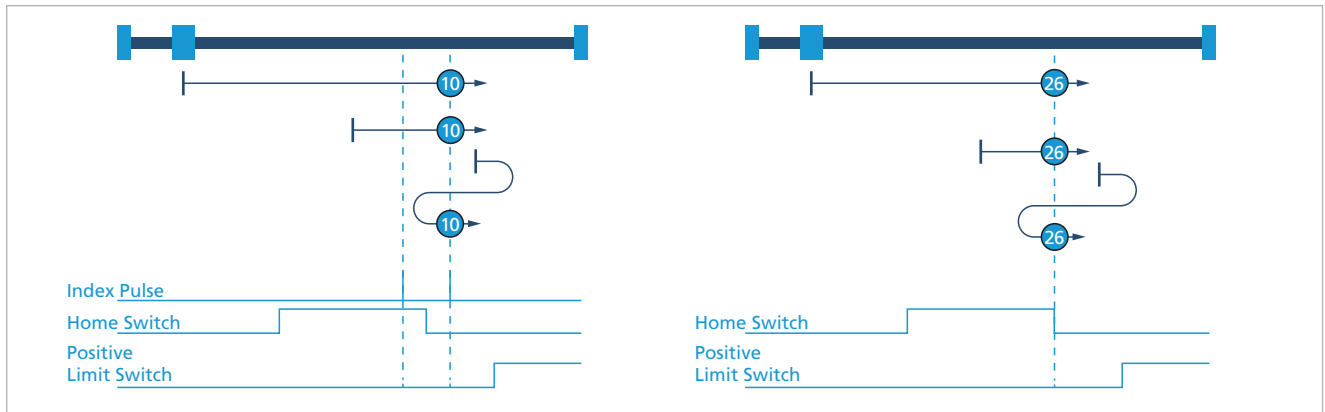


Fig. 17: Homing methods 10 and 26

- **Methods 11 and 27:**  
Homing at the falling flank at the top. Starts in a negative direction if the switch is inactive.

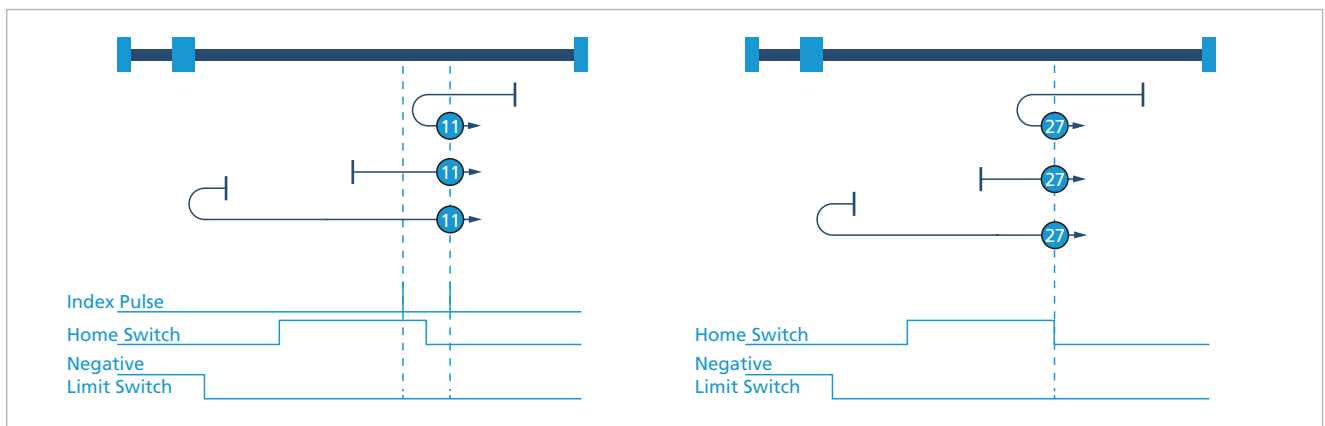


Fig. 18: Homing methods 11 and 27

## Functional description

- Methods 12 and 28:

Homing at the rising flank at the top. Starts in a negative direction if the switch is inactive.

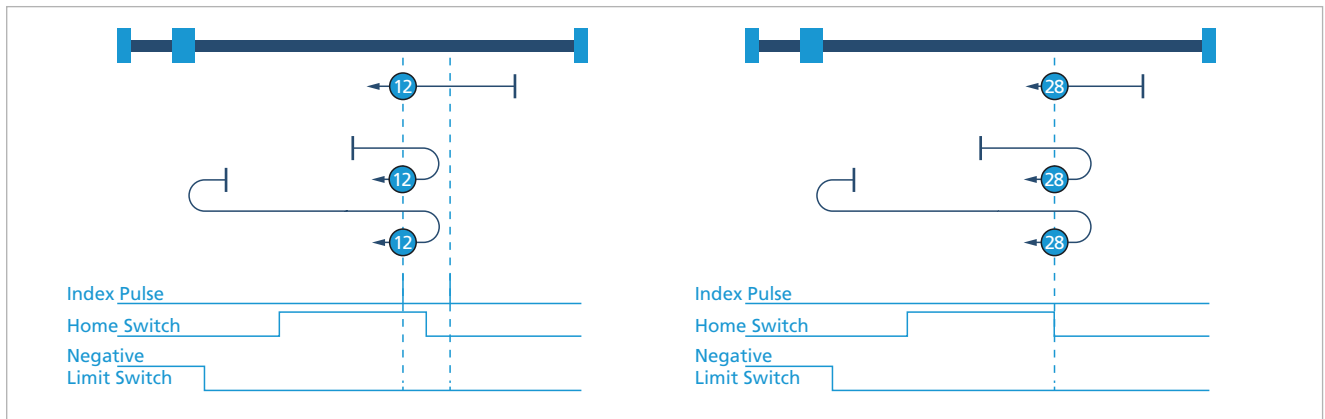


Fig. 19: Homing methods 12 and 28

- Methods 13 and 29:

Homing at the rising flank at the bottom. Always starts in the negative direction.

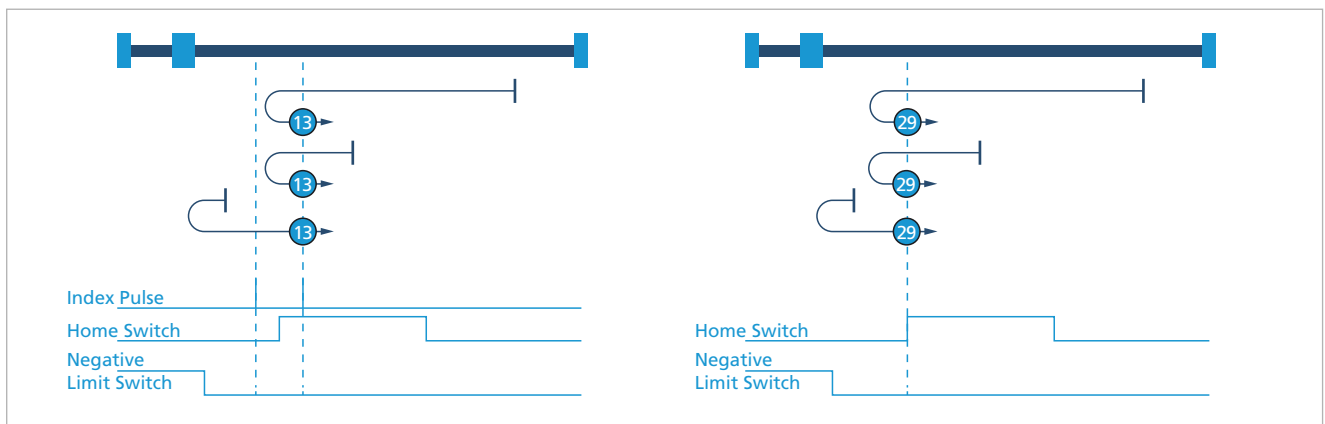


Fig. 20: Homing methods 13 and 29

## Functional description

- **Methods 14 and 30:**  
Homing at the falling flank at the bottom. Always starts in the negative direction.

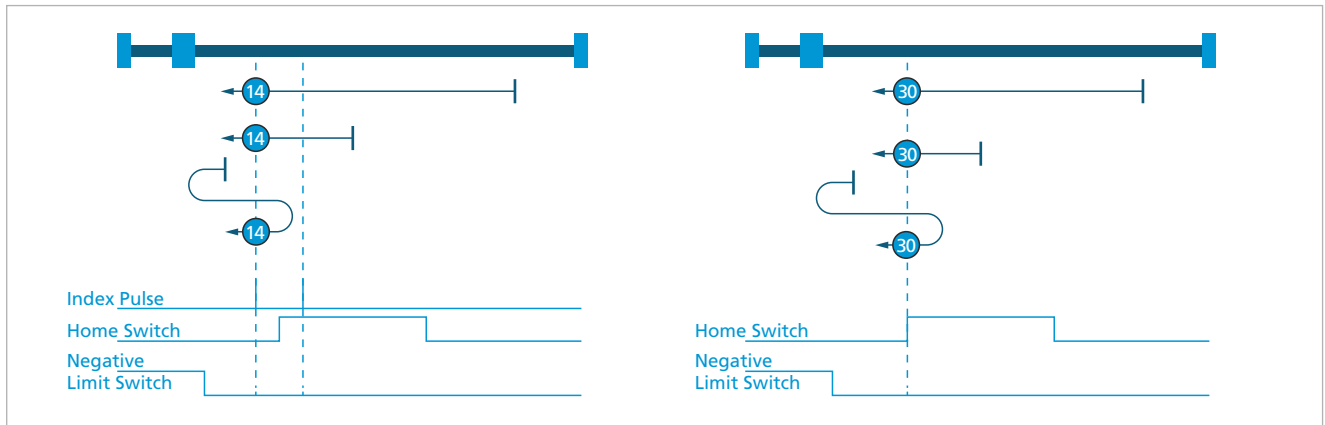


Fig. 21: Homing methods 14 and 30

### Methods 33 and 34

Homing at the index impulse. The drive moves in the negative (33) or positive (34) direction to the index impulse.

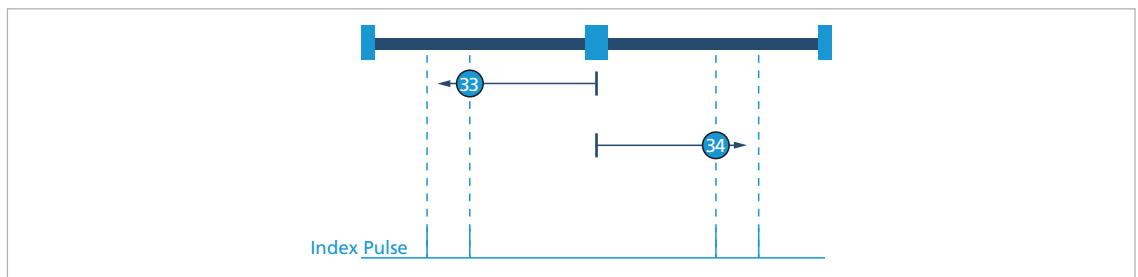


Fig. 22: Homing methods 33 and 34

### Method 35

The position counter is zeroed at the current position.

**i** Limit switches and homing switches are approached in velocity mode, an index pulse in positioning mode. The set range limits (0x607D), provided they are activated for the respective mode (0x2338), are also taken into account.

### Homing Speed (0x6099)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x6099	0x00	Number of Entries	U8	ro	2	Number of object entries
	0x01	Switch Seek Velocity	U32	rw	400	Speed during switch search [ $\text{min}^{-1}$ ]
	0x02	Homing Speed	U32	rw	100	Speed during search for zero [ $\text{min}^{-1}$ ]

### Homing Acceleration (0x609A)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x609A	0x00	Homing Acceleration	U32	rw	50	Acceleration during homing [ $1/\text{s}^2$ ]


## Functional description

---

### 4.5.2 Homing reference run

- ✓ NMT status *Operational*
- ✓ Drive status *Operation Enabled*
- ✓ Modes of operation (object 0x6060) set to Homing mode (6)
- 1. Assign the desired values to the following objects:
  - Homing limit switch (object 0x2310)
  - Homing Method (object 0x6098)
  - Homing speed (object 0x6099)
  - Homing Acceleration (object 0x609A)
- 2. Set bit 4 (Homing Operation Start) in the Controlword to 1.
- ↺ The drive responds with 0 at bit 12 and bit 10 of the Statusword.
- ↺ The drive will now start the reference run.
- ↺ When the homing position is reached and the reference run has been completed, bit 12 and bit 10 of the Statusword will be set to 1.

A currently ongoing reference run can be cancelled by resetting bit 4 in the Controlword, which is acknowledged with set bit 10 (Target Reached) and reset bit 12 (Homing Attained) in the Statusword.

 Before a further reference run can be performed, bit 4 in the Controlword must be reset once again. This causes bit 12 in the Statusword to be reset.

## Functional description

### 4.6 Profile Velocity Mode

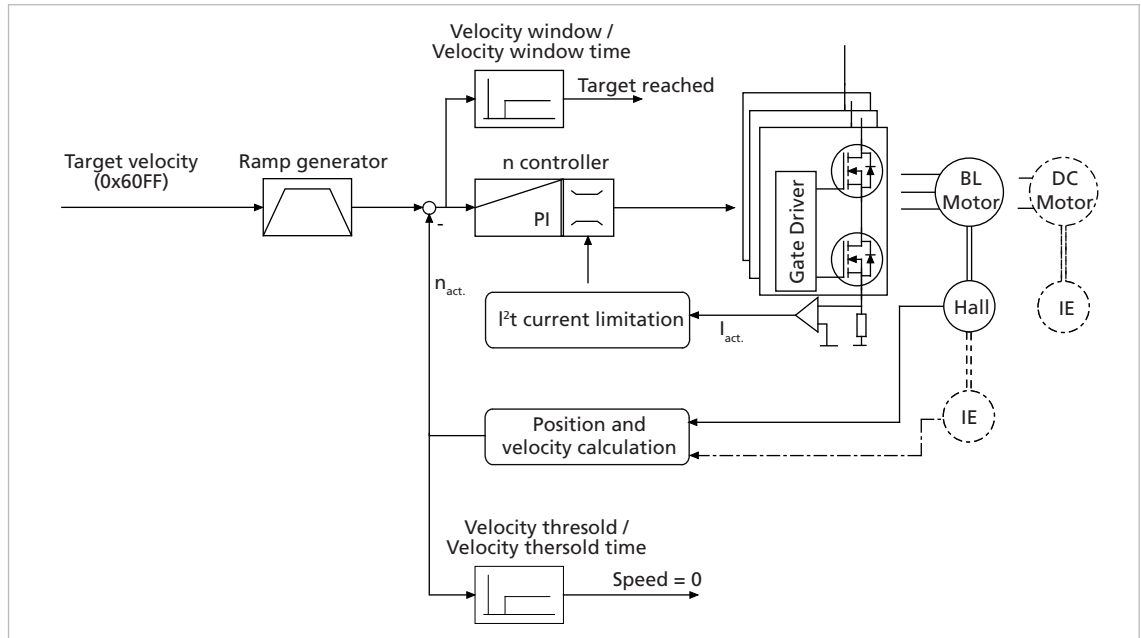


Fig. 23: Controller structure in Profile Velocity Mode

#### 4.6.1 Overview of operating mode

In the Profile Velocity Mode (PV) the speed of the drive is controlled by a PI controller. This ensures that the drive is operated without deviation from the specified values, provided it is not overloaded.

Prerequisites for the drive to be operated in PV Mode:

- The Profile Velocity Mode must be set in the Modes of Operation (0x6060) parameter.
- The drive is in *Operation Enabled* state via its state machine.
- Velocity and position controllers are set correctly.

The target velocity is set via the Target Velocity (0x60FF) object in the object dictionary. In Profile Velocity Mode the drive directly follows each new transferred setpoint value. At the same time, the set maximum values for acceleration, deceleration ramp and speed are also taken into account.

#### 4.6.2 Notification of the higher level control

Attainment of the target velocity is signalled by bit 10 (Target Reached) in the Statusword of the drive. A stopped drive is signalled via bit 12 (Speed = 0). If the transmission type for the particular PDO is set to 255, the PDO is transmitted asynchronously, triggered by the change in state.

## Functional description

### 4.6.3 Basic settings

The Velocity Control Parameter Set (0x2331) object can be used to set the proportional gain and the I term for the velocity controller.

#### Velocity Control Parameter Set (0x2331)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x2331	0x00	Number of Entries	U8	ro	2	Number of object entries
	0x01	Proportional Term POR	U16	rw	a)	Proportional gain of speed controller
	0x02	Integral Term I	U16	rw	a)	Integral term of speed controller

a) *Dependent on the configuration of the Motion Controller*

The sampling rate can be set between 1 and 20 as a multiple of the internal sampling rate using the Sampling Rate (0x2330.01) object.

The internal sampling rate is 0.2 ms. For MCDC and MCBL-AES the sampling rate is 0.1 ms.

#### Filter Settings (0x2330)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x2330	0x00	Number of Entries	U8	ro	2	Number of object entries
	0x01	Sampling Rate	U16	rw	1	Sampling rate factor
	0x02	Gain scheduling	U16	rw	0	1: Reduced controller gain in target corridor during positioning

### 4.6.4 Actual velocity value

In BL motors the current velocity is determined by evaluating the analog hall sensor signals. Alternatively, an external incremental encoder can be used for the velocity recording.

The sensor type and the resolution of the external encoder are set via object 0x2351. In DC motors, the velocity is always determined using the incremental encoder.

### 4.6.5 Additional settings

#### 4.6.5.1 Movement limits

The Software Position Limits (0x607D) can also be activated for velocity mode via the General Settings (0x2338) object.

#### 4.6.5.2 Ramp generator

After specifying a new target speed using the Target Velocity (0x60FF) object, the drive is accelerated or braked to the new speed in the Profile Velocity Mode using the acceleration deposited in the Profile Acceleration (0x6083) object. The parameter is valid in both directions.

#### 4.6.5.3 Current limitation

The peak and continuous current limiting values (object 0x2333) can be used to protect the drive against overload.



## Functional description

### 4.6.6 Motion control commands

A velocity set-point is specified using the Target Velocity (0x60FF) object. Provided the drive is in *Operation Enable* state (see chap. 4.2, p. 37), the drive is accelerated directly to the new target velocity.

The parameter Velocity Window (0x606D) is used to define a window around the target velocity, within which the target velocity is signalled as being reached, if the velocity remains within the target window for at least the time in using the parameter Velocity Window Time (0x606E).

The attained target velocity is signalled in the Statusword by bit 10 (Target Reached).

The parameter Velocity Threshold (0x606F) is used to define a threshold value for the velocity, below which the drive is signalled as being at a standstill, if the velocity remains below the threshold value for at least the time defined using the parameter and Velocity Threshold Time (0x6070).

Stoppage is signalled in the Statusword by bit 12 (Speed=0).

#### Target Velocity (0x60FF)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x60FF	0x00	Target Velocity	I32	rw	–	Speed set value [ $\text{min}^{-1}$ ]

The last set target velocity can be queried using the Velocity Demand Value (0x606B) object. The current velocity value can be queried using the Velocity Actual Value (0x606C) object (see chap. 6.3, p. 107).

### 4.6.7 Complex motion profiles

Evaluation of bits 10 (Target Reached) and 12 (Speed = 0) in the Statusword can be used to deliberately shut down specific velocity profiles. The acceleration is defined using the Profile Acceleration (0x6083) object.

**i** The position limits for Velocity Mode (0x2338.02) are not activated, if the drive is not to stop at the set range limits.

Ensure that the maximum velocity (0x607F) is set to a value smaller than the required target velocity.

## 4.7 Cyclic Synchronous Position Mode

### 4.7.1 Overview of operating mode

In Cyclic Synchronous Position Mode (CSP), the drive immediately positions itself to the passed target position. Unlike Profile Position Mode, no acknowledgement via Controlword/Statusword is necessary. The drive immediately moves to the passed positions.

Prerequisite for this is that only small changes in the set value that come quickly one after the next may be specified.

Multi-axis controls are a typical application case for this operating mode. These must precisely specify the position of each axis and, thus, each drive, at all times in order to avoid collisions.

## Functional description

The Cyclic Synchronous Position Mode is provided in numerous FAULHABER controllers, e.g., in the integrated controllers of the COD series (e.g., 2232...BX4 COD, 2250...BX4 COD) or the separate controllers (MCDC 3002 x CO, MCBL 3002 CO). Whether a given controller actually supports synchronous position mode CSP can be read in object 0x6502. If bit 7 is set, CSP is implemented.

Prerequisites for operating the drive in CSP Mode:

- The Cyclic Synchronous Position Mode operating mode is set in parameter Modes of Operation (0x6060).
- The drive is in the *Operation Enabled* NMT state.

The control structure in Cyclic Synchronous Position Mode is the same as in Profile Position Mode (see chap. 4.4.1, p. 50).

The target position is always specified as an absolute value. Relative positions are not possible.

Limitations set for the drive, such as Max Profile Velocity or deceleration and acceleration ramps, remain active.

### 4.7.2 Inputs and outputs

Inputs from the point of view of the controller:

- Target Position (0x607A)

Outputs:

- Motor Current Actual Value (0x6078)
- Velocity Actual Value (0x606C)
- Position Actual Value (0x6064)

### 4.7.3 Statusword and Controlword

#### Statusword

In Cyclic Synchronous Position operating mode the following operating mode-specific bits are assigned to the Statusword:

Tab. 20: *Operating mode-specific bits of the statusword (Cyclic Synchronous Position mode)*

Bit	Function	Description
10	Reserved	0: No function
12	Drive Follows the Command Value	1: The drive follows the operating value, the target position is used as an input to the position control
13	FollowingError	0: The actual position follows the instructions without a following error. 1: Permissible range for the following error exceeded.

#### Controlword

In Cyclic Synchronous Position operating mode the Controlword is not assigned to any operating mode-specific bits.

## Functional description

### 4.7.4 Settings

The following objects must be set when using this operating mode:

- Target Position (0x607A)
- Operating mode (0x6060 = 8)

Fig. 24 shows all the objects that are effective in this operating mode. The objects shown additionally permit optional settings within this operating mode.

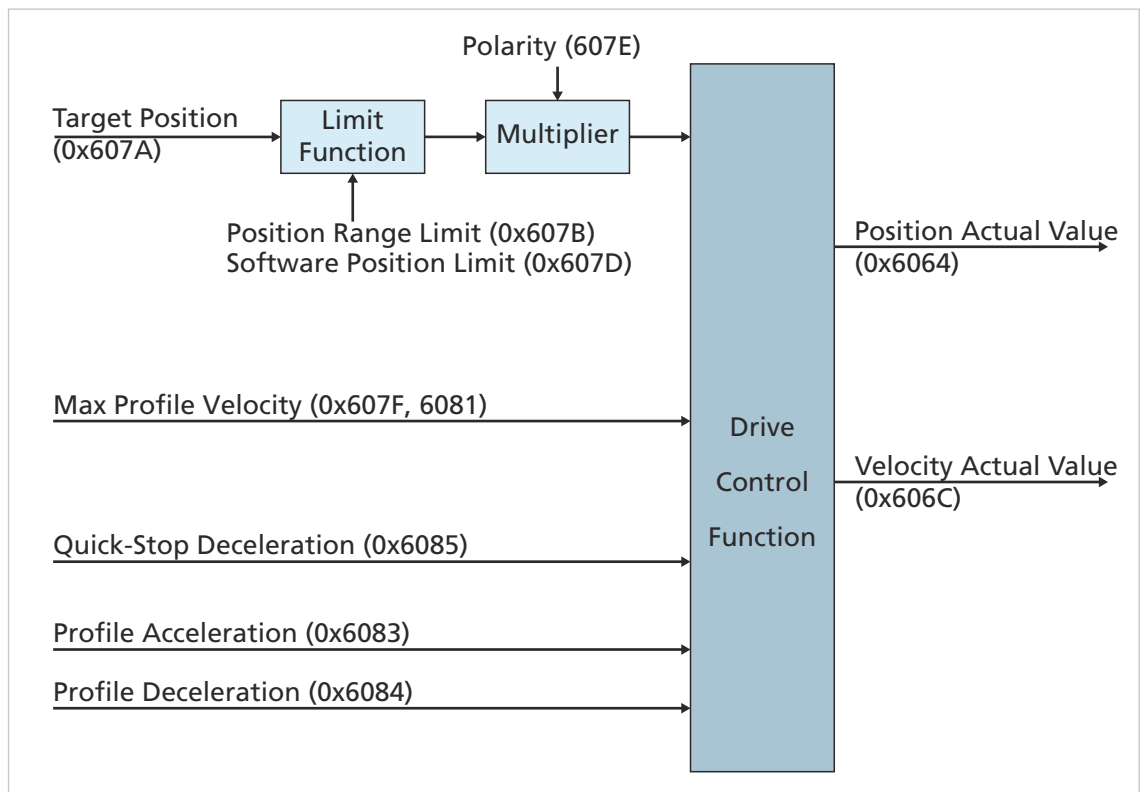


Fig. 24: Overview of all the objects that are effective in CSP operating mode

## Functional description

### 4.8 Inputs/outputs

#### 4.8.1 Limit switch connections and switching level

The following connections can be used as reference and limit switch inputs:

- AnIn
- Fault
- 3. Input
- 4., 5. Input (only MDCD)

The zero crossing of the Hall sensor signals is also available as index pulse for BL motors. The index pulse occurs once or twice per revolution depending on the motor type (two poles or four poles). The index pulse of an external encoder can also be connected to the fault pin, enabling the actual position to be exactly zeroed.

The AnIn and Fault connections are designed as interrupt inputs, which means that they are edge-triggered. All other inputs are not edge-triggered. The signal here must be applied for at least 500 µs to enable it to be reliably detected. The maximum response time to level changes at all inputs is 500 µs.

##### 4.8.1.1 Digital input configuration

###### Switching level

The Input Threshold Level (0x2316) object can be used to set the switching level of all digital inputs to 5V-TTL-compatible or 24V-PLC-compatible (default). The precise details of the respective switching thresholds and the approved voltage ranges are given in the data sheet of the control used.


###### Input Threshold Level (0x2316)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x2316	0x00	Input Threshold Level	U8	rw	1	Switching level <ul style="list-style-type: none"> <li>■ 0: 5V-TTL</li> <li>■ 1: 24V-PLC</li> </ul>

###### Limit switch and homing switch setting

The available digital inputs can each be configured as limit switches or homing switches for use within a DSP402 homing method. The upper and lower limit switches are additionally used as range limit switches, beyond which the drive cannot move (hard blocking).

If lower and upper limit switches are not used for a DSP402 Homing Method method, their Switch Polarity can be defined using the Switch Polarity parameter (rising or falling edge valid). By default, homing methods 1, 2, 17 and 18 assume a positively switching limit switch. If, on the other hand, a negative switching limit switch is to be used the required polarity must be set here accordingly and in addition the Polarity for Homing Limit parameter must be set to 1.

 The input configuration cannot be changed in Homing Mode. For this switch to Profile Position Mode or Profile Velocity Mode.

## Functional description

### Digital Input Settings (0x2310)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x2310	0x00	Number of Entries	U8	ro	6	Number of object entries
	0x01	Negative Limit Switch	U8	rw	0	Selection of the lower limit switch
	0x02	Positive Limit Switch	U8	rw	0	Selection of the upper limit switch
	0x03	Homing Switch	U8	rw	0x07 / 0x1F <sup>a)</sup>	Homing switch
	0x05	Switch Polarity	U8	rw	0x07 / 0x1F <sup>a)</sup>	Polarity of the limit switches <ul style="list-style-type: none"> <li>▪ 1: Positive edge valid</li> <li>▪ 0: Negative edge valid</li> </ul>
	0x06	Polarity for Homing Limit	U8	rw	0	Also use the polarity of the limit switches for DSP402 limit switch homing method

a) BL controller/MCDC

The digital input settings are made with the following bit mask:

Bit:	0	1	2	3	4	5	6	7
Input:	Analogue input	Fault pin	3. Input	4. Input (MCDC only)	5. Input (MCDC only)	–	–	–

### Explanation

Subindex	Name	Explanation
0x01	Negative Limit Switch	Here the input is given, at which the lower limit switch for homing methods 1 and 17 or for a hard blocking function is connected. If the limit switch is activated the drive is stopped and can now only be moved back out of the limit switch in the opposite direction.
0x02	Positive Limit Switch	Here the input is given, at which the upper limit switch for homing methods 2 and 18 or for a hard blocking function is connected. If the limit switch is activated the drive is stopped and can now only be moved back out of the limit switch in the opposite direction.
0x03	Homing Switch	Here the input is given, at which the homing switch for homing methods 3 to 14 and 19 to 30 is connected. Polarity (subindex 0x05) cannot be used here.
0x05	Switch Polarity	The polarity of the notify switch and the hard blocking limit switch can be set here. If the polarity is to be changed with homing methods 1, 2, 17 and 18 too, subindex 0x06 must be set to 1 beforehand..
0x06	Polarity for Homing Limit	Here it is possible to give whether the polarity settings under subindex 0x05 are to be used for the homing methods 1, 2, 17 and 18. In general, the setting can only be set for all inputs (no bitmask coding).

For a description of the homing methods, see chap. 4.5.1, p. 55.

The limit switch functions for the fault pin are only accepted if this is configured as reference input (fault pin function 4). Ensure you always save the setting with SAVE via object 0x2315.01.

In the default configuration all inputs are configured as homing inputs. A homing switch can therefore be connected at each input. However, it is recommended that only the homing input at which the reference switch is actually connected be given.

## Functional description

If a homing method is run without defining the necessary switches in 0x2310 beforehand the homing run does not start.

### 4.8.2 Special functions of the fault pin

The error connection (fault pin) can be configured as input or output for different tasks using the Fault-Pin Settings (0x2315) object:

#### Fault-Pin Settings (0x2315)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x2315	0x00	Number of Entries	U8	ro	3	Number of object entries
	0x01	Fault-Pin Function	U8	rw	0	Function of fault pin <ul style="list-style-type: none"> <li>0: Error Output</li> <li>2: Digital Output</li> <li>4: Reference Input</li> <li>5: Position Output</li> </ul>
	0x03	Digital Output Status	U8	rw/ro <sup>a)</sup>	a)	Change the state of the pin in the Digital Output function <ul style="list-style-type: none"> <li>0: Clear Output</li> <li>1: Set Output</li> <li>2: Toggle Output</li> </ul>

a) *Dependent on the configuration of the Motion Controller*

#### Fault-Pin Function (0x2315.01)

Value	Function	Description
0	Error Output	Fault pin as error output.
2	Digital Output	Fault pin as digital output. The output is set to low level.
4	Reference Input	Fault pin as reference or limit switch input.
5	Position Output	Fault pin as output for display of the condition "target position reached".

In the function as output the connection is designed as an open collector. This means that when the output is set the connection is drawn to low level.

#### 4.8.2.1 Fault pin as error output

In the Error Output function the output is set as soon as an error occurs in the FAULHABER error register and the Errout Mask (0x2321.03) is set to 1 for the corresponding error (see chap. 3.10, p. 34).

##### Additional settings

##### Delayed signalling

In order to hide the transient occurrence of errors, for example, during the acceleration phase, the Error Handling (0x2322.01) object can be used to set an error delay, which indicates how long an error must be queued before it is displayed at the error output (siehe chap. 3.10, p. 34). The error delay affects the errors Continuous Over Current, Deviation and Over Voltage.

##### Example:

Wait 2 seconds before displaying error:

- Error Delay = 200 (object 0x2322.01)

## Functional description

### 4.8.2.2 Fault pin as digital output

In the Digital Output function the fault pin can be used as a universal digital output. The digital output can be set or deleted via object 0x2315.03:

#### Digital Output Status (0x2315.03)

Value	Function	Description
0	Clear Output	Set digital output to low level
1	Set Output	Set digital output to high level
2	Toggle Output	Switch to digital output

### 4.8.2.3 Fault pin as reference input

In the Reference Input function the fault pin has the function of a digital input and can be used as a limit switch or homing switch according to object 0x2310 or can be used to connect the index pulse of an incremental encoder..

### 4.8.2.4 Fault pin as "Position reached" output

In the Position Output function the output is set if the target position has been reached in Profile Position Mode, according to the conditions in Position Window (0x6067) and Position Window Time (0x6068). The output is reset with the next positioning start command.



#### NOTICE!

##### Damage to the electronics

If a voltage is applied to the fault pin while it is not configured as an input the electronics can be damaged.

- ▶ Configure the fault pin as an input first before applying external voltage.

## 4.8.3 Query the input states

The state of the digital inputs can be queried via the Digital Input Status (0x2311) object as direct input level (subindex 0x02) or polarity evaluated (subindex 0x01) in accordance with the entry under 0x2310.05. The state is displayed according to the bit mask of object 0x2310 (see chap. 4.8.1, p. 68).

#### Digital Input Status (0x2311)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x2311	0x00	Number of Entries	U8	ro	4	Number of object entries
	0x01	Input Status	U8	ro	0	Logical states of the digital inputs
	0x02	Input Level	U8	ro	0	Physical states of the digital inputs

The voltage applied at the analog input and at the other inputs can be queried in millivolts or in digits via the Analog Input Status (0x2313) object and Analog Input Status Raw (0x2314) object. The analog input can therefore also supply a measurement signal for the higher-level control.

## Functional description

### Analog Input Status (0x2313)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x2313	0x00	Number of Entries	U8	ro	3/5 <sup>a)</sup>	Number of object entries
	0x01	Input 1 ADC Value	I16	ro	-	Voltage at input 1 [mV] (AnIn)
	0x03	Input 3 ADC Value	I16	ro	-	Voltage at input 3 [mV] (3rd In)
	0x04	Input 4 ADC Value	I16	ro	-	Voltage at input 4 [mV] (MCDC only)
	0x05	Input 5 ADC Value	I16	ro	-	Voltage at input 5 [mV] (MCDC only)

a) BL controller/MCDC

### Analog Input Status Raw (0x2314)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x2314	0x00	Number of Entries	U8	ro	8	Number of object entries
	0x01	Input 1 ADC value raw	I16	ro	-	Digital value at input 1
	0x02	Input 2 ADC value raw	I16	ro	-	Digital value at input 2
	0x03	Input 3 ADC value raw	I16	ro	-	Digital value at input 3
	0x04	Input 4 ADC value raw	I16	ro	-	Digital value at input 4
	0x05	Input 5 ADC value raw	I16	ro	-	Digital value at input 5
	0x06	Input 6 ADC value raw	I16	ro	-	Digital value at input 6
	0x07	Input 7 ADC value raw	I16	ro	-	Digital value at input 7
	0x08	Input 8 ADC value raw	I16	ro	-	Digital value at input 8

This object can also be used to read out the current raw values of the internally used digital inputs.

**i** The objects for querying the input states can be mapped in PDOs. The PDOs can then be queried cyclically via SYNC or RTR. Automatic sending of a PDOs in the event of a state change is not possible, as PDOs can only be sent automatically if the Statusword is changed.



## Functional description

### 4.9 Query of the device state

The Device Status (0x2323) object is available for monitoring the current device status (current temperatures and temperature thresholds in °C).

#### Device Status (0x2323)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x2323	0x00	Number of Entries	U8	ro	4	Number of object entries
	0x01	Housing Temperature	U16	ro	0	Housing temperature [°C]
	0x02	Internal Temperature	U16	ro	0	Coil and MOSFET temperature [°C]
	0x03	Max. Temperature Limit	U16	ro	0	Upper temperature threshold [°C]
	0x04	Min. Temperature Limit	U16	ro	0	Lower temperature threshold [°C]

The values of the upper and lower temperature threshold show the switching on and switching off threshold of the integrated overtemperature protection (see chap. 4.10.4, p. 79).

## Functional description

### 4.10 Technical information

#### 4.10.1 Ramp generator

In all modes the set-point is controlled by the ramp generator.

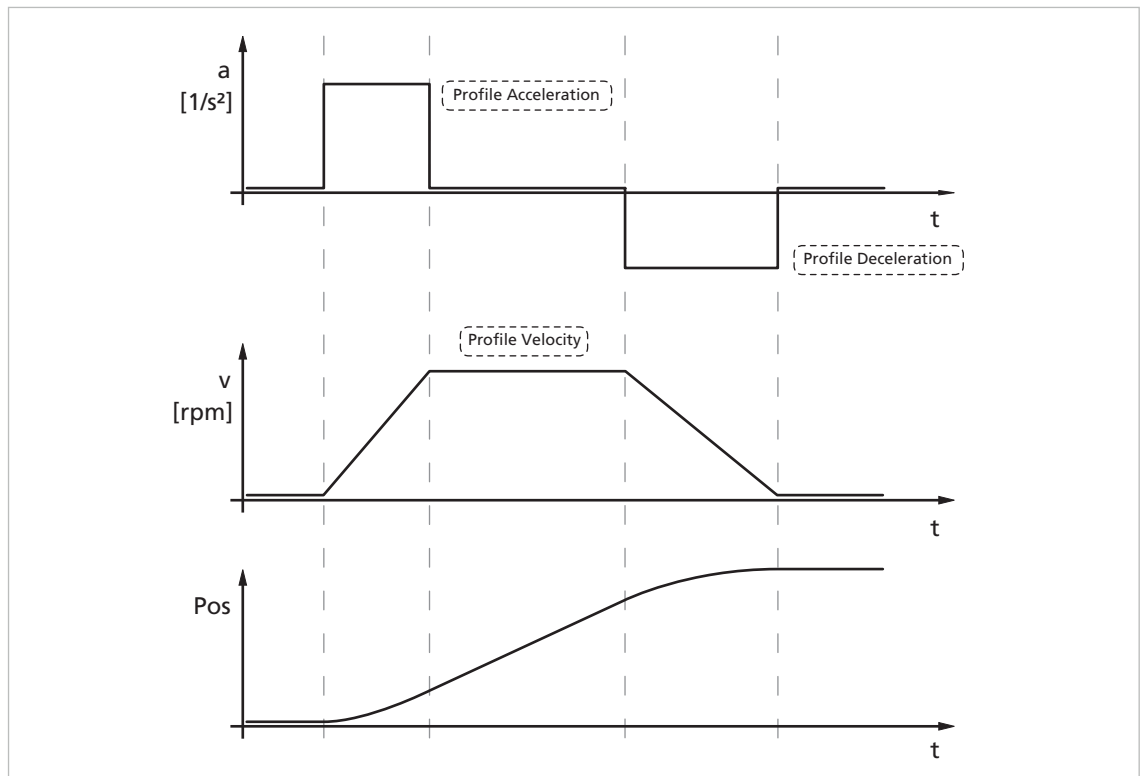


Fig. 25: Basic ramp generator function

This can be used to separately set the parameters for maximum acceleration (Profile Acceleration), maximum delay (Profile Deceleration) and maximum speed (Profile Velocity) for specific applications.

##### 4.10.1.1 Basic settings

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x6081	0x00	Profile Velocity	U32	rw	a)	Maximum speed in $\text{min}^{-1}$
0x6083	0x00	Profile Acceleration	U32	rw	30 000	Maximum acceleration [ $1/\text{s}^2$ ]
0x6084	0x00	Profile Deceleration	U32	rw	30 000	Maximum braking rate [ $1/\text{s}^2$ ]

a) Dependent on the delivery configuration of the Motion Controller

## Functional description

### 4.10.1.2 Ramp generator in Profile Velocity Mode

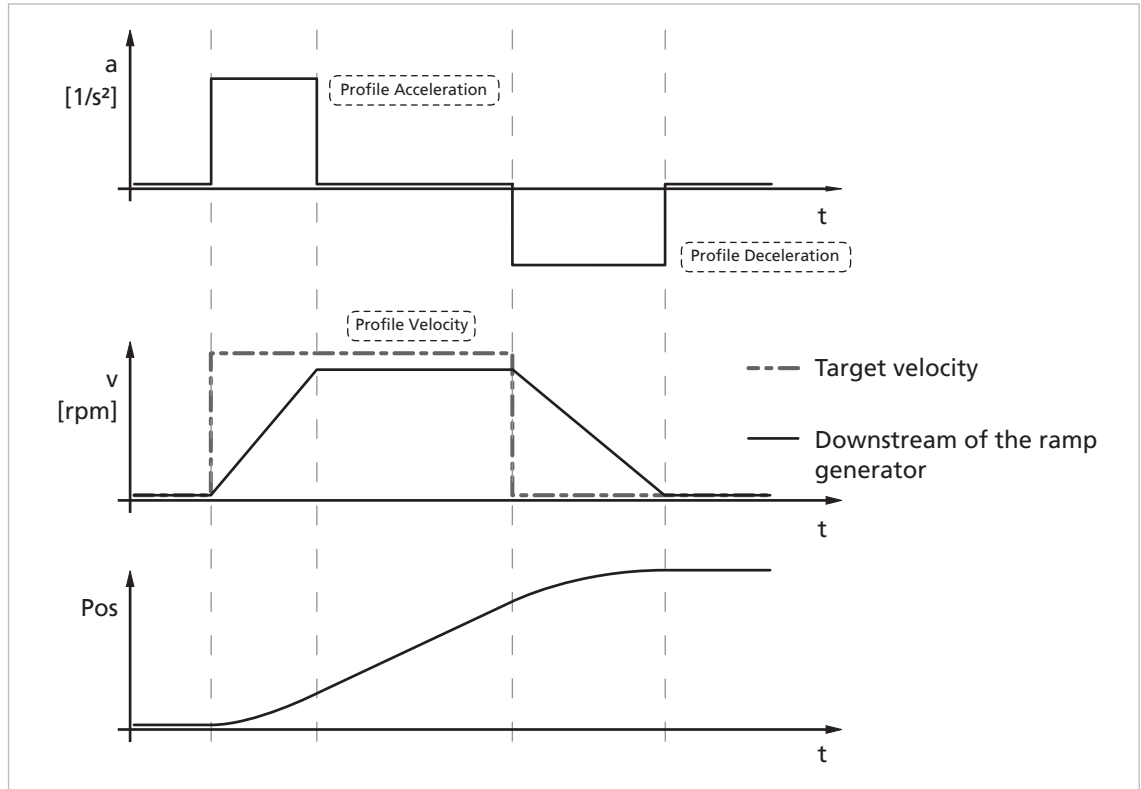


Fig. 26: Intervention of the ramp generator in speed mode

In speed mode the ramp generator acts like a filter on the target speed. The set value is limited to the Profile Velocity value and the set value changes are limited according to the Profile Acceleration and Profile Deceleration.

## Functional description

### 4.10.1.3 Ramp generator in Profile Position Mode

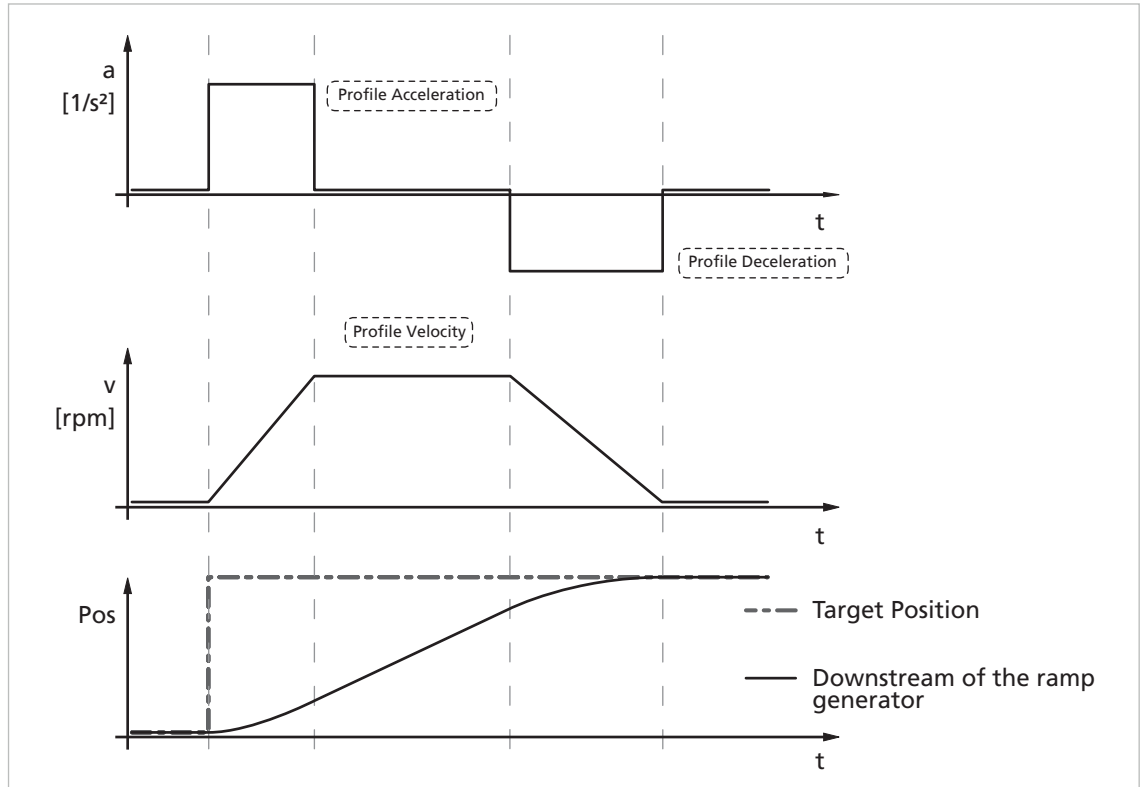


Fig. 27: Intervention of the ramp generator in positioning mode

In positioning mode a preset speed is determined by the position controller from the difference between the target position and actual position.

In the ramp generator, the preset speed output by the position controller is limited to the Profile Velocity and accelerations are limited according to the acceleration ramp Profile Acceleration.

In positioning mode the deceleration process is not extended as, before reaching the limit position, the speed has to be reduced so that the target position can be reached without overshooting.

According to the equation of motion:

$$2a \cdot s = v^2 \rightarrow v_{\max} = \sqrt{2a \cdot s}$$

a: acceleration [m/s<sup>2</sup>]

v: velocity [m/s]

s: remaining distance [m]

the maximum speed  $v_{\max}$  must be limited proportional to the remaining distance.

The allowable delay, or rather the technically possible delay depending on the motor and inertia of the load, is set here using the parameter Profile Deceleration.

## Functional description

### 4.10.2 Sinus commutation

The outstanding feature of FAULHABER motion controllers for brushless motors is their so-called sinus commutation. This means that the specified rotating field is always ideally positioned relative to the rotor. As a result, torque fluctuations can be reduced to a minimum, even at very low speeds. In addition, the motor runs particularly quietly.

The sinus commutation is further enhanced by so-called flat-top modulation, which enables more modulation. As a result, higher no-load speeds are possible.

The parameter Pure Sinus Commutation in the General Settings object can even be used to set the system so that the sinus commutation switches to block commutation in the upper speed range. This full modulation enables the complete speed range of the motor to be utilised.

#### General Settings (0x2338)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x2338	0x00	Number of Entries	U8	ro	3	Number of object entries
	0x01	Pure Sinus Commutation	U16	rw	1	<ul style="list-style-type: none"> <li>0: Full modulation</li> <li>1: Limitation to sinus form (not MDCD)</li> </ul>
	0x02	Activate Position Limits in Velocity Mode	U16	rw	0	1: Also use set positioning limits in speed mode
	0x03	Activate Position Limits in Position Mode	U16	rw	1	0: No range limits in positioning mode

### 4.10.3 Current controller and $I^2t$ current limitation

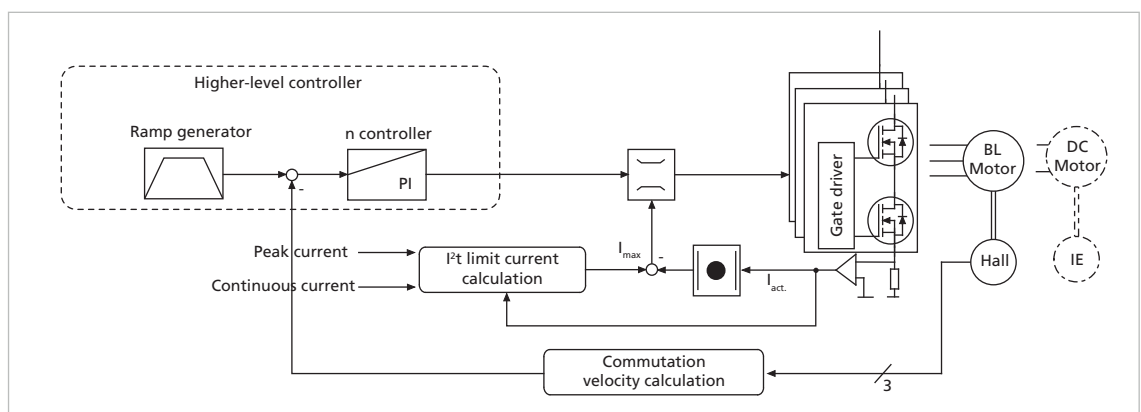


Fig. 28: Intervention of the current limiting controller

The FAULHABER Motion Controllers are equipped with an integral current controller, which enables torque limitation.

The current controller operates as a limitation controller. Depending on the previous loading, the  $I^2t$  current limitation limits to the allowable peak current or continuous current. As soon as the motor current exceeds the currently allowed maximum value the current controller limits the voltage.

Due to its design as a current limiting controller, current control in the thermally relaxed state has no effect on the dynamic of the velocity control. The time response of this limitation can be adjusted using the parameter CI.

## Functional description

The default values for CI limit the current to the allowable value after around 5 ms.

### Current Control Parameter Set (0x2333)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x2333	0x00	Number of Entries	U8	ro	3	Number of object entries
	0x01	Continuous Current Limit	U16	rw	a)	Continuous current limit [mA]
	0x02	Peak Current Limit	U16	rw	a)	Peak current limit [mA]
	0x03	Integral Term CI	U16	rw	a)	Integral term of current controller

a) *Dependent on the configuration of the Motion Controller*

These values are already preset for integrated units. For external controls these values are suitably preassigned for the motor and controller by selecting a motor type in the Motion Manager's Motor Wizard.

### Mode of operation of the current controller

When the motor starts, the peak current is preset as the set-point for the current controller. As the load increases, the current in the motor constantly increases until it finally reaches the peak current. The current controller then comes into operation and limits the current to this set-point.

A thermal current model operating in parallel calculates a model temperature from the actually flowing current. If this model temperature exceeds a critical value, continuous current is switched to and the motor current is regulated to this. Only when the load becomes so small that the temperature falls below the critical model temperature is peak current permitted again.

The aim of this so-called  $I^2t$  current limiting is not to heat the motor above the thermally allowable temperature by selecting a suitable continuous current. On the other hand, a high load should be temporarily possible in order to enable very dynamic movements.

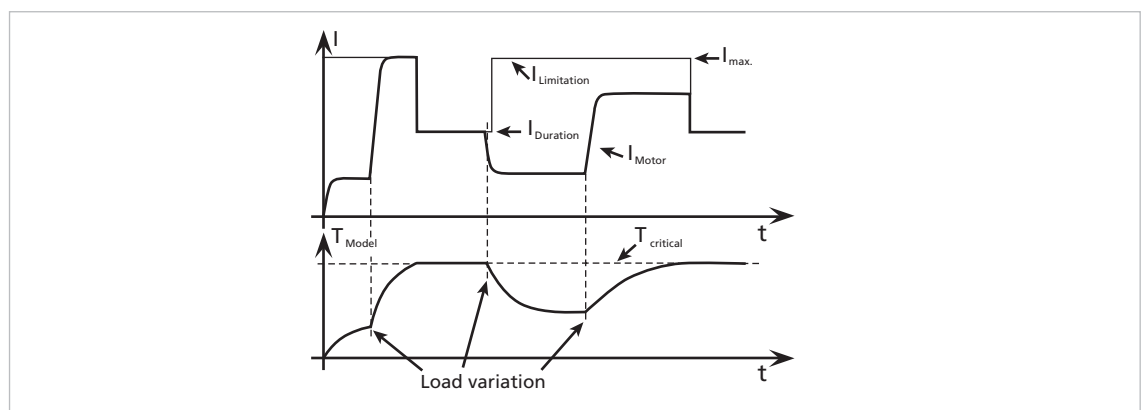


Fig. 29: *Function of the  $I^2t$  current limitation*

The currently used current limiting value (peak or continuous current) can be queried using the Actual Current Limit object:

### Actual Current Limit (0x2334)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x2334	0x00	Actual Current Limit	U16	ro	–	Currently used current limitation value

## Functional description


### 4.10.4 Overtemperature protection

If the MOSFET temperature of the external controllers or the coil temperature of the drives with integrated controller exceeds a preset limit value more than one second, the motor is switched off.

The FAULHABER Error Mask (0x2321) can be used to set the further response to an overtemperature error (EMCY, fault state or error output).

The following conditions must be fulfilled in order to reactivate the motor:

- Temperature below the preset limit value
- Target velocity set to  $0 \text{ min}^{-1}$
- Actual motor speed  $< 50 \text{ min}^{-1}$

 The housing temperature is measured and the power loss concluded from the current measurement. The MOSFET or coil temperature is calculated from these values via a thermal model. In most applications, this method represents a thermal motor protection device.

### 4.10.5 Under-voltage monitoring

If the supply voltage falls below the lower voltage threshold, the power stage is switched off. The Motion Controller remains active. When the voltage returns within the permissible range, the power stage is switched on again immediately.


### 4.10.6 Overvoltage regulation

If the motor is operated as a generator, it produces energy. Usually power supply units are not able to feed this energy back into the power line. For this reason, the supply voltage at the motor increases, and depending on the speed, the allowable maximum voltage may be exceeded.

In order to avoid irreparable damage to components, FAULHABER motion controllers for brushless motors contain a controller which adjusts the displacement angle if a limit voltage (32 V) is exceeded. Motion controllers for DC motors contain a ballast circuit which is activated if a limit voltage (32 V) is exceeded. As a result, the energy generated in the motor is converted, and the voltage of the electronics remains limited to 32 V. This method protects the drive during generating operation and rapid braking.

### 4.10.7 Setting the controller parameters

The preset controller parameters must be optimised in order to optimally adjust the controller to the respective application.

 The digital controller operates with a sampling rate of  $200 \mu\text{s}$ , with  $100 \mu\text{s}$  for MCDC and MCBL-AES. When needed the sampling rate can be increased up to  $2 \text{ ms}$  via the parameter Sampling Rate (0x2330.01).

## Functional description

### 4.10.7.1 Default behaviour

Without further settings, the gain set in the Proportional Term parameter POR is effective for the velocity controller in Profile Velocity Mode.

In Profile Position Mode the gain set via the Proportional Term parameter POR is increased within the target corridor by the value of the Derivative Term parameter PD. This enables faster adjustment to the stoppage in the target position without having to over-stimulate the controller during the transition phenomena. To this end, the parameter PD must be set carefully and should typically be a maximum of 50% of the base value POR. Otherwise there is a risk of instability.

### 4.10.7.2 Available controller parameters

#### Filter Settings (0x2330)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x2330	0x00	Number of Entries	U8	ro	2	Number of object entries
	0x01	Sampling Rate	U16	rw	1	Sampling rate factor
	0x02	Gain scheduling	U16	rw	0	1: Reduced controller gain in target corridor during positioning

#### Gain Scheduling

If Gain Scheduling (0x2330.02) is activated the controller gain POR is reduced successively in positioning mode, as soon as the drive is located within the target corridor (0x6067). This enables a much "gentler" stoppage to be achieved in the target position. As soon as the drive leaves the target corridor, POR is immediately increased back to the set value.

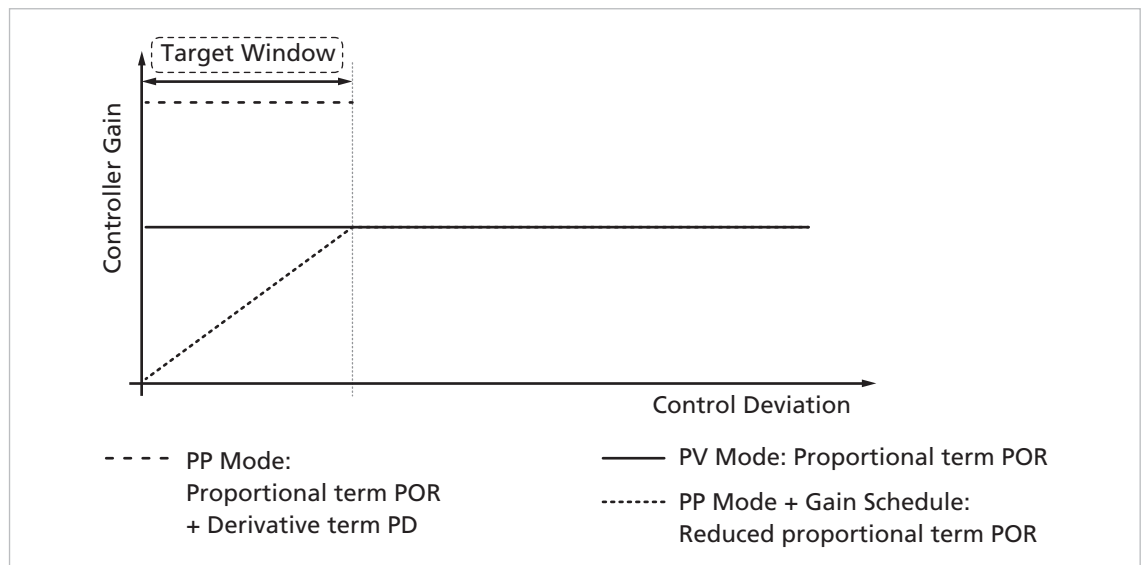


Fig. 30: Controller gain at activated Gain Scheduling



The Gain Scheduling function only becomes active for sampling rates with a factor larger than 3 (sampling rate > 3).



## Functional description

### Velocity Control Parameter Set (0x2331)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x2331	0x00	Number of Entries	U8	ro	2	Number of object entries
	0x01	Proportional Term POR	U16	rw	a)	Proportional gain of speed controller
	0x02	Integral Term I	U16	rw	a)	Integral term of speed controller

a) *Dependent on the configuration of the Motion Controller*

### Position Control Parameter Set (0x2332)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x2332	0x00	Number of Entries	U8	ro	2	Number of object entries
	0x01	Proportional Term PP	U16	rw	a)	Proportional gain of position controller
	0x02	Derivative Term PD	U16	rw	a)	Differential term of position controller

a) *Dependent on the configuration of the Motion Controller*

In the case of integrated units these values are already preset, however, they can be adjusted to the driving load using the Motion Manager's Motor Wizard. These values are suitably preassigned for external controls by selecting a motor type in the Motion Manager's Motor Wizard.

The Tool **Controller Tuning** in Motion Manager can be used to further adjust several controller parameters, in order to optimally adjust the controller to the respective application.

#### 4.10.7.3 Possible procedure

1. Perform the default settings of the Motor Wizard.
2. Optimise the velocity controller:
  - Using the Tool **Controller Tuning**, make velocity jumps between 1/3 and 2/3 of the maximum velocity and at the same time increase the controller gain POR gradually, until the controller becomes unstable.
  - Reduce the controller gain again until reliable stability exists.

Under certain circumstances it may be necessary to optimise the integral term I accordingly.

3. Optimise the position controller:
  - Specify appropriate motion profiles for the application using the Tool **Controller Tuning**.
  - If the system does not function stably with these settings, stability can be achieved by reducing the I term of the velocity controller or reducing the P term of the position controller.
  - Increase the P term of the position controller gradually up to the system's stability limit.
  - Restore the stability either by increasing the D term of the position controller or by reducing the I term of the velocity controller.

## Commissioning

### 5 Commissioning

The drive unit must be connected to a PC via a CAN adapter or a host control with CAN interface in order to make the basic settings for commissioning.

**i** Connection of the CAN interface is described in the technical manual. For the communication setup, ensure that the same transfer rate is set for all nodes (siehe chap. 5.1, p. 82) and the terminating resistances are used.

#### 5.1 Communication settings

FAULHABER Motion Controllers are delivered without valid node address (node ID = 255) and with automatic baud rate detection (AutoBaud) set.

**i** In network mode the network transmission rate in use should be set as the fixed rate.

##### 5.1.1 Setting via the CAN network

For setting via the CAN network the FAULHABER Motion Manager or other configuration tool which supports the LSS protocol (Layer Setting Service and Protocol) to CiA 305 is required.

**i** The FAULHABER Motion Manager must be installed on a PC with a supported CAN interface.

There are two ways of setting the communication parameters:

- An individual drive is connected at the CAN interface of the configuration tool:  
The “LSS Switch Mode Global” without further data allows the drive to be switched to configuration mode, in order to set the node number and Baud rate.
- The drive to be configured is connected via the CAN interface within a network to the configuration tool:  
The “LSS Switch Mode Selective” allows the desired drive to be addressed by inputting the LSS address (Vendor ID, Product code, Revision number, Serial number) and switched to configuration mode, in order to set the node number and Baud rate.

FAULHABER drives of the range MC V3.0 require the following input:

- Vendor ID: 327
- Product code: 3150
- Serial number: See the product sticker

0.0 can always be transferred for the revision number, as this value is ignored in the protocol.

As well as the setting of the node number and Baud rate, the LSS protocol also supports the reading of the LSS addresses of units that are connected and the reading of the node ID setting.

The identifier 0x7E5 is used (by the master) and 0x7E4 (by the slave) for LSS communication.

## Commissioning

After configuration, the Motion Controller saves the set parameters in the EEPROM. They remain available after switching off and on again.

For a detailed description of the LSS protocol please refer to the document CiA 305.

### 5.1.1.1 Setting the node number

- Node numbers 1 to 127 can be set.
- The node ID 255 (0xFF) marks the node as unconfigured. After it is switched on, the node is in the LSS-Init status, until a valid node number is assigned. After a valid node number has been assigned to the node, the NMT initialisation continues.

### 5.1.1.2 Setting the Baud rate

- If the automatic Baud rate detection (AutoBaud) is active, the drive can be used in a network with any transmission rate in accordance with Tab. 21. The Baud rate of the network is detected after no more than 24 telegrams (3 per Baud rate) on the bus cable. The drive then sets itself to match the network Baud rate.
- If the automatic Baud rate detection is active, telegrams cannot be processed until the Baud rate has been detected. If the automatic Baud rate detection is active, it takes correspondingly longer to boot up the system.
- A fixed Baud rate in accordance with Tab. 21 can be set by inputting the index 0 to 4.

Tab. 21: Bit timing parameters

Baud rate	Index
1000 kBit/s	00
800 kBit/s	01
500 kBit/s	02
250 kBit/s	03
125 kBit/s	04
AutoBaud	255

### 5.1.2 Reading the node number

The object 0x2400.01 can be used to read the current setting of the Baud rate (AutoBaud or fixed Baud rate).

Tab. 22: CAN Baud rate index and node number

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x2400	0x00	Baudrate Set	U8	ro	0xFF	Set baud rate

## Commissioning

---

### 5.2 Basic settings

For external Motion Controllers, several basic settings have to be made during the initial start-up to adjust the controller to the connected motor.



#### **NOTICE!**

##### **Destruction of components**

- ▶ Perform the basic settings described in the following.

The following basic settings must be made for external Motion Controllers:

- Motor type or motor data of the connected motor
- Resolution of an external encoder, if used
- Current limitation values, adjusted to the motor type and application
- Controller parameters, adjusted to the motor type and application

In addition, FAULHABER Motion Manager can be used to synchronise the hall sensor signals for smooth start-up and optimisation of the phase angle for the best efficiency.

The configuration must then be adjusted to the respective application for all motion controllers (integrated and external). In particular, the following basic settings are important:

- Operating mode
- Current limiting values
- Controller parameters
- Function of the digital inputs/outputs



#### **NOTICE!**

##### **Destruction of components**

- ▶ If using the Fault Pin as input, program the desired function before applying external voltage.

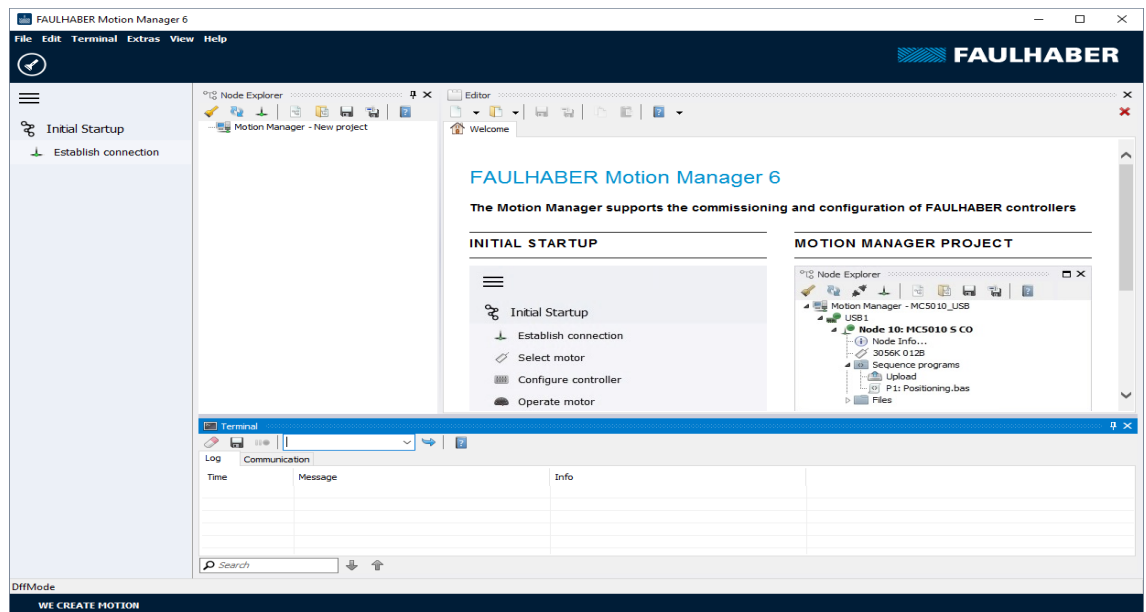
Configuration of these parameters with the help of the FAULHABER Motion Manager is explained in greater detail in the following chapter.

## Commissioning

### 5.3 Configuration using the Motion Manager

FAULHABER Motion Manager PC software provides a simple option for configuring the drive unit and for performing initial tests and optimisation.

The software is available for Microsoft Windows and can be downloaded free of charge from the FAULHABER internet site: [www.faulhaber.com/motionmanager](http://www.faulhaber.com/motionmanager)



Motion controllers with connected motor must be equipped with current limitation values suitable for the motor and suitable controller parameters before being started up.

The motor selection Wizard is available for selecting the motor and the suitable basic parameters.

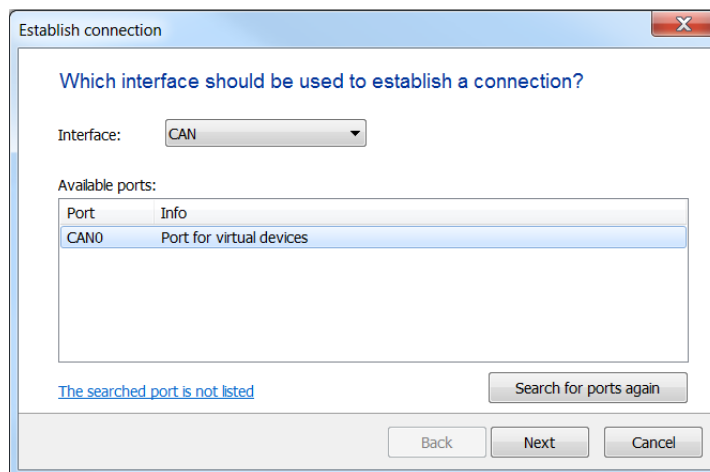
Other settings, e.g. for the function of the fault pin, can be made under the **Configuration – Drive Functions** menu item, where a convenient dialog is provided (see chap. 5.3.3, p. 87). The configuration dialog is also available in the quick access bar of the Motion Manager.

The Tool **Controller Tuning**, with which the controller parameters of the speed and positioning controller can be adjusted to the application, is also provided.

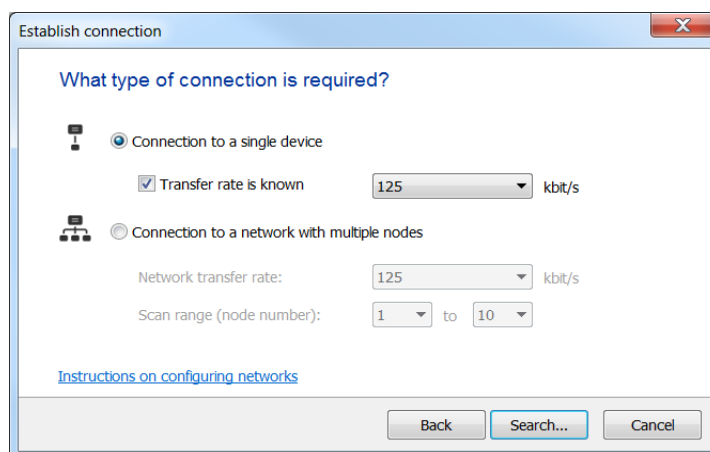
## Commissioning

### 5.3.1 Setting the connection

In order to communicate with the control in question, a connection with the control must be established by the PC on which the Motion Manager is installed. A wizard is available for setting up the communication connection via a one of the supported interfaces. The wizard appears automatically when a new project is created. It can be called up at any time by pressing the **Establish connection** button in the quick access toolbar or via the **Terminal** menu.

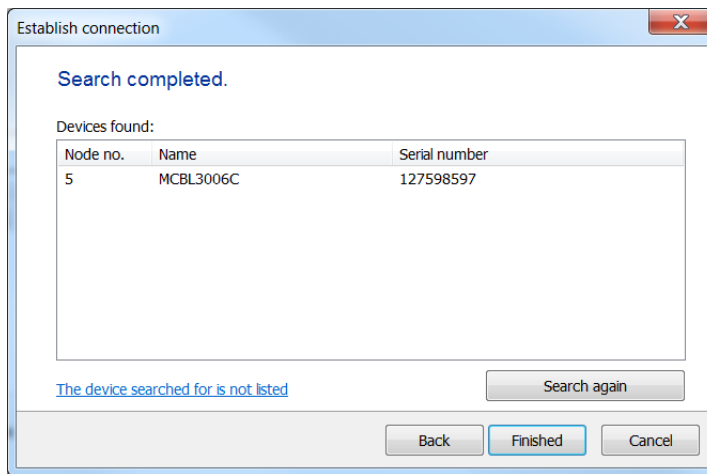


1. Select **CAN** interface.
2. Under **Available ports** select the desired interface connection.
3. Press **Next** button.



4. Select the type of connection and the baud rate.  
Information on the supported CAN interfaces is given in the instruction manual of the Motion Manager or you can contact FAULHABER for information.
5. Press **Search...** button.  
Devices which are already set to a suitable baud rate are then found and displayed by the Motion Manager.

## Commissioning



6. Press **Finished** button.

Devices that have not yet been configured are displayed as “unconfigured LSS node” in the node explorer. Then a valid node number can be set via **Configuration - Connection parameter**.

### 5.3.2 Selecting the motor

External motion controllers must be adjusted to the connected motor. The motor selection wizard is provided for this purpose; it can be opened via the quick access toolbar of the Motion Manager or via **Configuration** under **Initial Startup - Select motor**.

After selecting the required FAULHABER motor from a list and setting the sensor type used, as well as entering an inertia factor for the load to be operated, in addition to the motor and current limiting values, suitable controller parameters are also determined and transferred to the drive.

Refer to the Motion Manager instruction manual for details of how to use the motor selection wizard.


### 5.3.3 Configuring the drive

The motor selection wizard has already set sensible default settings for the motor/sensor combination selected.

A configuration dialog with several pages for further drive configuration and adjustment to the required application is available in the Motion Manager’s quick access bar or under the menu item: **Configuration – Drive Functions...**

No settings are transferred to the drive until the **Send** button is pressed. The current state of the drive is also read back and the dialog is updated accordingly. Invalid combinations of settings are corrected at the same time, as they are not accepted by the drive.

The settings are permanently saved in the drive using the **SAVE** button.

 In addition, settings can be made using the CANopen object browser (menu **Configuration – Object Browser**), which shows the whole object dictionary. The PDO mapping dialog can be used to adjust the PDO data layout to the required application (menu **Configuration – PDO Mapping**).

## Commissioning

### 5.3.3.1 Performing the basic settings

Within the scope of the commissioning, the following settings must be made in the **Basic Settings** tab:

- Operating mode
- Positioning range limits
- Velocity range limit
- Encoder settings
- Commutation settings for BL motors

#### Operating modes

The FAULHABER Motion Controller supports the following operating modes:

- Profile Position Mode (position control)
- Profile Velocity Mode (velocity control)
- Cyclic Synchronous Position Mode (position control)
- Homing Mode (reference run)

#### Positioning range limits

The movement range can be monitored and limited in various operating modes. The limits of this movement range can be given in unit of the actual position.

Object 0x2338.02 of the General Settings can be used to activate the range limits for Profile Velocity Mode too. By default the monitoring is only active in Profile Position Mode and in Homing Mode.

#### Encoder type and optimisation

If an incremental encoder attached to the motor is to be evaluated its effective resolution must be given for 4 edge evaluation. If using the internal encoder, no further inputs are necessary.

A button, with which a wizard for optimisation to the connected motor can be started, is available for adjusting Hall sensor signals and phase angles to the connected motor for externally connected BL motors with analogue Hall sensors.



Ensure that the motor can freely rotate before starting the encoder optimisation.

#### Commutation setting for BL motors

By default the Motion Controller for BL motors uses pure sinus commutation. This means the motor runs with the lowest possible losses and noise.

Alternatively, at higher velocities it is possible to use the object 0x2338.01 of the General Settings to also allow overriding of the output signals similar to block commutation. The whole velocity range of the drive can be used as a result.



On changing between pure sinus commutation and operation with block commutation in the upper velocity range the controller gain is also increased accordingly.



## Commissioning

### 5.3.3.2 Factor Group

Conversion factors of internal user-defined position values can be given in the **Factor Group** tab.

#### Conversion factors for position values

Most position details are preset and displayed in user units. The objects of the Factor Group are available for converting internal units into user units. Factors for a given gear reduction and a feed rate can be given here. The conversion factor for position values is determined from these depending on the respective encoder resolution.

### 5.3.3.3 Setting the controller parameters

Changes to the default set controller and current limiting parameters can be made in the **Velocity Controller** and **Position Controller** tabs of the drive configuration dialog.

In addition, under the **Configuration – Control Parameters...** menu item, there is another dialog in which the parameters can be changed online and the result can be observed directly or can be recorded using the trace function in Motion Manager.

#### Current controller

The Continuous Current Limit parameter can be used to give the thermally allowable continuous current for the application.

Motors and the Motion Controller can be overloaded within certain limits. Therefore, higher currents can also be allowed for dynamic processes. The maximum peak current value is given via the Peak Current Limit parameter.

Depending on the drive's load, the internal current monitoring limits the output current to the Peak Current or the allowable Continuous Current.



#### NOTICE!

##### Destruction of the motor

- ▶ Do **not** set the Continuous Current Limit higher than the thermally allowable continuous current of the motor according to its data sheet.
- ▶ Do **not** set the Peak Current Limit higher than the maximum peak output current of the existing electronics.

The current controller of the Motion Controller operates as a current limiting controller and therefore in an unlimited case has no effect on the dynamics of the velocity control. The speed of the limiting can be set using the parameter CI. If using the default values for your motor, the current is limited to the allowable value after around 5 ms.

If a FAULHABER motor was selected via the Motor Wizard, parameters are already set with which the motor can be operated safely.

Further details are given in chap. 4.10.3, p. 77.

#### Velocity controller

The velocity controller is designed as a PI controller. The following parameters can be set:

- Sampling Rate as multiples of the basic sampling rate of the drive
- Proportional gain POR
- Integral component I

If a FAULHABER motor was selected via the Motor Wizard, parameters are already set with which the motor can be operated safely.

## Commissioning

If the motor is exposed to additional loads, the inertia of the load must be compensated for by a higher proportional term and if necessary slower sampling. In most applications the integral term can remain unchanged.

Further notes see chap. 4.10.7, p. 79.

### Ramp generator

The ramp generator limits the velocity change at the input of the velocity controller via the Profile Acceleration and Profile Deceleration parameters and the maximum default speed via the Profile Velocity parameter.

The Profile Acceleration and Profile Velocity parameters can be freely selected depending on the application. The Profile Deceleration parameter is used to specify the deceleration behaviour in positioning mode. For large loads, the deceleration ramp must be limited using the Profile Deceleration parameter, in order to achieve dead beat (overshoot-free) run-in in the target position.

Further notes see chap. 4.10.1, p. 74.

### Position controller

The position controller is designed as a proportional controller. An additional D term also acts, but only within the target corridor.

The proportional term uses the position deviation in increments to calculate the maximum default velocity for the secondary velocity controller. The ramp generator is used to additionally limit the acceleration and maximum velocity.

Dead beat run-in in the target position can be preferentially achieved by adjusting the deceleration ramp to the load. For a well-attenuated transient condition in the limit position, the parameter PP must be reduced proportionally to the load inertia.

Further notes see chap. 4.10.7, p. 79.

### Target corridor

The Position Window and Position Window Time parameters define a range about the target position and a dwell time within this range, until the „Target Reached“ is set in the Statusword. If the transmission type is set to 255 for the PDO that contains the Statusword (default setting), the target position is signalled via an asynchronously sent PDO. Within this corridor the D term of the position controller is active and the ramp generator is inactive.

### 5.3.3.4 Error handling

In the **Error handling** tab of the drive configuration dialog it is possible to set how the Motion Controller is to respond to certain fault events.

Basically, errors can be signalled via an emergency telegram and via the error output. It is also possible to set that the drive is to switch to DSP402 error state if certain errors occur.

The Error Delay parameter specifies how long an overcurrent, overvoltage or velocity deviation error has to be queued before it is signalled..

#### Maximum permissible velocity deviation

In the Error Handling object the Deviation parameter can be used to specify a maximum permissible velocity for the velocity controller. If this barrier is exceeded for longer than the time set with the Error Delay parameter, an error is signalled via the fault pin or via a CANopen emergency message, depending on the setting in the Error Mask object.

## Commissioning

### 5.3.3.5 Setting inputs/outputs and homing

The **Inputs/outputs and Homing** tab of the drive configuration dialog can be used to specify the function of the digital inputs and outputs and to define the homing settings.

#### Input level and edge

The switching thresholds of the digital inputs are either directly 5 V TTL compatible or are adjusted to the switching level of 24 V PCS outputs. The setting is made via the Input Threshold Level parameter.

Precise details of the thresholds are given in the drive's data sheet.

#### Function of the fault pin

The fault pin can be used both as an input and as an output. The required function can be selected using the **Fault Pin Function** radio buttons.



#### NOTICE!

##### Destruction of the motor

- ▶ Do **not** connect voltage to the fault pin, if the fault pin is configured as a digital output (ERRROUT/DIGOUT).

For the default function as a fault output, the delay time can be specified via the Error Delay parameter in order to suppress the response, e.g. to individual short overcurrent pulses.

In the Position Output function the output displays reaching of the target position as a digital signal (low means target position is reached).

In the Digital Output function the output can be set and deleted from a higher-level control via the digital output status parameter (object 0x2315.03).

In the Reference Input function the fault pin can be used as a reference input for connecting a homing or reference switch.

#### Function of the digital inputs

The displayed input matrix can be used to set use of the available digital inputs:

- Limit switches are limiting switches that when activated block the respective movement direction. The polarity bit can be used to set whether the positive or the negative edge is to be valid for the activation.
- Homing switches are reference switches for resetting the position to zero with certain homing methods. The polarity is predefined by the selected homing method.
- Limit switches can also be used like homing switches to reset the position to zero with certain homing methods. The polarity is also predefined by the selected homing method. However, if, in deviation from DSP402, the polarity is to be changed, this can be set using the polarity bit, provided the **Use polarity for homing mode limit switches** function is activated.

## Commissioning

### 5.3.4 Managing the data set

#### Saving parameters

The settings of a drive can be saved as a backup or as a file for configuration of other drives.

The Motion Manager provides the option of reading out the current drive configuration via the object browser and saving it as an XDC file (XML device configuration file).

#### Transferring parameters to the drive

In the Motion Manager, previously saved XDC files can be opened in the CANopen object browser where they can be edited if necessary and transferred to the drive.



Run the `SAVE` command to permanently save a transferred parameter set in the drive.

### 5.3.5 Diagnosis

#### 5.3.5.1 Status display

The status display is used for continuous checking of the main operating states.

Internal states, error flags and the state of the digital inputs are signalled. In addition, the internally measured housing temperature, the Statusword and further actual values are displayed.

The display is updated by Motion Manager via cyclical querying of the state data.

#### Internal states

The following partially autonomous states of the Motion Controller are displayed:

- Error flags
- Housing temperature
- States of the digital inputs

The current limiting flag is set if the maximum current has been set to Continuous Current Limit by the  $I^2t$  monitoring.

#### Device state (DSP402)

The individual bits of the DSP402 Statusword, the current actual position and the actual velocity are displayed.

#### 5.3.5.2 Trace function

Motion Manager provides the tool **Graphic Analysis** as an additional diagnosis tool with which the internal parameters can be graphically recorded. This enables the dynamic behaviour of the drive to be monitored, which is useful, e.g. for optimisation of the controller parameters.

## Parameter description

### 6 Parameter description

#### 6.1 Communication objects acc. to CiA 301

##### Device Type

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x1000	0x00	Device Type	U32	ro	0x00420192	Indication of the device type

Contains information on the device type, coded in two 16-bit fields:

- Byte MSB (Most Significant Byte): additional information = 0x192 (402d)
- Byte LSB (Least Significant Byte): 0x42 (servo drive, type-specific PDO mapping)

##### Error Register

For the description of the error register, see chap. 3.6, p. 25.

##### Predefined Error Field (error log)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x1003	0x00	Number of Errors	U8	rw	–	Number of errors stored
	0x01	Error Field	U32	ro	–	Last error
	0x02	Error Field	U32	ro	–	Error before last

The error log contains the coding of the last errors that occurred.

- MSB byte: error code, FAULHABER error
- LSB byte: error code, FAULHABER error

The meaning of the error codes is described in chap. 3.6, p. 25.

Writing a 0 to the subindex 0 clears down the error log.

##### COB-ID SYNC

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x1005	0x00	COB ID SYNC	U32	rw	0x80	CAN-object identifier of the SYNC object

##### Manufacturer's Device Name

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x1008	0x00	Manufacturer's Device Name	Vis string	const	–	Device name

Use the segmented SDO protocol to read out the Manufacturer's Device Name.

## Parameter description

### Manufacturer's Hardware Version

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x1009	0x00	Manufacturer's Hardware Version	Vis string	const	–	Hardware version

Use the segmented SDO protocol to read out the Manufacturer Hardware Version.

### Manufacturer's Software Version

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x100A	0x00	Manufacturer's Software Version	Vis string	const	–	Software version

The segmented SDO protocol must be used to read the manufacturer's software version.

### Guard Time

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x100C	0x00	Guard Time	U16	rw	0	Monitoring time for node guarding

Specification of the Guard Time in milliseconds. The value 0 switches node guarding off.

### Life Time Factor

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x100D	0x00	Life Time Factor	U8	rw	0	Time factor for node guarding

The Life Time Factor multiplied by the Guard Time gives the Life Time for the Node Guarding (see chap. 3.8, p. 28). The value 0 switches the Node Guarding off.

### Store Parameters

Tab. 23: Save parameters

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x1010	0x00	Number of entries	U8	ro	9	Number of object entries
	0x01	Save All Parameters	U32	rw	1	Saves all parameters
	0x02	Save comm parameters	U32	rw	1	Save the communication parameters (object directory entries 0x0000 to 0x1FFF)
	0x03	Save app parameters	U32	rw	1	Save the application parameters (object dictionary entries 0x2000 to 0x6FFF)

The "Save Parameters" object saves the configuration parameters into the flash memory. Read access supplies information about the save options. Writing the "Save" signature to the respective subindex initiates the save procedure.

Tab. 24: Signature "save"

Signature	ISO 8 859 ("ASCII")	hex
MSB	e	65 h
	v	76 h
	a	61 h
LSB	s	73 h

## Parameter description



### NOTICE!

The flash memory is designed to accommodate 10,000 write cycles. If this command is executed more than 10,000 times, the correct operation of the flash memory can no longer be guaranteed.

- ▶ Avoid performing frequent saves.
- ▶ After 10,000 save cycles, replace the device.

### Restore Default Parameters

Tab. 25: Restoring the parameters

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x1011	0x00	Number of entries	U8	ro	6	Number of object entries
	0x01	Restore all default parameters	U32	rw	1	Restore all factory settings
	0x02	Restore comm default parameters	U32	rw	1	Restore all factory settings for communications parameters (0x0000 to 0x1FFF)
	0x03	Restore app default parameters	U32	rw	1	Restore all factory settings for application parameters (from 0x2000)
	0x04	Reload user parameters	U32	rw	1	Restore the user's last saved settings for application parameters (from 0x2000)
	0x05	Reload application parameters 1	U32	rw	1	Application parameter set 1 for immediate changes
	0x06	Reload application parameters 2	U32	rw	1	Application parameter set 2 for immediate changes

The object "Restore Default Parameters" loads the standard configuration parameters. The standard configuration parameters are either those as delivered or those last saved. Read access supplies information about the restore options. Writing the "Load" signature to the respective subindex initiates the restore procedure:

Tab. 26: "Load" signature

Signature	ISO 8859 ("ASCII")	hex
MSB	d	64 h
	a	61 h
	o	6Fh
LSB	l	6Ch



The status as delivered may be loaded only when the output stage is switched off.

### COB-ID Emergency Message

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x1014	0x00	COB-ID EMCY	U32	rw	0x80 + node ID	CAN object identifier of the emergency object

### Consumer Heartbeat Time

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x1016	0x00	Number of entries	U8	ro	1	Number of object entries
	0x01	Consumer Heartbeat Time	U32	rw	0	Heartbeat monitoring time

## Parameter description

### Producer Heartbeat Time

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x1017	0x00	Producer Heartbeat Time	U16	rw	0	Heartbeat send time interval

The Producer Heartbeat Time object contains the producer heartbeat time interval in milliseconds. If the value is set to 0, the producer heartbeat function is deactivated.

### Identity Object

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x1018	0x00	Number of entries	U8	ro	4	Number of object entries
	0x01	Vendor ID	U32	ro	327	Manufacturer's code number (FAULHABER: 327)
	0x02	Product Code	U32	ro	3 150	Product code number
	0x04	Serial number	U32	ro	–	Serial number

### Error Behaviour

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x1029	0x00	Number of entries	U8	ro	1	Number of object entries
	0x01	Communication Error	U8	rw	0	Behaviour in the event of communication errors 0 = Pre-operational status 1 = No change of status 2 = Stopped status

In the event of a serious communications error the Motion Controller switches to the *Pre-Operational* NMT status. Subindex 1 allows the behaviour in the event of a serious communications error to be changed.

### Server SDO Parameters

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x1200	0x00	Number of entries	U8	ro	2	Number of object entries
	0x01	COB ID Client to Server (rx)	U32	ro	0x600 + node ID	CAN object identifier for the server RxSDO
	0x02	COB ID Server to Client (tx)	U32	ro	0x580 + node ID	CAN object identifier for the server TxSDO

### Receive PDO1 Parameter

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x1400	0x00	Number of entries	U8	ro	2	Number of object entries
	0x01	COB ID Used by RxPDO1	U32	rw	0x200 + node ID	CAN object identifier for the server RxPDO1
	0x02	Transmission Type	U8	rw	255 (asynchr.)	Type of PDO transmission



## Parameter description

### Receive PDO2 Parameter

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x1401	0x00	Number of entries	U8	ro	2	Number of object entries
	0x01	COB ID Used by RxPDO2	U32	rw	0x300 + node ID	CAN object identifier for the server RxPDO2
	0x02	Transmission Type	U8	rw	255 (asynchr.)	Type of PDO transmission

### Receive PDO3 Parameter

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x1402	0x00	Number of entries	U8	ro	2	Number of object entries
	0x01	COB ID Used by RxPDO3	U32	rw	0x400 + node ID	CAN object identifier for the server RxPDO3
	0x02	Transmission Type	U8	rw	255 (asynchr.)	Type of PDO transmission

### Receive PDO4 Parameter

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x1403	0x00	Number of entries	U8	ro	2	Number of object entries
	0x01	COB ID Used by RxPDO4	U32	rw	0x500 + node ID	CAN object identifier for the server RxPDO4
	0x02	Transmission Type	U8	rw	255 (asynchr.)	Type of PDO transmission

### Receive PDO1 Mapping

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x1600	0x00	Number of Mapped Objects	U8	ro	1	Number of mapped objects
	0x01	RxPDO1 Mapping Entry 1	U32	rw	0x60400010	Pointer to the 16-bit Controlword (0x6040)
	0x02	RxPDO1 Mapping Entry 2	U32	rw	0	
	0x03	RxPDO1 Mapping Entry 3	U32	rw	0	
	0x04	RxPDO1 Mapping Entry 4	U32	rw	0	

### Receive PDO2 Mapping

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x1601	0x00	Number of Mapped Objects	U8	ro	2	Number of mapped objects
	0x01	RxPDO2 Mapping Entry 1	U32	rw	0x60400010	Pointer to the 16-bit Controlword (0x6040)
	0x02	RxPDO2 Mapping Entry 2	U32	rw	0x607A0020	Pointer to the 32-bit Target Position (0x607A)
	0x03	RxPDO2 Mapping Entry 3	U32	rw	0	
	0x04	RxPDO2 Mapping Entry 4	U32	rw	0	

## Parameter description

### Receive PDO3 Mapping

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x1602	0x00	Number of Mapped Objects	U8	ro	2	Number of mapped objects
	0x01	RxPDO3 Mapping Entry 1	U32	rw	0x60400010	Pointer to the 16-bit Controlword (0x6040)
	0x02	RxPDO3 Mapping Entry 2	U32	rw	0x60FF0020	Pointer to the 32-bit Target Velocity (0x60FF)
	0x03	RxPDO3 Mapping Entry 3	U32	rw	0	
	0x04	RxPDO3 Mapping Entry 4	U32	rw	0	

### Receive PDO4 Mapping

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x1603	0x00	Number of Mapped Objects	U8	ro	2	Number of mapped objects
	0x01	RxPDO4 Mapping Entry 1	U32	rw	0x60400010	Pointer to the 16-bit Controlword (0x6040)
	0x02	RxPDO4 Mapping Entry 2	U32	rw	0x257A0020	Pointer to the 32-bit Target Position Internal Value (0x257A)
	0x03	RxPDO4 Mapping Entry 3	U32	rw	0	
	0x04	RxPDO4 Mapping Entry 4	U32	rw	0	

### Transmit PDO1 Parameter

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x1800	0x00	Number of entries	U8	ro	2	Number of object entries
	0x01	COB ID Used by TxPDO1	U32	rw	0x180 + node ID	CAN object identifier of the TxPDO1
	0x02	Transmission Type	U8	rw	255 (asynchr.)	Type of PDO transmission
	0x05	Event timer	U16	rw	0	Minimum time between TxPDO transmission <ul style="list-style-type: none"> <li>▪ 0: Switched off</li> <li>▪ 5...65 000: time in milliseconds</li> </ul>

### Transmit PDO2 Parameter

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x1801	0x00	Number of entries	U8	ro	2	Number of object entries
	0x01	COB ID Used by TxPDO2	U32	rw	0x280 + node ID	CAN object identifier of the TxPDO2
	0x02	Transmission Type	U8	rw	255 (asynchr.)	Type of PDO transmission
	0x05	Event timer	U16	rw	0	Minimum time between TxPDO transmission <ul style="list-style-type: none"> <li>▪ 0: Switched off</li> <li>▪ 5...65 000: time in milliseconds</li> </ul>

## Parameter description

### Transmit PDO3 Parameter

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x1802	0x00	Number of entries	U8	ro	2	Number of object entries
	0x01	COB ID Used by TxPDO3	U32	rw	0x380 + node ID	CAN object identifier of the TxPDO3
	0x02	Transmission Type	U8	rw	255 (asynchr.)	Type of PDO transmission
	0x05	Event timer	U16	rw	0	Minimum time between TxPDO transmission <ul style="list-style-type: none"> <li>▪ 0: Switched off</li> <li>▪ 5...65 000: time in milliseconds</li> </ul>

### Transmit PDO4 Parameter

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x1803	0x00	Number of entries	U8	ro	2	Number of object entries
	0x01	COB ID Used by TxPDO4	U32	rw	0x480 + node ID	CAN object identifier of the TxPDO4
	0x02	Transmission Type	U8	rw	255 (asynchr.)	Type of PDO transmission
	0x05	Event timer	U16	rw	0	Minimum time between TxPDO transmission <ul style="list-style-type: none"> <li>▪ 0: Switched off</li> <li>▪ 5...65 000: time in milliseconds</li> </ul>

### Transmit PDO1 Mapping

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x1A00	0x00	Number of Mapped Objects	U8	rw	1	Number of mapped objects
	0x01	TxPDO1 Mapping Entry 1	U32	rw	0x60410010	Pointer to the 16-bit Statusword (0x6041)
	0x02	TxPDO1 Mapping Entry 2	U32	rw	0	
	0x03	TxPDO1 Mapping Entry 3	U32	rw	0	
	0x04	TxPDO1 Mapping Entry 4	U32	rw	0	

### Transmit PDO2 Mapping

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x1A01	0x00	Number of Mapped Objects	U8	rw	2	Number of mapped objects
	0x01	TxPDO2 Mapping Entry 1	U32	rw	0x60410010	Pointer to the 16-bit Statusword (0x6041)
	0x02	TxPDO2 Mapping Entry 2	U32	rw	0x60640020	Pointer to the 32-bit Position Actual Value (0x6064)
	0x03	TxPDO2 Mapping Entry 3	U32	rw	0	
	0x04	TxPDO2 Mapping Entry 4	U32	rw	0	

## Parameter description

### Transmit PDO3 Mapping

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x1A02	0x00	Number of Mapped Objects	U8	rw	2	Number of mapped objects
	0x01	TxPDO3 Mapping Entry 1	U32	rw	0x60410010	Pointer to the 16-bit Statusword (0x6041)
	0x02	TxPDO3 Mapping Entry 2	U32	rw	0x606C0020	Pointer to the 32-bit Velocity Actual Value (0x606C)
	0x03	TxPDO3 Mapping Entry 3	U32	rw	0	
	0x04	TxPDO3 Mapping Entry 4	U32	rw	0	

### Transmit PDO4 Mapping

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x1A03	0x00	Number of Mapped Objects	U8	rw	2	Number of mapped objects
	0x01	TxPDO4 Mapping Entry 1	U32	rw	0x60410010	Pointer to the 32-bit Position Actual value (0x6064)
	0x02	TxPDO4 Mapping Entry 2	U32	rw	0x606C0020	Pointer to the 32-bit Velocity Actual Value (0x6077)
	0x03	TxPDO4 Mapping Entry 3	U32	rw	0	
	0x04	TxPDO4 Mapping Entry 4	U32	rw	0	

## Parameter description

### 6.2 Manufacturer-specific objects

#### Digital Input Settings (0x2310)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x2310	0x00	Number of Entries	U8	ro	6	Number of object entries
	0x01	Negative Limit Switch	U8	rw	0	Selection of the lower limit switch
	0x02	Positive Limit Switch	U8	rw	0	Selection of the upper limit switch
	0x03	Homing Switch	U8	rw	0x07 / 0x1F <sup>a)</sup>	Homing switch
	0x05	Switch Polarity	U8	rw	0x07 / 0x1F <sup>a)</sup>	Polarity of the limit switches <ul style="list-style-type: none"> <li>▪ 1: Positive edge valid</li> <li>▪ 0: Negative edge valid</li> </ul>
	0x06	Polarity for Homing Limit	U8	rw	0	Also use the polarity of the limit switches for DSP402 limit switch homing method

a) BL controller/MCDC

Settings of the digital inputs in accordance with the bit mask in chap. 4.8, p. 68.

#### Digital Input Status (0x2311)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x2311	0x00	Number of Entries	U8	ro	4	Number of object entries
	0x01	Input Status	U8	ro	0	Logical states of the digital inputs
	0x02	Input Level	U8	ro	0	Physical states of the digital inputs

States of the digital inputs in accordance with the bit mask in chap. 4.8, p. 68.

#### Analog Input Status (0x2313)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x2313	0x00	Number of Entries	U8	ro	3/5 <sup>a)</sup>	Number of object entries
	0x01	Input 1 ADC Value	I16	ro	–	Voltage at input 1 [mV] (AnIn)
	0x03	Input 3 ADC Value	I16	ro	–	Voltage at input 3 [mV] (3rd In)
	0x04	Input 4 ADC Value	I16	ro	–	Voltage at input 4 [mV] (MCDC only)
	0x05	Input 5 ADC Value	I16	ro	–	Voltage at input 5 [mV] (MCDC only)

a) BL controller/MCDC

## Parameter description

### Analog Input Status Raw (0x2314)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x2314	0x00	Number of Entries	U8	ro	8	Number of object entries
	0x01	Input 1 ADC value raw	I16	ro	-	Digital value at input 1
	0x02	Input 2 ADC value raw	I16	ro	-	Digital value at input 2
	0x03	Input 3 ADC value raw	I16	ro	-	Digital value at input 3
	0x04	Input 4 ADC value raw	I16	ro	-	Digital value at input 4
	0x05	Input 5 ADC value raw	I16	ro	-	Digital value at input 5
	0x06	Input 6 ADC value raw	I16	ro	-	Digital value at input 6
	0x07	Input 7 ADC value raw	I16	ro	-	Digital value at input 7
	0x08	Input 8 ADC value raw	I16	ro	-	Digital value at input 8

Raw values read in at the internally used analogue inputs.

### Fault-Pin Settings (0x2315)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x2315	0x00	Number of Entries	U8	ro	3	Number of object entries
	0x01	Fault-Pin Function	U8	rw	0	Function of fault pin <ul style="list-style-type: none"> <li>0: Error Output</li> <li>2: Digital Output</li> <li>4: Reference Input</li> <li>5: Position Output</li> </ul>
	0x03	Digital Output Status	U8	rw/ro <sup>a)</sup>	a)	Change the state of the pin in the Digital Output function <ul style="list-style-type: none"> <li>0: Clear Output</li> <li>1: Set Output</li> <li>2: Toggle Output</li> </ul>

a) *Dependent on the configuration of the Motion Controller*

### Input Threshold Level (0x2316)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x2316	0x00	Input Threshold Level	U8	rw	1	Switching level <ul style="list-style-type: none"> <li>0: 5V-TTL</li> <li>1: 24V-PLC</li> </ul>

### FAULHABER error register (0x2320)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x2320	0x00	Fault Register	U16	ro	-	FAULHABER error register

## Parameter description

### Error Mask (0x2321)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x2321	0x00	Number of Entries	U8	ro	3	Number of object entries
	0x01	Emergency Mask	U16	rw	0xFFFF	Errors for which an error message is sent
	0x02	Fault Mask	U16	rw	0	Errors for which the state machine of the drive switches into <i>Fault Reaction Active</i> state
	0x03	Errout Mask	U16	rw	0x00FF	Errors for which the error output pin is set

The error coding described in chap. 3.10, p. 34 applies for the FAULHABER error register and the Error Mask.

### Error Handling (0x2322)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x2322	0x00	Number of Entries	U8	ro	2	Number of object entries
	0x01	Error Delay	U16	rw	200	Error delay time in 1/100 s
	0x02	Deviation	U16	rw	30 000	Permissible speed deviation in rpm

Additional settings for error handling.

### Device Status (0x2323)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x2323	0x00	Number of Entries	U8	ro	4	Number of object entries
	0x01	Housing Temperature	U16	ro	0	Housing temperature [°C]
	0x02	Internal Temperature	U16	ro	0	Coil and MOSFET temperature [°C]
	0x03	Max. Temperature Limit	U16	ro	0	Upper temperature threshold [°C]
	0x04	Min. Temperature Limit	U16	ro	0	Lower temperature threshold [°C]

Output of the current device state.

### Filter Settings (0x2330)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x2330	0x00	Number of Entries	U8	ro	2	Number of object entries
	0x01	Sampling Rate	U16	rw	1	Sampling rate factor
	0x02	Gain scheduling	U16	rw	0	1: Reduced controller gain in target corridor during positioning

Controller basic setting.

## Parameter description

### Velocity Control Parameter Set (0x2331)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x2331	0x00	Number of Entries	U8	ro	2	Number of object entries
	0x01	Proportional Term POR	U16	rw	a)	Proportional gain of speed controller
	0x02	Integral Term I	U16	rw	a)	Integral term of speed controller

a) *Dependent on the configuration of the Motion Controller*

Parameters of speed controller.

### Position Control Parameter Set (0x2332)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x2332	0x00	Number of Entries	U8	ro	2	Number of object entries
	0x01	Proportional Term PP	U16	rw	a)	Proportional gain of position controller
	0x02	Derivative Term PD	U16	rw	a)	Differential term of position controller

a) *Dependent on the configuration of the Motion Controller*

Parameters of position controller.

### Current Control Parameter Set (0x2333)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x2333	0x00	Number of Entries	U8	ro	3	Number of object entries
	0x01	Continuous Current Limit	U16	rw	a)	Continuous current limit [mA]
	0x02	Peak Current Limit	U16	rw	a)	Peak current limit [mA]
	0x03	Integral Term CI	U16	rw	a)	Integral term of current controller

a) *Dependent on the configuration of the Motion Controller*

Current limitation values and parameters of current controller.

### Actual Current Limit (0x2334)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x2334	0x00	Actual Current Limit	U16	ro	–	Currently used current limitation value



## Parameter description

### General Settings (0x2338)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x2338	0x00	Number of Entries	U8	ro	3	Number of object entries
	0x01	Pure Sinus Commutation	U16	rw	1	<ul style="list-style-type: none"> <li>▪ 0: Full modulation</li> <li>▪ 1: Limitation to sinus form (not MCDC)</li> </ul>
	0x02	Activate Position Limits in Velocity Mode	U16	rw	0	1: Also use set positioning limits in speed mode
	0x03	Activate Position Limits in Position Mode	U16	rw	1	0: No range limits in positioning mode

General settings for commutation and the range limits.

In Profile Position Mode, the range limits are active by default. If no mechanical limits are present in the application, it is also possible to deactivate the limits in order to perform unlimited relative positioning operations. In this case, counting is continued at 0 if the upper or lower limit is exceeded.

### Motor Data (0x2350)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x2350	0x00	Number of Entries	U8	ro	5	Number of object entries
	0x01	Speed Constant $K_N$	U16	rw	<sup>a)</sup>	Speed constant
	0x02	Terminal Resistance $R_M$	U16	rw	<sup>a)</sup>	Connection resistance
	0x03	Pole Number	U16	rw	2/4 <sup>a)</sup>	Number of poles on BL motors (not MCDC)
	0x05	Thermal Time Constant $TW1$	U16	rw	<sup>a)</sup>	Thermal time constant 1

<sup>a)</sup> Dependent on the delivery configuration of the Motion Controller

Data sheet values of the connected motor.

### Encoder Data (0x2351)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x2351	0x00	Number of Entries	U8	ro	3	Number of object entries
	0x01	Sensor Type	U8	rw	0	<ul style="list-style-type: none"> <li>▪ 0: Analogue Hall (int. encoder)</li> <li>▪ 1: Incremental encoder (ext.)</li> <li>▪ 10: Incremental + Hall</li> <li>▪ 104: AES-4096 absolute encoder (not MCDC)</li> </ul>
	0x02	Resolution External Encoder	U32	rw	2 048	4-edge resolution of an externally connected incremental encoder
	0x03	Resolution Internal Encoder	U32	ro	3 000	Resolution of internal Hall sensor encoder (not MCDC)

Setting of the encoder to be used.

## Parameter description

### Velocity Actual Value Unfiltered (0x2361)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x2361	0x00	Velocity Actual Value Unfiltered	I16	ro	–	Unfiltered actual speed

### Baudrate Set (0x2400)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x2400	0x00	Baudrate Set	U8	ro	0xFF	Set baud rate

This object can be used to query which baud rate is set. The query returns the index of the set baud rate, or 0xFF if AutoBaud is set.

Baud rate	Index	Baud rate	Index
1000 KBit	0	250 KBit	3
800 KBit	1	125 KBit	4
500 KBit	2	AutoBaud	0xFF

### Position Demand Internal Value (0x2562)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x2562	0x00	Position Demand Internal Value	I32	rw	–	Last target position in internal units

### Target Position Internal Value (0x257A)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x257A	0x00	Target Position Internal Value	I32	rw	–	Target position in internal units

### Position Range Limit Internal Value (0x257B)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x257B	0x00	Number of Entries	U8	ro	2	Number of object entries
	0x01	Minimum Position Range Limit	I32	ro	$-1.8 \cdot 10^6$	Internal lower positioning range limit in internal units
	0x02	Maximum Position Range Limit	I32	ro	$1.8 \cdot 10^6$	Internal upper positioning range limit in internal units

### Software Position Limit Internal Value (0x257D)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x257D	0x00	Number of Entries	U8	ro	2	Number of object entries
	0x01	Minimum Position Limit	I32	ro	$-1.8 \cdot 10^6$	Lower positioning range limit in internal units
	0x02	Maximum Position Limit	I32	ro	$1.8 \cdot 10^6$	Upper positioning range limit in internal units

## Parameter description

### 6.3 Objects of the drive profile acc. to CiA 402

#### Controlword (0x6040)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x6040	0x00	Controlword	U16	rw	0	Controlword

The bits in the Controlword are described in chap. 4.2, p. 37.

#### Statusword (0x6041)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x6041	0x00	Statusword	U16	ro	0	Statusword

The bits in the Statusword are described in chap. 4.2, p. 37.

#### Modes of Operation (0x6060)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x6060	0x00	Modes of Operation	I8	rw	1	Select the operating mode 0: controller not activated 1: Profile Position (position control) 3: Profile Velocity (speed control) 6: Homing (reference run) 8: Cyclic Synchronous Position (position control)

#### Modes of Operation Display (0x6061)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x6061	0x00	Modes of Operation Display	I8	ro	1	Display of the selected operating mode

The meaning of the return values corresponds to the values of the object 0x6060.

#### Position Demand Value (0x6062)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x6062	0x00	Position Demand Value	I32	ro	–	Position set value in user-defined units

#### Position Actual Internal Value (0x6063)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x6063	0x00	Position Actual Internal Value	I32	ro	–	Position actual value in internal units

#### Position Actual Value (0x6064)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x6064	0x00	Position Actual Value	I32	ro	–	Position actual value in user-defined units

#### Position Window (0x6067)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x6067	0x00	Position Window	U32	rw	32	Corridor around the target position in user-defined units

## Parameter description

### Position Window Time (0x6068)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x6068	0x00	Position Window Time	U16	rw	48	Minimum residence time within the corridor in PP operating mode, until the target position is reported as achieved.

### Velocity Demand Value (0x606B)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x606B	0x00	Velocity Demand Value	I32	ro	–	Target speed in $\text{min}^{-1}$

### Velocity Actual Value (0x606C)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x606C	0x00	Velocity Actual Value	I32	ro	–	Actual speed in $\text{min}^{-1}$

### Velocity Window (0x606D)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x606D	0x00	Velocity Window	U16	rw	20	Corridor around the set speed in user-defined units

### Velocity Window Time (0x606E)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x606E	0x00	Velocity Window Time	U16	rw	200	Minimum residence time within the corridor [ms]

### Velocity Threshold (0x606F)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x606F	0x00	Velocity Threshold	U16	rw	20	Speed threshold value in $\text{min}^{-1}$

### Velocity Threshold Time (0x6070)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x6070	0x00	Velocity Threshold Time	U16	rw	48	Monitoring time [ms]. If the speed lies outside the corridor for longer than is listed here, the speed is reported as not equal to 0.

### Current Actual Value (0x6078)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x6078	0x00	Current Actual Value	I16	ro	–	Current motor standby current in mA

### Target Position (0x607A)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x607A	0x00	Target Position	I32	rw	–	Position set value in user-defined units

## Parameter description

### Position Range Limit (0x607B)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x607B	0x00	Number of Entries	U8	ro	2	Number of object entries
	0x01	Min Position Range Limit	I32	ro	$-1.8 \cdot 10^9$	Lower limit of the position range in user-defined units
	0x02	Max Position Range Limit	I32	ro	$1.8 \cdot 10^9$	Upper limit of the position range in user-defined units

### Homing Offset (0x607C)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x607C	0x00	Min Range Limit	I32	rw	0	Offset of the zero position relative to the position of the reference switch

### Software Position Limit (0x607D)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x607D	0x00	Number of Entries	U8	ro	2	Number of object entries
	0x01	Min Position Limit	I32	rw	$-1.8 \cdot 10^9$	Lower limit of the position range
	0x02	Max Position Limit	I32	rw	$1.8 \cdot 10^9$	Upper limit of the position range

### Polarity (0x607E)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x607E	0x00	Polarity	U8	rw	0	Bit-coded

This object can be used to globally change the direction of rotation, i.e. the default and actual values for position and speed are multiplied by  $-1$ :

- Bit 7 = 1: negative direction of rotation in positioning mode
- Bit 6 = 1: negative direction of rotation in speed mode

### Max Profile Velocity (0x607F)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x607F	0x00	Max Profile Velocity	U32	rw	a)	Maximum permitted speed [ $\text{min}^{-1}$ ]

a) Dependent on the delivery configuration of the Motion Controller

### Profile Velocity (0x6081)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x6081	0x00	Profile Velocity	U32	rw	a)	Maximum speed in $\text{min}^{-1}$

a) Dependent on the delivery configuration of the Motion Controller

### Profile Acceleration (0x6083)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x6083	0x00	Profile Acceleration	U32	rw	30 000	Maximum acceleration [ $1/\text{s}^2$ ]

## Parameter description

### Profile Deceleration (0x6084)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x6084	0x00	Profile Deceleration	U32	rw	30 000	Maximum braking rate [ $1/s^2$ ]

### Quick Stop Deceleration (0x6085)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x6085	0x00	Quick Stop Deceleration	U32	rw	30 000	Braking ramp value at Quick stop [ $1/s^2$ ]

### Position Encoder Resolution (0x608F)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x608F	0x00	Number of Entries	U8	ro	2	Number of object entries
	0x01	Encoder Increments	U32	ro	3 000/ 2 048 <sup>a)</sup>	Encoder increments
	0x02	Motor Revolutions	U32	ro	1	Number of motor revolutions with the pulse number named in subindex 0x01

a) BL controller/MCDC

The value of the encoder resolution is loaded from the settings in the Encoder Data (0x2351) object and cannot be changed here.

### Gear Ratio (0x6091)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x6091	0x00	Number of Entries	U8	ro	2	Number of object entries
	0x01	Motor Revolutions	U32	rw	1	Number of motor revolutions
	0x02	Shaft Revolutions	U32	rw	1	Number of revolutions of the output shaft

### Feed Constant (0x6092)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x6092	0x00	Number of Entries	U8	ro	2	Number of object entries
	0x01	Feed	U32	rw	3 000/ 2 048 <sup>a)</sup>	Feed in user units
	0x02	Shaft Revolutions	U32	rw	1	Number of revolutions of the output shaft

a) BL controller/MCDC

The Gear Ratio and Feed Constant factors can be used to specify a gear ratio and a feed value for converting into user units (see chap. 4.3, p. 46).

On delivery the user units are set to the default encoder resolution (3 000 or 2 048 increments per revolution).

## Parameter description

### Position Factor (0x6093)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x6093	0x00	Number of Entries	U8	ro	2	Number of object entries
	0x01	Numerator	U32	ro	1	Numerator of the position factor
	0x02	Divisor	U32	ro	1	Denominator (divisor) of the position factor

$$\text{Internal position} = \frac{\text{Position in user units} \cdot \text{Position factor numerator}}{\text{Position factor denominator}}$$

The position factor is calculated from the Position Encoder Resolution (0x608F), Gear Ratio (0x6091) and Feed Constant (0x6092) and cannot be changed here (see chap. 4.3, p. 46).

### Homing Method (0x6098)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x6098	0x00	Homing Method	I8	rw	20	Homing method

### Homing Speed (0x6099)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x6099	0x00	Number of Entries	U8	ro	2	Number of object entries
	0x01	Switch Seek Velocity	U32	rw	400	Speed during switch search [ $\text{min}^{-1}$ ]
	0x02	Homing Speed	U32	rw	100	Speed during search for zero [ $\text{min}^{-1}$ ]

### Homing Acceleration (0x609A)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x609A	0x00	Homing Acceleration	U32	rw	50	Acceleration during homing [ $1/\text{s}^2$ ]

### Control Effort (0x60FA)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x60FA	0x00	Control Effort	U32	ro	–	Controller output

### Digital Inputs (0x60FD)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x60FD	0x00	Control Effort	U32	ro	–	State of digital inputs

Using the following bits, this object indicates which switch is switched on or off:

- Bit 0: Negative Limit Switch
- Bit 1: Positive Limit Switch
- Bit 2: Homing Switch

### Target Velocity (0x60FF)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x60FF	0x00	Target Velocity	I32	rw	–	Speed set value [ $\text{min}^{-1}$ ]

## Parameter description

### Supported Drive Modes (0x6502)

Index	Subindex	Name	Type	Attr.	Default value	Meaning
0x6502	0x00	Supported Drive Modes	U32	ro	0x25	Supported operating modes

The Supported Drive Modes object indicates the operating modes supported by the FAULHABER Motion Control Systems:

- Bit 0: Profile Position Mode (PP)
- Bit 2: Profile Velocity Mode (PV)
- Bit 5: Homing Mode (HM)
- Bit 7: Cyclic Synchronous Position (CSP)



