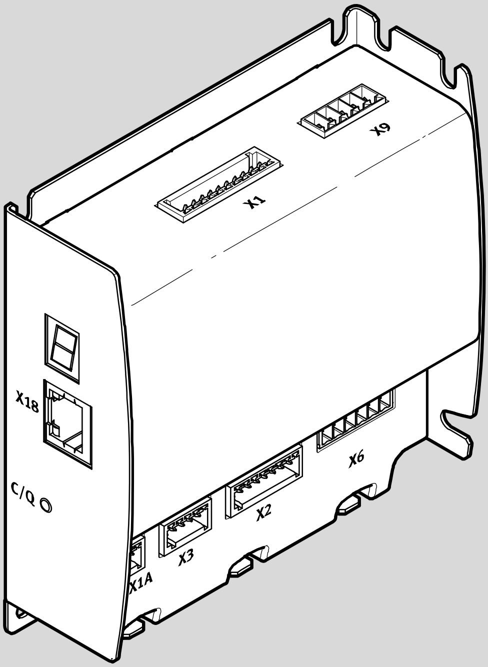
Motor controller

CMMO-ST-C5-1-LKP



Description

Motor controller with interface for
- IO-Link
- I-Port
- Modbus TCP

Device profile FHPP

8071674
2017-05b
[8071676]

Original instructions
GDCP-CMMO-ST-LK-C-HP-EN

Identification of hazards and instructions on how to prevent them:

Warning
Hazards that can cause death or serious injuries.

Caution
Hazards that can cause minor injuries or serious material damage.

Other symbols:

Note
Material damage or loss of function.

Recommendations, tips, references to other documentation.

Essential or useful accessories.

Information on environmentally sound usage.

Text designations:
- • Activities that may be carried out in any order.
- 1. Activities that should be carried out in the order stated.
- – General lists.
- ➔ Result of an action/References to more detailed information.

Software identification:

‹xxx›	Buttons in the software
[xxx] [xxx]	References to menu and sub-menu structures in the software
FCT […] [xxx]	FCT plug-in menu for components in the “Workplace” window
FCT menu [xxx]	FCT-main menu

Table of Contents – CMMO-ST-C5-1-LKP

Motor controller documentation 7
Version status 8
Service 8

1 FHPP with motor controller CMMO-ST 9

1.1 FHPP overview 9
1.2 Interfaces 10

2 IO-Link 11

2.1 IO-Link/I-Port and digital I/O interface [X1] 11
2.2 Parameterisation of IO-Link device 12
2.2.1 Parameterisation with the FCT plug-in CMMO-ST 12
2.2.2 Parameterisation with the integrated web server 12
2.3 Configuration IO-Link master 13
2.3.1 Example of CMMO-ST at S7 1200 14

3 I-Port 16

3.1 IO-Link/I-Port and digital I/O interface [X1] 16
3.2 Parameterisation of I-Port device 17
3.2.1 Parameterisation with the FCT plug-in CMMO-ST 17
3.2.2 Parameterisation with the integrated web server 17
3.3 Configuration of I-Port master 18

4 Modbus TCP 19

4.1 Modbus TCP interface [X18] 20
4.1.1 Pin allocation and cable specifications 20
4.2 Parameterisation of Modbus-TCP user 21
4.2.1 Parameterisation with the FCT plug-in CMMO-ST 21
4.2.2 Parameterisation with the integrated web server 22
4.3 Modbus master configuration 22
4.3.1 IP address 22
4.3.2 Address assignment and Modbus operations 22
4.3.3 Data objects for Modbus command “Read Device Identification” 28
4.3.4 Monitoring functions 28

5 Sequence control and I/O data **29**

5.1 Setpoint specification (FHPP operating modes) 29
5.1.1 Switching the FHPP operating mode 29
5.1.2 Record selection 29
5.1.3 Direct application 29
5.2 FHPP finite state machine 30
5.2.1 Create ready status 31
5.2.2 Positioning 32
5.2.3 Examples of control and status bytes 35
5.3 Configuration of the I/O data 40
5.3.1 Concept 40
5.3.2 I/O data (byte 1 ... 8) in the various FHPP operating modes 41
5.4 Assignment of the control bytes and status bytes (overview) 42
5.4.1 Description of the control bytes 43
5.4.2 Description of the status bytes 47

6 Control via FHPP **52**

6.1 Dimension reference system for electric drives 52
6.2 Homing run 52
6.2.1 Homing for electric drives 52
6.2.2 Methods of homing 53
6.3 Jogging 54
6.4 Teaching via fieldbus 55
6.5 Execute Record 57
6.5.1 Record selection flow diagrams 59
6.5.2 Record structure 62
6.5.3 Record linking (PNU 402) 63
6.6 Execute direct mode 64
6.6.1 Direct mode sequence 67
6.7 Monitoring of the drive behaviour 70
6.7.1 “Motion Complete” message 70
6.7.2 “Following error” message 71
6.7.3 “Standstill monitoring” message 72
6.7.4 Comparators 74

7 Diagnostics **76**

7.1 Diagnostic messages 76
7.1.1 Classification and error responses 76
7.1.2 Display of a diagnostic event 77
7.1.3 Diagnostic memory 78

7.2 Fault detection and elimination 79
7.2.1 Acknowledge error 79
7.2.2 Parameterisation of the diagnostic messages and fault clearance 80

A Technical appendix 81

A.1 Increments 81
A.1.1 Encoder increments [EINC] 81
A.1.2 Interface increments [SINC] 81
A.2 Conversion factors 82

B Reference paramete 83

B.1 General FHPP parameter structure 83
B.2 Access protection and master control 84
B.3 Overview of FHPP parameters 85
B.3.1 Device data 85
B.3.2 Diagnostics 86
B.3.3 Process Data 87
B.3.4 Record list 88
B.3.5 Project Data 90
B.3.6 Factor group 92
B.3.7 Axis parameters: electric drives 1 93
B.4 Descriptions of FHPP parameters 96
B.4.1 Representation of the parameter entries 96
B.4.2 Device data – version numbers 97
B.4.3 Device data – identification 98
B.4.4 Device data – MMI parameters 100
B.4.5 Diagnostic parameters 102
B.4.6 Process data – general process data 110
B.4.7 Process data – FHPP-data 112
B.4.8 Record list – record data 113
B.4.9 Record list – record messages 122
B.4.10 Project data – general project data 125
B.4.11 Project data – force/torque mode 126
B.4.12 Project data –teach mode 127
B.4.13 Project data – FHPP-direct mode 128
B.4.14 Project data – jog mode 130
B.4.15 Project data - direct mode position 131
B.4.16 Project data – force direct mode 132
B.4.17 Project data – rotational speed direct mode 133
B.4.18 Project data – direct mode general 134
B.4.19 Factor group 137
B.4.20 Axis parameters: electrical Drives 1 – mechanical parameters 138

B.4.21 Axis parameter: electrical drives 1 – homing parameters 140
B.4.22 Axis parameter: electrical drives 1 – controller parameters 142
B.4.23 Axis parameters: electric drives 1 – electronic rating plate 145
B.4.24 Axis parameters: electric drives 1 – standstill monitoring 146
B.4.25 Axis parameters: electric drives 1 – following error monitoring 147
B.4.26 Axis parameters: electric drives 1 – motor data 147
B.4.27 Axis parameters: electric drives 1 – temperature data 147
B.4.28 Axis parameters: electric drives 1 – general drive data 148

C Festo Parameter Channel (FPC) 150

C.1 FPC for cyclical I/O data 150
C.2 Overview of EFPC 150
C.2.1 EFPC structure 150
C.2.2 FPCC and FPCS – transmission mode, request and response ID 151
C.3 Parameter transmission (PNUs, internal objects) 152
C.3.1 Structure of EFPC in parameter transmission 152
C.3.2 Sequence of parameter transmission 152
C.3.3 Example of parameter transmission 152
C.3.4 Error codes 153
C.4 Parameter file transmission 154
C.4.1 Structure of EFPC in parameter file transmission 154
C.4.2 Package ID 154
C.4.3 Parameter file and user data package 155
C.4.4 Checking and activation of the parameter file 156
C.4.5 Sequence of parameter file transmission 156
C.4.6 Examples of parameter file transmission 157
C.4.7 Error codes 162

D Diagnostic messages 164

E Terms and abbreviations 179

Index 181

Motor controller documentation

This documentation (GDCP-CMMO-ST-LK-C-HP-...) describes the Festo Handling und Positioning Profile (FHPP) for the motor controller CMMO-ST-C5-1-LKP. The full description of the motor controller includes the following documents:

Always observe the general safety regulations for the motor controller in the equipment and functional description of the motor controller GDCP-CMMO-ST-LK-SY-... ➔ Tab. 1.

Designation	Contents
Condensed documentation CMMO-ST-LK...	Brief equipment and functional description of the motor controller for initial information
Manual GDCP-CMMO-ST-LK-SY-...	Equipment and functional description of the motor controller – Mounting – Commissioning via web server/Festo Configuration Tool (FCT) – Technical data
Manual GDCP-CMMO-ST-LK-C-HP-...	Control and parameterisation of the motor controller with the device profile FHPP via: – IO-Link – I-Port – Modbus TCP
Manual GDCP-CMMO-ST-LK-S1-...	Use of the STO safety function ("Safe Torque Off")
Help system for the FCT software	Descriptions of the Festo Configuration Tool (FCT) for commissioning and parameterisation of: – configurable axis/motor combinations – positioning systems in Festo's Optimised Motion Series (OMS)
Special documentation CMMO-ST_UL-...	Requirements for operating the product in the USA and Canada in accordance with certification by Underwriters Laboratories Inc. (UL).

Tab. 1 Documentation for the motor controller

Additional information about the product:

- CMMO-ST-Quickguide-...:brief description of the initial commissioning and diagnostics of positioning systems in Festo's Optimised Motion Series (OMS) with the web server of the CMMO-ST
- Overview of accessories (catalogue) ➔ www.festo.com/catalogue
- Operating instructions for configurable actuators and the positioning systems from Festo (e.g. EPCO) ➔ www.festo.com/sp
- Parameter lists: Default settings of the commissioning parameters for positioning systems in Festo's Optimised Motion Series (OMS)
- Function elements (CODESYS, ...) ➔ www.festo.com/sp
- Certificates, declaration of conformity ➔ www.festo.com/sp

Target group

This documentation is intended exclusively for technicians trained in control and automation technology, who have experience in installation, commissioning, programming and diagnostics of positioning systems.

Version status

This documentation refers to the following version of the motor controller:

– Firmware: V 1.5.x and higher
– FCT plug-in: CMMO-ST V 1.5.x and later

The following details are displayed in the software with an active online connection:

– Firmware version and MAC-ID ➔ "Info" tab of the integrated web server
– Hardware version, firmware version ➔ FCT ("Controller" page)
  If at this time there is no online connection, the information from the most recent connection is displayed.

Additional version details, e.g. amendment: ➔ product labelling of the motor controller

Note

Before using a newer firmware version:

• Check whether a newer corresponding version of the FCT plug-in or user documentation is available ➔ www.festo.com/sp.

Service

Please consult your regional Festo contact if you have any technical problems.

1 FHPP with motor controller CMMO-ST

1.1 FHPP overview

Tailored to the target applications for handling and positioning tasks, Festo has developed an optimised device profile, the "Festo Handling and Positioning Profile (FHPP)".
The FHPP permits a uniform control and parameterisation for the various motor controllers from Festo, independent of the connection to different control devices.
To do this, it defines for the user, largely uniformly,

- operating modes
- I/O data structure
- parameter objects
- sequence control

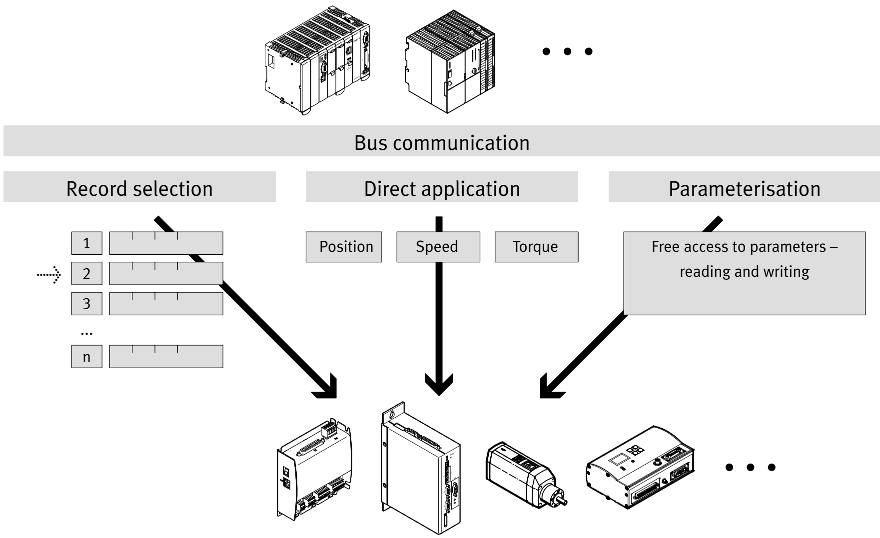


Fig. 1.1 Principle of FHPP

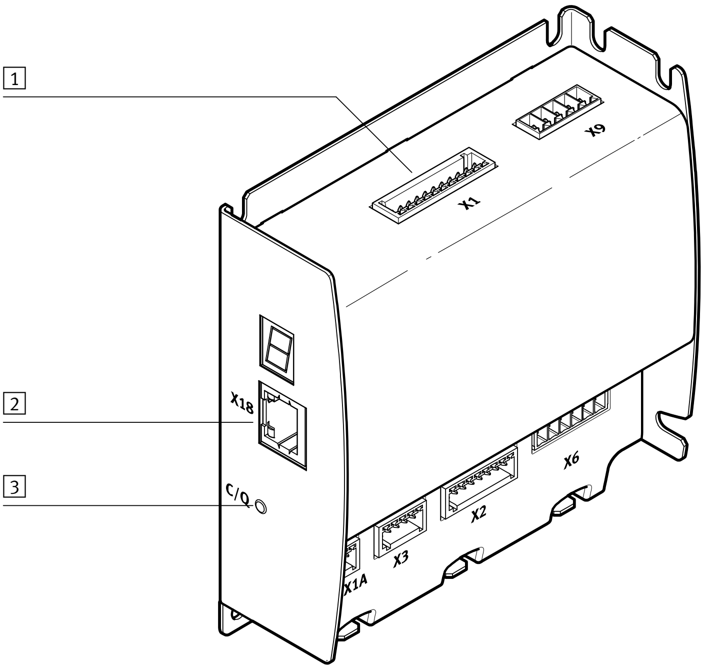
Control and status data (FHPP Standard)
Communication takes place via 8-byte control and status data. Functions and status messages required in operation can be written and read directly.
Parameterisation (FPC)
The controller can access the parameter values of the motor controller via the parameter channel. A further 8 bytes of I/O data are used for this purpose.

1.2 Interfaces

Control and parameterisation via FHPP is supported in CMMO-ST-C5-1-LKP through various fieldbus interfaces corresponding to Tab. 1.1.

Connection	Interface	Description
IO-Link	[X1] – IO-Link/I-Port and digital I/Os	→ Chapter 2
I-Port	[X1] – IO-Link/I-Port and digital I/Os	→ Chapter 3
Modbus TCP	[X18] – Ethernet interface	→ Chapter 4

Tab. 1.1 Interfaces for FHPP



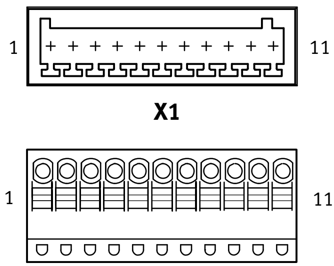
1 [X1] – IO-Link/I-Port and digital I/Os
2 [X18] Ethernet interface
3 Link/activity-LED C/Q

Fig. 1.2 Motor controller CMMO-ST-C5-1-LKP

2 IO-Link

The communication system IO-Link is used to exchange serial data from decentralised function modules (devices) at the field level.
IO-Link is a standardised I/O technology (IEC 61131-9) for exchanging serial data with sensors and actuators bi-directionally via a 3-wire connection. The motor controller is an IO-Link device in accordance with the IO-Link Interface Specification Version 1.1 [IOL].
The LED C/Q on the motor controller displays the status of the IO-Link connection.

2.1 IO-Link/I-Port and digital I/O interface [X1]

Connection	Pin	Function	
	1	+24 V (OUT)	Output +24 V[1)] e.g. supply of a potential-free relay contact for the controller enable input
	2	0 V (GND)	Reference potential for output signals
	3	DOUT2	Output 2, parameterisable
	4	DOUT1	Output 1, parameterisable
	5	READY	Output Ready
	6	ENABLE	Controller enable input[2)]
	7	–	No function, not connected internally[3)]
	8	–	
	9	L–	0 volt (GND)
	10	C/Q	IO-Link/I-Port signal
	11	L+	24 volt supply of the IO-Link IC, not connected to the logic supply at X9

1) Not overload-proof, max. 100 mA
2) Required signals for controller enable can be parameterised (FCT) ➔ section 2.2.1
3) Pins can be used for the 4th and 5th conductor of the I-Port / IO-Link cable

Tab. 2.1 Port X1 I/O interface, pin 9 ... 11 allocated for IO-Link

2.2 Parameterisation of IO-Link device

Before connecting the motor controller to the I-Link master, parameterise the controller interface and device profile:

- with the FCT plug-in CMMO-ST ➔ section 2.2.1
- with the integrated web server ➔ section 2.2.2

Connecting the motor controller to the PC ➔ equipment and functional description of the motor controller, GDCP-CMMO-ST-LK-SY-....

2.2.1 Parameterisation with the FCT plug-in CMMO-ST

1. Create drive configuration ➔ Help for the FCT plug-in CMMO-ST
2. On the application data page (Application Data), determine the control interface (Control Interface):
   - "IO-Link"
3. Optionally determine on the controller page (Controller):
   - Enable with (Enabled by), determination of the required signals for controller enable:
     - "Fieldbus" (Fieldbus) – factory setting
     - "Digital input 'Enable' and fieldbus" (Digital Input 'Enable' and Fieldbus)
4. On the fieldbus page (Fieldbus), determine the device profile (Device Profile):
   - "FHPP standard"
   - "FHPP standard + FPC"
5. Establish an online connection.
6. Activate device control (Device Control).
7. Download and save (Store) the parameters.

A restart is required after changing and storing the following parameters with the FCT plug-in to make the settings active:

- control interface (Modbus, IO-Link, I-Port)
- device profile (FHPP standard, FHPP standard + FPC)

After parameterisation and restart of the motor controller, the IO-Link master can be configured ➔ section 2.3.

2.2.2 Parameterisation with the integrated web server

1. Call up online connection with the web browser: "http://192.168.178.1/" (factory setting)
2. To parameterise and store, activate device control (Device Control).
3. In the Control Interface tab, determine and save the control interface (Save):
   - "IO-Link"
4. In the FHPP Profile tab, determine and save the device profile (Save):
   - "FHPP channel"
   - "FHPP + FPC channel"

After parameterisation, the IO-Link master can be configured ➔ section 2.3.

2.3 Configuration IO-Link master

To create the IO-Link connection, configure the motor controller in the IO-Link master.

Steps for configuration of the IO-Link master ➔ documentation on the used configuration program (CODESYS, TIA-Portal, STEP 7, ...).

The IODD files include all necessary information on configuration:

IODD files	Device profile
Festo-CMMO-ST-C5-1-LKP_FHPP-xxxxxxxx-IODD1.1.xml	FHPP standard (8 I/O bytes)
Festo-CMMO-ST-C5-1-LKP_FHPP_and_FPC-xxxxxxxx-IODD1.1.xml	FHPP standard + FPC (16 I/O bytes)
(xxxxxxxx = date)	

Tab. 2.2 IODD files

The motor controller supports the IO-Link specification V1.1 with the following characteristics:
– Cyclical IO-Link data 8 or 16 I/O bytes.
– Device-specific errors and warnings are reported to the IO-Link master through the “Event management”.
– SIO mode is not supported.
– Transmission rate 230.4 KBaud.
– No support of the parameter server of the IO-Link master (2048 bytes for parameters of the motor controller are not sufficient).
  The upload and download of all parameters to the controller can be implemented via EFPC with appropriate functional modules or function blocks ➔ appendix C.4

Current IODD files, functional modules or function blocks ➔ www.festo.com/sp

2.3.1 Example of CMMO-ST at S7 1200

The following lists an example of steps for connecting a CMMO-ST to an S7 1200 as IO-Link master.

Specific steps for configuration of the IO-Link master:
➔ Documentation for the module.
➔ Documentation on the used configuration program.

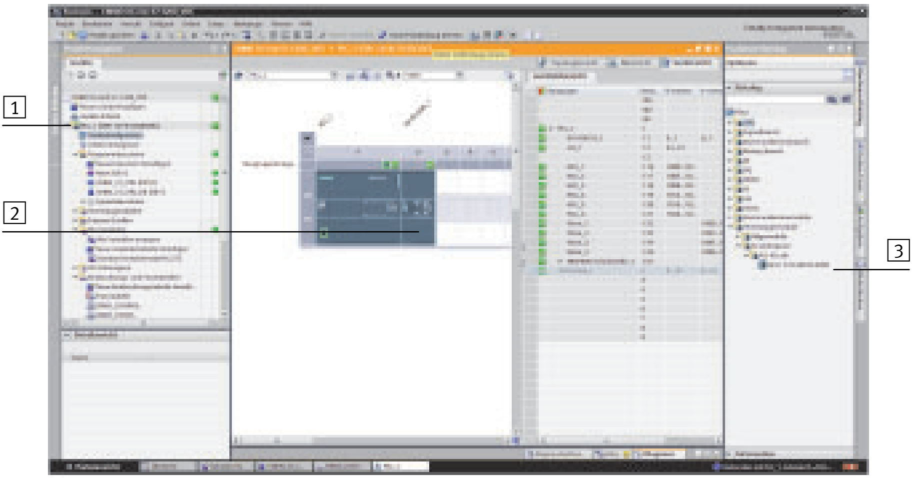
Requirements

– TIA-Portal V13
– S7 PCT V3.3 must support the configuration of IO-Link 1.1

Typical steps

The following steps are normally necessary to connect a CMMO-ST to an S7 1200 as IO-Link master.

1. Create new project in the TIA-Portal.
2. Open project view.
3. Add new device (the S7 must support the IO-Link master ➔ documentation on the S7)
4. Set IP address for the CPU.
5. Select PLC, then select slot in mounting rack for the IO-Link master.
6. In the “Hardware catalogue” window under “Technology modules”, select the IO-Link master and take it over for the slot.

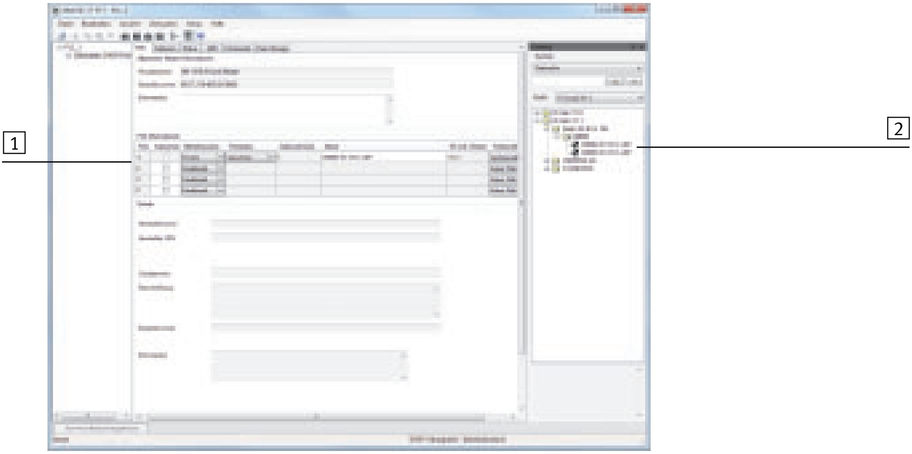


1 PLC
2 Slot in mounting rack
3 “Catalogue” window for selection of the IO-Link master

Fig. 2.1 Example TIA-Portal – configure S7

7. Load configuration into controller.
8. Start the DeviceTool via the context menu of the IO-Link master.

9. In the dialogue, select the PC interface.
10. Menu [Extras] [Import IODD], then select IODD file and import it.



[1] Port [2] Catalogue, IO-Link 1.1

Fig. 2.2 Example DeviceTool, assign IODD to the port

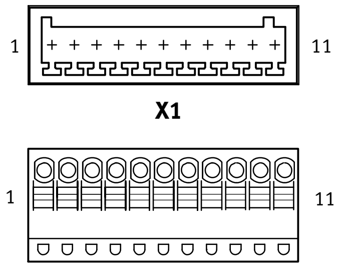
11. In the catalogue under “IO-Link 1.1”, “Festo AG & Co. KG”, “CMMO”, select the desired variant (standard FHPP or with FPC) and assign the IODD to the used port.
12. Load configuration into the device.
13. End DeviceTool, save changes. The IO-Link connection is now active.

Subsequently, load the FHPP_Positions_Library_TIA from Festo, for example under “Libraries”, and assign the addresses of the variables table to the module inputs (I_ADRESS, O_ADRESS)
➔ description/Help file for the Library.

3 I-Port

The Festo-specific I-Port interface is used for exchange of serial data from decentralised function modules (devices) at field level.
The LED C/Q on the motor controller displays the status of the I-Port connection.

3.1 IO-Link/I-Port and digital I/O interface [X1]

Port	Pin	Function	
	1	+24 V (OUT)	Output +24 V[1] e.g. supply of a potential-free relay contact for the controller enable input
	2	0 V (GND)	Reference potential for output signals
	3	DOUT2	Output 2, parameterisable
	4	DOUT1	Output 1, parameterisable
	5	READY	Output Ready
	6	ENABLE	Controller enable input[2]
	7	–	No function, not connected internally[3]
	8	–	
	9	L–	0 volt (GND)
	10	C/Q	IO-Link/I-Port signal
	11	L+	24 volt supply of the I-Port IC, not connected to the logic supply at X9

1) Not overload-proof, max. 100 mA

2) Required signals for controller enable can be parameterised (FCT) ➔ section 3.2.1

3) Pins can be used for the 4th and 5th conductor of the I-Port / IO-Link cable

Tab. 3.1 Port X1 I/O interface, pin 9 ... 11 allocated for IO-Link

3.2 Parameterisation of I-Port device

Before connecting the motor controller to the I-Port master, parameterise the controller interface and equipment profile:
– with the FCT plug-in CMMO-ST ➔ section 3.2.1
– with the integrated web server ➔ section 3.2.2

Connecting the motor controller to the PC ➔ equipment and functional description of the motor controller, GDCP-CMMO-ST-LK-SY-....

3.2.1 Parameterisation with the FCT plug-in CMMO-ST

1. Create drive configuration ➔ Help for the FCT plug-in CMMO-ST.
2. On the application data page (Application Data), determine the control interface (Control Interface):
   – "I-Port"
3. Optionally, determine the following on the controller page (Controller):
   – Enable with (Enabled by), determination of the required signals for controller enable:
     – "Fieldbus" (Fieldbus) – factory setting
     – "Digital input 'Enable' and fieldbus" (Digital Input 'Enable' and Fieldbus)
4. On the fieldbus page (Fieldbus), determine the device profile (Device Profile):
   – "FHPP standard"
   – "FHPP standard + FPC"
5. Establish an online connection.
6. Activate device control (Device Control).
7. Download and save (Store) the parameters.

A restart is required after changing and storing the following parameters with the FCT plug-in to make the settings active:
– control interface (Modbus, IO-Link, I-Port)
– device profile (FHPP standard, FHPP standard + FPC)

After parameterisation and restart of the motor controller, the I-Port master can be configured ➔ section 3.3.

3.2.2 Parameterisation with the integrated web server

1. Call up online connection with the web browser: "http://192.168.178.1/" (factory setting)
2. To parameterise and store, activate device control (Device Control).
3. In the Control Interface tab, determine and save the control interface (Save):
   – "I-Port"
4. In the FHPP Profile tab, determine and save the device profile (Save):
   – "FHPP channel"
   – "FHPP + FPC channel"

After parameterisation, the I-Port master can be configured ➔ section 3.3.

3.3 Configuration of I-Port master

The following I-Port masters support the motor controller:

I-Port master	Supported I-Ports and data size	Special features
CPX-CTEL	4 x I-Port, total max. 32 bytes I and 32 bytes O	With the “Automatic configuration” setting, the data size can be freely distributed (2x16 or 1x16 and 2x8 or 4x8). If the “tool change mode” of the used CPX-CTEL is not supported, the motor controller must be switched on before the CPX-CTEL.
CTEU-PB	2 x I-Port, each 16 bytes I and 16 bytes O	Device description file GSD has module identifier for I-Port
CTEU-EC	2 x I-Port, each 16 bytes I and 16 bytes O	Device description file ESI has module identifier for I-Port
CTEU-CO	(2 x I-Port, each 16 bytes I and 16 bytes O)	Support in preparation

Tab. 3.2 Supported I-Port master

The I-Port connection does not have to be configured from most masters.
For some I-Port masters, device description files are available for the respective fieldbus.

Specific module support of the I-Port devices through current GSD and ESI files, functional modules or function blocks ➔ www.festo.com/sp

4 Modbus TCP

Modbus is an open communication protocol based on the master-slave architecture. It is an established standard for communication via Ethernet-TCP/IP in automation technology.

The basic function of Modbus TCP is described in IEC 61158.
The standard port for Modbus TCP is 502.

The Ethernet control interface is used parallel to the Ethernet parameterisation interface (FCT, web server). A maximum of **one** Modbus TCP connection at a time is possible.
After the TCP connection has been made, it is normally kept open and only disconnected by the motor controller in case of error, with a timeout set or through the counterpart station.
Communication with the FCT and the web server remains possible.

Data Encoding

Modbus TCP uses "Big Endian" transmission sequence. The "most significant byte" is sent first. The actual data (Modbus: "tab") are processed word-by-word (2 bytes). It may therefore be necessary to "turn" these 2 bytes on the controller. This applies to the operations (function codes): 0x03, 0x10, 0x17 ➔ section 4.3.2.
This already takes place through the module if provided by Festo.

Modbus telegram

In general, a Modbus telegram is constructed correspondingly ➔ Tab. 4.1 (the higher-value byte is always sent first).
If, for example, the CMMO is to be accessed by the computer via Modbus, the transaction identifier, protocol identifier, message length and unit identifier must additionally be sent at the beginning before the function code is sent.

The assignment can be visualised and tested with the help of the "Modbus TCP Client".
➔ www.festo.com/sp, search for "Modbus TCP Client"

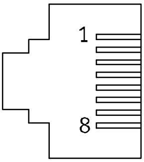
Byte no.	Number of bytes	Function	Comment	
1	2	Transaction number	Freely selectable. Returned again in the answer.	High-order byte
2				Low-order byte
3	2	Protocol identifier	Always 0	High-order byte
4				Low-order byte
5	2	Number of bytes still to follow	= n + 2, whereby n is the number of data points from byte 9.	High-order byte
6				Low-order byte
7	1	Address (unit identifier, slave ID)	Can be ignored (e.g. set to 0).	–
8	1	Function code	➔ section 4.3.2	–
9 …	n	Data	➔ section 4.3.2	–

Tab. 4.1 Structure of Modbus telegram

4.1 Modbus TCP interface [X18]

The Modbus connection is established via the Ethernet interface [X18] as an RJ45 socket. This can be used in parallel for 2 additional TCP connections (one for the FCT parameter software and one for the web server). As a Modbus/TCP user, the motor controller can be reached via the same IP address as is used by FCT or the web server.

4.1.1 Pin allocation and cable specifications

	Pin	Specification	
	1	Transmission signal+ (TX+)	Wire pair 3
	2	Transmission signal- (TX-)	Wire pair 3
	3	Receiver signal+ (RX+)	Wire pair 2
	4	–	Wire pair 1
	5	–	Wire pair 1
	6	Receiver signal– (RX–)	Wire pair 2
	7	–	Wire pair 4
	8	–	Wire pair 4
	–	Housing	Screening

Tab. 4.2 Allocation [X18]

Type and design of cable

Shielded twisted-pair STP, Cat.5 cables must be used for cabling.

4.2 Parameterisation of Modbus-TCP user

Before connecting the motor controller to the Modbus master, parameterise the controller interface, device profile, TCP-Port and Timeout:

– with the FCT plug-in CMMO-ST ➔ section 4.2.1
– with the integrated web server ➔ section 4.2.2

Connecting the motor controller to the PC ➔ equipment and functional description of the motor controller, GDCP-CMMO-ST-LK-SY-....

4.2.1 Parameterisation with the FCT plug-in CMMO-ST

1. Create drive configuration ➔ Help for the FCT plug-in CMMO-ST.
2. On the application data page (Application Data), determine the control interface (Control Interface):
   – “Modbus/TCP”
3. Optionally determine on the controller page (Controller):
   – Enable with (Enabled by), determination of the required signals for controller enable:
     – “Fieldbus” (Fieldbus) – factory setting
     – “Digital input ‘Enable’ and fieldbus” (Digital Input 'Enable' and Fieldbus)
4. On the fieldbus page (Fieldbus), operation parameters tab (Operation Parameters), determine:
   – Device profile (Device Profile):
     – “FHPP standard”
     – “FHPP standard + FPC”
   – Optionally, change TCP-Port (factory setting TCP-Port 502)
   – Optionally activate timeout (Timeout) (factory setting: 100 ms, not activated) ➔ section 4.3.4
5. Establish an online connection.
6. Activate device control (Device Control).
7. Download and save (Store) the parameters.
8. Optionally, on the Controller page, network settings tab (Network Settings), change the network settings (Setup network settings):
   – “DHCP server active” (DHCP server active, factory setting)
   – “Obtain IP address automatically” (Obtain an IP adress automatically)
   – “Use the following IP address” (fixed setting IP address, subnet mask and standard gateway)

A restart is required after changing and storing the following parameters with the FCT plug-in to make the settings active:

– control interface (Modbus, IO-Link, I-Port)
– interface parameters (device profile, TCP-Port)
– Network settings

After parameterisation and restart of the motor controller, the Modbus master can be configured ➔ section 4.3.

4.2.2 Parameterisation with the integrated web server

1. Call up online connection with web browser: “http://192.168.178.1/”
2. To parameterise and store, activate device control (Device Control).
3. In the Control Interface tab, set and save the control interface (Save):
   – “MODBUS”
4. In the FHPP Profile tab, set and save the device profile (Save):
   – “FHPP channel”
   – “FHPP + FPC channel”
5. In the Network tab, determine and save the network settings (Save):
   – “DHCP server active”
   – “Obtain an IP adress automatically”
   – “Use the following IP adress” (fixed setting IP address, subnet mask and standard gateway)

After parameterisation, the Modbus master can be configured ➔ section 4.3.

4.3 Modbus master configuration

4.3.1 IP address

The IP address of the motor controller as a Modbus/TCP user is identical to the IP address set in the FCT or web server.

4.3.2 Address assignment and Modbus operations

The following operations (Modbus transactions) are supported:

– Read Holding Registers (0x03)
– Read Exception Status (0x07)
– Write Multiple Registers (0x10)
– Read/Write Multiple Registers (0x17)
– Read Device Identification (0x2B)

The start address is always “0”; the byte sequence is always “Big endian”.
Tab. 4.3 shows the supported Modbus commands.

Modbus command	Significance			
Read/write multiple registers	Read and write the process data			
	Read/write multiple registers request (0x17)			
	Field	**Bytes**	**Values**	**Byte no.**
	Function code	1	0x17	8
	Start address read	2	0x0000	9, 10
	Quantity of registers read	2	0x0004: FHPP standard 0x0008: FHPP standard + FPC	11, 12
	Start address write	2	0x0000	13, 14
	Quantity of registers write	2	0x0004: FHPP standard 0x0008: FHPP standard + FPC	15, 16
	Byte count write	1	0x08: FHPP standard 0x10: FHPP standard + FPC	17
	Registers values write	8, 16	FHPP standard process output telegram O FHPP standard + FPC process output telegram O	18 ...
	Read/write multiple registers response (0x17)			
	Field	**Bytes**	**Values**	**Byte no.**
	Function code	1	0x17	8
	Byte count	1	0x08: FHPP standard 0x10: FHPP standard + FPC	9
	Register value	8, 16	FHPP standard process input telegram I FHPP standard + FPC process input telegram I	10 ...
	Read/write multiple registers exception (0x97)			
	Field	**Bytes**	**Values**	**Byte no.**
	Error code	1	0x97	8
	Exception code	1	0x01: illegal function 0x02: illegal data address 0x03: illegal data value 0x04: server device failure	9

Modbus command	Significance			
Read holding registers	Read the process data			
	Read holding registers request (0x03)			
	Field	**Bytes**	**Values**	**Byte no.**
	Function code	1	0x03	8
	Start address	2	0x0000	9, 10
	Quantity of registers	2	0x0004: FHPP standard 0x0008: FHPP standard + FPC	11, 12
	Read holding registers response (0x03)			
	Field	**Bytes**	**Values**	**Byte no.**
	Function code	1	0x03	8
	Byte count	1	0x08: FHPP standard 0x10: FHPP standard + FPC	9
	Register value	8, 16	FHPP standard I/O and FPC	10 ...
	Read holding registers exception (0x83)			
	Field	**Bytes**	**Values**	**Byte no.**
	Error code	1	0x83	8
	Exception code	1	0x01: illegal function 0x02: illegal data address 0x03: illegal data value 0x04: server device failure	9

<table>
<tr><th>Modbus command</th><th colspan="4">Significance</th></tr>
<tr><td rowspan="22">Write multiple registers</td><td colspan="4">Write the process data</td></tr>
<tr><td colspan="4">Write multiple registers request (0x10)</td></tr>
<tr><td>Field</td><td>Bytes</td><td>Values</td><td>Byte no.</td></tr>
<tr><td>Function code</td><td>1</td><td>0x10</td><td>8</td></tr>
<tr><td>Start address</td><td>2</td><td>0x0000</td><td>9, 10</td></tr>
<tr><td>Quantity of registers</td><td>2</td><td>0x0004: FHPP standard
0x0008: FHPP standard + FPC</td><td>11, 12</td></tr>
<tr><td>Byte count</td><td>1</td><td>0x08: FHPP standard
0x10: FHPP standard + FPC</td><td>13</td></tr>
<tr><td>Register value</td><td>8, 16</td><td>FHPP standard process output telegram O
FHPP FHPP standard + FPC process output telegram O</td><td>14 ...</td></tr>
<tr><td colspan="4">Write multiple registers respone (0x10)</td></tr>
<tr><td>Field</td><td>Bytes</td><td>Values</td><td>Byte no.</td></tr>
<tr><td>Function code</td><td>1</td><td>0x10</td><td>8</td></tr>
<tr><td>Start address</td><td>2</td><td>0x0000</td><td>9, 10</td></tr>
<tr><td>Quantity of registers</td><td>2</td><td>0x0004: FHPP standard
0x0008: FHPP standard + FPC</td><td>11, 12</td></tr>
<tr><td colspan="3">Write multiple registers exception (0x90)</td><td></td></tr>
<tr><td>Field</td><td>Bytes</td><td>Values</td><td>Byte no.</td></tr>
<tr><td>Error code</td><td>1</td><td>0x90</td><td>8</td></tr>
<tr><td>Exception code</td><td>1</td><td>0x01: illegal function
0x02: illegal data address
0x03: illegal data value
0x04: server device failure</td><td>9</td></tr>
</table>

Modbus command	Significance			
Read exception status	Read the fault number			
	Read exception status request (0x07)			
	Field	**Bytes**	**Values**	**Byte no.**
	Function code	1	0x07	8
	Read exception status response (0x07)			
	Field	**Bytes**	**Values**	**Byte no.**
	Function code	1	0x07	8
	Output data	1	0x01 ... 0xFF: Exception status (fault number) 0x00: No fault	9
	Read exception status exception (0x87)			
	Field	**Bytes**	**Values**	**Byte no.**
	Error code	1	0x87	8
	Exception code	1	0x01: illegal function 0x02: illegal data address 0x03: illegal data value 0x04: server device failure	9

Modbus command	Significance
Read device identification	Read the device data

Read device identification request (0x2B)			
Field	**Bytes**	**Values**	**Byte no.**
Function code	1	0x2B	8
MEI type	1	0x0E	9
Read device ID code	1	0x01: basic device identification 0x02: regular device identification	10
Object ID	1	0x00: (first object to be transferred)	11

Read device identification response (0x2B)			
Field	**Bytes**	**Values**	**Byte no.**
Function code	1	0x2B	8
MEI Type	1	0x0E	9
Read device ID code	1	Same as request field	10
Conformity level	1	0x01: basic device identification 0x02: regular device identification	11
More follows	1	0x00: no more objects	12
Next object ID	1	0x00	13
No. of objects	1	Number of objects in this message	14
Object 1	1	➔ Section 4.3.3, Tab. 4.4	15 …
…	…		
Object n	1		

Read device identification exception (0xAB)			
Field	**Bytes**	**Values**	**Byte no.**
Error code	1	0xAB	8
Exception code	1	0x01: illegal function 0x02: illegal data address 0x03: illegal data value 0x04: server device failure	9

Tab. 4.3 Overview of Modbus function codes

4.3.3 Data objects for Modbus command “Read Device Identification”

Object ID		Object Name	Access	Content
Basic	0x00	VendorName	R	Manufacturer name
	0x01	ProductCode	R	Product code
	0x02	MajorMinorRevision	R	Firmware version
Regular	0x00	VendorName	R	Manufacturer name
	0x01	ProductCode	R	Product code
	0x02	MajorMinorRevision	R	Firmware version
	0x03	VendorURL	R	Web address
	0x04	ProductName	R	Product name
	0x06	UserApplicationName	R	Project name

Tab. 4.4 Data objects for Modbus command “Read Device Identification”

4.3.4 Monitoring functions

TCP/IP connection monitoring (node guard, timeout)
The motor controller supports the TCP/IP connection monitoring.
Node guarding is connection monitoring at the application level. The node guard timeout is reset with each Modbus client message. If the client application no longer reacts or no more new messages are received within the timeout, the error reaction “Timeout MODBUS TCP/IP” is triggered.

The timeout time for connection monitoring can be entered between 0 and 5000 ms ➔ section 4.2.
Values between 0 and 100 ms can be entered, but are limited internally to a minimum of 100 ms.
A value of 0 deactivates the timeout.

In case of a timeout, the fault message 47h or 48h is triggered ➔ appendix D.
The error response is adjustable from “warning” to “immediate shut-off of the output stage”.

5 Sequence control and I/O data

5.1 Setpoint specification (FHPP operating modes)

The FHPP operating modes differ as regards their contents and the significance of the cyclic I/O data and in the functions which can be accessed in the motor controller.

Operating mode	Description
Record selection	A specific number of positioning records can be saved in the motor controller. A record contains all the parameters which are specified for a positioning job. The record number is transferred to the cyclic I/O data as the setpoint or actual value.
Direct application	The positioning task is transferred directly in the I/O telegram. The most important setpoint values (position, velocity, torque) are transmitted here. Supplementary parameters (e.g. acceleration) are defined by the parameterisation.

Tab. 5.1 Overview of FHPP operating modes with motor controller CMMO-ST

5.1.1 Switching the FHPP operating mode

The FHPP operating mode is switched by the CCON control byte (see below) and a feedback signal returned in the SCON status word. Switching between record selection and direct application is only permitted in the "ready" status ➔ section 5.2, Fig. 5.1.

5.1.2 Record selection

Each motor controller has a specific number of records, which contain all the information needed for one positioning job. The record number that the motor controller is to process at the next start is transferred in the controller's output data. The motor controller reports the last-executed record number in the input data of the controller. The positioning job itself does not need to be active.
The motor controller does not support an automatic mode, i.e. no user program. The motor controller cannot accomplish any significant tasks as stand alone – close coupling to the controller is always necessary. However, it is possible to link various records and execute them one after the other with the help of a start command. It is also possible to execute record switching before the target position is reached.

In this way, positioning profiles can be created without the time delays, which arise from the transfer via the fieldbus and the cycle time of the controller.

5.1.3 Direct application

In the direct application, positioning tasks are formulated directly in the controller's output data.
The typical application calculates the target setpoint values dynamically. This makes it possible to adjust the system to different workpiece sizes, for example, without having to re-parametrise the record list. The positioning data are managed completely in the controller and sent directly to the motor controller.

5.2 FHPP finite state machine

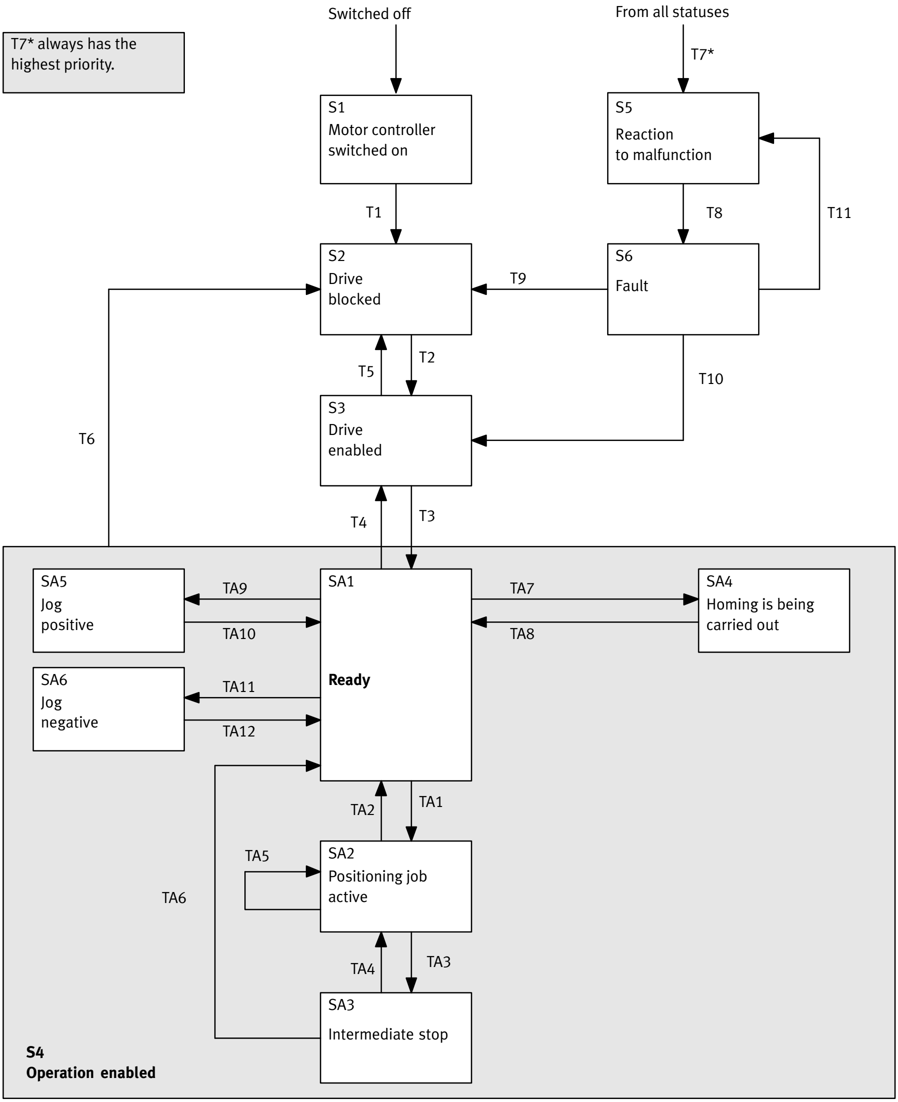


Fig. 5.1 Finite state machine

You can find the explanation of the control and status bytes (CCON, SCON, ...) in
➔ section 5.3.

Notes on the "operation enabled" status
The transition T3 changes to status S4, which itself contains its own sub-finite state machine, the statuses of which are marked with "SAx" and the transitions with "TAx" ➔ Fig. 5.1.
This enables an equivalent circuit diagram (➔ Fig. 5.2) to be used, in which the internal statuses SAx are omitted.
Transitions T4, T6 and T7* are executed from every sub-status SAx and automatically have a higher priority than any transition TAx.

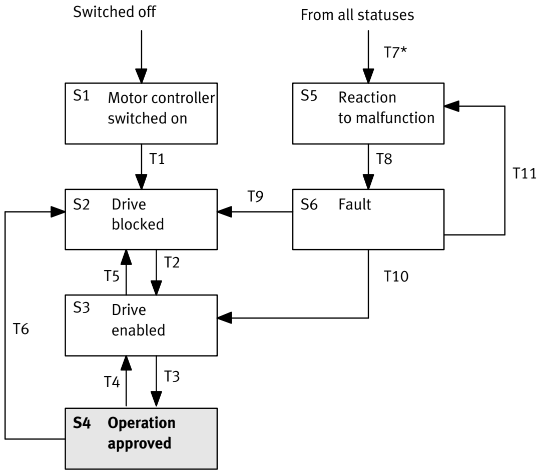


Fig. 5.2 Finite state machine equivalent circuit diagram

Reaction to malfunctions
T7 ("malfunction recognised") has the highest priority ("*"). T7 is then executed from S5 + S6 if an error with a higher priority occurs. This means that a serious error can displace a less serious error.

5.2.1 Create ready status

If parameterised (➔ PNU 128), the digital input signal ENABLE [X1.6] is also required to create the ready status.
Information on the digital inputs ➔ description GDCP-CMMO-ST-SY-...

T	Internal conditions	Actions of the user 1)
T1	Drive is switched on. An error cannot be determined.	
T2	Load voltage applied. Controller has master control.	Enable drive, activate CCON.ENABLE = 1 ➔ CCON = xxx0.xxx**1**
T3		Enable operation CCON.STOP = 1 CCON.ENABLE = 1 ➔ CCON = xxx0.xx**1**1
T4		Block operation CCON.STOP = 0 ➔ CCON = xxx0.xx**0**1

1) Key: P = rising edge (positive), N = falling edge (negative), x = any

T	Internal conditions	Actions of the user 1)
T5		Deactivate drive CCON.ENABLE = 0 ➔ CCON = xxx0.xxx**0**
T6		Deactivate drive CCON.ENABLE = 0 ➔ CCON = xxx0.xxx**0**
T7*	Malfunction recognised.	
T8	Reaction to malfunction completed; drive stopped.	
T9	There is no longer a malfunction. It was a serious error.	Acknowledge malfunction CCON.RESET = 0 ⟶ 1 CCON.ENABLE = 0 ➔ CCON = xxx0.**P**xx0
T10	There is no longer a malfunction. It was a simple error. Note: T10 permits acknowledgement of malfunctions without having to switch off the controller.	Acknowledge malfunction CCON.RESET = 0 ⟶ 1 CCON.ENABLE = 0 ➔ CCON = xxx0.**P**xx1
T11	Malfunction still exists.	Acknowledge malfunction CCON.RESET = 0 ⟶ 1 ➔ CCON = xxx0.**P**xxx

1) Key: P = rising edge (positive), N = falling edge (negative), x = any

Tab. 5.2 Status transitions while achieving ready status

5.2.2 Positioning

In principle: The transitions T4, T6 and T7* always have priority!

T	Internal conditions	Actions of the user 1)
TA1	Homing is present.	Start positioning task CPOS.START = 0 ⟶ 1 CPOS.HALT = 1 ➔ CPOS = 0xx0.00**P1**
TA2	Motion Complete = 1 The current record is completed. The next record is not processed automatically.	None, positioning job has been completed
TA3	Motion Complete = 0 Positioning job not yet completed.	Trigger intermediate stop CPOS.HALT = 1 ⟶ 0 ➔ CPOS = 0xxx.xxx**N**

1) Key: P = rising edge (positive), N = falling edge (negative), x = any

T	Internal conditions	Actions of the user 1)
TA4	Internal status “intermediate stop”	Continue positioning task CPOS.HALT = 1 CPOS.START = 0 ⟶ 1 CPOS.CLEAR = 0 ➔ CPOS = 0**0**xx.xx**P1**
TA5	Record selection, record sequencing: – A single record is finished. – The next record should be processed automatically.	Subsequent record is running ➔ CPOS = 0xxx.xxx**1**
	Record selection, record sequencing: – A new positioning job has arrived and is to interrupt the existing job	New positioning task interrupts the existing one CPOS.START = 0 ⟶ 1 CPOS.HALT = 1 ➔ CPOS = 0xx0.00**P1**
	Direct application: – A new positioning task has arrived.	New positioning task interrupts the existing one CPOS.START = 0 ⟶ 1 CPOS.HALT = 1 ➔ CPOS = 0xxx.xx**P1**
TA6		Delete remaining path CPOS.CLEAR = 0 ⟶ 1 ➔ CPOS = 0**P**xx.xxxx
TA7		Start homing CPOS.START = 0 ⟶ 1 CPOS.HALT = 1 ➔ CPOS = 0xx0.0**P**x**1**
TA8	Referencing finished or halt	Homing completed none
	None	Homing interrupted Only for halt: CPOS.HALT = 1 ⟶ 0 ➔ CPOS = 0xxx.xxx**N**
TA9		Jog positive CPOS.JOGP = 0 ⟶ 1 CPOS.HALT = 1 ➔ CPOS = 0xx0.**P**xx**1**

1) Key: P = rising edge (positive), N = falling edge (negative), x = any

T	Internal conditions	Actions of the user [1)]
TA10		Positively end jogging Either CPOS.JOGP = 1 ⟶ 0 ➔ CPOS = 0xxx.**N**xx1 or CPOS.HALT = 1 ⟶ 0 ➔ CPOS = 0xxx.xxx**N**
TA11		Jog negative CPOS.JOGN = 0 ⟶ 1 CPOS.HALT = 1 ➔ CPOS = 0xx**P**.0xx1
TA12		End jogging negatively Either CPOS.JOGN = 1 ⟶ 0 ➔ CPOS = 0xx**N**.xxx1 or CPOS.HALT = 1 ⟶ 0 ➔ CPOS = 0xxx.xxx**N**

1) Key: P = rising edge (positive), N = falling edge (negative), x = any

Tab. 5.3 Status transitions at positioning

FHPP operating mode	Notes on special features
Record selection	No restrictions.
Direct application	TA2: The condition that no new record may be processed no longer applies. TA5: A new record can be started at any time.

Tab. 5.4 Special features dependent on FHPP operating mode

5.2.3 Examples of control and status bytes

On the following pages you will find typical examples of control and status bytes:
Example 1: Establish ready status – record selection, Tab. 5.5
Example 2: Establish ready status – direct application, Tab. 5.6
Example 3: Fault handling, Tab. 5.7
Example 4: Homing, Tab. 5.8
Example 5: Positioning record selection, Tab. 5.9
Example 6: Positioning direct application, Tab. 5.10

Information on the finite state machine ➔ section 5.2.
For all examples: If parameterised (➔ PNU 128), the digital input signal ENABLE [X1.6] is also required to create the ready status.
Information on the digital inputs ➔ description GDCP-CMMO-ST-SY-...

Example 1: Establish ready status – record selection

Step	Control bytes (job) 1)		Status bytes (response) 1)	
1.1 Basic status	CCON	= $0000.0x00_b$	SCON	= 0001.0000_b
	CPOS	= 0000.0000_b	SPOS	= 0000.0100_b
1.2 Block device control for FCT (optional)	CCON.LOCK	= 1	SCON.FCT/MMI	= 0
	→ CCON	= $0010.0x00_b$	→ SCON	= 0001.0000_b
	→ CPOS	= 0000.0000_b	→ SPOS	= 0000.0100_b
1.3 Enable drive, enable operation	CCON.ENABLE	= 1	SCON.ENABLED	= 1
	CCON.STOP	= 1	SCON.OPEN	= 1
	CCON.OPM1	= 0	SCON.OPM1	= 0
	CCON.OPM2	= 0	SCON.OPM2	= 0
	CPOS.HALT	= 1	SPOS.HALT	= 1
	→ CCON	= $0010.0x11_b$	→ SCON	= 0001.0011_b
	→ CPOS	= 0000.0001_b	→ SPOS	= 0000.0101_b

1) Key: P = rising edge (positive), N = falling edge (negative), x = any

Tab. 5.5 Control and status bytes - “Establish ready status – record selection”

Description of the steps:

1.1 Initial status of the drive when the supply voltage has been switched on. → Step 1.2 or 1.3

1.2 Block device control for FCT.
Optionally, acceptance of device control by the FCT can be blocked with CCON.LOCK = 1. → Step 1.3

1.3 Enable drive in record selection mode. → Homing: Example 4, Tab. 5.8.

If there are malfunctions after switching on or after setting CCON.ENABLE.
→ Fault handling ➔ example 3, Tab. 5.7.

Example 2: Establish ready status – direct application

Step	Control bytes (job) 1)		Status bytes (response) 1)	
2.1 Basic status	CCON	= $0000.0x00_b$	SCON	= 0001.0000_b
	CPOS	= 0000.0000_b	SPOS	= 0000.0100_b
2.2 Block device control for FCT (optional)	CCON.LOCK	= 1	SCON.FCT/MMI	= 0
2.3 Enable drive, enable operation	CCON.ENABLE	= 1	SCON.ENABLED	= 1
	CCON.STOP	= 1	SCON.OPEN	= 1
	CCON.OPM1	= 1	SCON.OPM1	= 1
	CCON.OPM2	= 0	SCON.OPM2	= 0
	CPOS.HALT	= 1	SPOS.HALT	= 1

1) Key: P = rising edge (positive), N = falling edge (negative), x = any

Tab. 5.6 Control and status bytes “Establish ready status – direct application”

Description of the steps:

2.1 Initial status when the supply voltage has been switched on. ⟶ Step 2.2 or 2.3

2.2 Block device control for FCT. Optionally, acceptance of device control by the FCT can be blocked with CCON.LOCK = 1. ⟶ Step 2.3

2.3 Enable drive in direct application. ⟶ Homing: Example 4, Tab. 5.8.

If there are malfunctions after switching on or after setting CCON.ENABLE.
⟶ Fault handling ➔ example 3, Tab. 5.7.
Warnings do not have to be acknowledged; these are automatically deleted after some seconds when their cause has been remedied.

Example 3: fault handling

Step	Control bytes (job) 1)		Status bytes (response) 1)	
3.1 Errors	CCON	= $xxx0.xxxx_b$	SCON	= $xxxx.1xxx_b$
	CPOS	= $0xxx.xxxx_b$	SPOS	= $xxxx.x0xx_b$
3.1 Warning	CCON	= $xxx0.xxxx_b$	SCON	= $xxxx.x1xx_b$
	CPOS	= $0xxx.xxxx_b$	SPOS	= $xxxx.x0xx_b$
3.3 Acknowledge malfunction with CCON.RESET	CCON.ENABLE	= 1	SCON.ENABLED	= 1
	CCON.RESET	= P	SCON.FAULT	= 0
			SCON.WARN	= 0
			SPOS.ACK	= 0
			SPOS.MC	= 1

1) Key: P = rising edge (positive), N = falling edge (negative), x = any

Tab. 5.7 Control and status bytes “Malfunction handling”

Description of the steps:

3.1 An error is shown with SCON.FAULT. ⟶ Positioning job is no longer possible.

3.2 A warning is shown with SCON.WARN. ⟶ Positioning job remains possible.

3.3 Acknowledge malfunction with rising edge at CCON.RESET. ⟶ Malfunction bit SCON.FAULT or SCON.WARN is reset, ⟶ SPOS.MC is set, ⟶ drive is ready for operation

Example 4: Homing (requires status S4)

Step	Control bytes (job) 1)		Status bytes (response) 1)	
4.1 Start reference travel	CCON.ENABLE	= 1	SCON.ENABLED	= 1
	CCON.STOP	= 1	SCON.OPEN	= 1
	CPOS.HALT	= 1	SPOS.HALT	= 1
	CPOS.HOM	= P	SPOS.ACK	= 1
			SPOS.MC	= 0
4.2 Reference travel is running	CPOS.HOM	= 1	SPOS.MOV	= 1
4.3 Reference travel ended			SPOS.MC	= 1
			SPOS.REF	= 1

1) Key: P = rising edge (positive), N = falling edge (negative), x = any

Tab. 5.8 Control and status bytes “Homing”

Description of the steps:

4.1 A rising edge at CPOS.HOM, (start homing) starts homing. The start is confirmed with SPOS.ACK (Acknowledge start) as long as CPOS.HOM is set.

4.2 Movement of the axis is shown with SPOS.MOV.

4.3 After successful homing, SPOS.B2 MC (Motion Complete) and SPOS.REF are set.

Example 5: Positioning record selection (requires status S4)

Step	Control bytes (job) [1]		Status bytes (response) [1]	
5.1 Record number pre-selection (control byte 3)	Record no.	1 ... 64	Previous record no.	1 ... 64
5.2 Start job	CCON.ENABLE	= 1	SCON.ENABLED	= 1
	CCON.STOP	= 1	SCON.OPEN	= 1
	CPOS.HALT	= 1	SPOS.HALT	= 1
	CPOS.START	= P	SPOS.ACK	= 1
			SPOS.MC	= 0
5.3 Job is running	CPOS.START	= 1	SPOS.MOV	= 1
	Record no.	1 ... 64	Current record no.	1 ... 64
5.4 Job ended	CPOS.START	= 0	SPOS.ACK	= 0
			SPOS.MC	= 1
			SPOS.MOV	= 0

1) Key: P = rising edge (positive), N = falling edge (negative), x = any

Tab. 5.9 Control and status bytes “Positioning record selection”

Description of the steps:

(Steps 5.1 .... 5.4 conditional sequence)

When the ready status is established and homing has been carried out, a positioning job can be started.

5.1 Preselect record number: Byte 3 of the output data

0 = Homing

1 ... 64 = Programmable positioning records

5.2 With CPOS.B1 (START, start job) the preselected positioning job will be started. The start is confirmed with SPOS.ACK (Acknowledge start) as long as CPOS.START is set.

5.3 Movement of the axis is shown with SPOS.MOV.

5.4 At the end of the positioning task, SPOS.MC will be set.

Example 6: Positioning direct application (requires status S4)

Step	Control bytes (job) 1)		Status bytes (response) 1)	
6.1 Preselect position (bytes 5 ... 8) and speed (byte 4)	Speed preselection	0 ... 100 (%)	Speed acknowledgment	0 ... 100 (%)
	Setpoint position	[SINC]	Current position	[SINC]
6.2 Start job	CCON.ENABLE	= 1	SCON.ENABLED	= 1
	CCON.STOP	= 1	SCON.OPEN	= 1
	CPOS.HALT	= 1	SPOS.HALT	= 1
	CDIR.ABS	= S	SDIR.ABS	= S
	CPOS.START	= P	SPOS.ACK	= 1
			SPOS.MC	= 0
6.3 Job is running	CPOS.START	= 1	SPOS.MOV	= 1
6.4 Job ended	CPOS.START	= 0	SPOS.ACK	= 0
			SPOS.MC	= 1
			SPOS.MOV	= 0

1) Key: P = rising edge (positive), N = falling edge (negative), x = any, S= travel condition: 0= absolute; 1 = relative

Tab. 5.10 Control and status bytes for “Positioning direct application”

Description of the steps:

(Step 6.1 ... 6.4 conditional sequence)

When the ready status is achieved and homing has been carried out, a setpoint position must be preselected.

6.1 The setpoint position [SINC] is transferred in bytes 5 ... 8 of the output word.
The setpoint speed [% of the speed basic value] is transferred in byte 4
(0 = no speed; 255 = max. speed, limited internally to 100 %).

6.2 With CPOS.START, the preselected positioning task will be started. The start is confirmed with SPOS.ACK as long as CPOS.START is set.

6.3 Movement of the axis is shown with SPOS.MOV.

6.4 At the end of the positioning task, SPOS.MC is set.

5.3 Configuration of the I/O data

5.3.1 Concept

The FHPP protocol essentially provides 8 bytes for input and output data. The first byte is fixed. It remains intact in each FHPP operating mode and controls enabling of the motor controller and the FHPP operating modes. The other bytes are dependent on the selected FHPP operating mode. Additional control or status bytes and target and actual values can be transmitted here.
In the cyclic data, additional data are permissible to transmit parameters in accordance with the FPC protocol.
A controller exchanges the following data via FHPP:

- Control and status data (8 bytes):
  - Control and status bytes
  - Record number or setpoint position in the output data
  - Feedback of actual position and record number in the input data
  - Additional mode-dependent setpoint and actual values
- If required, additional input and output data (8 bytes) can be used for FPC parameterisation
  ➔ appendix C.

i If applicable, observe the specification in the bus master for the representation of words and double words (Intel/Motorola). For example, the representation via Modbus uses the “big endian” representation (high-order byte first).

5.3.2 I/O data (byte 1 ... 8) in the various FHPP operating modes

Record selection

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Output data	CCON	CPOS	Record number	Reserved	Reserved			
Input data	SCON	SPOS	Record number	RSB	Current position			

Direct application

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Output data	CCON	CPOS	CDIR	Setpoint value1	Setpoint value2			
Input data	SCON	SPOS	SDIR	Actual value1	Actual value2			

Optional: extended I/O data (byte 9 ... 16) for parameterisation in accordance with EFPC (➔ section C.1):

EFPC

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Output data	FPCC	Control and user data independent of the transmission mode ➔ section C.2.2						
Input data	FPCS							

Tab. 5.11 EFPC structure in general

5.4 Assignment of the control bytes and status bytes (overview)

Assignment of the control bytes (overview)								
CCON (All)	**B7** **OPM2**	**B6** **OPM1**	**B5** **LOCK**	**B4** **–**	**B3** **RESET**	**B2** **BRAKE**	**B1** **STOP**	**B0** **ENABLE**
	FHPP operating mode selection		Block FCT access	–	Acknow-ledge malfunc-tion	Release brake	Stop	Enable drive
CPOS (All)	**B7** **–**	**B6** **CLEAR**	**B5** **TEACH**	**B4** **JOGN**	**B3** **JOGP**	**B2** **HOM**	**B1** **START**	**B0** **HALT**
	–	Delete remain-ing path	Teach value	Jog neg-ative	Jog posit-ive	Start homing	Start po-sitioning task	Halt
CDIR (Direct applica-tion)	**B7** **–**	**B6** **–**	**B5** **XLIM**	**B4** **–**	**B3** **–**	**B2** **COM2**	**B1** **COM1**	**B0** **ABS**
	–	–	Deactiv-ate stroke limit value.	–	–	Control mode (position, force, speed, ...)		Absolute/ relative

Tab. 5.12 Overview, assignment of the control bytes

Assignment of the status bytes (overview)								
SCON (All)	**B7** **OPM2**	**B6** **OPM1**	**B5** **FCT/MMI**	**B4** **VLOAD**	**B3** **FAULT**	**B2** **WARN**	**B1** **OPEN**	**B0** **ENABLED**
	Feedback on FHPP operating mode		FCT device control	Load voltage applied	Fault	Warning	Opera-tion en-abled	Drive enabled
SPOS (All)	**B7** **REF**	**B6** **STILL**	**B5** **FOLERR**	**B4** **MOV**	**B3** **TEACH**	**B2** **MC**	**B1** **ACK**	**B0** **HALT**
	Drive re-ferenced	Standstill monito-ring	Following error	Axis is moving	Acknow-ledge teaching or sampling	Motion Com-plete	Acknow-ledge start	Halt
SDIR (Direct applica-tion)	**B7** **–**	**B6** **–**	**B5** **XLIM**	**B4** **VLIM**	**B3** **–**	**B2** **COM2**	**B1** **COM1**	**B0** **ABS**
	–	–	Stroke limit reached	Speed limit reached	–	Feedback control mode (position, force, speed)		Absolute/ relative

Tab. 5.13 Overview, assignment of the status bytes

5.4.1 Description of the control bytes

CCON controls statuses in all FHPP operating modes.

<table>
<tr><th colspan="8">Control byte 1 (CCON)</th></tr>
<tr><th>Bit</th><th>DE</th><th>EN</th><th colspan="5">Description</th></tr>
<tr><td rowspan="2">B0
ENABLE</td><td rowspan="2">Enable drive</td><td rowspan="2">Enable Drive</td><td>= 1:</td><td colspan="4">Enable drive (controller).</td></tr>
<tr><td>= 0:</td><td colspan="4">Drive (controller) blocked.
The ongoing order is stopped (Quick Stop).</td></tr>
<tr><td rowspan="2">B1
STOP</td><td rowspan="2">Stop</td><td rowspan="2">Stop</td><td>= 1:</td><td colspan="4">Enable operation.</td></tr>
<tr><td>= 0:</td><td colspan="4">STOP active (cancel positioning job + stop). The drive stops with quick stop deceleration; the positioning job is reset.</td></tr>
<tr><td rowspan="3">B2
BRAKE</td><td rowspan="3">Release brake</td><td rowspan="3">Open Brake</td><td>= 1:</td><td colspan="4">Release brake.</td></tr>
<tr><td>= 0:</td><td colspan="4">Activate brake.</td></tr>
<tr><td colspan="5">Note: It is only possible to release the brake if the controller is blocked. As soon as the controller is enabled, it has priority over the brake control system.</td></tr>
<tr><td>B3
RESET</td><td>Acknowledge malfunction</td><td>Reset Fault</td><td colspan="5">A malfunction is acknowledged with a rising edge and the malfunction value is deleted.</td></tr>
<tr><td>B4
–</td><td>–</td><td>–</td><td colspan="5">Reserved, must be at 0.</td></tr>
<tr><td rowspan="3">B5
LOCK</td><td rowspan="3">Block FCT access</td><td rowspan="3">Lock FCT Access</td><td colspan="5">Controls access to the local (integrated) parameterisation interface of the motor controller.</td></tr>
<tr><td>= 1:</td><td colspan="4">The software (FCT) cannot take over the device control (HMI control) (may only monitor the motor controller).</td></tr>
<tr><td>= 0:</td><td colspan="4">The software (FCT) can take over the device control (HMI control) (to change parameters or control inputs).</td></tr>
<tr><td rowspan="2">B6
OPM1</td><td rowspan="6">Operating mode selection</td><td rowspan="6">Select Operating Mode</td><td colspan="5">Determining the FHPP operating mode.</td></tr>
<tr><td>No.</td><td>Bit 7</td><td>Bit 6</td><td colspan="2">Operating mode</td></tr>
<tr><td rowspan="4">B7
OPM2</td><td>0</td><td>0</td><td>0</td><td colspan="2">Record selection</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td colspan="2">Direct application</td></tr>
<tr><td>2</td><td>1</td><td>0</td><td colspan="2">Reserved</td></tr>
<tr><td>3</td><td>1</td><td>1</td><td colspan="2">Reserved</td></tr>
</table>

Tab. 5.14 Control byte 1

CPOS controls the positioning sequences in the FHPP operating modes “record selection” and “direct application” as soon as the drive is enabled.

Control byte 2 (CPOS)			
Bit	DE	EN	Description
B0 **HALT**	Halt	**Halt**	= 1: Halt is not requested. = 0: Halt activated (interrupt positioning job). The axis stops with a defined braking ramp. In positioning mode, the positioning task remains active (intermediate stop); the task can be continued with CPOS.START or completed with CPOS.CLEAR. The task is completed in speed and force mode.
B1 **START**	Start positioning task	**Start** Positioning Task	With a rising edge **rising edge**, the current setpoint data are accepted and a positioning task started.
B2 **HOM**	Start homing	Start **Hom**ing	A **rising edge** starts homing with the set parameters.
B3 **JOGP**	Jog positive	**Jog p**ositive	The drive moves at the specified speed or rotational speed in the direction of larger actual values, as long as the bit is set. The movement begins with the rising edge and ends with the falling edge.
B4 **JOGN**	Jog negative	**Jog n**egative	The drive moves at the specified speed or rotational speed in the direction of smaller actual values, as long as the bit is set. The movement begins with the rising edge and ends with the falling edge.
B5 **TEACH**	Teach value	**Teach** actual Value	With a **falling edge**, the current actual value is transferred to the nominal value register of the currently addressed positioning record. The teach target is defined with PNU 520. If the teach target is part of a position set (position, position comparator), in direct mode the record number is determined in PNU 400:1; in record selection, the record number is transmitted into byte 3 of the cyclical data ➔ section 6.4.
B6 **CLEAR**	Delete remaining path	**Clear** Remaining Position	In the “Halt” status, a **rising edge** causes the positioning task to be deleted and a transition to the “Ready” status.
B7 **–**	–	–	Reserved, must be at 0.

Tab. 5.15 Control byte 2

In direct application, CDIR specifies the type of positioning task.

<table>
<tr><th colspan="7">Control byte 3 (CDIR) – direct application</th></tr>
<tr><th>Bit</th><th>DE</th><th>EN</th><th colspan="4">Description</th></tr>
<tr><td rowspan="3">B0
ABS</td><td rowspan="3">Absolute/
relative</td><td rowspan="3">Absolute/
Relative</td><td colspan="4">= 1: Setpoint value is relative to last setpoint value.</td></tr>
<tr><td colspan="4">= 0: Setpoint value is absolute.</td></tr>
<tr><td colspan="4">Considered only in positioning mode (COM1/2 = 00). Whether travel is relative to the last setpoint or actual value can be set in PNU 524.</td></tr>
<tr><td rowspan="2">B1
COM1</td><td rowspan="5">Control mode</td><td rowspan="5">Control Mode</td><td>No.</td><td>Bit 2</td><td>Bit 1</td><td>Control mode</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>Positioning mode</td></tr>
<tr><td rowspan="3">B2
COM2</td><td>1</td><td>0</td><td>1</td><td>Power mode (torque, current)</td></tr>
<tr><td>2</td><td>1</td><td>0</td><td>Speed mode (rotational speed)</td></tr>
<tr><td>3</td><td>1</td><td>1</td><td>Reserved</td></tr>
<tr><td>B3
–</td><td></td><td>–</td><td colspan="4">Reserved, must be at 0.</td></tr>
<tr><td>B4
–</td><td></td><td>–</td><td colspan="4">Reserved, must be at 0.</td></tr>
<tr><td rowspan="3">B5
XLIM</td><td rowspan="3">Deactivate stroke limit value</td><td rowspan="3">stroke (X-)
LIMit inactive</td><td colspan="4">= 1: Stroke monitoring not active</td></tr>
<tr><td colspan="4">= 0: Stroke monitoring active</td></tr>
<tr><td colspan="4">Considered only with force mode or speed mode (COM1/2 = 01 or 10)</td></tr>
<tr><td>B6
–</td><td></td><td>–</td><td colspan="4">Reserved, must be at 0.</td></tr>
<tr><td>B7
–</td><td></td><td>–</td><td colspan="4">Reserved, must be at 0.</td></tr>
</table>

Tab. 5.16 Control byte 3 – direct application

<table>
<tr><th colspan="4">Control byte 4 (setpoint value 1) – direct application</th></tr>
<tr><th>Bit</th><th>DE</th><th>EN</th><th>Description</th></tr>
<tr><td rowspan="6">B0 ... 7</td><td colspan="3">Preselection in positioning mode</td></tr>
<tr><td>Speed</td><td>Velocity</td><td>Speed [% of the basic value] ➔ PNU 540</td></tr>
<tr><td colspan="3">Preselection in force/torque mode</td></tr>
<tr><td>Speed</td><td>Velocity</td><td>Speed [% of the basic value] ➔ PNU 540</td></tr>
<tr><td colspan="3">Preselection in speed mode</td></tr>
<tr><td>Speed ramp</td><td>Velocity ramp</td><td>Speed ramp [% of the basic value] ➔ PNU 560</td></tr>
</table>

Tab. 5.17 Control byte 4 – direct application

Control bytes 5 … 8 (setpoint value 2) – direct application			
Bit	DE	EN	Description
B0 … 31	Preselection in positioning mode, 32-bit number		
	Position	Position	Position [SINC] ➔ appendix A.2
	Preselection in force/torque mode, 32-bit number		
	Torque	Torque	Setpoint torque [% of the basic value] ➔ PNU 555
	Preselection in speed mode, 32-bit number		
	Speed	Velocity	Speed [SINC/s] ➔ appendix A.2

Tab. 5.18 Control bytes 5 … 8 – direct application

Control byte 3 (setpoint value 1) – record selection			
Bit	DE	EN	Description
B0 … 7	Record number	Record number	Preselection of the record number.

Tab. 5.19 Control byte 3 – record selection

Control bytes 4 … 8 (reserved) – record selection			
Bit	DE	EN	Description
B0 … 31	–	–	Reserved (= 0)

Tab. 5.20 Control bytes 4 … 8 – record selection

5.4.2 Description of the status bytes

<table>
<tr><th colspan="7">Status byte 1 (SCON)</th></tr>
<tr><th>Bit</th><th>DE</th><th>EN</th><th colspan="4">Description</th></tr>
<tr><td rowspan="2">B0
ENABLED</td><td rowspan="2">Drive enabled</td><td rowspan="2">Drive Enabled</td><td colspan="4">= 1: Drive (controller) is enabled.</td></tr>
<tr><td colspan="4">= 0: Drive blocked, controller not active.</td></tr>
<tr><td rowspan="2">B1
OPEN</td><td rowspan="2">Operation enabled</td><td rowspan="2">Operation Enabled</td><td colspan="4">= 1: Operation enabled, positioning possible.</td></tr>
<tr><td colspan="4">= 0: Stop active.</td></tr>
<tr><td rowspan="2">B2
WARN</td><td rowspan="2">Warning</td><td rowspan="2">Warning</td><td colspan="4">= 1: Warning is present.</td></tr>
<tr><td colspan="4">= 0: No warning present.</td></tr>
<tr><td rowspan="2">B3
FAULT</td><td rowspan="2">Fault</td><td rowspan="2">Fault</td><td colspan="4">= 1: Fault present.</td></tr>
<tr><td colspan="4">= 0: Fault not present or fault reaction active.</td></tr>
<tr><td rowspan="2">B4
VLOAD</td><td rowspan="2">Load voltage applied</td><td rowspan="2">Load Voltage is Applied</td><td colspan="4">= 1: Load voltage applied.</td></tr>
<tr><td colspan="4">= 0: Load voltage not applied.</td></tr>
<tr><td rowspan="3">B5
FCT/MMI</td><td rowspan="3">Device control by FCT/MMI</td><td rowspan="3">Software Access by FCT/MMI</td><td colspan="4">Device control (refer to PNU 125, section B.4.4)</td></tr>
<tr><td colspan="4">= 1: Device control through fieldbus not possible.</td></tr>
<tr><td colspan="4">= 0: Device control through fieldbus possible.</td></tr>
<tr><td rowspan="2">B6
OPM1</td><td rowspan="6">Operating mode feed-back</td><td rowspan="6">Display Oper-ating Mode</td><td colspan="4">Feedback on FHPP operating mode.</td></tr>
<tr><td>No.</td><td>Bit 7</td><td>Bit 6</td><td>Operating mode</td></tr>
<tr><td rowspan="4">B7
OPM2</td><td>0</td><td>0</td><td>0</td><td>Record selection</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>Direct application</td></tr>
<tr><td>2</td><td>1</td><td>0</td><td>Reserved</td></tr>
<tr><td>3</td><td>1</td><td>1</td><td>Reserved</td></tr>
</table>

Tab. 5.21 Status byte 1

Status byte 2 (SPOS)				
Bit	DE	EN	Description	
B0 **HALT**	Halt	**Halt**	= 1:	Halt is not active; axis can be moved.
			= 0:	Halt is active.
B1 **ACK**	Acknowledge start	**Ack**nowledge Start	= 1:	Start executed (homing, jogging, positioning)
			= 0:	Ready for start (homing, jogging, positioning)
B2 **MC**	Motion Complete	**M**otion **C**omplete	= 1:	Positioning task completed, possibly with error
			= 0:	Positioning task active
			Note: MC is set only after device is switched on (status “Drive blocked”).	
B3 **TEACH**	Acknowledge teach/ sampling	Acknowledge **Teach**/ Sampling	= 1:	Teaching carried out, actual value has been transferred
			= 0:	Ready for teaching
B4 **MOV**	Axis is moving	Axis is **Mov**ing	= 1:	Speed of the axis >= limit value
			= 0:	Speed of the axis < limit value
B5 **FOLERR**	Following error	**FOL**owing **ER**-**R**or	= 1:	Following error active
			= 0:	No following error
B6 **STILL**	Standstill monitoring	Stand**still** Control	= 1:	Axis has left the tolerance window after MC
			= 0:	After MC, axis remains in tolerance window
B7 **REF**	Drive referenced	Axis **Ref**erenced	= 1:	Homing information available; homing does not need to be carried out
			= 0:	Homing must be executed

Tab. 5.22 Status byte 2

The SDIR status byte acknowledges positioning mode.

<table>
<tr><th colspan="7">Status byte 3 (SDIR) – direct application</th></tr>
<tr><th>Bit</th><th>DE</th><th>EN</th><th colspan="4">Description</th></tr>
<tr><td rowspan="2">B0
ABS</td><td rowspan="2">Absolute/
relative</td><td rowspan="2">Absolute/
Relative</td><td>= 1:</td><td colspan="3">Setpoint value is relative to last setpoint value.
Whether travel is relative to the last setpoint or actual value can be set in PNU 524.</td></tr>
<tr><td>= 0:</td><td colspan="3">Setpoint value is absolute.</td></tr>
<tr><td rowspan="2">B1
COM1</td><td rowspan="5">Control mode
feedback</td><td rowspan="5">Control Mode
Feedback</td><td>No.</td><td>Bit 2</td><td>Bit 1</td><td>Control mode</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>Positioning mode</td></tr>
<tr><td rowspan="3">B2
COM2</td><td>1</td><td>0</td><td>1</td><td>Force/torque mode (current)</td></tr>
<tr><td>2</td><td>1</td><td>0</td><td>Speed mode (rotational speed)</td></tr>
<tr><td>3</td><td>1</td><td>1</td><td>Reserved</td></tr>
<tr><td>B3
–</td><td>–</td><td>–</td><td colspan="4">Reserved = 0</td></tr>
<tr><td rowspan="2">B4
VLIM</td><td rowspan="2">Speed limit
reached</td><td rowspan="2">velocity (V-)
LIMit reached</td><td>= 1:</td><td colspan="3">Speed limit reached</td></tr>
<tr><td>= 0:</td><td colspan="3">Speed limit not reached</td></tr>
<tr><td rowspan="2">B5
XLIM</td><td rowspan="2">Stroke limit
reached</td><td rowspan="2">stroke (X-)
LIMit reached</td><td>= 1:</td><td colspan="3">Stroke limit reached</td></tr>
<tr><td>= 0:</td><td colspan="3">Stroke limit not reached</td></tr>
<tr><td>B6
–</td><td>–</td><td>–</td><td colspan="4">Reserved = 0</td></tr>
<tr><td>B7
–</td><td>–</td><td>–</td><td colspan="4">Reserved = 0</td></tr>
</table>

Tab. 5.23 Status byte 3 – direct application

<table>
<tr><th colspan="4">Status byte 4 (actual value 1) – direct application</th></tr>
<tr><th>Bit</th><th>DE</th><th>EN</th><th>Description</th></tr>
<tr><td rowspan="7">B0 ... 7</td><td colspan="3">Feedback in positioning mode</td></tr>
<tr><td>Speed</td><td>Velocity</td><td>Speed [% of the basic value] ➔ PNU 540</td></tr>
<tr><td colspan="3">Feedback in force/torque operation, dependent on parameterisation ➔ PNU 523:7</td></tr>
<tr><td>Speed</td><td>Velocity</td><td>Speed [% of the basic value] ➔ PNU 540</td></tr>
<tr><td>Torque</td><td>Torque</td><td>Torque [% of the basic value] ➔ PNU 555</td></tr>
<tr><td colspan="3">Feedback in speed mode</td></tr>
<tr><td>–</td><td>–</td><td>No function, = 0</td></tr>
</table>

Tab. 5.24 Status byte 4 – direct application

Status bytes 5 … 8 (actual value 2) – direct application			
Bit	DE	EN	Description
B0 … 31	Feedback in positioning mode, 32-bit number		
	Position	Position	Position [SINC] ➔ appendix A.2
	Feedback in force/torque operation, dependent on parameterisation ➔ PNU 523:8		
	Position	Position	Position [SINC] ➔ appendix A.2
	Torque	Torque	Torque [% of the basic value] ➔ PNU 555
	Feedback in speed mode		
	Speed	Velocity	Speed as absolute value [SINC/s]

Tab. 5.25 Status bytes 5 … 8 – direct application

Status byte 3 (record number) – record selection			
Bit	DE	EN	Description
B0 … 7	Record number	Record number	Feedback of record number.

Tab. 5.26 Status byte 3 – record selection

Status byte 4 (RSB) – record selection			
Bit	DE	EN	Description
B0 **RC1**	1st record chaining executed	**1**st **R**ecord **C**haining Done	= 1: The first step enabling condition has been achieved. = 0: A step enabling condition was not configured or not achieved.
B1 **RCC**	Record chaining completed	**R**ecord **C**haining **C**omplete	Valid as soon as MC is present. = 1: Record chain was processed up to the end. = 0: Record chaining aborted. At least one step enabling condition has not been achieved.
B2 **–**	–	–	Reserved, = 0
B3 **–**	–	–	Reserved = 0
B4 **VLIM**	Speed limit reached	velocity (**V**-) **LIM**it reached	= 1: Speed limit reached = 0: Speed limit not reached
B5 **XLIM**	Stroke limit reached	stroke (**X**-) **LIM**it reached	= 1: Stroke limit reached = 0: Stroke limit not reached
B6 **–**	–	–	Reserved = 0
B7 **–**	–	–	Reserved = 0

Tab. 5.27 Status byte 4 – record selection

Status bytes 5 … 8 (position) – record selection			
Bit	DE	EN	Description
B0 … 31	Position	Position	Feedback of position [SINC] ➔ appendix A.2, 32-bit number

Tab. 5.28 Status bytes 5 … 8 – record selection

6 Control via FHPP

6.1 Dimension reference system for electric drives

Information on the dimension reference system ➔ equipment and functional description of the motor controller, GDCP-CMMO-ST-LK-SY-....

6.2 Homing run

With the motor controller CMMO-ST, a homing run must always be performed after power ON (switch on “control section” power supply).

6.2.1 Homing for electric drives

The drive can be referenced with respect to the current position, a reference switch or a stop.
The motor current increases when the drive reaches a stop. Since the drive must not permanently control against the stop, it must move at least one millimetre back into the stroke range. This can take place through selection of a homing method with travel to the zero pulse or through travel to a project zero point off-set away from the stop.

Procedure:

1. Search for the homing point of the configured homing method.
2. Set axis zero point: current position = 0 – offset project zero point.
3. Optional parameterisation: Run relative to the reference point around the “Offset axis zero point”.

Overview of parameters and I/Os in homing		
Homing parameters	**Name**	**PNU**
➔ Page140	Offset axis zero point	1010
	Homing method	1011
	Speeds	1012
	Acceleration/delay	1013
	Max. torque for homing:	1015
	Stop detection speed limit	1016
	Stop damping time	1017
Start (FHPP)	CPOS.HOM = rising edge: start homing (for record selection: record 0 = homing, start with CPOS.START)	
Acknowledgement (FHPP)	SPOS.ACK = rising edge: acknowledge start	
	SPOS.REF = drive homed	
Requirement	Device control through controller/fieldbus	
	Motor controller in the status “Operation enabled”	
	No command for jogging is present	

Tab. 6.1 Parameters and I/Os in homing

6.2.2 Methods of homing

The homing method to be selected depends on the parameterised axis, application and condition of the system.

The homing methods are oriented on CANopen CiA 402.

Accuracy of the homing point

To increase the absolute positioning accuracy, the zero pulse of the incremental encoder can be used for evaluation.

Software end positions

The software end positions are deactivated with the start of homing and activated again after homing is completed.

The following homing methods are possible, depending on the motor configuration.

Motor/operation/reference switch	Possible homing methods	
Motor with encoder in controlled operation (closed loop) without reference switch	-35	Current position
	33	Current position + index – negative direction
	34	Current position + index – positive direction
	-17	Stop – negative direction
	-18	Stop – positive direction
Motor with encoder in controlled operation (closed loop) with reference switch	-35	Current position
	33	Current position + index – negative direction
	34	Current position + index – positive direction
	-17	Stop – negative direction
	-18	Stop – positive direction
	27	Reference switch – positive direction
	23	Reference switch – negative direction
	11	Reference switch + index – negative direction
	7	Reference switch + index – positive direction
Motor without encoder or in controlled operation (open loop) without reference switch	-35	Current position
Motor without encoder or in controlled operation (open loop) with reference switch	-35	Current position
	27	Reference switch – negative direction
	23	Reference switch – positive direction

Tab. 6.2 Possible homing methods

Information on the sequence of homing methods ➔ equipment and functional description of the motor controller, GDCP-CMMO-ST-LK-SY-....

6.3 Jogging

In the “Operation enabled” status, the drive can be travelled with the function “Jog positive” or “Jog negative”. This function is usually used for:

– Moving to teach positions
– Moving the drive out of the way (e.g. after a system error)
– Manual travel as a normal operating mode (manually operated feed)

Sequence

1. When one of the signals “Jog positive” or “Jog negative” is set, the drive starts to move slowly. Due to the slow speed (creep speed), a position can be approached very accurately.
2. If the signal remains set for longer than the parameterised period (phase 1). acceleration takes place until the parameterised fast speed (max. speed) is reached. In this way, large strokes can be traversed quickly.
3. If the signal changes to 0, the drive will be braked with the maximum set deceleration.
4. Only if the drive is referenced:
   If the drive reaches a software end position, it will stop automatically. The software end position is not passed; the path for stopping is taken into account according to the deceleration ramp set. The jog mode can also only be exited here again after Jogging = 0.

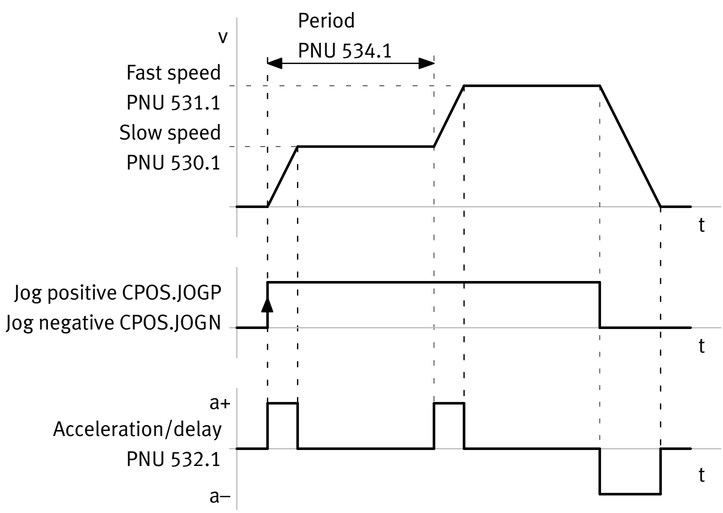


Fig. 6.1 Timing diagram jog operation (“Jog positive” shown as an example)

Overview of parameters and I/Os during jogging		
Parameters involved	**Name of the parameter**	**PNU**
➔ Section B.4.10	Speed slow – phase 1	530
	Max. speed – phase 2	531
	Acceleration/delay	532
	Crawling duration phase 1	534
	Following error message window (jog operation)	538
	Following error delay time	539
Start (FHPP)	CPOS.JOGP = rising edge: jog positive (larger actual values)	
	CPOS.JOGN = rising edge: jog negative (smaller actual values)	
Acknowledgement (FHPP)	SPOS.MOV = 1: Drive moves	
	SPOS.MC = 0: (motion complete)	
Requirement	Device control through controller/fieldbus	
	Motor controller in the status “Operation enabled”	

Tab. 6.3 Parameters and I/Os during jog mode

6.4 Teaching via fieldbus

Absolute position values can be taught via the fieldbus. Previously taught position values will then be overwritten. Teaching of relative position records, speed records or force records in not possible and results in warning 0x40 “Last teaching not successful”.
Note: The drive must not stand still for teaching. Of course, imprecisions of several millimetres are possible even at low speeds due to the normal cycle times of the motor controller, data transmission and the higher-order controller. The speed must be set during teaching in such a way that the position is detected accurately enough.

Sequence

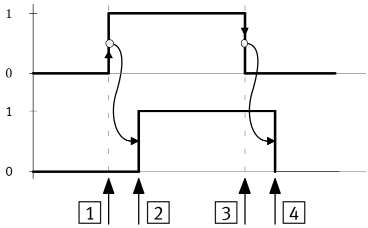
1. The drive will be moved to the desired position via the jogging mode or manually. This can be accomplished in jogging mode by positioning (or by moving manually in the “Drive blocked” status in the case of motors with an encoder).
2. Parameterise desired teaching target. For “setpoint position in position sets” and the position comparators, specify the record number in PNU 400 (direct mode) or in control byte 3 (record selection).

Teach target (PNU 520)	is taught
= 1 (specification)	Setpoint position in position set[1)] ➔ PNU 404
= 2	Axis zero point ➔ PNU 1010
= 3	Project zero point ➔ PNU 500
= 4	Lower software end position ➔ PNU 501.1
= 5	Upper software end position ➔ PNU 501.2
= 6	Position comparator upper limit[1)] ➔ PNU 430
= 7	Position comparator lower limit[1)] ➔ PNU 431

1) Record number in direct mode via PNU 400.1 “Setpoint record number”; in case of record selection via record number, specify in control byte 3

Tab. 6.4 Overview of teach targets

3. Teaching takes place via the handshake of the bits in the control and status bytes CPOS/SPOS:



1 Controller: Prepare teaching
2 Motor controller: Ready for teaching
3 Controller: Teach now
4 Motor controller: Value accepted

Fig. 6.2 Handshake during teaching

Taught values, like all written parameters, must be permanently stored by writing PNU 127:2 with the value 1 so they will be secure in case of power failure.

Overview of parameters and I/Os when teaching		
Parameters involved	**Parameter**	**PNU**
➔ Sections B.4.8, B.4.10	Teach target	520
	Record number	400
	Offset project zero point	500
	Software end positions	501
	Axis zero point offset (electric drives)	1010
Start (FHPP)	CPOS.TEACH = N (falling edge, negative): Teach value	
Acknowledgement (FHPP)	SPOS.TEACH = N (falling edge, negative): Value accepted	
Requirement	Device control through controller/fieldbus	
	Motor controller in the status “Operation enabled”	

Tab. 6.5 Parameters and I/Os when teaching

6.5 Execute Record

A record can be started in the “Operation enabled” status. This function is usually used for:
- selection-free approach to positions in the record list by the controller
- selection-free running of speeds and forces of the record list by the controller
- processing a positioning profile by linking records
- known target positions that seldom change (recipe change)

Sequence
1. Set the desired record number in the controller's output data. Until the start, the motor controller continues to reply with the number of the record last processed.
2. With a rising edge at CPOS.START, the controller accepts the record number and starts the positioning job.
3. The motor controller signals with the rising edge at SPOS.ACK that the output data of the controller has been taken over and the positioning task is now active. The positioning command continues to be executed, even if CPOS.START is reset to zero.
4. When the record is concluded, SPOS.MC is set.

Causes of errors in application:
- Homing has not been carried out.
- The target position has not been reached.
- The record number is invalid.
- The record has not been initialised.

With record chaining ➔ section 6.5.3:
If a new speed and/or target position is specified in the movement, the remaining path to the target position must be large enough to reach the destination with the braking ramp that was set.
If this destination cannot be reached with the parameterised speed, acceleration or deceleration, fault message 0x25 (path calculation) is reported.

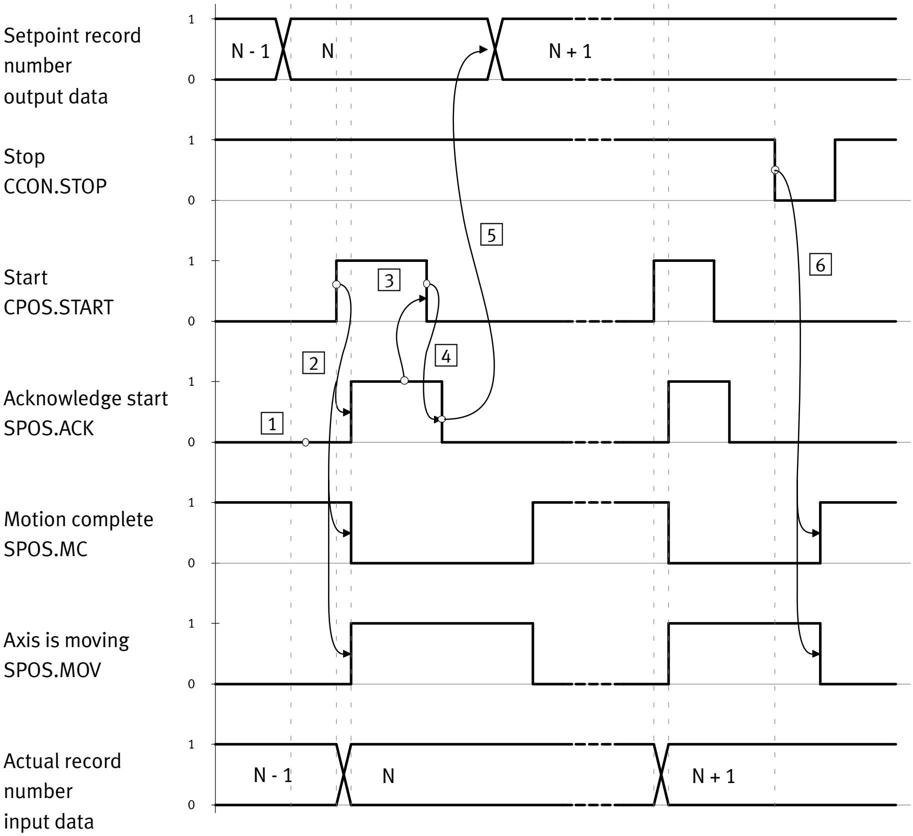
Overview of parameters and I/Os in record selection		
Parameters involved **➔ Section B.4.8**	**Parameter**	**PNU**
	Record number	400
	All parameters of the record data ➔ section 6.5.2, Tab. 6.7	401 ... 442
Start (FHPP)	Record number in control byte 3 CPOS.START = rising edge: start Jogging and referencing have priority.	
Acknowledgement (FHPP)	SPOS.MC = 0: Motion Complete	
	SPOS.ACK = rising edge: acknowledge start	
	SPOS.MOV = 1: Drive moves	
Requirement	Device control through controller/fieldbus	
	Motor controller in the status "Operation enabled"	
	Valid record number is present	

Tab. 6.6 Parameters and I/Os with record selection

6.5.1 Record selection flow diagrams

Fig. 6.3, Fig. 6.4 and Fig. 6.5 show the flow diagram for starting and stopping of a record.

Record start/stop



1 Requirement: “Start acknowledgment” = 0

2 Rising edge at “Start” causes the new record number N to be accepted and “Start acknowledgment” to be set.

3 As soon as “Start acknowledgement” is recognised by the controller, “Start” may be set to 0 again.

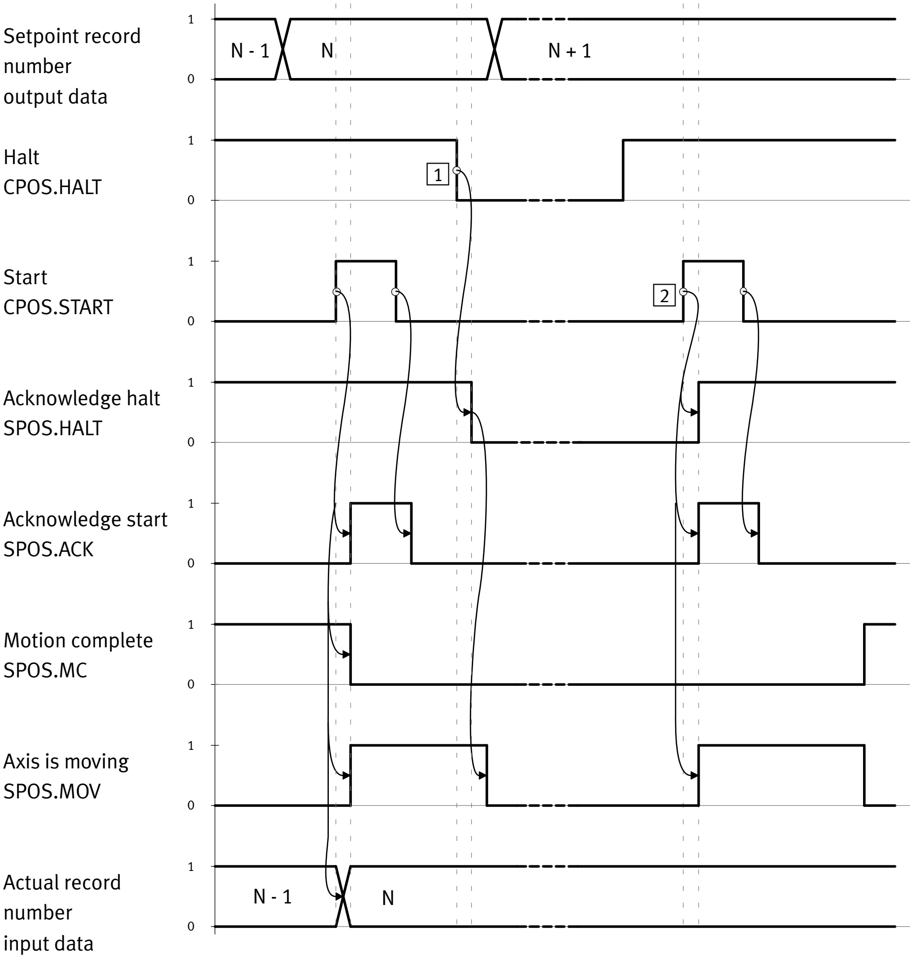
4 The motor controller reacts with a trailing edge at “Start acknowledgment”.

5 As soon as “Start acknowledgment” is recognized by the controller, it can create the next record number.

6 A currently running positioning task can be stopped with “Stop”.

Fig. 6.3 Flow diagram, record start/stop

Stop record with halt and continue

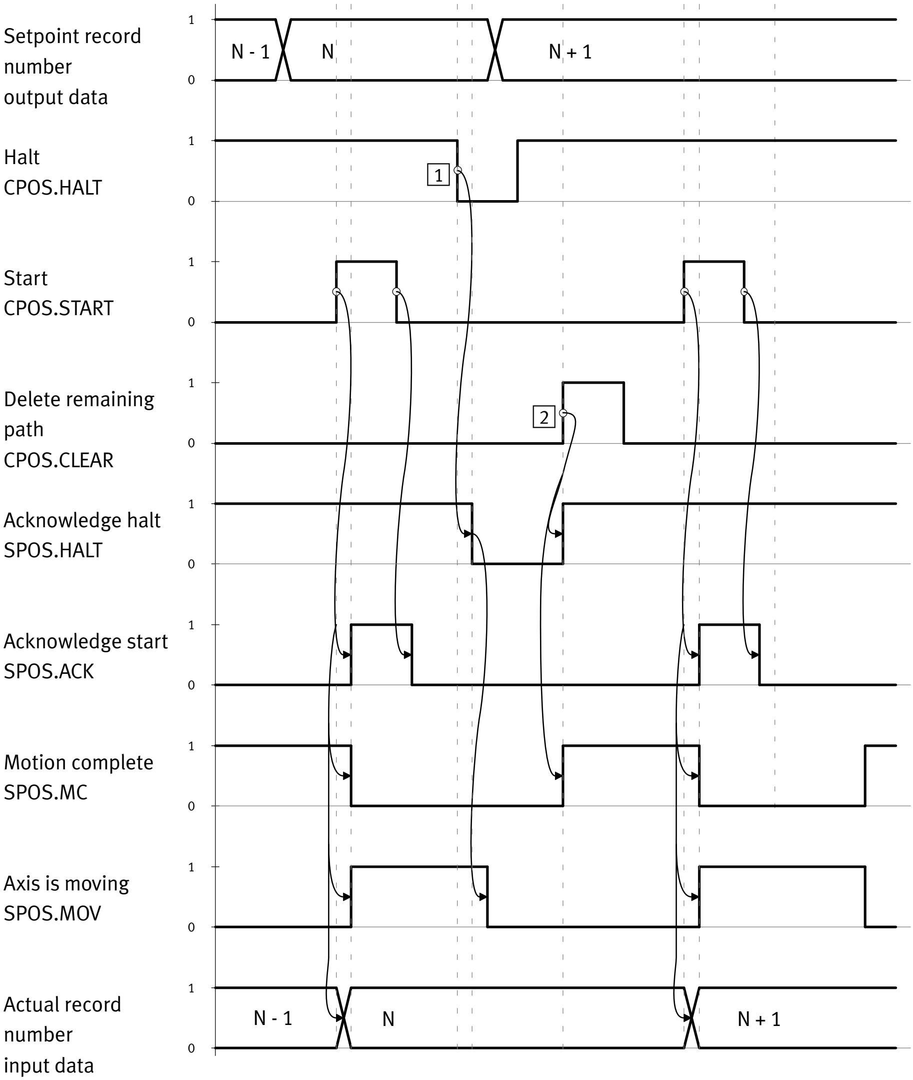


1 Record is stopped with "halt"; actual record number N is retained; "Motion Complete" remains reset

2 Rising edge at "Start" starts record N again; "Confirm halt" is set

Fig. 6.4 Stop and continue flow diagram for record with halt

Stop record with halt and delete remaining path



1 Stop record

2 Delete remaining path

Fig. 6.5 Flow diagram for stop record with halt and delete remaining path

6.5.2 Record structure

A positioning task in record select mode is described by a record made up of setpoint values. Every setpoint value is addressed via its own parameter number (PNU). A record consists of the setpoint values with the same subindex.

PNU	Name	Description
401	Record control byte 1	Setting for positioning task: absolute/relative
402	Record control byte 2	Record control: settings for record chaining
404	Setpoint value position	Target position setpoint value
406	Speed	Setpoint speed
407	Acceleration	Nominal acceleration
408	Deceleration	Setpoint deceleration
409	Jerk acceleration	Setpoint value of the jerk during acceleration
410	Load	Load that is moved in addition to the basic load
416	Record chaining target	Record number that is jumped to if the step enabling condition is active
417	Jerk deceleration	Setpoint value of the jerk during deceleration
418	Torque limitation	Setpoint value of max. force
421	Record control byte 3	Specific behaviour of the record during ongoing positioning
423	Final speed	Setpoint value of the final speed at the end of the record.
424	Max. deviation	Setpoint value for max. deviation.
425	MC with record sequencing	Signal status “Motion Complete” (MC) with record sequencing
426	Start delay	Setpoint value for the start delay time
427	Stroke limit	Setpoint value for the path related to the start position
428	Factor torque pilot control	Share of torque pilot control
430	Position comparator, min.	Lower limit of position comparator
431	Position comparator, max.	Upper limit of position comparator
432	Position comparator damping time	Setpoint value for position comparator damping time
433	Velocity comparator, min.	Velocity comparator lower limit
434	Velocity comparator, max.	Velocity comparator upper limit
435	Velocity comparator damping time	Setpoint value for velocity comparator damping time
436	Force comparator, min.	Lower limit of force comparator
437	Force comparator, max.	Upper limit of force comparator
438	Force comparator damping time	Setpoint value for force comparator damping time
439	Time comparator, min.	Lower limit of time comparator
440	Time comparator, max.	Upper limit of time comparator

PNU	Name	Description
441	Setpoint value speed	Setpoint value of target speed
442	Setpoint value force	Setpoint value of target speed

Tab. 6.7 Parameters for positioning record

6.5.3 Record linking (PNU 402)

Record selection mode allows multiple positioning jobs to be linked. This means that, starting at CPOS.START, several records are automatically executed one after the other. This allows a travel profile to be defined, such as switching to another speed after a position is reached.
To do this, the user sets a condition in the record control byte and the entry of the subsequent record number in PNU 416 to define that the subsequent record is automatically executed after the current record.

Record control byte 2 (PNU 402)	
Bit 0 ... 6	Numerical value 0 ... 128: step enabling condition as a list ➔ Tab. 6.9
Bit 7	Reserved

Tab. 6.8 Settings for record chaining

Step enabling condition for automatic record chaining (bit 0 ... 6)			
Value	Com-mand	Condition	Description
0	Inactive	End of the sequence	No record sequencing.
1	MC active	Motion Complete	The preselection value from PNU 426 is interpreted as a delay in milliseconds. Continuing takes place once the target setpoint value is reached, i.e. once the internal MC condition is fulfilled.
20	PosC active	Position comparator active	Continuation takes place if the limit of the position comparator has been reached.
21	VC active	Velocity comparator active	Continuation takes place if the limit of the velocity comparator has been reached.
22	FC active	Force comparator active	Continuation takes place if the limit of the force comparator has been reached.
23	TC active	Time comparator active	Continuation takes place if the limit of the time comparator has been reached.

Tab. 6.9 Step enabling conditions

6.6 Execute direct mode

In the status “Operation enabled” (direct mode), a job is formulated directly in the I/O data and transmitted via the fieldbus. Some of the setpoint values are reserved in the controller.
The function is used in the following situations:

– Selection-free approach to positions within the working stroke
– Unknown target positions in project engineering
– Frequent changes of the target position (e.g. many different workpiece positions)
– Corresponding jobs in force mode or velocity mode

If short wait times are not critical, it is possible to implement a positioning profile externally, controlled by the controller, by linking positions.

Causes of errors in application

– No homing carried out
– Target position cannot be reached or lies outside the software end positions

Overview of parameters and I/Os in direct application of positioning mode			
Parameters involved		**Parameter**	**PNU**
	FHPP direct mode ➔ B.4.13	FHPP setpoint/actual values	523
		Settings of the FHPP direct mode	524
	Direct mode position ➔ B.4.15	Basic value speed[1]	540
		Acceleration	541
		Deceleration	542
		Jerk acceleration	543
		Load	544
		Jerk deceleration	547
		Final speed	548
		Following error (direct mode position)	549
	FHPP direct mode general ➔ B.4.18	Torque limitation (not force mode)	581
		Start delay	582
		Start condition	583
Start (FHPP)		CPOS.START = rising edge: start	
		CDIR.ABS = absolute/relative setpoint value	
		CDIR.COM1/2 = control mode ➔ section 5.3	
Acknowledgement (FHPP)		SPOS.MC = 0: Motion Complete	
		SPOS.ACK = rising edge: acknowledge start	
		SPOS.MOV = 1: Drive moves	
Requirement		Device control through controller/fieldbus	
		Motor controller in the status “Operation enabled”	

1) The controller transfers a percentage value in the control bytes, which is multiplied by the base value to get the setpoint value

Tab. 6.10 Parameters and I/Os in direct application of positioning mode

Overview of parameters and I/Os in direct application of force mode		
Parameters involved	**Parameter**	**PNU**
FHPP direct mode ➔ B.4.13	FHPP setpoint/actual values	523
	Settings of the FHPP direct mode	524
Direct mode position ➔ B.4.15	Basic value speed[1)]	540
	Load	544
Force direct mode ➔ B.4.16	Message window for force reached	552
	Basic value force	555
FHPP direct mode general ➔ B.4.18	Start delay	582
	Start condition	583
Electric drives ➔ B.4.23	Nominal motor torque	1036
Start (FHPP)	CPOS.START = rising edge: start	
	CDIR.ABS = absolute/relative setpoint value	
	CDIR.COM1/2 = control mode ➔ section 5.3	
Acknowledgement (FHPP)	SPOS.MC = 0: Motion Complete	
	SPOS.ACK = rising edge: acknowledge start	
	SPOS.MOV = 1: Drive moves	
Requirement	Device control through controller/fieldbus	
	Motor controller in the status “Operation enabled”	

1) The controller transfers a percentage value in the control bytes, which is multiplied by the base value to get the setpoint value

Tab. 6.11 Parameters and I/Os in direct application of force mode

Overview of parameters and I/Os in direct application of speed mode		
Parameters involved	**Parameter**	**PNU**
FHPP direct mode ➔ B.4.13	FHPP setpoint/actual values	523
	Settings of the FHPP direct mode	524
Direct mode position ➔ B.4.15	Jerk acceleration	543
	Load	544
	Jerk deceleration	547
Direct mode speed ➔ B.4.17	Basic value acceleration[1)]	560
	Message window for speed reached	561
	Stroke limit (speed adjustment)	566
	Message window for deviation (speed adjustment)	568
FHPP direct mode general ➔ B.4.18	Torque limitation (not force mode)	581
	Start delay	582
	Start condition	583
Start (FHPP)	CPOS.START = rising edge: start	
	CDIR.ABS = absolute/relative setpoint value	
	CDIR.COM1/2 = control mode ➔ section 5.3	
Acknowledgement (FHPP)	SPOS.MC = 0: Motion Complete	
	SPOS.ACK = rising edge: acknowledge start	
	SPOS.MOV = 1: Drive moves	
Requirement	Device control through controller/fieldbus	
	Motor controller in the status “Operation enabled”	

1) The controller transfers a percentage value in the control bytes, which is multiplied by the base value to get the setpoint value

Tab. 6.12 Parameters and I/Os in direct application of speed mode

6.6.1 Direct mode sequence

1. Set the desired setpoint values and positioning conditions (absolute/relative, ...) in the output data.
2. With a rising edge at Start (CPOS.START), the motor controller accepts the setpoint values and starts the positioning job.
3. Depending on the parameterisation of the start condition PNU 583:
   – Ignore (default): A new start command is ignored as long as the last job is not finished
   – Interrupt: After the start, a new setpoint value can be started at any time. It is not necessary to wait for the “Motion Complete” (MC) signal
   – Wait: Start of the new job after Motion Complete
4. Once the last setpoint value has been reached, the signal “MC” (SPOS.MC) is set.

Start of the job

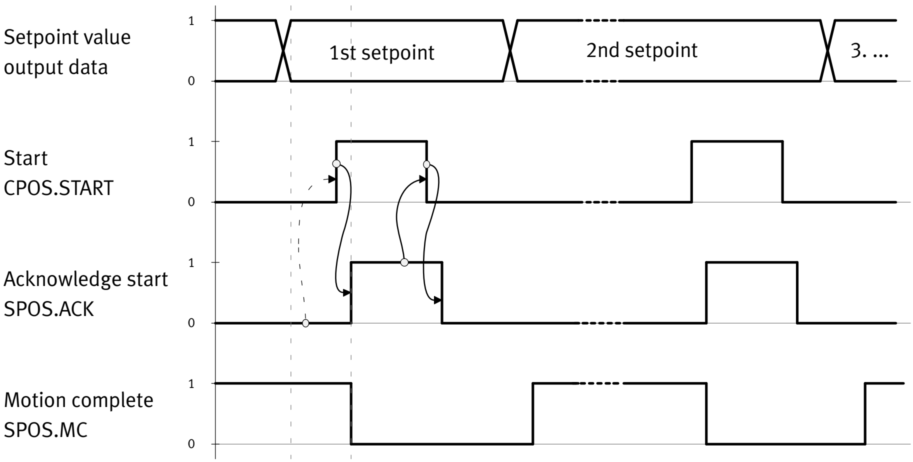


Fig. 6.6 Start of the positioning task

The sequence of the remaining control and status bits as well as the functions Halt and Stop reacts corresponding to the record selection function ➔ Fig. 6.3, Fig. 6.4 and Fig. 6.5.

Positioning mode

The positioning mode is specified by determination of the control mode with the bits CDIR.COM1/2.
The position setpoint value is dependent, absolutely or relatively, on the bit CDIR.ABS.
The setpoint values are specified as follows:

Bytes	Content	Value
Setpoint values		
Control byte 4	Setpoint value 1	Speed [% of the basic value] ➔ PNU 540
Control bytes 5 … 8	Setpoint value 2	Position [SINC], 32-bit number ➔ appendix A.2
Actual values		
Status byte 4	Actual value 1	Speed [% of the basic value] ➔ PNU 540
Status bytes 5 … 8	Actual value 2	Position [SINC], 32-bit number ➔ appendix A.2

Tab. 6.13 Setpoint and actual values – direct application positioning mode

After the setpoint specification, travel to the position in accordance with the setpoint values begins with the start signal (start bit) and the active position control mode is displayed via the SDIR.COM1/2 bits.
The signal “Motion Complete” (MC) reports “Position reached” in this control mode.

Speed mode (speed adjustment)

The speed mode is specified by determination of the control mode with the bits CDIR.COM1/2. The speed setpoint value is always absolute; CDIR.ABS is ignored.
The setpoint values are specified as follows:

Bytes	Content	Value
Setpoint values		
Control byte 4	Setpoint value 1	Speed ramp [% of the basic value] ➔ PNU 560
Control bytes 5 … 8	Setpoint value 2	Speed [SINC/s] ➔ appendix A.2
Actual values		
Status byte 4	Actual value 1	No function, = 0
Status bytes 5 … 8	Actual value 2	Speed as absolute value [SINC/s]

Tab. 6.14 Setpoint and actual values – direct application speed adjustment

After the setpoint specification, with the start signal (start bit), the speed is built up in the direction indicated by the prefix of the setpoint value 2 and the active speed control mode is displayed via the SDIR.COM1/2 bits.
The signal SPOS.MC (Motion Complete) in this control mode reports “target speed reached”.

Force/torque mode (current control)

The force/torque mode is specified by determination of the control mode with the bits CDIR.COM1/2. The drive first stops with the position controlled.
The setpoint values are specified as follows:

Bytes	Content	Value
Setpoint values		
Control byte 4	Setpoint value 1	Speed [% of the basic value] ➔ PNU 540
Control bytes 5 … 8	Setpoint value 2	Setpoint torque [% of the basic value] ➔ PNU 555
Actual values, dependent on the parameterisation ➔ PNU 523		
Status byte 4	Actual value 1	Speed [% of the basic value] ➔ PNU 540
		Torque [% of the basic value] ➔ PNU 555 (factory setting)
Status bytes 5 … 8	Actual value 2	Position [SINC] ➔ appendix A.2 (factory setting)
		Torque [% of the basic value] ➔ PNU 555

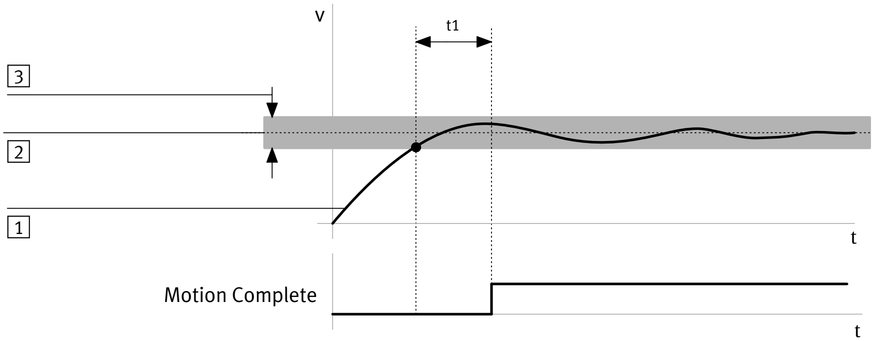
Tab. 6.15 Setpoint and actual values – direct application speed adjustment

After the setpoint specification, with the start signal (start bit), the torque is built up in the direction indicated by the prefix of the setpoint value 2, and the active torque control mode is displayed via the SDIR.COM1/2 bits.
The signal SPOS.MC (Motion Complete) in this control mode reports “carried out/done” or “Actual force = Setpoint force”.

6.7 Monitoring of the drive behaviour

6.7.1 “Motion Complete” message

“Motion Complete” signals the end of an order. A window is defined for each order type (position, velocity or force mode). As soon as the actual value of the target variable is in the target window for the duration of the parameterised damping time, the message Motion Complete (task ended) is triggered.



t1: Damping time, Motion Complete

1 Actual speed
2 Setpoint speed
3 Time window, Motion Complete

Fig. 6.7 Motion Complete – example of velocity mode

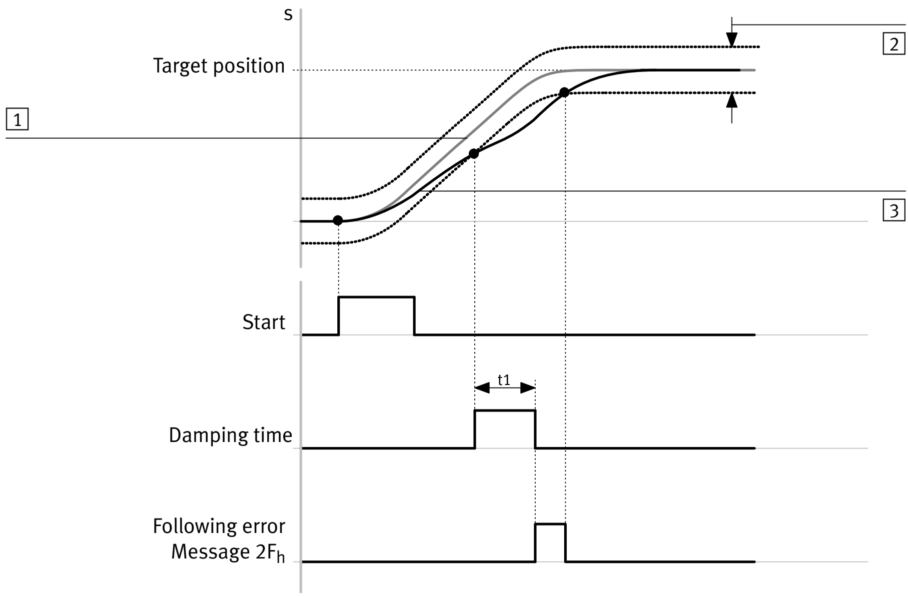
Overview of parameters and I/Os in Motion Complete		
Parameters involved	**Parameters**	**PNU**
Force direct mode ➔ B.4.16	Force reached message window (target window for force mode in record selection/direct mode)	552
Direct mode speed ➔ B.4.17	Speed reached message window (target window for speed mode in record selection/direct mode)	561
Controller parameter (positioning mode direct mode) ➔ B.4.22	Target reached message window (target window for positioning mode with record selection/direct mode)	1022
	Target reached damping time (Motion Complete damping time, applicable for all task types)	1023
Acknowledgement (FHPP)	SPOS.MC = 1: Motion Complete	

Tab. 6.16 Parameter and I/Os in Motion Complete

6.7.2 “Following error” message

In position and speed mode, exceeding of the max. permissible following error can be monitored, e.g. in the case of sluggishness or overload of the drive.

A theoretical progression is calculated from the parameters of a job before it is executed (➔ Fig. 6.8, 1). While carrying out an order, the variance between the calculated setpoint and the current actual value is monitored. The permitted difference (max. permissible following error) is determined in parameterisation. The message is enabled after the damping time has expired if the difference between the setpoint and actual value of the current controlled variable (path, speed) lies outside the permitted difference.



t1: Damping time for following error

1 Setpoint positioning process
2 Max. following error – PNU 424, 549, 568
3 Actual positioning process

Fig. 6.8 Timing diagram: message “Following error” – position control example, following error parameterised as warning

The error management of FCT permits parameterising the reaction to this message ($2F_h$) (➔ FCT error management). If the following error has been configured as a warning, the message is automatically deleted when the actual value is again within the following error window.

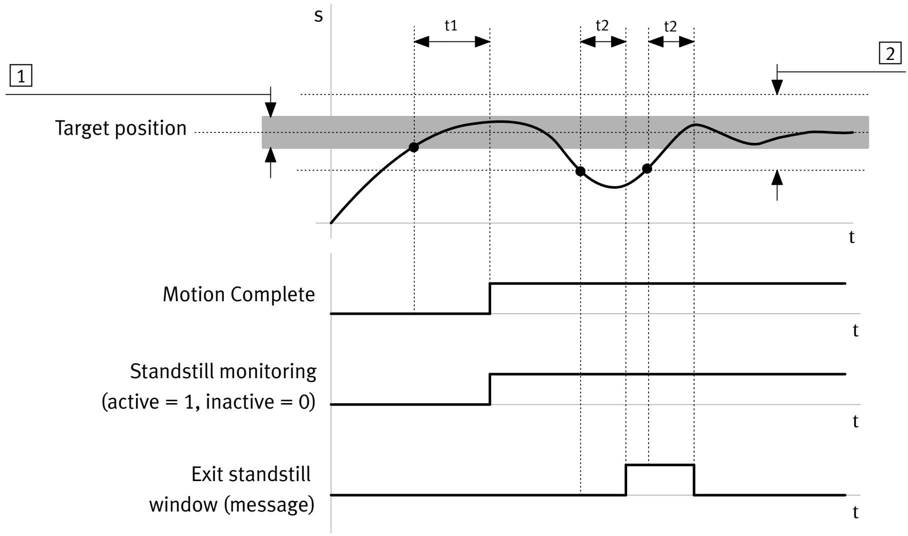
Overview of parameters and I/Os in following error			
Parameters involved		**Parameters**	**PNU**
	Record data ➔ B.4.8	Max. deviation (max. following error for positioning mode or speed mode in record selection)	424
	Direct mode position ➔ B.4.15	Following error message window (max. following error for positioning mode in direct mode)	549
	Direct mode speed ➔ B.4.17	Deviation message window (max. following error for speed mode in direct mode)	568
	Following error monitoring ➔ B.4.22	Delay time for following error (damping time following error message for all jobs)	1045
Acknowledgement (FHPP)		SPOS.FOLERR = 1: following error	

Tab. 6.17 Parameters and I/Os in following error

6.7.3 “Standstill monitoring” message

The standstill monitoring function checks in positioning mode if the actuator is within the standstill window of the target position for the duration of the damping time (➔ Fig. 6.9).
Standstill monitoring is automatically activated after the target position is reached (“Motion Complete”). Standstill monitoring can be suppressed, if required, by setting the standstill window to the value “0”.
During standstill monitoring, if the actual position of the drive leaves the standstill window for the duration of the standstill monitoring time, e.g. due to external forces, the motor controller reacts as follows:

– The “standstill monitoring” diagnostic message is triggered.
  The reaction to the diagnostic message can be parameterised with the FCT error management. If the diagnostic message is parameterised as a warning, the message is automatically deleted as soon as the actual position is within the standstill window again or a new job is started.
– The position controller attempts to return the actuator to the standstill window.



t1: Damping time, Motion Complete

t2: Damping time, standstill monitoring

1 Target window – PNU 1022

2 Standstill window

Fig. 6.9 Standstill monitoring - example

Overview of parameters and I/Os in standstill monitoring		
Parameters involved	**Parameters**	**PNU**
Controller parameter (positioning mode direct mode) ➔ B.4.22	Target reached message window (time window, Motion Complete)	1022
	Target reached damping time (Motion Complete damping time)	1023
Standstill monitoring ➔ B.4.24	Standstill message window (standstill window)	1042
	Standstill delay time (damping time, standstill monitoring)	1043
Acknowledgement (FHPP)	SPOS.STILL = 1: Exit standstill window	

Tab. 6.18 Parameters and I/Os in standstill monitoring

6.7.4 Comparators

Comparators are used to check if a value lies within a defined range of values (window). Comparators are used:

– for control of record chaining (➔ chapter 6.5.3)
– for messaging on a digital output (if configured ➔ equipment and functional description of the motor controller, GDCP-CMMO-ST-LK-SY-...)
– for messaging via PNU 312

The window is defined by a lower and an upper limit value. If the monitored value is within this window, the related comparator message is enabled. If a time can be indicated for the comparator, the monitored value must be within this window for the stipulated length of time. The message is inactive outside this window.

A plausibility check takes place: If the lower limit value is larger than the upper limit value, the comparator message is never active.
The limits are specified for negative ranges of values with algebraic symbol. That algebraic symbol indicates the direction. Example “Position comparator”:
-50 mm (= minimum) ≤ actual position ≤ -40 mm (= maximum).

Comparator	Description
Time	The message is enabled if the elapsed time since the start of the order lies within the window.
Position	The limit values must lie within the permissible range between the software end positions. Even the relative position limits are always specified in absolute values (reference to the zero point). The message is active if the actual value for the parameterised time is within the window.
Speed	The message is active if the actual value for the parameterised time is within the window.
Force[1)]	The limits are specified between -1000 to +1000 ‰ related to the force base value PNU 555. The message is active if the actual value for the parameterised time is within the window.

1) Only present in closed-loop operation.

Tab. 6.19 Comparators

Overview of parameters for the comparators		
Parameters involved	**PNU**	
Status message FHPP	General process data ➔ B.4.6	
Status of comparator outputs	312	
Comparators	Record selection ➔ B.4.9	Direct mode ➔ B.4.18
Position comparator, min.	430	585
Position comparator, max.	431	586
Position comparator, damping time	432	587
Speed comparator, min.	433	588
Speed comparator, max.	434	589
Speed comparator, damping time	435	590
Force comparator, min.	436	591
Force comparator, max.	437	592
Force comparator, damping time	438	593
Time comparator, min.	439	594
Time comparator, max.	440	595

Tab. 6.20 Parameters and I/Os in standstill monitoring

7 Diagnostics

7.1 Diagnostic messages

7.1.1 Classification and error responses

The diagnostic messages of the motor controller are classified as errors, warnings or information.

Type	Description	FHPP
Error (Error)	When an error occurs, the motor controller changes to the error status (DOUT READY 1➔ 0). Errors always generate an error reaction that has an effect on the behaviour of the drive, e.g. stop behaviour, switching off the output stage (➔ Tab. 7.2). To restore the ready status, errors require: – elimination of the error cause – an acknowledgment or a restart (reset)	SCON.FAULT bit is set
Warning (Warning)	Warnings have no influence on the behaviour of the drive and do not need to be acknowledged. To avoid a subsequent error: Clarify the cause of the warning and eliminate it.	The SCON.WARN bit is set. If the cause has been eliminated, the bit is automatically deleted again.
Information (Information)	Information messages have no influence on the behaviour of the drive and do not need to be acknowledged.	—

Tab. 7.1 Classification of the diagnostic messages

Error response	Description
Free run-out (Free-wheeling)	– The output stage is switched off. – The drive then gradually comes to rest.
Quick stop deceleration (QS deceleration)	– The movement is stopped immediately with the parameterised quick stop deceleration. – The output stage can then optionally [2)] be switched off.
Job deceleration (Command deceleration)	– The movement is stopped immediately with the deceleration used in the current job. – The output stage can then optionally [2)] be switched off.
End job (Finish command)	– The current job is executed until the target is reached (Motion Complete). – The output stage can then optionally [2)] be switched off.

2) Parameterisation with PNU 234 ➔ B.4.5 or with FCT

Tab. 7.2 Error response (stop behaviour)

Parameterisable diagnostic messages can be adapted through the error management ➔ B.4.5, PNU 242 and 246:
- Classification as error, warning or information
- Selection of the error reaction (stop behaviour, switching off of the output stage)
- Entry in the diagnostic memory

7.1.2 Display of a diagnostic event

Dependent on the type of the corresponding message, a diagnostic event is shown through display of the device status or designation of the message or hex code ➔ 7.2.

Type	Display	
Error	7-segment	Hex code
	FCT	Online tab device status: status "error", designation
	Web server	Status "error"
Warning	7-segment	Hex code
	FCT	Online tab device status: status "warning", designation
	Web server	Status "warning"

Tab. 7.3 Display of a diagnostic message

Information on current messages can be read via FHPP.

PNU	Description	
205	Device fault	Read the active fault with the highest priority.
220	Current fault messages	Read all existing faults.
221	Current warning messages	Read all existing warnings.
230	Current fault can be acknowledged	Read the acknowledgment type of the currently highest priority fault.

Tab. 7.4 Reading out diagnostic messages

Additionally, diagnostic events can be read from the diagnostic memory. Messages of type "information" are not displayed and can only be read via FCT or web server. Additional information on the diagnostic memory ➔ 7.1.3.

7.1.3 Diagnostic memory

The motor controller has a permanent diagnostic memory for logging diagnostic messages. The diagnostic memory is designed as a ring memory with a capacity of 200 diagnostic messages.
The following information is included in the diagnostic messages of the diagnostic memory:

Information	PNU	Description
Counter (Counter)	–1)	Counter number of the diagnostic message
Type (Type)	200	Classification of the diagnostic message ➔ Tab. 7.1
Number (No.)	201	Hexadecimal number of the message (0x = hex prefix) ➔ 7.2.2
Message (Message)	–1)	Brief description of the diagnostic message
Timestamp (Timestamp)	202	Time of the diagnostic message in the form "HH.MM.SS:nnn (HH = hours, MM = minutes, SS = seconds, nnn = milliseconds). Time base is the respective switch-on time of the motor controller.
Additional information (Additional Info)	203	Additional information for Festo Service in case of complex faults

1) Not available via FHPP

Tab. 7.5 Structure of diagnostic messages

The diagnostic messages are written one after the other in the diagnostic memory. The entry is optional for parameterisable diagnostic messages ➔ 7.2.2. If the diagnostic message has reached the maximum capacity, the oldest diagnostic message is overwritten by the newest one.

Structure of the diagnostic memory in the FHPP				
PNU1)	**200**	**201**	**202**	**203**
Content	Diagnostic event	Diagnostic number	Timestamp	Additional information
Format	uint8	uint16	uint32	uint32
Subindex 1	Newest saved diagnostic message			
Subindex 2	2nd saved diagnostic message			
Subindex ...	...			
Subindex 200	200th saved diagnostic message			

1) ➔ B.4.5

Tab. 7.6 Components of the diagnostic memory in PNU 200 ... 203

The diagnostic memory can be deleted as needed with web server, FCT or FHPP (PNU 204.3). During deletion, the switch-on event 3Dh (start-up event) is generated and entered in the diagnostic memory. The counter is not reset.

7.2 Fault detection and elimination

List of diagnostic messages ➔ appendix D.

7.2.1 Acknowledge error

Acknowledgeable errors

For acknowledgeable errors, the ready status can be restored after elimination of the cause of error (Reset), e.g. load voltage error. Some errors do not require elimination of the cause of error and can be acknowledged immediately, e.g. following error.

Acknowledge error via ...	
FCT	Button ‹Brake›
Web server	Button ‹Reset Error›
FHPP	Rising edge at the control byte CCON.RESET

Tab. 7.7 Acknowledge error

Non-acknowledgeable errors

For non-acknowledgeable errors, the ready status can be restored, after elimination of the cause of error, only through a restart of the motor controller:

- Restart via FCT or FHPP (software reset)
- Alternatively: Switch logic voltage off and back on.

Restart via ...	
FCT	Command [Component] [Online] [Restart Controller]
FHPP	Writing PNU 127:3 with the value 16

Tab. 7.8 Restart of the motor controller (software reset)

7.2.2 Parameterisation of the diagnostic messages and fault clearance

Term	Significance	PNU[1)]
No.	Number of the diagnostic message in hexadecimal notation.	–[2)]
Classifiable as …	**F**/W/I = fault/warning/information (➔ Tab. 7.1) Indicates which classification is possible for a diagnostic message. The factory setting is printed in bold (here F). If a classification is not possible, this is indicated by dashes. Example: "F/–/–" means that the diagnostic message can only be classified as a fault.	238/246
Diagnostic memory	Indicates whether an entry is made in the diagnostic memory or if this can be parameterised in FCT (always/optional).	238/246
Acknowledgement option	Information on an error's ability to be acknowledged:[2)] – acknowledgeable: acknowledgement via FCT, Webserver or FHPP – not acknowledgeable: restart of the motor controller (software reset); Alternatively: Switch logic voltage off and back on	–[2)]
Error response	For every diagnostic message, provides the parameterisable error responses as Code letters (A to G) (➔ Tab. 7.10). Code letters for the factory response settings are printed in bold.	234/242

1) Permitted values (bit mask) / current values

2) Not available via FHPP

Tab. 7.9 Parameterisation of the diagnostic messages (explanations for the tables of the diagnostic messages)

Code letters for the parameterisable error responses	
A	Free outlet – no braking ramp, turn off output stage
B	Quick-Stop deceleration - quick-stop braking ramp, turn off output stage
C	Job deceleration - braking ramp of the current job, turn off output stage
D	End job – carry out job up to Motion complete; switch off output stage
E	Quick-stop deceleration - quick-stop braking ramp, do **not** turn off output stage
F	Job deceleration - braking ramp of the current job, do **not** turn off output stage
G	End job – continue to carry out job to Motion complete; do **not** switch off output stage

Tab. 7.10 Error responses (code letters)

A Technical appendix

A.1 Increments

A.1.1 Encoder increments [EINC]

The motor controller works with encoder increments [EINC] in the drive control area (e.g. in the path generator).

A.1.2 Interface increments [SINC]

In contrast, so-called interface increments [SINC] are used at all user interfaces and in the area of internal data management. This avoids rounding errors when writing and reading values.

Size of a SINC

Interface increments are at first dimensionless, i.e, they have no defined unit or size. The unit, that is, the size of the interface increment (SINC), is established in the factor group object (powers of ten exponent PNU 600 and unit of measurement PNU 601):

Factor group objects (Factor Group)

Name	PNU	Object	Type	Access
Position powers of ten exponent (Position Notation Index)	600	Var	int8	rw2
Position unit of measurement (Position Dimension Index)	601	Var	uint8	rw2

Tab. A.1 Factor group overview

During parameterisation in FCT, you can use commonly used units for length specifications, such as millimetres or inches. You do not need any interface increments here.

Parameterise the drive completely in FCT and then read the objects of the factor group (powers of ten exponent PNU 600 and unit of measurement PNU 601).

Example:

Powers of ten exponent PNU 600 = -7
Unit of measurement PNU 601 = metre (0x01)
Calculation:

– 1 SINC: 1 * 10^{-7} m = 0.1 µm
– 10,000 SINC: 10,000 * 10^{-7} m = 1 mm

A.2 Conversion factors

The reference of the interface increments [SINC] to the encoder increments [EINC] is created via the following mechanical data of the axis parameter and used for determining the internal conversion factor.

Name	PNU	Object	Type	Access
Reversal of direction (Polarity)	1000	Var	int8	rw2
Encoder resolution (Encoder Resolution)	1001	Array	uint32	ro
Gear ratio (Gear Ratio)	1002	Array	uint32	rw2
Feed constant (Feed Constant)	1003	Array	uint32	rw2
Axis parameter (Axis Parameter)	1005	Array	uint32	rw2

Tab. A.2 Overview of parameters involved

B Reference paramete

B.1 General FHPP parameter structure

The motor controller includes a parameter set with the following structure.

Group	PNU range	Description	Page
Device data	100 … 199	Device identification and device-specific settings, e.g. version numbers.	85
Diagnostics	200 … 299	Diagnostic events and diagnostic memory. Fault numbers, timestamp, incoming/outgoing event.	86
Process data	300 … 399	Current setpoint and actual values, local digital inputs and outputs, e.g. status data.	87
Record list	400 … 499	A record includes all the setpoint value parameters required for a positioning procedure.	88
Project data	500 … 599	Fundamental project settings, e.g. maximum speed, acceleration, deceleration, project zero point offset. These parameters are the basis for the record list.	90
Factor group	600 … 699	Parameters for unit conversion.	92
Axis data: electric drives 1	1000 … 1099	All axis-specific parameters for electric drives, e.g. gear ratio, feed constant, reference parameters.	93

Tab. B.1 FHPP parameter structure

B.2 Access protection and master control

The user can prevent the drive from being operated simultaneously through the controller and the Festo Configuration Tool (FCT). The CCON.LOCK bit (block FCT access) and SCON.FCT/MMI bit (FCT master control) are used for this.

Prevent operation through FCT: CCON.LOCK
By setting the CCON.LOCK control bit, the controller prevents the FCT from taking over master control. So if the LOCK is set, FCT cannot write parameters or control the drive, execute homing, etc.
The controller is programmed not to issue this release until the user carries out the corresponding action. This generally causes exit from automatic operation. This means that the controller programmer can ensure that the controller always knows when it has control over the drive.
Important: The lock is active if the CCON.LOCK has a logic 1. The user who does not need this type of interlock can always leave it at 0.

Acknowledgment, higher-order control with FCT: SCON.FCT/MMI
This bit informs the controller that the drive is controlled by the FCT and that the controller no longer has any control over the drive. This bit does not need to be evaluated. The controller can react by transferring to stop or manual operation.

B.3 Overview of FHPP parameters

The following tables (Tab. B.2 ... Tab. B.8) show the available components.
The parameters are described in sections B.4.2 ... B.4.28.

General instructions on the parameter names: The names are mostly based on the CANopen device profile CiA 402. Some names may vary from product to product while the function remains the same (e.g. in FCT). Examples: rotational speed and speed, or torque and force.

B.3.1 Device data

Group/name	PNU	Subindex	Type
Version numbers → B.4.2, page 97			
Hardware version of the manufacturer (Manufacturer Hardware Version)	100	1	uint16
Firmware design of the manufacturer (Manufacturer Firmware Version)	101	1 ... 4	uint16
FHPP version (Version FHPP)	102	1	uint16
Required software version (Required Software Version)	104	1	uint16
Identification → B.4.3, page 98			
Type of controller (Controller Type)	115	1 ... 5	uint8
Manufacturer's device name (Manufacturer Device Name)	120	1 ... 30	char
User's device name (User Device Name)	121	1 ... 30	char
Name of the drive manufacturer (Drive Manufacturer Name)	122	1 ... 30	char
HTTP address of the manufacturer (HTTP Drive Catalog Address)	123	1 ... 30	char
Festo order number (Festo Order Number)	124	1 ... 30	char
MMI parameters → B.4.4, page 100			
Master control (Controllogic)	125	1	uint8
Controller data memory (Data Memory Control)	127	1 ... 4	uint8
Controller enable signals (Control Enable Signals)	128	1	uint8

Tab. B.2 Device data

B.3.2 Diagnostics

Group/name	PNU	Subindex	Type
Diagnostic parameters → B.4.5, page 102			
Diagnostic event (Diagnostics Event)	200	1 ... 200	uint8
Diagnostic number (Diagnostics Number)	201	1 ... 200	uint16
Time stamp (Time Stamp)	202	1 ... 200	uint32
Additional information (Additional Information)	203	1 ... 200	uint32
Diagnostic memory parameter (Diagnostics Memory Parameter)	204	3, 4	uint8
Device fault (Device Fault)	205	1	uint16
Current fault messages (Actual Malfunction Messages)	220	1 ... 32	uint32
Current warning messages (Actual Warning Messages)	221	1 ... 32	uint32
Current fault acknowledgeable (Actual Acknowledged Malfunction)	230	1	uint8
Permitted error response 1 (Permissible Error Reaction 1)	234	1 ... 255	uint16
Permitted fault handling 1 (Permissible Malfunction Handling 1)	238	1 ... 255	uint16
Error reaction 1 (Error Reaction 1)	242	1 ... 255	uint16
Fault handling 1 (Malfunction Handling 1)	246	1 ... 255	uint16
Safety status (Safety State)	280	1	uint8

Tab. B.3 Diagnostics

B.3.3 Process Data

Group/name	PNU	Subindex	Type
General process data ➔ B.4.6, page 110			
Position values (Position Values)	300	1 … 3	int32
Force values (Force Values)	301	1 … 3	int16
Local digital inputs (Local Digital Inputs)	303	1	uint32
Local digital outputs (Local Digital Outputs)	304	1	uint32
Speed values (Velocity Values)	310	1 … 3	int32
Status of comparator outputs (Status Comparator Outputs)	312	1	uint8
FHPP data ➔ B.4.7, page 112			
FHPP status information (FHPP State Information)	320	1, 2	uint32/ int32
FHPP control information (FHPP Control Information)	321	1, 2	uint32/ int32

Tab. B.4 Process data

B.3.4 Record list

Group/name	PNU	Subindex	Type
Record data → B.4.8, page 113			
Record status (Record State)	400	1, 2	uint8
Record control byte 1 (Record Control Byte 1)	401	1 … 64	uint8
Record control byte 2 (Record Control Byte 2)	402	1 … 64	uint8
Setpoint value (Setpoint Value)	404	1 … 64	int32
Speed (Velocity)	406	1 … 64	int32
Acceleration (Acceleration)	407	1 … 64	int32
Deceleration (Deceleration)	408	1 … 64	int32
Jerk acceleration (Jerk Acceleration)	409	1 … 64	uint32
Load (Load)	410	1 … 64	uint32
Record sequencing target (Record Following Position)	416	1 … 64	uint8
Jerk deceleration (Jerk Deceleration)	417	1 … 64	uint32
Torque limitation (Torque Limitation)	418	1 … 64	int16
Record control byte 3 (Record Control Byte 3)	421	1 … 64	uint8
End speed (Final Velocity)	423	1 … 64	int32
Max. deviation (Max. Deviation)	424	1 … 64	int32
MC with record sequencing (MC During Record Continuation)	425	1 … 64	uint8
Start delay (Start Delay)	426	1 … 64	uint32
Stroke limit (Stroke Limit)	427	1 … 64	int32
Factor torque pilot control (Torque feed forward control factor)	428	1 … 64	uint16

Group/name	PNU	Subindex	Type
Record messages ➔ B.4.9, page 122			
Position comparator, min. (Position Comparator, Min.)	430	1 ... 64	int32
Positions comparator, max. (Position Comparator, Max.)	431	1 ... 64	int32
Position comparator damping time (Position Comparator, Window Time)	432	1 ... 64	uint16
Velocity comparator, min. (Velocity Comparator, Min.)	433	1 ... 64	int32
Velocity comparator, max. (Velocity Comparator, Max.)	434	1 ... 64	int32
Velocity comparator damping time (Velocity Comparator, Window Time)	435	1 ... 64	uint16
Force comparator, min. (Force Comparator, Min.)	436	1 ... 64	int16
Force comparator, max. (Force Comparator, Max.)	437	1 ... 64	int16
Force comparator damping time (Force Comparator, Window Time)	438	1 ... 64	uint16
Time comparator, min. (Time Comparator, Min.)	439	1 ... 64	uint32
Time comparator, max. (Time Comparator, Max.)	440	1 ... 64	uint32
Setpoint value speed (Setpoint Value Velocity)	441	1 ... 64	int32
Setpoint value force (Setpoint Value Force)	442	1 ... 64	int16

Tab. B.5 Record list

B.3.5 Project Data

Group/name	PNU	Subindex	Type
General project data ➔ B.4.10, page 125			
Project zero point (Project Zero Point)	500	1	int32
Software end positions (Software Position Limits)	501	1, 2	int32
Max. permitted speed (Max. Velocity)	502	1	int32
Max. permitted acceleration (Max. Acceleration)	503	1	int32
Force/torque mode ➔ B.4.11, page 126			
Stroke limiter (Stroke Limitation)	510	1	int32
Max. permitted force (Max. Force)	512	1	int32
Teach mode ➔ B.4.12, page 127			
Teach target (Teach Target)	520	1	uint8
FHPP direct mode ➔ B.4.13, page 128			
FHPP setpoint/actual values (FHPP Setpoint and actual values)	523	1 ... 12	uint32
FHPP direct mode settings (FHPP Direct Mode Settings)	524	1	uint8
Jog mode ➔ B.4.14, page 130			
Speed slow – phase 1 (Velocity Slow – Phase 1)	530	1	int32
Speed fast – phase 2 (Velocity Fast – Phase 2)	531	1	int32
Acceleration/deceleration (Acceleration/Deceleration)	532	1	int32
Time period phase 1 (Time Phase 1)	534	1	uint16
Following error message window (Following Error Window)	538	1	int32
Following error delay time (Following Error Timeout)	539	1	uint16

Group/name	PNU	Subindex	Type
Direct mode position ➔ B.4.15, page 131			
Base value speed (Base Value Velocity)	540	1	int32
Acceleration (Acceleration)	541	1	int32
Deceleration (Deceleration)	542	1	int32
Jerk acceleration (Jerk Acceleration)	543	1	uint32
Load (Mass)	544	1	uint32
Jerk deceleration (Jerk Deceleration)	547	1	uint32
End speed (Final Velocity)	548	1	int32
Following error message window (Following Error Window)	549	1	int32
Force direct mode ➔ B.4.16, page 132			
Force message window reached (Force Target Window)	552	1	int16
Basic value force (Base Value Force)	555	1	uint32
Direct mode speed ➔ B.4.17, page 133			
Basic value acceleration (Base Value Acceleration)	560	1	int32
Speed message window reached (Velocity Target Window)	561	1	int32
Stroke limiter (Stroke Limitation)	566	1	int32
Message window for deviation (Velocity Difference Error Window)	568	1	int32

Group/name	PNU	Subindex	Type
Direct mode, general ➔ B.4.18, page 134			
Torque limitation (Torque Limitation)	581	1	int16
Start delay (Start Delay)	582	1	uint32
Start condition (Start Condition)	583	1	uint8
Position comparator, min. (Position Comparator, Min.)	585	1	int32
Position comparator, max. (Position Comparator, Max.)	586	1	int32
Position comparator damping time (Position Comparator, Window Time)	587	1	uint16
Velocity comparator, min. (Velocity Comparator, Min.)	588	1	int32
Velocity comparator, max. (Velocity Comparator, Max.)	589	1	int32
Velocity comparator damping time (Velocity Comparator, Window Time)	590	1	uint16
Force comparator, min. (Force Comparator, Min.)	591	1	int16
Force comparator, max. (Force Comparator, Max.)	592	1	int16
Force comparator damping time (Force Comparator, Windowime)	593	1	uint16
Time comparator, min. (Time Comparator, Min.)	594	1	uint32
Time comparator, max. (Time Comparator, Max.)	595	1	uint32

Tab. B.6 Project Data

B.3.6 Factor group

Group/name	PNU	Subindex	Type
Factor group ➔ B.4.19, page 137			
Position powers of ten exponent (Position Notation Index)	600	1	int8
Position unit of measurement (Position Dimension Index)	601	1	uint8

Tab. B.7 Factor group

B.3.7 Axis parameters: electric drives 1

Group/name	PNU	Subindex	Type
Mechanical parameters ➔ B.4.20, page 138			
Reversal of direction (Polarity)	1000	1	int8
Encoder resolution (Encoder Resolution)	1001	1, 2	uint32
Gear ratio (Gear Ratio)	1002	1, 2	uint32
Feed constant (Feed Constant)	1003	1, 2	uint32
Axis parameter (Axis Parameter)	1005	2, 3	uint32
Homing parameters ➔ B.4.21, page 140			
Offset axis zero point (Offset Axis Zero Point)	1010	1	int32
Homing method (Homing Method)	1011	1	int8
Speeds (Velocities)	1012	1 … 3	int32
Acceleration/deceleration (Acceleration/Deceleration)	1013	1	int32
Max. torque (Max. Torque)	1015	1	int16
Speed limit, stop detection (Block Detection Velocity Limit)	1016	1	int32
Damping time, stop detection (Block Detection Window Time)	1017	1	uint16

Controller parameter → B.4.22, page 142			
Target message window reached (Position Target Window)	1022	1	int32
Damping time for target reached (Position Window Time)	1023	1	uint16
Position control parameter (Position Control Parameter Set)	1024	1 … 7	uint32
I^2t parameters (I^2t Parameter)	1025	1, 2	uint32
I^2t limit values (I^2t Limits)	1026	1, 2	uint16
Current I^2t value (Actual I^2t Value)	1027	1	uint16
Quick stop deceleration (Quick Stop Deceleration)	1029	1	int32
Electronic rating plate → B.4.23, page 145			
Motor type (Motor Type)	1030	1	uint16
Max. current (Max. Current)	1034	1	int32
Motor nominal current (Motor Rated Current)	1035	1	int32
Motor nominal torque (Motor Rated Torque)	1036	1	int32
Standstill monitoring → B.4.24, page 146			
Setpoint position (Setpoint Position)	1040	1	int32
Current position (Position Actual Value)	1041	1	int32
Standstill message window (Standstill Position Window)	1042	1	int32
Standstill delay time (Standstill Window Timeout)	1043	1	uint16
Following error monitoring → B.4.25, page 147			
Following error delay time (Following Error Timeout)	1045	1	uint16
Motor data → B.4.26, page 147			
Current motor current (Actual Current)	1059	1	int32

Temperature data ➔ B.4.27, page 147			
Current temperature CPU (Actual Temperature CPU)	1063	1	int8
Min./max. temperature CPU (Min./Max. Temperature CPU)	1065	1, 2	int8
Current temperature output stage (Actual Temperature Output Stage)	1066	1	int8
Min./max. temperature output stage (Min./Max. Temperature Output Stage)	1068	1, 2	int8
General drive data ➔ B.4.28, page 148			
Tool load/basic mass load (Tool Load/Ground Mass)	1071	1	uint32
Current intermediate circuit voltage (Actual Intermediate Circuit Voltage)	1073	1	uint32
Current control section voltage (Actual Control Section Voltage)	1074	1	uint32
Current string currents (Actual Phase Current)	1075	1 ... 3	int32
Torque pilot control (Torque Feed Forward Control)	1080	1	uint16

Tab. B.8 Axis parameters: electric drives 1

B.4 Descriptions of FHPP parameters

B.4.1 Representation of the parameter entries

[1] **PNU 1001**	[2] **Encoder resolution (Encoder Resolution)**			
[3] Subindex 1, 2	[4] Class: array	[5] Data type: uint32	[6] FW ...	[7] Access: ro
[8] The encoder resolution is the ratio ...				

[9] Subindex 1	[10] Encoder increments (Encoder Increments)
[11] Dependent on the encoder used, default: 0x000007D0 (2000)	

[9] Subindex 2	[10] Motor revolutions (Motor Revolutions)
[11] Fixed: 0x00000001 (1)	

[1] Parameter number (PNU)
[2] Parameter name
[3] List of the subindices of the parameter (1: no subindex, simple variable)
[4] Class (Class):
- Var: contains only one value
- Array: contains several values
- Struct: summary of several variables

[5] Data type (Data type):
Values without sign (8, 16, 32 bit)
- uint8: 0 ... 255
- uint16: 0 ... 65,535
- uint32: 0 ... 4,294,967,295

Values with sign (8, 16, 32 bit)
- int8: –128 ... 127
- int16: –32,768 ... 32,767
- int32: –2,147,483,648 ... 2,147,483,647

Character (8 bit)
- char: 0 ... 255 (ASCII)

[6] Valid from firmware version (... = all)
[7] Access (Read/write permission):
- ro: Read only
- wo: Write only
- rw1: Read and write with energised output stage
- rw2: Read and write only with switched-off output stage.

[8] Description of the parameter
[9] Subindex number
[10] Name of the subindex
[11] Description of the subindex

Fig. B.1 Representation of the parameter entries

B.4.2 Device data – version numbers

PNU 100	Hardware version of the manufacturer (Manufacturer Hardware Version)			
Subindex 1	Class: Array	Data type: uint16	FW ...	Access: ro
Reading the hardware version. Coding of the hardware version number contains the revision numbers (creation date) of the plugged-in printed circuit boards.				
Creation date format (2nd byte/1st byte)				
Byte	Significance			
1 (LSB)	Year			
2 (MSB)	Month			

Tab. B.9 PNU 100

PNU 101	Firmware version of the manufacturer (Manufacturer Firmware Version)			
Subindex 1 ... 4	Class: Array	Data type: uint16	FW ...	Access: ro
Reading of the firmware version. Coding of the firmware version number of the device consists of the 4 numerals (e.g. “1.2.3.4”) of the subindexes.				
Subindex 1	Main version number (Major Version Number)			
1st numeral of the firmware version				
Subindex 2	Secondary version number (Minor Version Number)			
2nd numeral of the firmware version				
Subindex 3	Revision number (Revision Number)			
3rd numeral of the firmware version				
Subindex 4	Build number (Build Number)			
4th numeral of the firmware version				

Tab. B.10 PNU 101

PNU 102	FHPP version (Version FHPP)			
Subindex 1	Class: Var	Data type: uint16	FW ...	Access: ro
Reading of the FHPP version. The FHPP version number of the device consists of 4 numerals (e.g. “xxyy”).				
Format (16 bit, BCD)				
Numerals	Significance			
xx	Main version number			
yy	Secondary version number			

Tab. B.11 PNU 102

PNU 104	Required software version (Required Software Version)			
Subindex 1	Class: Var	Data type: uint16	FW ...	Access: ro
Reading of the FCT version, which is required for operation of the firmware. The min. version number of the Festo Configuration Tool (FCT) consists of 4 numerals (e.g. “xxyy”).				
Format (16 bit, BCD)				
Numerals	Significance			
xx	Main version number			
yy	Secondary version number			

Tab. B.12 PNU 104

B.4.3 Device data – identification

PNU 115	Controller type (Controller Type)			
Subindex 1 ... 5	Class: Array	Data type: uint8	FW ...	Access: ro
Reading of the configuration of the motor controller.				
Subindex 1	Motor technology (Motor Technology)			
Technology of the motor				
Value	Significance			
0x02 (2)	Stepper motor (-ST)			
Subindex 2	Nominal current class (Nominal Current Class)			
Nominal current of the motor controller				
Value	Significance			
0x02 (2)	5 A (-C5)			
Subindex 3	Voltage class (Voltage Class)			
Voltage class of the motor controller				
Value	Significance			
0x01 (1)	24 V (-1)			
Subindex 4	Fieldbus interface (Field Bus Interface)			
Bus interface of the motor controller				
Value	Significance			
0x09 (9)	IO-Link			
Subindex 5	Design of digital inputs/outputs (Digital In/Outputs)			
Value	Significance			
0x01 (1)	PNP (-P)			

Tab. B.13 PNU 115

PNU 120	Manufacturer's device name (Manufacturer Device Name)			
Subindex 1 … 30	Class: Array	Data type: char	FW …	Access: ro
Reading of the manufacturer's designation of the drive (ASCII, 7-bit). Example: CMMO-ST-C5-1-LKP. Unused characters are filled with zero (00_h='\0').				

Tab. B.14 PNU 120

PNU 121	User's device name (User Device Name)			
Subindex 1 … 30	Class: Array	Data type: char	FW …	Access: rw1
Reading or writing of the user designation of the drive (ASCII, 7-bit). Unused characters are filled with zero (00_h='\0').				

Tab. B.15 PNU 121

PNU 122	Name of the drive manufacturer (Drive Manufacturer Name)			
Subindex 1 … 30	Class: Array	Data type: char	FW …	Access: ro
Reading of the drive manufacturer's name (ASCII, 7-bit). Fixed: "Festo AG & Co. KG" Unused characters are filled with zero (00_h='\0').				

Tab. B.16 PNU 122

PNU 123	HTTP address of manufacturer (HTTP Drive Catalog Address)			
Subindex 1 … 30	Class: Array	Data type: char	FW …	Access: ro
Reading of the manufacturer's Internet address (ASCII, 7-bit). Fixed: "http://www.festo.com" Unused characters are filled with zero (00_h='\0').				

Tab. B.17 PNU 123

PNU 124	Festo order number (Festo Order Number)			
Subindex 1 … 30	Class: Array	Data type: char	FW …	Access: ro
Reading of the Festo order number/order code (ASCII, 7-bit). With this specification, the user can order an identical device. Unused characters are filled with zero (00h='\0').				

Tab. B.18 PNU 124

B.4.4 Device data – MMI parameters

PNU 125	Master control (Control logic)			
Subindex 1	Class: Var	Data type: uint8	FW ...	Access: rw1

Reading or parameterisation of the master control via the drive. The control interface that currently has master control can enable the drive and start or stop it (control).

Control interfaces:

– Festo Configuration Tool (FCT): Ethernet
– Fieldbus: IO-Link, I-Port or Modbus

In addition to the parameterised control interface, the following conditions must be fulfilled:

– STO channels (STO1/STO2) [X3.2/3] = 24 V
– Controller enable corresponding to the parameterised enable logic (only fieldbus or fieldbus and digital input) ➔ PNU 128

The controller can reserve master control exclusively with CCON.LOCK = 1.

Reading		
Value	Significance	SCON.FCT/MMI
0x00 (0)	Master control with Festo Configuration Tool (FCT) or web server	1
0x01 (1)	Fieldbus has master control Presetting after every Power ON (switch on “control section” power supply) or restart controller (FCT).	0

Writing		
Value	Significance	SCON.FCT/MMI
0x01 (1)	Fieldbus has master control ➔ Master control cannot be withdrawn from FCT ➔ error 17. ➔ Master control is withdrawn from the web server.	0

Tab. B.19 PNU 125

PNU 127	Data memory control (Data Memory Control)			
Subindex 1 … 4	Class: Struct	Data type: uint8	FW …	Access: rw2
Reading or writing of the commands for permanent data storage (EEPROM). Reading returns the fixed value that has to be written to trigger the desired function.				
Subindex 1	Delete EEPROM (Delete EEPROM)			
When the object is written and after Power OFF (switch off "control section" power supply) or controller restart (FCT), the data in EEPROM are deleted.				
Value	Significance			
0x10 (16)	Data in the EEPROM are deleted and the factory settings loaded.			
Note All user-specific settings will be lost on deletion. The factory settings are loaded in the boot process (after Power ON or controller restart (FCT)). • After deletion, always carry out an initial commissioning.				
Subindex 2	Store data (Save Data)			
By writing the object, the data in EEPROM will be overwritten with the current user-specific settings.				
Value	Significance			
0x01 (1)	User-specific data are stored in the EEPROM.			
Subindex 3	Reset device (Reset Device)			
By writing the object, the data are read from the EEPROM and taken over as the current settings (EEPROM is not cleared; it is in the same status as after Power OFF/ON of the"control section" power supply).				
Value	Significance			
0x10 (16)	Reset device (restart of the firmware without changing the data)			
Subindex 4	Load parameter file (Load Parameter Data)			
Through writing of the object, parameter values are loaded from the parameter file (permanent data memory of the motor controller).				
Value	Significance			
0x10 (16)	Load parameter values from parameter file			

Tab. B.20 PNU 127

PNU 128	Controller enable signals (Controller Enable Signals)			
Subindex 1	Class: Var	Data type: uint8	FW ...	Access: rw2

Reading of writing of the signals required for controller enable. The signals are AND-linked, i.e. all signals must be active so that the controller switches on the output stage.

Value	Required enable signals
0	Communication Control Enable
1	Digital input + Communication Control Enable

Communication Control Enable: e.g. controller enable via fieldbus with CCON.ENABLE or FCT-Freigabe

Tab. B.21 PNU 128

B.4.5 Diagnostic parameters

For a description of how the diagnostic memory functions ➔ section 7.1.3.

PNU 200	Diagnostic event (Diagnostics Event)			
Subindex 1 ... 200	Class: Array	Data type: uint8	FW ...	Access: ro

Reading of the type of diagnostic events in the diagnostic memory.

Value	Significance
0x00 (0)	No malfunction (or fault message deleted)
0x01 (1)	Incoming fault
0x04 (4)	Overflow of the timestamp (reserved)
0x05 (5)	Warning
0x07 (7)	Switching on
0x09 (9)	Information

Subindex 1	Event 1 (Event 1)
Type of latest/current diagnostic message	
Subindex 2	**Event 2 (Event 2)**
Type of second saved diagnostic message	
Subindex 3 ... 200	**Event 3 ... 200 (Event 3 ... 200)**
Type of 3rd ... 200th saved diagnostic message	

Tab. B.22 PNU 200

PNU 201	Diagnostic number (Diagnostics Number)			
Subindex 1 ... 200	Class: Array	Data type: uint16	FW ...	Access: ro
Reading of the detailed specifications in the diagnostic event of the diagnostic numbers. For faults and warnings, this is the exact fault number; for configuration events, it is the function performed, etc. In the case of an invalid diagnostic entry, the value 0xFFFF is returned.				
Subindex 1	Event 1 (Event 1)			
Latest/current diagnostic message				
Subindex 2	Event 2 (Event 2)			
2nd saved diagnostic message				
Subindex 3 ... 200	Event 3 ... 200 (Event 3 ... 200)			
3rd ... 200th saved diagnostic message				

Tab. B.23 PNU 201

PNU 202	Time stamp (Time Stamp)			
Subindex 1 ... 200	Class: Array	Data type: uint32	FW ...	Access: ro
Reading of the time [ms] of the diagnostic events since Power ON. The time stamp has the format hh.mm.ss:nnn (hh = hours, mm = minutes, ss = seconds, nnn = milliseconds). In case of overflow, the value of the timestamp jumps from 0xFFFFFFFF to 0 and a new switch-on event (error message 0x3d) is written in the diagnostic memory.				
Subindex 1	Event 1 (Event 1)			
Time of the latest/current diagnostic message				
Subindex 2	Event 2 (Event 2)			
Time of the 2nd saved diagnostic message				
Subindex 3 ... 200	Event 3 ... 200 (Event 3 ... 200)			
Time of the 3rd ... 200th saved diagnostic message				

Tab. B.24 PNU 202

PNU 203	**Additional information (Additional Information)**			
Subindex 1 … 200	Class: Array	Data type: uint32	FW …	Access: ro
Reading of the additional information for FCT or service personnel.				
Subindex 1	Event 1 (Event 1)			
Additional information for newest/current diagnostic message				
Subindex 2	Event 2 (Event 2)			
Additional information for the 2nd saved diagnostic message				
Subindex 3 … 200	Event 3 … 200 (Event 3 … 200)			
Additional information for the 3rd … 200th saved diagnostic message				

Tab. B.25 PNU 203

PNU 204	**Diagnostic memory parameter (Diagnostics Memory Parameter)**			
Subindex 3, 4	Class: Struct	Data type: uint8	FW …	Access: ro, wo
Reading or deleting of the diagnostic memory.				
Subindex 3	Clear diagnostic memory (Delete Memory)			Access: wo
Clearing the diagnostic memory.				
Value	Significance			
1	Diagnostic memory is cleared			
Subindex 4	Number of entries (Number of Entries)			Access: ro
Read out the number of valid entries in the diagnostic memory				
Value	Significance			
0 … 200	Number of			

Tab. B.26 PNU 204

PNU 205	**Device fault (Device Fault)**			
Subindex 1	Class: Var	Data type: uint16	FW …	Access: ro
Read the active fault with the highest priority. If no fault is present, 0xFFFF (65535) is returned.				

Tab. B.27 PNU 205

PNU 220	**Current fault messages (Actual Malfunction Messages)**			
Subindex 1 … 32	Class: Array	Data type: uint32	FW …	Access: ro
Reading of all existing faults. While the diagnostic memory depicts the history, here it can be determined which faults are now present. Each diagnostic number becomes a bit number. The parameter values cannot be written. Errors cannot be acknowledged with this PNU. If the bit is set, the respective fault is active.				
Subindex 1	0 entry (0th Entry)			
Diagnostic numbers 0 … 31				
Subindex 2	1st entry (1st Entry)			
Diagnostic numbers 32 … 63				
…				
Subindex 4	31st entry (31th Entry)			
Diagnostic numbers 992 … 1023				

Tab. B.28 PNU 220

PNU 221	**Current warning messages (Actual Warning Messages)**			
Subindex 1 … 32	Class: Array	Data type: uint32	FW …	Access: ro
Read all existing warnings. While the diagnostic memory depicts the history, here it can be determined which warnings are now present. Each diagnostic number becomes a bit number. The parameter values cannot be written. Warnings cannot be deleted via this PNU. If the bit is set, the respective warning is active.				
Subindex 1	0 entry (0th Entry)			
Diagnostic numbers 0 … 31				
Subindex 2	1st entry (1st Entry)			
Diagnostic numbers 32 … 63				
…				
Subindex 32	31st entry (31th Entry)			
Diagnostic numbers 992 … 1023				

Tab. B.29 PNU 221

PNU 230	Current fault acknowledgeable (Actual Acknowledged Malfunction)			
Subindex 1	Class: Var	Data type: uint8	FW ...	Access: ro
Read the acknowledgment type of the currently highest priority fault.				

Value	Significance
0x00 (0)	The fault cannot be acknowledged.
0x01 (1)	The fault is still active; the fault can be cleared only after fault clearance.
0x02 (2)	The fault can be acknowledged immediately.
0xFF (255)	No fault is present at all.

Tab. B.30 PNU 230

PNU 234	Permitted error response 1 (Permissible Error Reaction 1)			
Subindex 1 ... 255	Class: Array	Data type: uint16	FW ...	Access: ro
Reading of the permitted error responses for the faults 0 ....254. The parameter is implemented as a bitfield. A value of 0x0037 means, for example, that the error responses 1, 2, 4, 16 and 32 can be parameterised. For unassigned diagnostic numbers, the value 65535 (0xFFFF) is returned.				

Value	Significance
Output stage OFF:	
0x0001 (1)	A: No deceleration ramp
0x0002 (2)	B: After quick-stop deceleration ramp (EMERGENCY STOP)
0x0004 (4)	C: After deceleration ramp (HALT)
0x0008 (8)	D: End after positioning record
Output stage ON:	
0x0010 (16)	E: After quick-stop deceleration ramp (EMERGENCY STOP)
0x0020 (32)	F: After deceleration ramp (HALT)
0x0040 (64)	G: End after positioning record

Subindex 1	Fault number 0 (Malfunction Number 0)
Error response for the fault number 0.	
Subindex 2	Fault number 1 (Malfunction Number 1)
Error response for the fault number 1.	
Subindex 3 ... 255	Fault number 2 ... 254 (Malfunction Number 2 ... 254)
Error responses for the fault numbers 2 ... 254.	

Tab. B.31 PNU 234

PNU 238	**Permitted fault handling 1 (Permissible Malfunction Handling 1)**			
Subindex 1 … 255	Class: Array	Data type: uint16	FW …	Access: ro
Reading of the permitted fault handlings for the faults 0 ....254. The parameter is implemented as a bit mask. If one of the bits is 1, this means that the corresponding bit in the related configuration parameter PNU 246 can be modified. For unassigned diagnostic numbers, the value 65535 (0xFFFF) is returned.				
Subindex 1	Fault number 0 (Malfunction Number 0)			
Fault handling for the fault number 0.				
Subindex 2	Fault number 1 (Malfunction Number 1)			
Fault handling for the fault number 1.				
Subindex 3 … 255	Fault number 2 … 254 (Malfunction Number 2 … 254)			
Fault handlings for the fault numbers 2 … 254.				

Bit	Value	Significance
0 … 4	–	Reserved
5	0	Fault or warning cannot be parameterised
	1	Error or warning can be parameterised
6	0	Information not parameterisable
	1	Information parameterisable
7	0	Diagnostic memory not parameterisable
	1	Diagnostic memory parameterisable
8 … 15	–	Reserved

Tab. B.32 PNU 238

PNU 242	**Error response 1 (Error Reaction 1)**			
Subindex 1 … 255	Class: Array	Data type: uint16	FW …	Access: rw2
Reading or parameterisation of the current error response for the faults 0 ....254. Definition of the fault response and permitted error response ➔ PNU 234.				
Subindex 1	Fault number 0 (Malfunction Number 0)			
Error response for the fault number 0.				
Subindex 2	Fault number 1 (Malfunction Number 1)			
Error response for the fault number 1.				
Subindex 3 … 255	Fault number 2 … 254 (Malfunction Number 2 … 254)			
Error responses for the fault numbers 2 … 254.				

Tab. B.33 PNU 242

PNU 246	**Fault handling 1 (Malfunction Handling 1)**			
Subindex 1 … 255	Class: Array	Data type: uint16	FW …	Access: rw2
Reading or parameterisation of the current fault handling for the faults 0 ….254. Permitted fault handling ➔ PNU 238.				

Bit	Value	Significance
0 … 4	–	Reserved
5	0	W: Fault is parameterised as a warning
	1	F: Fault is parameterised as an error
6	0	Fault can be parameterised as an error or warning (bit 5)
	1	I: Fault is parameterised as information
7	0	No entry in the diagnostic memory
	1	Save in diagnostic memory
8 … 15	–	Reserved

Subindex 1	Fault number 0 (Malfunction Number 0)
Error response for the fault number 0.	
Subindex 2	Fault number 1 (Malfunction Number 1)
Error response for the fault number 1.	
Subindex 3 … 255	Fault number 2 … 254 (Malfunction Number 2 … 254)
Error responses for the fault numbers 2 … 254.	

Tab. B.34 PNU 246

PNU 280	**Safety status (Safety State)**			
Subindex 1	Class: Var	Data type: uint8	FW ...	Access: ro

Reading of the enable status of the hardware.
The following enable statuses are required for operation:

Bit	Value	Significance
0	0	One or both STO channels = 0 V
	1	Both STO channels = 24 V
1	Controller enable via fieldbus[1)]	
	1	Always = 1
	Controller enable via digital input + fieldbus[1)]	
	0	ENABLE (controller enable) [X1.6] = 0 V
	1	ENABLE (controller enable) [X1.6] = 24 V
2 ... 7	Reserved (= 1)	

Note
Only when all bits = 1 can the status be switched to “Ready”.

1) Parameterisation of the controller enable via ➔ PNU 128 or FCT

Tab. B.35 PNU 280

B.4.6 Process data – general process data

PNU 300	Position values (Position Values)			
Subindex 1 ... 3	Class: Array	Data type: int32	FW ...	Access: ro
Reading of the current position values [SINC] of the position controller.				
Subindex 1	Current position (Actual Position)			
Current actual position of the position controller.				
Subindex 2	Current setpoint position (Actual Setpoint Position)			
Current setpoint position of the position controller.				
Subindex 3	Current following error (Actual Following Error)			
Current setpoint value deviation of the position controller.				

Tab. B.36 PNU 300

PNU 301	Force values (Force Values)			
Subindex 1 ... 3	Class: Array	Data type: int16	FW ...	Access: ro
Reading of the current force values [‰ force basic value, PNU 555] of the force regulator.				
Subindex 1	Current value (Actual Value)			
Current actual value of the force regulator.				
Subindex 2	Current setpoint value (Actual Setpoint Value)			
Current setpoint value of the force regulator.				
Subindex 3	Current deviation (Actual Control Deviation)			
Current setpoint value deviation of the force regulator.				

Tab. B.37 PNU 301

PNU 303	Local digital inputs (Local Digital Inputs)			
Subindex 1	Class: Var	Data type: uint32	FW ...	**Access: ro**
Reading of the actual status of the local digital inputs.				
Bit	Significance			
0 ... 8	Reserved			
9	ENABLE (controller enable) [X1.6]			
10 ... 32	Reserved			

Tab. B.38 PNU 303

PNU 304	**Local digital outputs (Local Digital Outputs)**			
Subindex 1	Class: Var	Data type: uint32	FW ...	**Access: ro**
Reading of the actual status of the local digital outputs.				
Bit	Significance			
0 ... 4	Reserved			
5	DOUT1 (output 1, parameterisable) [X1.4]			
6	DOUT2 (output 2, parameterisable) [X1.3]			
7, 8	Reserved			
9	READY (ready for operation) [X1.5]			
10 ... 31	Reserved			

Tab. B.39 PNU 304

PNU 310	**Speed values (Velocity Values)**			
Subindex 1 ... 3	Class: Array	Data type: int32	FW ...	Access: ro
Reading of the current speed values of the speed regulator.				
Subindex 1	Current speed (Actual Velocity)			
Current actual value of the speed regulator.				
Subindex 2	Current setpoint speed (Actual Nominal Velocity)			
Current setpoint value of the speed regulator				
Subindex 3	Current deviation (Actual Control Deviation)			
Current setpoint value deviation of the speed regulator.				

Tab. B.40 PNU 310

PNU 312	**Status of comparator outputs (Status Comparator Outputs)**			
Subindex 1	Class: Var	Data type: uint8	FW ...	**Access: ro**
Reading of the actual status of the comparators for various variables. If the corresponding bit equals 1, this means that the variable (at least corresponding to the duration of the related damping time) is within the area defined from the min. and max. value.				
Bit	Control mode			
0	Position comparator			
1	Velocity comparator			
2	Force comparator			
3	Time comparator			
4 ... 7	Reserved			

Tab. B.41 PNU 312

B.4.7 Process data – FHPP-data

PNU 320	**FHPP status information (FHPP State Information)**			
Subindex 1, 2	Class: Struct	Data type: uint32/int32	FW ...	Access: ro
Reading of the status data (input data).				
Subindex 1	FHPP status byte 1 ... 4 (FHPP State Byte 1... 4)			Data type: uint32
Status information on byte 1 ... 4 (e.g. SCON, SPOS, ...)				
Subindex 2	FHPP status byte 5 ... 8 (FHPP State Byte 5... 8)			Data type: int32
Status information on byte 5 ... 8 (actual value 2)				

Tab. B.42 PNU 320

PNU 321	**FHPP control information (FHPP Control Information)**			
Subindex 1, 2	Class: Struct	Data type: uint32/int32	FW ...	Access: ro
Reading of the control data (output data).				
Subindex 1	FHPP control byte 1 ... 4 (FHPP Control Byte 1... 4)			Data type: uint32
Control information on byte 1 ... 4 (e.g. CCON, CPOS, ...)				
Subindex 2	FHPP control byte 5 ... 8 (FHPP Control Byte 5... 8)			Data type: int32
Control information on byte 5 ... 8 (setpoint value 2)				

Tab. B.43 PNU 321

B.4.8 Record list – record data

With FHPP, record selection for reading and writing is done via the subindex of the PNUs 401 … 427. The active record for teaching is selected via PNU 400.

PNU	Designation	Data type	Subindex
401	**RCB1 (record control byte 1)**	uint8	1 … 64
402	**RCB2 (record control byte 2)**	uint8	1 … 64
404	**Setpoint value position**	int32	1 … 64
406	**Speed**	int32	1 … 64
407	**Acceleration**	int32	1 … 64
408	**Deceleration**	int32	1 … 64
409	**Jerk acceleration**	uint32	1 … 64
410	**Load**	uint32	1 … 64
416	**Following position**	uint8	1 … 64
417	**Jerk deceleration**	uint32	1 … 64
418	**Torque limitation**	int16	1 … 64
421	**RCB3 (record control byte 3)**	uint8	1 … 64
423	**Final speed**	int32	1 … 64
424	**Maximum Offset**	int32	1 … 64
425	**MC with record sequencing**	uint8	1 … 64
426	**Start Delay**	uint32	1 … 64
427	**Stroke limit**	int32	1 … 64
428	**Factor torque pilot control**	uint16	1 … 64
430	**Position comparator, min.**	int32	1 … 64
431	**Position comparator, max.**	int32	1 … 64
432	**Position comparator, damping time**	uint16	1 … 64
433	**Velocity comparator, min.**	int32	1 … 64
434	**Velocity comparator, max.**	int32	1 … 64
435	**Velocity comparator, damping time**	uint16	1 … 64
436	**Force comparator, min.**	int16	1 … 64
437	**Force comparator, max.**	int16	1 … 64
438	**Force comparator, damping time**	uint16	1 … 64
439	**Time comparator, min.**	uint32	1 … 64
440	**Time comparator, max.**	uint32	1 … 64
441	**Setpoint value speed**	int32	1 … 64
442	**Setpoint value force**	int16	1 … 64

Tab. B.44 Structure of the record list – record data for FHPP

PNU 400	**Record status (Record State)**			
Subindex 1, 2	Class: Struct	Data type: uint8	FW ...	Access: rw1, ro
Reading or parameterisation of the currently selected record.				
Subindex 1	Setpoint record number (Demand Record Number)			Access: rw1
The entry includes the number of the target record in whose parameter the current position is entered as soon as the Teach bit is set ➔ PNU 520				
Subindex 2	Current record number (Actual Record Number)			Access: ro
It is also valid if the drive is not in the record selection mode (Teach!). In record selection mode, this parameter is transmitted in the cyclic I/O data.				

Tab. B.45 PNU 400

<table>
<tr><th colspan="2">PNU 401</th><th colspan="4">Record control byte 1 (Record Control Byte 1)</th></tr>
<tr><td colspan="2">Subindex 1 ... 64</td><td colspan="2">Class: Array</td><td>Data type: uint8</td><td>FW ... | Access: rw1</td></tr>
<tr><td colspan="6">Reading or parameterisation of the record control byte 1 (RCB1).
The record control byte defines the type of a position set (positioning, speed, force) and includes the most important settings.</td></tr>
<tr><td>Designation</td><td>Bit</td><td colspan="2">Value</td><td colspan="2">Significance</td></tr>
<tr><td rowspan="3">ABS</td><td rowspan="3">0</td><td colspan="2">Binary</td><td colspan="2">Selection of the positioning type.
(Considered only in positioning mode (COM1/2 = 00))</td></tr>
<tr><td colspan="2">0</td><td colspan="2">Setpoint value is absolute</td></tr>
<tr><td colspan="2">1</td><td colspan="2">Setpoint value is relative</td></tr>
<tr><td rowspan="5">COM1/2</td><td rowspan="5">1, 2</td><td>Bit 2</td><td>Bit 1</td><td colspan="2">Selection of the control mode.</td></tr>
<tr><td>0</td><td>0</td><td colspan="2">Positioning mode</td></tr>
<tr><td>0</td><td>1</td><td colspan="2">Force/torque mode</td></tr>
<tr><td>1</td><td>0</td><td colspan="2">Rotational speed/speed mode</td></tr>
<tr><td>1</td><td>1</td><td colspan="2">Invalid record</td></tr>
<tr><td>–</td><td>3</td><td colspan="2">–</td><td colspan="2">Reserved</td></tr>
<tr><td rowspan="3">REL</td><td rowspan="3">4</td><td colspan="2">Binary</td><td colspan="2">Selection of the point of reference for the setpoint value.
(Considered only in positioning mode (COM1/2 = 00))</td></tr>
<tr><td colspan="2">0</td><td colspan="2">Setpoint value is relative to last setpoint value/target</td></tr>
<tr><td colspan="2">1</td><td colspan="2">Setpoint value is relative to the last actual value/actual position</td></tr>
<tr><td rowspan="3">XLIM</td><td rowspan="3">5</td><td colspan="2">Binary</td><td colspan="2">Activation of stroke monitoring.
(Considered only for force/torque mode or rotational speed/speed mode (COM1/2 = 01 or 10))</td></tr>
<tr><td colspan="2">0</td><td colspan="2">Stroke monitoring active</td></tr>
<tr><td colspan="2">1</td><td colspan="2">Stroke monitoring not active</td></tr>
<tr><td>FAST</td><td>6</td><td colspan="2">–</td><td colspan="2">Not supported/reserved</td></tr>
<tr><td>–</td><td>7</td><td colspan="2">–</td><td colspan="2">Reserved</td></tr>
<tr><td colspan="2">Subindex 1 ... 64</td><td colspan="4">Record 1 ... 64 (Record 1 ... 64)</td></tr>
<tr><td colspan="6">Record control byte 1 of the record 1 ... 64.</td></tr>
</table>

Tab. B.46 PNU 401

PNU 402	Record control byte 2 (Record Control Byte 2)			
Subindex 1 ... 64	Class: Array	Data type: uint8	FW ...	Access: rw1
Reading or parameterisation of the record control byte 2 (RCB2). The record control byte includes conditional record sequencing.				

Bit	Value	Significance
0 ... 6	Decimal	Step enabling condition for automatic record chaining.
	0	No record sequencing
	1	MC (Motion Complete)
	20	Position Comparator
	21	Velocity Comparator
	22	Force Comparator
	23	Time Comparator
7	Reserved (= 0!)	

Subindex 1 ... 64	Record 1 ... 64 (Record 1 ... 64)
Record control byte 2 of the record 1 ... 64.	

Tab. B.47 PNU 402

PNU 404	Setpoint value (Setpoint Value)			
Subindex 1 ... 64	Class: Array	Data type: int32	FW ...	Access: rw1
Reading or writing of the target position.				
Subindex 1 ... 64	Record 1 ... 64 (Record 1 ... 64)			
Setpoint value of the record 1 ... 64.				

Tab. B.48 PNU 404

PNU 406	Speed (Velocity)			
Subindex 1 ... 64	Class: Array	Data type: int32	FW ...	Access: rw1
Reading or parameterisation of the max. speed [SINC/s]. The speed is always specified positively. When travelling in a negative direction, the value is automatically negated. – Position record: max. speed – Speed record: without function – Force record: max. speed				
Subindex 1 ... 64	Record 1 ... 64 (Record 1 ... 64)			
Max. speed of the record 1 ... 64.				

Tab. B.49 PNU 406

PNU 407	Acceleration (Acceleration)			
Subindex 1 … 64	Class: Array	Data type: int32	FW …	Access: rw1
Reading or parameterisation of the max. acceleration [SINC/s^2].				
Subindex 1 … 64	Record 1 … 64 (Record 1 … 64)			
Max. acceleration of the record 1 … 64.				

Tab. B.50 PNU 407

PNU 408	Deceleration (Deceleration)			
Subindex 1 … 64	Class: Array	Data type: int32	FW …	Access: rw1
Reading or parameterisation of the max. deceleration [SINC/s^2].				
Subindex 1 … 64	Record 1 … 64 (Record 1 … 64)			
Max. deceleration of the record 1 … 64.				

Tab. B.51 PNU 408

PNU 409	Jerk acceleration (Jerk Acceleration)			
Subindex 1 … 64	Class: Array	Data type: uint32	FW …	Access: rw1
Reading or parameterisation of the max. jerk [(SINC/s^3)/10] during acceleration. The value 0 is interpreted as max. jerk.				
Subindex 1 … 64	Record 1 … 64 (Record 1 … 64)			
Max. jerk acceleration value of the record 1 … 64.				

Tab. B.52 PNU 409

PNU 410	Load (Mass)			
Subindex 1 … 64	Class: Array	Data type: uint32	FW …	Access: rw1
Reading or parameterisation of the load that is moved in addition to the basic load during positioning. – Linear axis: [g] – Rotative axis: [kgm^2 * 10^{-7}]				
Subindex 1 … 64	Record 1 … 64 (Record 1 … 64)			
Load of the record 1 … 64.				

Tab. B.53 PNU 410

PNU 416	Record sequencing target (Record Following Position)			
Subindex 1 … 64	Class: Array	Data type: uint8	FW …	Access: rw1
Reading or writing of the record number that is jumped to if the step enabling condition is met.				
Subindex 1 … 64	Record 1 … 64 (Record 1 … 64)			
Record sequencing target of the record 1 … 64.				

Tab. B.54 PNU 416

PNU 417	Jerk deceleration (Jerk Deceleration)			
Subindex 1 … 64	Class: Array	Data type: uint32	FW …	Access: rw1
Reading or parameterisation of the max. jerk [(SINC/s^3)/10] during deceleration. The value 0 is interpreted as max. jerk. Force record: no function				
Subindex 1 … 64	Record 1 … 64 (Record 1 … 64)			
Max. jerk deceleration value of the record 1 … 64.				

Tab. B.55 PNU 417

PNU 418	Torque limitation (Torque Limitation)			
Subindex 1 … 64	Class: Array	Data type: int16	FW …	Access: rw1
Reading or parameterisation of the max. force [‰ force basic value, PNU 555]. – 0 ‰ = no motor current (0 A) – 1000 ‰ = force basic value, PNU 555				
Subindex 1 … 64	Record 1 … 64 (Record 1 … 64)			
Max. force of the record 1 … 64.				

Tab. B.56 PNU 418

PNU 421	Record control byte 3 (Record Control Byte 3)			
Subindex 1 ... 64	Class: Array	Data type: uint8	FW ...	Access: rw1
Reading or parameterisation of the record control byte 3 (RCB3). The record control byte controls the specific behaviour of the record (start condition for start commands during active jobs).				

Bit	Value		Significance
0, 1	Bit 1	Bit 0	Start command options
	0	0	Ignore: Ignore start command
	0	1	Interrupt: Switch immediately to the new job
	1	0	Wait: Start of the new job after Motion Complete (attachment of the record to the ongoing job)
	1	1	Reserved
2 ... 7	–		Reserved

Subindex 1 ... 64	Record 1 ... 64 (Record 1 ... 64)
Record control byte 3 of the record 1 ... 64.	

Tab. B.57 PNU 421

PNU 423	End speed (Final Velocity)			
Subindex 1 ... 64	Class: Array	Data type: int32	FW ...	Access: rw1
Reading or parameterisation of the speed [SINC/s] at the end of the record. – Position record: end speed – Speed record: setpoint speed – Force record: no function				
Subindex 1 ... 64	Record 1 ... 64 (Record 1 ... 64)			
End speed of the record 1 ... 64.				

Tab. B.58 PNU 423

PNU 424	Max. deviation (Max. Deviation)			
Subindex 1 ... 64	Class: Array	Data type: int32	FW ...	Access: rw1
Reading or parameterisation of the max. deviation. – Position record: max. following error [SINC] – Speed record: max deviation from the setpoint speed [SINC/s] – Force record: no function				
Subindex 1 ... 64	Record 1 ... 64 (Record 1 ... 64)			
End speed of the record 1 ... 64.				

Tab. B.59 PNU 424

PNU 425	MC with record sequencing (MC During Record Continuation)			
Subindex 1 ... 64	Class: Array	Data type: uint8	FW ...	Access: rw1
Reading or parameterisation of Motion Complete (MC) with record sequencing.				
Value	Significance			
0	No Motion Complete (MC) is output.			
1	A Motion Complete (MC) is output.			
Subindex 1 ... 64	Record 1 ... 64 (Record 1 ... 64)			
MC with record sequencing of the record 1 ... 64.				

Tab. B.60 PNU 425

PNU 426	Start delay (Start Delay)			
Subindex 1 ... 64	Class: Array	Data type: uint32	FW ...	Access: rw1
Reading or parameterisation of the start delay times [ms]. The time is started with the Start command. After the time has elapsed, the record starts to travel.				
Subindex 1 ... 64	Record 1 ... 64 (Record 1 ... 64)			
Start delay of the record 1 ... 64.				

Tab. B.61 PNU 426

PNU 427	Stroke limit (Stroke Limit)			
Subindex 1 ... 64	Class: Array	Data type: int32	FW ...	Access: rw1
Reading or parameterisation of the max. displacement (stroke) [SINC] relative to the start position which is travelled in the speed and force/torque mode. When the stroke limit is reached, the drive is braked via the Quick Stop ramp and stops in a position-controlled manner. Monitoring can be deactivated by setting the bit RCB1.B5 (PNU 401).				
Subindex 1 ... 64	Record 1 ... 64 (Record 1 ... 64)			
Stroke limit of the record 1 ... 64.				

Tab. B.62 PNU 427

PNU 428	Factor torque pilot control (Torque Feed Forward Control Factor)			
Subindex 1 … 64	Class: Array	Data type: uint16	FW …	Access: rw1
Reading or parameterisation of the torque pilot control proportion in the record mode [‰]. – 0 = inactive – 1000 = fully active The torque pilot control is added to the current controller setpoint value. The value is calculated from the acceleration. Compare also ➔ PNU 1080.				
Subindex 1 … 64	Record 1 … 64 (Record 1 … 64)			
Factor of the record 1 … 64.				

Tab. B.63 PNU 428

B.4.9 Record list – record messages

PNU 430	**Position comparator, min. (Position Comparator, Min.)**			
Subindex 1 … 64	Class: Array	Data type: int32	FW …	Access: rw1
Reading or parameterisation of the lower limit values [SINC] of the position comparator.				
Subindex 1 … 64	Record 1 … 64 (Record 1 … 64)			
Position comparator, min. of the record 1 … 64.				

Tab. B.64 PNU 430

PNU 431	**Position comparator, max. (Position Comparator, Max.)**			
Subindex 1 … 64	Class: Array	Data type: int32	FW …	Access: rw1
Reading or parameterisation of the upper limit values [SINC] of the position comparator.				
Subindex 1 … 64	Record 1 … 64 (Record 1 … 64)			
Position comparator, max. of the record 1 … 64.				

Tab. B.65 PNU 431

PNU 432	**Position comparator damping time (Position Comparator, Window Time)**			
Subindex 1 … 64	Class: Array	Data type: uint16	FW …	Access: rw1
Reading or parameterisation of the damping times [ms] of the position comparator.				
Subindex 1 … 64	Record 1 … 64 (Record 1 … 64)			
Position comparator damping time of the record 1 … 64.				

Tab. B.66 PNU 432

PNU 433	**Velocity comparator, min. (Velocity Comparator, Min.)**			
Subindex 1 … 64	Class: Array	Data type: int32	FW …	Access: rw1
Reading or parameterisation of the lower limit values [SINC/s] of the velocity comparator.				
Subindex 1 … 64	Record 1 … 64 (Record 1 … 64)			
Velocity comparator, min. of the record 1 … 64.				

Tab. B.67 PNU 433

PNU 434	Velocity comparator, max. (Velocity Comparator, Max.)			
Subindex 1 … 64	Class: Array	Data type: int32	FW …	Access: rw1
Reading or parameterisation of the upper limit values [SINC/s] of the velocity comparator.				
Subindex 1 … 64	Record 1 … 64 (Record 1 … 64)			
Velocity comparator, max. of the record 1 … 64.				

Tab. B.68 PNU 434

PNU 435	Velocity comparator damping time (Velocity Comparator, Window Time)			
Subindex 1 … 64	Class: Array	Data type: uint16	FW …	Access: rw1
Reading or parameterisation of the damping times [ms] of the velocity comparator.				
Subindex 1 … 64	Record 1 … 64 (Record 1 … 64)			
Velocity comparator, damping time of the record 1 … 64.				

Tab. B.69 PNU 435

PNU 436	Force comparator, min. (Force Comparator, Min.)			
Subindex 1 … 64	Class: Array	Data type: int16	FW …	Access: rw1
Reading or parameterisation of the lower limit values [‰ of the force basic value, PNU 555] of the force comparator.				
Subindex 1 … 64	Record 1 … 64 (Record 1 … 64)			
Force comparator, min. of the record 1 … 64.				

Tab. B.70 PNU 436

PNU 437	Force comparator, max. (Force Comparator, Max.)			
Subindex 1 … 64	Class: Array	Data type: int16	FW …	Access: rw1
Reading or parameterisation of the upper limit values [‰ of the force basic value, PNU 555] of the force comparator.				
Subindex 1 … 64	Record 1 … 64 (Record 1 … 64)			
Force comparator, max. of the record 1 … 64.				

Tab. B.71 PNU 437

PNU 438	Force comparator damping time (Force Comparator, Window Time)			
Subindex 1 … 64	Class: Array	Data type: uint16	FW …	Access: rw1
Reading or parameterisation of the damping times [ms] of the force comparator.				
Subindex 1 … 64	Record 1 … 64 (Record 1 … 64)			
Force comparator damping time of the record 1 … 64.				

Tab. B.72 PNU 438

PNU 439	Time comparator, min. (Time Comparator, Min.)			
Subindex 1 … 64	Class: Array	Data type: uint32	FW …	Access: rw1
Reading or parameterisation of the lower limit values [ms] of the time comparator.				
Subindex 1 … 64	Record 1 … 64 (Record 1 … 64)			
Time comparator, min. of the record 1 … 64.				

Tab. B.73 PNU 439

PNU 440	Time comparator, max. (Time Comparator, Max.)			
Subindex 1 … 64	Class: Array	Data type: uint32	FW …	Access: rw1
Reading or parameterisation of the upper limit values [ms] of the time comparator.				
Subindex 1 … 64	Record 1 … 64 (Record 1 … 64)			
Time comparator, max. of the record 1 … 64.				

Tab. B.74 PNU 440

PNU 441	Setpoint value speed (Setpoint Value Velocity)			
Subindex 1 … 64	Class: Array	Data type: int32	FW …	Access: rw1
Reading or parameterisation of the end speed of a speed record. The sign of the value determines the direction in which the speed is to be built up.				
Subindex 1 … 64	Record 1 … 64 (Record 1 … 64)			
End speed of the record 1 … 64.				

Tab. B.75 PNU 441

PNU 442	Setpoint value force (Setpoint Value Force)			
Subindex 1 … 64	Class: Array	Data type: int16	FW …	Access: rw1
Reading or parameterisation of the target force of a force record [‰ of the force basic value, PNU 555]. The sign of the value determines the direction in which the force is to be built up.				
Subindex 1 … 64	Record 1 … 64 (Record 1 … 64)			
Target force of the record 1 … 64.				

Tab. B.76 PNU 442

B.4.10 Project data – general project data

PNU 500	Project zero point (Project Zero Point)			
Subindex 1	Class: Var	Data type: int32	FW …	Access: rw1
Reading or parameterisation of the point of reference for position values in the application ➔ PNU 404. Offset of the axis zero point [SINC] to the project zero point.				

Tab. B.77 PNU 500

PNU 501	Software end positions (Software Position Limits)			
Subindex 1, 2	Class: Array	Data type: int32	FW …	Access: rw2
Reading or parameterisation of the software end positions [SINC]. A setpoint specification (position) outside the software end positions is not permissible and will result in an error. The offset to the axis zero point is entered. The software end positions are deactivated if both software end positions have the value = 0.				
Subindex 1	Lower limit value (Lower Limit)			
Lower software end position				
Subindex 2	Upper limit value (Upper Limit)			
Upper software end position				

Tab. B.78 PNU 501

PNU 502	Max. permitted speed (Max. Velocity)			
Subindex 1	Class: Var	Data type: int32	FW …	Access: rw2
Reading or parameterisation of the max. permitted speed [SINC/s]. This value limits the speed in all operating modes.				

Tab. B.79 PNU 502

PNU 503	Max. permitted acceleration (Max. Acceleration)			
Subindex 1	Class: Var	Data type: int32	FW ...	Access: rw2
Reading or parameterisation of the max. permitted acceleration [SINC/s²].				

Tab. B.80 PNU 503

B.4.11 Project data – force/torque mode

PNU 510	Stroke limiter (Stroke Limitation)			
Subindex 1	Class: Var	Data type: int32	FW ...	Access: rw1
Reading or parameterisation of the max. permitted displacement (stroke) [SINC] with active force control. With active force control, the actual position relative to the start position must not change by more than the amount specified in this parameter. In this way, you can ensure that the axis will not perform an uncontrolled movement if force control is activated by mistake (e.g. workpiece missing). Monitoring can be deactivated with CDIR.XLIM = 1.				

Tab. B.81 PNU 510

PNU 512	Max. permitted force (Max. Force)			
Subindex 1	Class: Var	Data type: int32	FW ...	Access: rw1
Reading or parameterisation of the max. current (force) [mA], with which the motor may be operated. The value is always positive. Internally, this limits the max. “positive” and “negative” current.				

Tab. B.82 PNU 512

B.4.12 Project data –teach mode

PNU 520	**Teach target (Teach Target)**			
Subindex 1	Class: Var	Data type: uint8	FW ...	Access: rw1
Reading or parameterisation of the Teach memory. With the next teach command, the actual position is written to the selected memory ➔ page 55.				

Value	Significance
0x01 (1)	Setpoint position in position set[1] ➔ PNU 404
0x02 (2)	Axis zero point ➔ PNU 1010
0x03 (3)	Project zero point ➔ PNU 500
0x04 (4)	Lower software end position ➔ PNU 501.1
0x05 (5)	Upper software end position ➔ PNU 501.2
0x06 (6)	Position comparator lower limit[1] ➔ PNU 430
0x07 (7)	Position comparator upper limit[1] ➔ PNU 431

1) Record number in direct mode via PNU 400.1 “Setpoint record number”; in case of record selection via record number, specify in control byte 3

Tab. B.83 PNU 520

B.4.13 Project data – FHPP-direct mode

PNU 523	**FHPP setpoint and actual values (FHPP Setpoint and actual values)**			
Subindex 1 ... 12	Class: Struct	Data type: uint32	FW ...	Access: rw1

Reading or parameterisation of the setpoint and actual values in the cyclical I/O data, dependent on the control mode.

Closed-loop control	Setpoint/ actual value	Subindex	Value	Description
Position	Setpoint value 1	1	0	Speed [% basic value] ➔ PNU 540
			1	Reserved
	Setpoint value 1	2	0	Position [SINC], 32-bit number ➔ appendix A.2
			1	Reserved
	Actual value 1	3	0	Speed [% basic value] ➔ PNU 540
			1	Reserved
	Actual value 2	4	0	Position [SINC], 32-bit number ➔ appendix A.2
			1	Reserved
Force/torque	Setpoint value 1	5	2	Speed [% basic value] ➔ PNU 540
			0, 1	Reserved
	Setpoint value 2	6	0	Setpoint torque [% basic value] ➔ PNU 555
			1	Reserved
	Actual value 1	7	0	Actual speed [SINC/s] ➔ appendix A.2
			1	Torque [% force basic value] ➔ PNU 555
	Actual value 2	8	0	Actual position [SINC] ➔ appendix A.2
			1	Torque [% force basic value] ➔ PNU 555
Speed	Setpoint value 1	9	0	Speed ramp [% basic value] ➔ PNU 560
			1	Reserved
	Setpoint value 2	10	0	Speed [SINC/s] ➔ appendix A.2
			1	Reserved
	Actual value 1	11	0	No function, = 0
			1	Reserved
	Actual value 2	12	0	Reserved
			1	Speed as absolute value [SINC/s]

Tab. B.84 PNU 523

PNU 524	**FHPP direct mode settings (FHPP Direct Mode Settings)**			
Subindex 1	Class: Var	Data type: uint8	FW ...	Access: rw1
Reading or parameterisation of the characterististics for the FHPP direct mode.				

Bit	Value	Significance
0	Binary	Relative positioning type
	0	Setpoint value is relative to the last setpoint/target position
	1	Setpoint value is relative to the current position (default)
1...7	–	Reserved

Tab. B.85 PNU 524

B.4.14 Project data – jog mode

PNU 530	**Speed slow – phase 1 (Velocity Slow – Phase 1)**			
Subindex 1	Class: Var	Data type: int32	FW ...	Access: rw2
Reading or parameterisation of the slow speed [SINC/s] for phase 1.				

Tab. B.86 PNU 530

PNU 531	**Speed fast – phase 2 (Velocity Fast – Phase 2)**			
Subindex 1	Class: Var	Data type: int32	FW ...	Access: rw2
Reading or parameterisation of the max. speed [SINC/s] for phase 2.				

Tab. B.87 PNU 531

PNU 532	**Acceleration/deceleration (Acceleration/Deceleration)**			
Subindex 1	Class: Var	Data type: int32	FW ...	Access: rw2
Reading or parameterisation of the acceleration/deceleration [SINC/s^2] during jogging.				

Tab. B.88 PNU 532

PNU 534	**Time period phase 1 (Time Phase 1)**			
Subindex 1	Class: Var	Data type: uint16	FW ...	Access: rw2
Reading or parameterisation of the time period [ms] for phase 1.				

Tab. B.89 PNU 534

PNU 538	**Following error message window (Following Error Window)**			
Subindex 1	Class: Var	Data type: int32	FW ...	Access: rw2
Reading or parameterisation of the max. permitted following error.				

Tab. B.90 PNU 538

PNU 539	**Following error delay time (Following Error Timeout)**			
Subindex 1	Class: Var	Data type: uint16	FW ...	Access: rw2
Reading or parameterisation of the damping time in [ms] of the following error monitoring.				

Tab. B.91 PNU 539

B.4.15 Project data - direct mode position

PNU 540	**Basic value speed (Base Value Velocity)**			
Subindex 1	Class: Var	Data type: int32	FW ...	Access: rw1
Reading or parameterisation of the basic value for speed [SINC/s]. The master transmits a percent value, which is multiplied by the basic value to calculate the final setpoint speed.				

Tab. B.92 PNU 540

PNU 541	**Acceleration (Acceleration)**			
Subindex 1	Class: Var	Data type: int32	FW ...	Access: rw1
Reading or parameterisation of the acceleration [SINC/s^2].				

Tab. B.93 PNU 541

PNU 542	**Deceleration (Deceleration)**			
Subindex 1	Class: Var	Data type: int32	FW ...	Access: rw1
Reading or parameterisation of the deceleration [SINC/s^2].				

Tab. B.94 PNU 542

PNU 543	**Jerk acceleration (Jerk Acceleration)**			
Subindex 1	Class: Var	Data type: uint32	FW ...	Access: rw1
Reading or parameterisation of the max. jerk [(SINC/s^3)/10] during acceleration. The value 0 is interpreted as max. jerk.				

Tab. B.95 PNU 543

PNU 544	**Load (Load)**			
Subindex 1	Class: Var	Data type: uint32	FW ...	Access: rw1
Reading or parameterisation of the load that is moved in addition to the basic load during positioning. – Linear axis: [g] – Rotative axis: [$kgm^2 * 10^{-7}$]				

Tab. B.96 PNU 544

PNU 547	Jerk deceleration (Jerk Deceleration)			
Subindex 1	Class: Var	Data type: uint32	FW ...	Access: rw1
Reading or parameterisation of the max. jerk [(SINC/s^3)/10] during deceleration. The value 0 is interpreted as max. jerk.				

Tab. B.97 PNU 547

PNU 548	End speed (Final Velocity)			
Subindex 1	Class: Var	Data type: int32	FW ...	Access: rw1
Reading or parameterisation of the speed [SINC/s] at the end of the record – Position record: end speed – Speed record: setpoint speed – Force record: no function				

Tab. B.98 PNU 548

PNU 549	Following error message window (Following Error Window)			
Subindex 1	Class: Var	Data type: int32	FW ...	Access: rw1
Reading or parameterisation of the permitted following error [SINC] in positioning mode				

Tab. B.99 PNU 549

B.4.16 Project data – force direct mode

PNU 552	Force message window reached (Force Target Window)			
Subindex 1	Class: Var	Data type: int16	FW ...	Access: rw2
Reading or parameterisation of the min./max. force of the force comparator in [‰ force basic value]. Force window [‰] for recognition of the setpoint force (max. difference between setpoint force and actual force).				

Tab. B.100 PNU 552

PNU 555	Basic value force (Base Value Force)			
Subindex 1	Class: Var	Data type: uint32	FW ...	Access: rw1
Basic value force in milliamperes [mA]. (The master transmits in the cyclic data a percentage value, which is multiplied by the basic value to calculate the final force.)				

Tab. B.101 PNU 555

B.4.17 Project data – rotational speed direct mode

PNU 560	**Basic value acceleration (Base Value Acceleration)**			
Subindex 1	Class: Var	Data type: int32	FW ...	Access: rw1
Reading or parameterisation of the acceleration basic value [SINC/s^2]. (The master transmits a percent value, which is multiplied by the basic value to calculate the final setpoint acceleration.)				

Tab. B.102 PNU 560

PNU 561	**Speed message window reached (Velocity Target Window)**			
Subindex 1	Class: Var	Data type: int32	FW ...	Access: rw2
Reading or parameterisation of the min./max. speed [SINC/s] of the velocity comparator. “Speed reached” message window for detection of a setpoint speed (max. difference between set-point speed and actual speed)				

Tab. B.103 PNU 561

PNU 566	**Stroke limiter (Stroke Limitation)**			
Subindex 1	Class: Var	Data type: int32	FW ...	Access: rw1
Reading or parameterisation of the max. permitted displacement (stroke) [SINC] with active speed adjustment. With active speed adjustment, the actual position relative to the start position must not change by more than the amount specified in this parameter. In this way, you can ensure that the axis will not perform an uncontrolled movement if speed adjustment is activated by mistake (e.g. workpiece missing). Monitoring can be deactivated by setting the bit CDIR.XLIM.				

Tab. B.104 PNU 566

PNU 568	**Message window for deviation (Velocity Difference Error Window)**			
Subindex 1	Class: Var	Data type: int32	FW ...	Access: rw1
Reading or parameterisation of the permitted displacement [SINC/s] with active speed adjustment				

Tab. B.105 PNU 568

B.4.18 Project data – direct mode general

PNU 581	Torque limitation (Torque Limitation)			
Subindex 1	Class: Var	Data type: int16	FW ...	Access: rw1
Reading or parameterisation of the max. force [‰ force basic value, PNU 555] for the direct mode position and speed. – 0 = no motor current (0 A) – 1000 = force basic value, PNU 555 The value is valid for the positive and negative direction of rotation.				

Tab. B.106 PNU 581

PNU 582	Start delay (Start Delay)			
Subindex 1	Class: Var	Data type: uint32	FW ...	Access: rw1
Reading or parameterisation of the start delay times [ms]. The time is started with the Start command. After the time has elapsed, travel is started.				

Tab. B.107 PNU 582

PNU 583	Start condition (Start Condition)			
Subindex 1	Class: Var	Data type: uint8	FW ...	Access: rw1
Reading or parameterisation of the start condition for start commands during active jobs.				

Value	Significance
0x00 (0)	Ignore: Ignore start command
0x01 (1)	Interrupt: Switch immediately to the new job
0x02 (2)	Wait: Start of the new job after Motion Complete

Tab. B.108 PNU 583

PNU 585	Position comparator, min. (Position Comparator, Min.)			
Subindex 1	Class: Var	Data type: int32	FW ...	Access: rw1
Reading or parameterisation of the lower limit value [SINC] of the position comparator.				

Tab. B.109 PNU 585

PNU 586	Position comparator, max. (Position Comparator, Max.)			
Subindex 1	Class: Var	Data type: int32	FW ...	Access: rw1
Reading or parameterisation of the upper limit value [SINC] of the position comparator.				

Tab. B.110 PNU 586

PNU 587	Position comparator damping time (Position Comparator, Window Time)			
Subindex 1	Class: Var	Data type: uint16	FW ...	Access: rw1
Reading or parameterisation of the damping time [ms] of the position comparator.				

Tab. B.111 PNU 587

PNU 588	Velocity comparator, min. (Velocity Comparator, Min.)			
Subindex 1	Class: Var	Data type: int32	FW ...	Access: rw1
Reading or parameterisation of the lower limit value [SINC/s] of the velocity comparator.				

Tab. B.112 PNU 588

PNU 589	Velocity comparator, max. (Velocity Comparator, Max.)			
Subindex 1	Class: Var	Data type: int32	FW ...	Access: rw1
Reading or parameterisation of the upper limit value [SINC/s] of the velocity comparator.				

Tab. B.113 PNU 589

PNU 590	Velocity comparator damping time (Velocity Comparator, Window Time)			
Subindex 1	Class: Var	Data type: uint16	FW ...	Access: rw1
Reading or parameterisation of the damping time [ms] of the speed comparator.				

Tab. B.114 PNU 590

PNU 591	Force comparator, min. (Force Comparator, Min.)			
Subindex 1	Class: Var	Data type: int16	FW ...	Access: rw1
Reading or parameterisation of the lower limit value [‰ basic value force, PNU 555] of the force comparator.				

Tab. B.115 PNU 591

PNU 592	Force comparator, max. (Force Comparator, Max.)			
Subindex 1	Class: Var	Data type: int16	FW ...	Access: rw1
Reading or parameterisation of the upper limit value [‰ basic value force, PNU 555] of the force comparator.				

Tab. B.116 PNU 592

PNU 593	Force comparator damping time (Force Comparator, Window Time)			
Subindex 1	Class: Var	Data type: uint16	FW ...	Access: rw1
Reading or parameterisation of the damping time [ms] of the force comparator.				

Tab. B.117 PNU 593

PNU 594	Time comparator, min. (Time Comparator, Min.)			
Subindex 1	Class: Var	Data type: uint32	FW ...	Access: rw1
Reading or parameterisation of the lower limit value [ms] of the time comparator.				

Tab. B.118 PNU 594

PNU 595	Time comparator, max. (Time Comparator, Max.)			
Subindex 1	Class: Var	Data type: uint32	FW ...	Access: rw1
Reading or parameterisation of the upper limit value [ms] of the time comparator.				

Tab. B.119 PNU 595

B.4.19 Factor group

PNU 600	**Position powers of ten exponent (Position Notation Index)**			
Subindex 1	Class: Var	Data type: int8	FW ...	Access: rw2
Reading or parameterisation of the powers of 10 exponent with the1 SINC converted to 1 basic unit value. **Example:** Power of 10 exponent = -7 Basic unit (0x01) = metre Calculation: – 1 SINC: 1 * 10^{-7} m = 0.1 µm – 10,000 SINC: 10,000 * 10^{-7}m = 1 mm				

Tab. B.120 PNU 600

PNU 601	**Position unit of measurement (Position Dimension Index)**			
Subindex 1	Class: Var	Data type: uint8	FW ...	Access: rw2
Reading or parameterisation of the system of measurement in relation to the basic unit.				

Value	Significance
0x00 (0)	Undefined/user specific
0x01 (1)	Metre (SI unit)
0x41 (65)	Degree
0xF0 (240)	Inch
0xF6 (246)	Revolutions

Tab. B.121 PNU 601

B.4.20 Axis parameters: electrical Drives 1 – mechanical parameters

PNU 1000	**Reversal of direction (Polarity)**			
Subindex 1	Class: Var	Data type: int8	FW ...	Access: rw2
Reading or parameterisation of the direction of rotation.				
Value	Significance			
0x00	Without reversal of direction (default).			
0x80	With reversal of direction of rotation (all encoder values are negated).			

Tab. B.122 PNU 1000

PNU 1001	**Encoder resolution (Encoder Resolution)**			
Subindex 1, 2	Class: Array	Data type: uint32	FW ...	Access: ro
Reading of the encoder resolution (ratio of encoder increments to motor revolutions). **Calculation of the encoder resolution:**				
$\text{Encoder resolution} = \frac{\text{Encoder increments}}{\text{Motor revolutions}}$				
Subindex 1	Encoder increments (Encoder Increments)			
Dependent on the encoder used, default: 0x000007D0 (2000)				
Subindex 2	Motor revolutions (Motor Revolutions)			
Fixed: 0x00000001 (1)				

Tab. B.123 PNU 1001

PNU 1002	Gear ratio (Gear Ratio)			
Subindex 1, 2	Class: Array	Data type: uint32	FW ...	Access: rw2
Reading or parameterisation of the gear ratio (Ratio of motor revolutions to spindle rotations of the internal gear unit ➔ page 82) **Calculation of the gear ratio:**				
$\text{Gear ratio} = \frac{\text{Motor revolutions}}{\text{Spindle rotations}}$				
The values for motor/spindle rotations are to be selected so that a whole number results.				
Subindex 1	Motor revolutions (Motor Revolutions)			
Numerator of the gear ratio.				
Subindex 2	Spindle rotations (Shaft Revolutions)			
Denominator of the gear ratio.				

Tab. B.124 PNU 1002

PNU 1003	Feed constant (Feed Constant)			
Subindex 1, 2	Class: Array	Data type: uint32	FW ...	Access: rw2
Reading or parameterisation of the feed constant [SINC] (Lead of the drive spindle per revolution ➔ page 82) **Calculation of the feed constant:**				
$\text{Feed constant} = \frac{\text{Feed}}{\text{Spindle rotations}}$				
Subindex 1	Feed (Feed)			
Numerator of the feed constant.				
Subindex 2	Spindle rotations (Shaft Revolutions)			
Denominator of the feed constant.				

Tab. B.125 PNU 1003

PNU 1005	Axis parameter (Axis Parameter)			
Subindex 2, 3	Class: Array	Data type: uint32	FW ...	Access: rw2
Reading or parameterisation of the gear ratio of the axis gear. Applicable exclusively for the external gear unit.				
Subindex 2	Axis gear, numerator (Axis Gear, Numerator)			
Numerator of the gear ratio.				
Subindex 3	Axis gear, denominator (Axis Gear, Denominator)			
Denominator of the gear ratio.				

Tab. B.126 PNU 1005

B.4.21 Axis parameter: electrical drives 1 – homing parameters

PNU 1010	Offset axis zero point (Offset Axis Zero Point)			
Subindex 1	Class: Var	Data type: int32	FW ...	Access: rw1
Reading or parameterisation of the offset axis zero point [SINC]. The offset for the axis zero point (home offset) defines the axis zero point (AZ) as a dimension reference point relative to the physical reference point (REF). The axis zero point (AZ) is the point of reference for the project zero point (PZ) and for the software end positions. All positioning operations refer to the project zero point (PZ) ➔ PNU 500. The axis zero point (AZ) is calculated as follows: AZ = REF + axis zero point offset				

Tab. B.127 PNU 1010

PNU 1011	Homing method (Homing Method)			
Subindex 1	Class: Var	Data type: int8	FW ...	Access: rw1
Reading or parameterisation of the homing method ➔ page 53.				

Tab. B.128 PNU 1011

PNU 1012	Speeds (Velocities)			
Subindex 1 ... 3	Class: Array	Data type: int32	FW ...	Access: rw2
Reading or parameterisation of the speeds [SINC/s] in the reference mode.				
Subindex 1	Search speed (Search Velocity)			
Speed when searching for the reference point (REF) near the reference switch or stop.				
Subindex 2	Travel speed (Drive Velocity)			
Speed when moving to the axis zero point (AZ).				
Subindex 3	Creep speed (Crawling Velocity)			
Creep speed for leaving the reference/limit switch.				

Tab. B.129 PNU 1012

PNU 1013	Acceleration/deceleration (Acceleration/Deceleration)			
Subindex 1	Class: Var	Data type: int32	FW ...	Access: rw2
Reading or parameterisation of the acceleration/deceleration [SINC/s^2] in reference mode.				

Tab. B.130 PNU 1013

PNU 1015	Max. torque (Max. Torque)			
Subindex 1	Class: Var	Data type: int16	FW ...	Access: rw2
Reading or parameterisation of the max. permitted torque [‰ force basic value, PNU 555] (via current limiter) in homing. If the value is reached for a specific time ➔ PNU 1017, the stop is detected as reference point and the drive travels to the axis zero point.				

Tab. B.131 PNU 1015

PNU 1016	Speed limit, stop detection (Block Detection Velocity Limit)			
Subindex 1	Class: Var	Data type: int32	FW ...	Access: rw2
Reading or parameterisation of the speed limit value for stop detection in homing (homing method: stop).				

Tab. B.132 PNU 1016

PNU 1017	Damping time, stop detection (Block Detection Window Time)			
Subindex 1	Class: Var	Data type: uint16	FW ...	Access: rw2
Reading or parameterisation of the damping time [ms] for stop detection in homing (homing method: stop).				

Tab. B.133 PNU 1017

B.4.22 Axis parameter: electrical drives 1 – controller parameters

PNU 1022	Target message window reached (Position Target Window)			
Subindex 1	Class: Var	Data type: int32	FW ...	Access: rw2
Reading or parameterisations of the target [SINC] by which the current position may deviate from the target position while still interpreted to be within the target window. The range of the message window is double that of the parameterised value. The setpoint/target position is in the centre of the window.				

Tab. B.134 PNU 1022

PNU 1023	Damping time for target reached (Position Target Window Time)			
Subindex 1	Class: Var	Data type: uint16	FW ...	Access: rw2
Reading or parameterisation of the damping time [ms]. The damping time begins when the target position window is reached. If the actual position has been in the target position window after the damping time has expired, the SPOS.MC bit is set.				

Tab. B.135 PNU 1023

PNU 1024	**Parameters of the controller (Position Control Parameter Set)**			
Subindex 1 … 7	Class: Struct	Data type: uint32	FW …	Access: rw1
Reading or parameterisation of the closed-loop control parameters.				
Subindex 1	Position gain (Gain Position)			Data type: uint32
Position controller gain.				
Subindex 2	Speed gain (Gain Velocity)			Data type: uint32
Gain of the speed controller.				
Subindex 3	I-proportion of speed (I-Fraction Velocity)			Data type: uint32
I-proportion of the speed controller.				
Subindex 4	Current gain (Gain Current)			Data type: uint32
Gain of the current regulator.				
Subindex 5	I-proportion, current (I-Fraction Current)			Data type: uint32
I-proportion of current regulator.				
Subindex 6	Time constant speed filter (Time Constant Velocity Filter)			Data type: uint32
Time constant for filtering the motor rotational speed.				
Subindex 7	Max. correction speed (Max. Correction Velocity)			Data type: int32
Max. speed contribution for correction of the following error.				

Tab. B.136 PNU 1024

PNU 1025	**I^2t-parameter (I^2t Parameter)**			
Subindex 1, 2	Class: Array	Data type: uint32	FW ...	Access: rw2
Reading or parameterisation of the I^2t integral [ms].				
Subindex 1	Motor time constant, rising I^2t integral (Motor Time Constant, Rising I^2t-Integral)			
Motor time constant of the ascending I^2t integral of the temperature monitoring of the motor.				
Subindex 2	Motor time constant, descending I^2t integral (Motor Time Constant, Falling I^2t-Integral)			
Motor time constant of the descending I^2t integral of the temperature monitoring of the motor. For protection of the motor, the motor current is automatically limited to the motor nominal current ➔ PNU 1035.				

Tab. B.137 PNU 1025

PNU 1026	**I^2t limit values (I^2t Limits)**			
Subindex 1, 2	Class: Struct	Data type: uint16	FW ...	Access: rw2/ro
Reading or parameterisation of the limit/threshold values [‰] of the I^2t-monitoring.				
Subindex 1	I^2t-warning threshold (I^2t Warning Level)			Access: rw2
Warning threshold of the I^2t monitoring of the motor.				
Subindex 2	I^2t-error limit (I^2t Error Limit)			Access: ro
Error limit value of the I^2t monitoring of the motor.				

Tab. B.138 PNU 1026

PNU 1027	**Current I^2t value (Actual I^2t Value)**			
Subindex 1	Class: Var	Data type: uint16	FW ...	Access: ro
Reading of the current fill level [‰] of the I^2t monitoring for the motor.				

Tab. B.139 PNU 1027

PNU 1029	**Quick stop deceleration (Quick Stop Deceleration)**			
Subindex 1	Class: Var	Data type: int32	FW ...	Access: rw2
Reading or parameterisation of the deceleration with Quick Stop [SINC/s^2].				

Tab. B.140 PNU 1029

B.4.23 Axis parameters: electric drives 1 – electronic rating plate

PNU 1030	**Motor type (Motor Type)**			
Subindex 1	Class: Var	Data type: uint16	FW ...	Access: ro
Reading of the motor type.				
Value	Significance			
0x0008 (8)	Stepper motor			

Tab. B.141 PNU 1030

PNU 1034	**Max. current (Max. Current)**			
Subindex 1	Class: Var	Data type: int32	FW ...	Access: ro
Reading of the max. motor current [mA]. The value is always positive. Internally, this limits the max. “positive” and “negative” current.				

Tab. B.142 PNU 1034

PNU 1035	**Motor nominal current (Motor Rated Current)**			
Subindex 1	Class: Var	Data type: int32	FW ...	Access: ro
Reading of the motor nominal current [mA] (rating plate specification).				

Tab. B.143 PNU 1035

PNU 1036	**Motor nominal torque (Motor Rated Torque)**			
Subindex 1	Class: Var	Data type: int32	FW ...	Access: rw2
Reading or parameterisation of the motor nominal torque [mNm].				

Tab. B.144 PNU 1036

B.4.24 Axis parameters: electric drives 1 – standstill monitoring

PNU 1040	**Setpoint position (Setpoint Position)**			
Subindex 1	Class: Var	Data type: int32	FW ...	Access: ro
Reading of the target position [SINC] of the last positioning task.				

Tab. B.145 PNU 1040

PNU 1041	**Current position (Position Actual Value)**			
Subindex 1	Class: Var	Data type: int32	FW ...	Access: ro
Read the current position [SINC] of the drive.				

Tab. B.146 PNU 1041

PNU 1042	**Standstill position window (Standstill Position Window)**			
Subindex 1	Class: Var	Data type: int32	FW ...	Access: rw2
Reading or parameterisation of the standstill position window [SINC]. Amount of the position by which the drive may move after MC until the standstill monitoring responds.				

Tab. B.147 PNU 1042

PNU 1043	**Standstill delay time (Standstill Window Time)**			
Subindex 1	Class: Var	Data type: uint16	FW ...	Access: rw2
Reading or parameterisation of the standstill monitoring time [ms]. Time during which the drive must be outside the standstill position window before standstill monitoring responds.				

Tab. B.148 PNU 1043

B.4.25 Axis parameters: electric drives 1 – following error monitoring

PNU 1045	**Following error delay time (Following Error Timeout)**			
Subindex 1	Class: Var	Data type: uint16	FW ...	Access: rw2
Reading or parameterisation of the damping time [ms] for detection of deviation (following error, speed). Time in which the difference between the setpoint and actual variable must be larger than the max. permissible deviation before a following error is output.				

Tab. B.149 PNU 1045

B.4.26 Axis parameters: electric drives 1 – motor data

PNU 1059	**Current motor current (Actual Motor Current)**			
Subindex 1	Class: Var	Data type: int32	FW ...	Access: ro
Reading of the current motor current [mA].				

Tab. B.150 PNU 1059

B.4.27 Axis parameters: electric drives 1 – temperature data

PNU 1063	**Current temperature CPU (Actual Temperature CPU)**			
Subindex 1	Class: Var	Data type: int8	FW ...	Access: ro
Reading of the current temperature [°C] of the main CPU.				

Tab. B.151 PNU 1063

PNU 1065	**Min./Max. temperature CPU (Min./Max. Temperature CPU)**			
Subindex 1, 2	Class: Array	Data type: int8	FW ...	Access: ro
Reading of the permitted temperature range [°C] of the main CPU.				
Subindex 1	Min. temperature CPU (Min. Temperature CPU)			
Min. temperature of the main CPU.				
Subindex 2	Max. temperature CPU (Max. Temperature CPU)			
Min. temperature of the main CPU.				

Tab. B.152 PNU 1065

PNU 1066	**Current temperature output stage (Actual Temperature Output Stage)**			
Subindex 1	Class: Var	Data type: int8	FW ...	Access: ro
Reading of the current temperature [°C] of the output stage (load section of the controller).				

Tab. B.153 PNU 1066

PNU 1068	Min./Max. temperature output stage (Min./Max. Temperature Output Stage)			
Subindex 1, 2	Class: Array	Data type: int8	FW ...	Access: ro
Reading of the permitted temperature range [°C] of the output stage (load section of the controller).				
Subindex 1	Min. temperature output stage (Min. Temperature Output Stage)			
Min. temperature of the output stage.				
Subindex 2	Max. temperature of the output stage (Max. Temperature Output Stage)			
Max. temperature of the output stage.				

Tab. B.154 PNU 1068

B.4.28 Axis parameters: electric drives 1 – general drive data

PNU 1071	Tool load/basic mass load (Tool Load/Ground Mass)			
Subindex 1	Class: Var	Data type: uint32	FW ...	Access: rw2
Reading or parameterisation of the tool load/basic mass load. Linear axis: moving basic mass load [g]. Rotative axis: basic mass moment of inertia at the gear unit output [kgm^2 * 10^{-7}].				

Tab. B.155 PNU 1071

PNU 1073	Current intermediate circuit voltage (Actual Intermediate Circuit Voltage)			
Subindex 1	Class: Var	Data type: uint32	FW ...	Access: ro
Reading of the current intermediate circuit voltage [mV] of the controller.				

Tab. B.156 PNU 1073

PNU 1074	Current control section voltage (Actual Control Section Voltage)			
Subindex 1	Class: Var	Data type: uint32	FW ...	Access: ro
Reading of the current control section voltage [mV] of the controller.				

Tab. B.157 PNU 1074

PNU 1075	Current string current (Actual Phase Current)			
Subindex 1 … 3	Class: Array	Data type: int32	FW …	Access: ro
Reading of the latest current flows [mA] in the individual motor strings.				
Subindex 1	Current string current 1 (Actual Phase Current 1)			
Latest current of the 1st motor string.				
Subindex 2	Current string current 2 (Actual Phase Current 2)			
Latest current of the 2nd motor string.				
Subindex 3	Current string current 3 (Actual Phase Current 3)			
Latest current of the 3rd motor string.				

Tab. B.158 PNU 1075

Overwriting of PNU 1080 (basic factor or weight-dependent factor) can lead to higher motor currents, which can result in oversteering during accelerations in the torque pilot control. Greater loads are hereby applied to the drive.
The factors are calculated from the parameters (motor, gear unit, feed constant, …) of the Festo Configuration Tools (FCT) and written to PNU 1080 and should not be changed.

PNU 1080	Torque pilot control (Torque Feed Forward Control)			
Subindex 1	Class: Var	Data type: uint16	FW …	Access: rw1
Reading or parameterisation of the torque pilot control proportion [‰] in the positioning and speed direct mode. – 0 = inactive – 1000 = fully active The torque pilot control is added to the current controller setpoint value. The value is calculated from the acceleration.				

Tab. B.159 PNU 1080

C Festo Parameter Channel (FPC)

C.1 FPC for cyclical I/O data

The FPC is used for transmission of parameters in the cyclical I/O data. An additional 8 I/O bytes have been added to the 8 bytes of I/O data of the FHPP standard for this purpose.

Data	**Byte 1 … 8**								**Byte 9 … 16**							
	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
O-data	FHPP standard control bytes								FPC control data							
I-data	FHPP standard status bytes								FPC status data							

Tab. C.1 Cyclical I/O data FHPP standard + FPC

The motor controller CMMO-ST only supports the extended functions of the Enhanced Festo Parameter Channel EFPC correspondingly ➔ section C.2.

C.2 Overview of EFPC

The EFPC extended parameter channel permits automated transmission of parameters and larger datasets in the form of a parameter file.

Modules with which transmission can be easily implemented are available for some selected controllers at ➔ www.festo.com/sp.

C.2.1 EFPC structure

The EFPC extended parameter channel uses the 8 bytes of the FPC.
The structure of the EFPC is shown in ➔ Tab. C.2.

Data	**Byte 1**	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**	**Byte 7**	**Byte 8**
O-data	FPCC	Control and status data independent of the transmission mode ➔ section C.2.2						
I-data	FPCS							

Tab. C.2 EFPC structure in general

In general, observe the specification in the bus master for representation of words and double words (Intel/Motorola). For example, the representation via Modbus uses the “big endian” representation (high-order byte first).

C.2.2 FPCC and FPCS – transmission mode, request and response ID

The transmission mode is switched via bits 4 to 7 of byte 1 correspondingly ➔ Tab. C.3.

FPCC/FPCS[1)]	Mode	Function
0001xxxx	Parameter	Transmission of PNUs (16 bit) ➔ section C.3
0100xxxx	File	Transmission of parameter file ➔ section C.4

1) Values not stated = reserved

Tab. C.3 FPCC/FPCS – coding of the transmission mode

Bits 0 to 3 of byte 1 include the request or response ID ➔ Tab. C.4 and Tab. C.5.

FPCC[2)]	Value	Function	Permitted with mode	
			Parameter	File
xxxx0000	0	No job	x	x
xxxx0100	4	Parameter file upload		x
xxxx0101	5	Parameter file download		x
xxxx0110	6	Request parameter value (array)	x	
xxxx1000	8	Modify parameter value (array double word)	x	

2) Values not stated = reserved

Tab. C.4 FPCC – coding request ID

FPCS[3)]	Value	Function	Permitted with mode	
			Parameter	File
xxxx0000	0	No reply	x	x
xxxx0011	3	Parameter file transmission active		x
xxxx0101	5	Parameter transmitted (array, double word)	x	
xxxx0111	7	Task cannot be carried out with error number (transmission of parameter or parameter file currently not possible)	x	x

3) Values not stated = reserved

Tab. C.5 FPCS – coding response-ID

C.3 Parameter transmission (PNUs, internal objects)

C.3.1 Structure of EFPC in parameter transmission

The structure of EFPC in parameter transmission is shown in ➔ Tab. C.6.

Data	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
O data	FPCC	Sub-index	Parameter number (PNU)		Parameter value (PWE)			
I data	FPCS	Sub-index	Parameter number (PNU)		Parameter value (PWE)/error code			

Tab. C.6 Structure of EFPC for parameter transmission

C.3.2 Sequence of parameter transmission

Parameter transmission takes place in the following sequence:

1. Start transmission.
2. Wait until the “Transmit parameters” acknowledgment is made.
3. Wait between two successive jobs, send task identifier 0 (no job, “zero request”) and response identifier 0 (no reply).
   This ensures that an “old” response is not interpreted as a “new” response.

Parallel to the transmission, the controller must evaluate possible errors.
Before and after the parameter transmission, the telegram “no job” is exchanged cyclically between the controller and motor controller.

Written parameters must be permanently stored by writing PNU 127:2 with the value 1 so they will be secure in case of power failure.

C.3.3 Example of parameter transmission

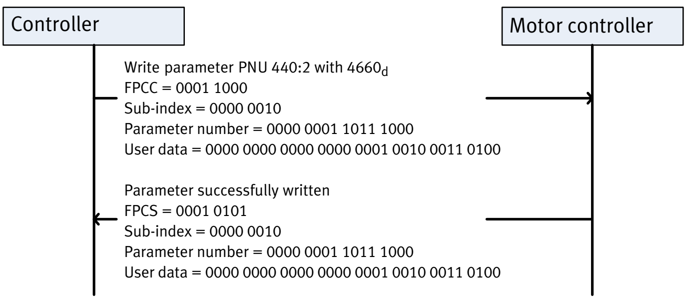


Fig. C.1 Example of parameter transmission sequence

C.3.4 Error codes

Errors are reported in the FPCS and the error code is transmitted in the user data.

Error code		Error
0	0x00	Invalid PNU
1	0x01	Parameter value cannot be altered
2	0x02	Lower or upper value limit exceeded
3	0x03	Faulty sub-index
11	0x0B	No supervising access
17	0x11	Task cannot be carried out in the operating status
101	0x65	Festo: ReqID is not supported
102	0x66	Festo: Parameter is write-only

Tab. C.7 Error codes in parameter transmission

C.4 Parameter file transmission

C.4.1 Structure of EFPC in parameter file transmission

The EFPC extended parameter channel permits automated transmission of all configurable parameters of a motor controller in the form of a parameter file.
As a result, the function of a parameter server can be implemented.
The procedure is in principle applicable for all controllers that support management of such files.

Modules with which transmission can be easily implemented are available for some selected controllers at ➔ www.festo.com/sp.

The structure of the EFPC in parameter file transmission is shown in ➔ Tab. C.8.

<table>
<tr><th>Data</th><th>Byte 1</th><th colspan="2">Byte 2</th><th>Byte 3</th><th>Byte 4</th><th>Byte 5</th><th>Byte 6</th><th>Byte 7</th><th>Byte 8</th></tr>
<tr><td rowspan="2">O-data</td><td rowspan="2">FPCC</td><td colspan="2">Package ID</td><td colspan="6" rowspan="2">User data package</td></tr>
<tr><td>3 controller bits</td><td>5 bit sequence number</td></tr>
<tr><td rowspan="2">I-data</td><td rowspan="2">FPCS</td><td colspan="2">Package ID</td><td colspan="6" rowspan="2">User data package</td></tr>
<tr><td>3 status bits</td><td>5 bit sequence number</td></tr>
</table>

Tab. C.8 Structure of EFPC for parameter file transmission

C.4.2 Package ID

The package ID is divided into 2 areas. The first 3 bits include controller and status information. The then following 5 bits include the sequence number of the data packets ➔ section C.4.5.

Control/status bits	Function	Content of user data
000xxxxx	Data transmission active	No data / user data package
001xxxxx	Start data transmission	No data / size of the parameter file in bytes
010xxxxx	Stop data transmission	No data
011xxxxx	Error	No data / error code

Tab. C.9 Package ID – 3 control or status bits

C.4.3 Parameter file and user data package

Parameter file

The parameter file functions as a data container of the complete parameter set of a motor controller. With it, parameter settings can be transmitted between identical motor controllers.
The parameter file consists of the three parts header, data range and CRC test value.

File beginning

256 bytes	Header
n bytes	Data range (objects)
2 bytes	CRC test value

File end

Fig. C.2 Structure of parameter file

With an upload of the parameter file, the parameter values are transmitted from the permanent data memory of the motor controller.
If the parameters of the motor controller were changed during execution time, although they are active they are not stored in the permanent memory of the motor controller and so are lost after a restart.
With PNU 127:2, the current parameterisation can be stored in the permanent data memory.

User data package

The parameter file is divided into 6-byte blocks for transmission and, put back together after transmission.
The LSB (lowest-order bit) of the user data is in byte 3 (Little Endian), that is, directly after the protocol header of the 8-byte EFPC. If no user data are present, all bits equal zero.
If all 6 bytes from the last transmitted data telegram are no longer needed, the remaining bytes are filled with zeros. The parameter file size, which was transmitted at the start, determines up to where the data have to be evaluated.

C.4.4 Checking and activation of the parameter file

The motor controller checks automatically whether a transmitted parameter file is compatible. This takes place both when the download is completed and the file is to be loaded as well as directly after transmission of the header information. If the parameter file is not compatible, the motor controller responds with an error and the corresponding error code.

Activating the transmitted parameter file

Through the download, the parameter file is stored in the memory of the motor controller (permanent memory), and the previously stored file (permanent memory) is deleted after the new file has been successfully checked. The currently active parameters remain uninfluenced by this.
The parameters of the new parameter file become **active only after**:

- Restart of the motor controller
  - PNU 127, subindex 3 "Reset Device" is set
  - The power supply is switched off/on
  - FCT plug-in menu [Component] [Online] [Restart controller] ([Component] [Online] [Restart Controller]).
- Setting of PNU 127, subindex 4 "Load parameter values from parameter file"
  But with this method, changed parameters of the error management are **not** taken over.

C.4.5 Sequence of parameter file transmission

Parameter file transmission takes place in the following sequence:

1. Start data transmission. The sequence number begins with 0.
2. Transmit user data packages and increment sequence number.
   As a handshake, the response is made with the same package ID after receipt of the data package. The data of the parameter file are transmitted beginning with sequence number 1. If the sequence number has reached its highest value 31, counting begins again at 0.
   The intermediate step through "no job" is eliminated in this EFPC variant, because the package ID changes in every new telegram.
3. Data transmission completed or stop data transmission.

Parallel to the transmission, the controller must evaluate possible errors.
Before and after data transmission, the telegram "no job" is exchanged cyclically between the controller and motor controller.
A stop or an error message can be written at any time in the control bits and interrupts the upload or download. A check of the sequence number then does not take place.

C.4.6 Examples of parameter file transmission

Parameter file upload – motor controller sends parameter file to the higher-order controller

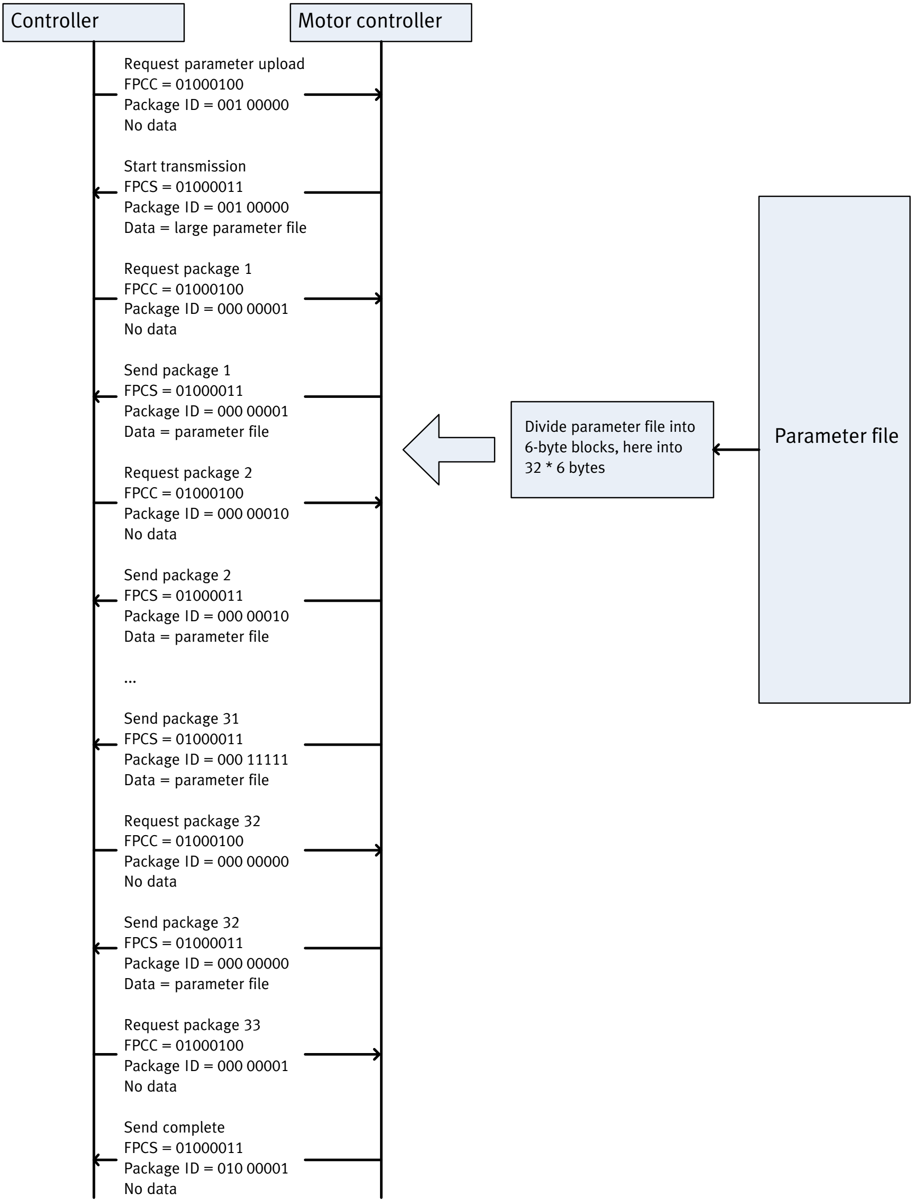


Fig. C.3 Sequence of parameter file upload

Parameter file download – higher-order controller sends parameter file to motor controller

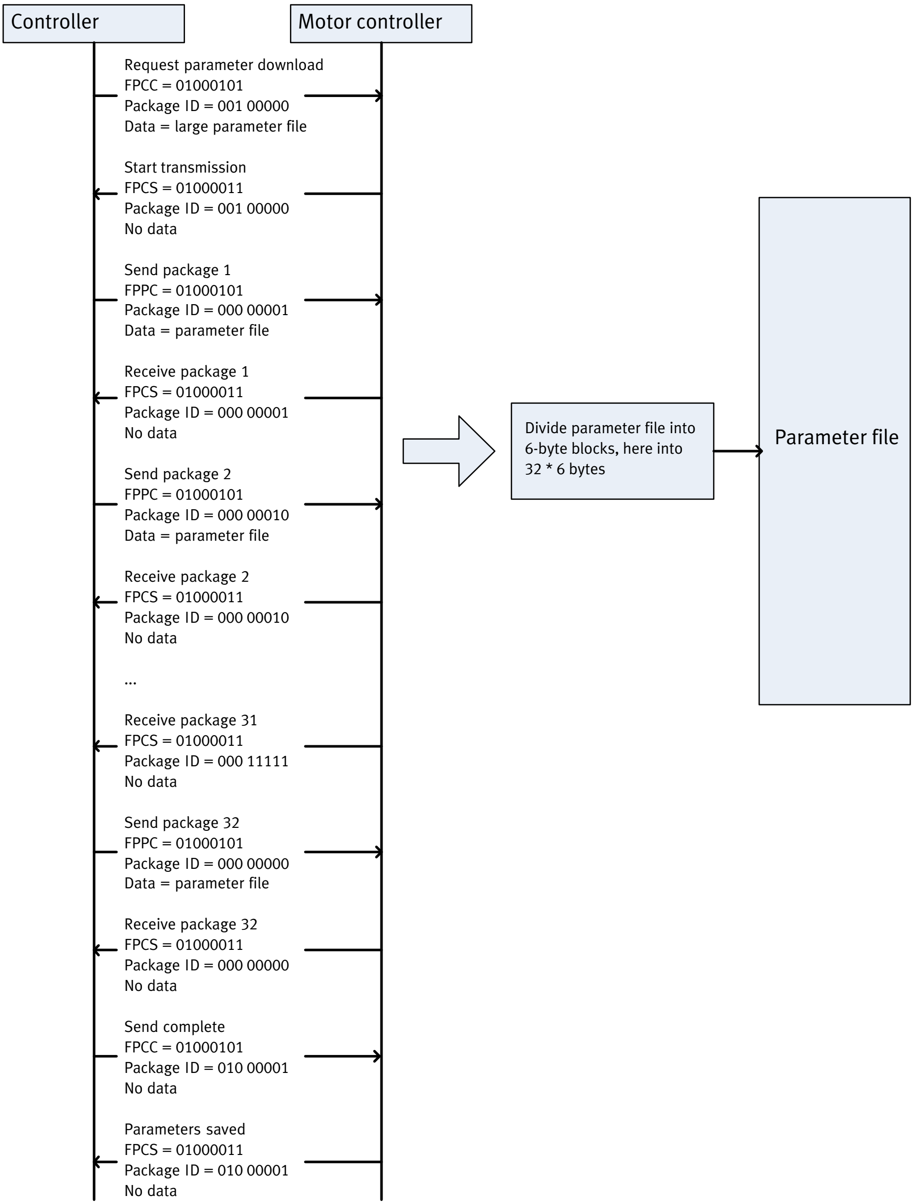


Fig. C.4 Sequence of parameter file download

Parameter file upload faulty

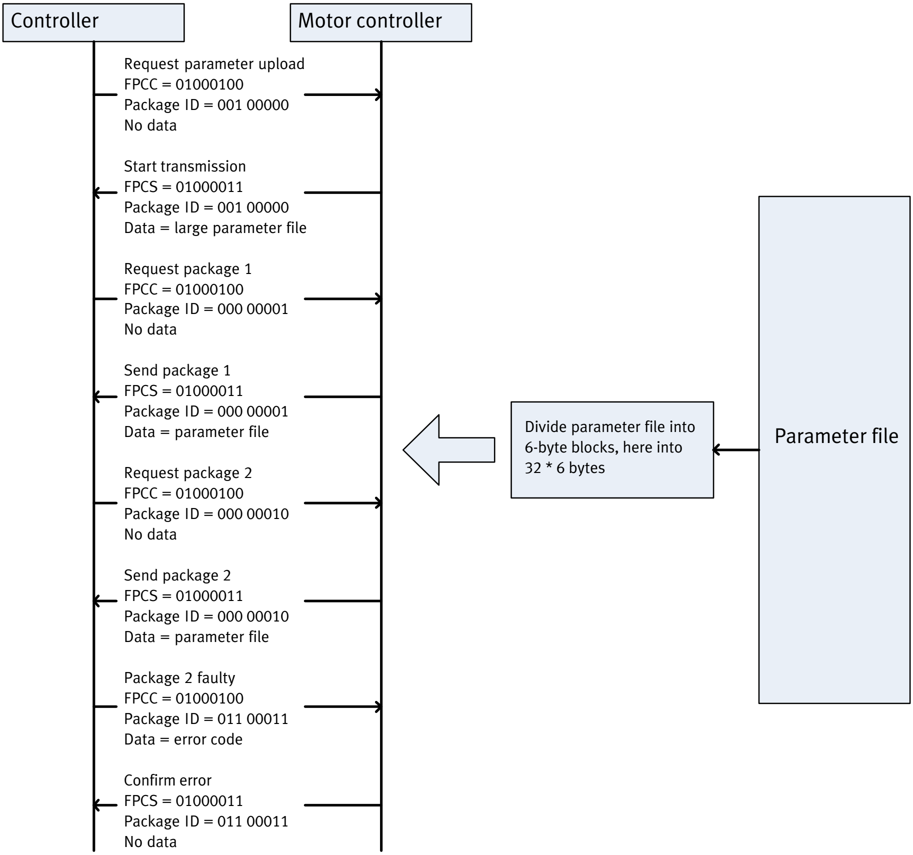


Fig. C.5 Error in parameter upload

Parameter file download faulty

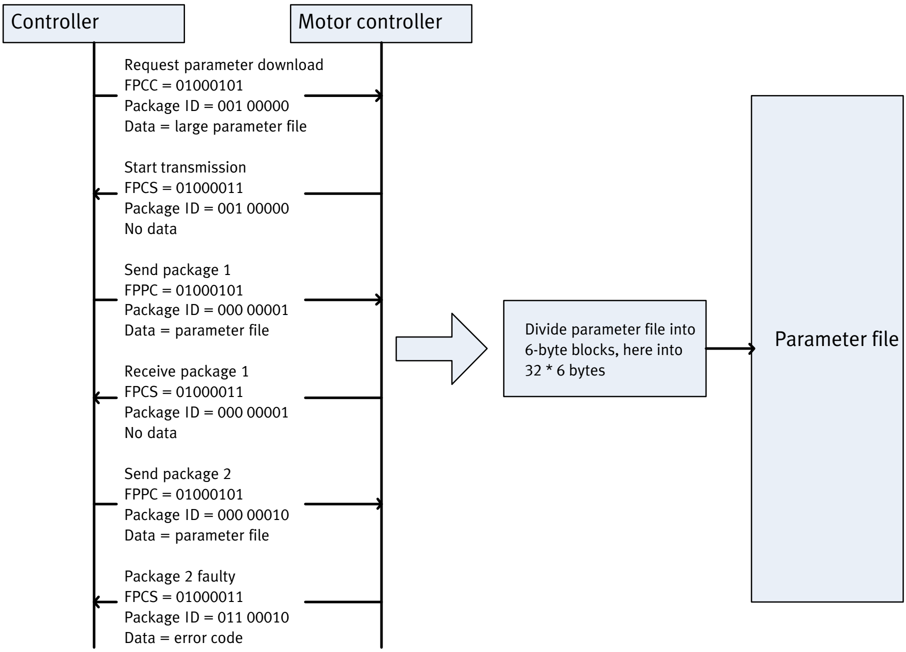


Fig. C.6 Error in parameter download

Parameter file download – FPCC is not supported

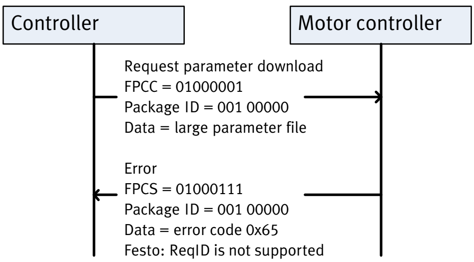


Fig. C.7 Error FPCC is not supported

The value in FPCC cannot be evaluated. The request ID included in the FPCC is not supported.

Parameter file download – EFPC is blocked

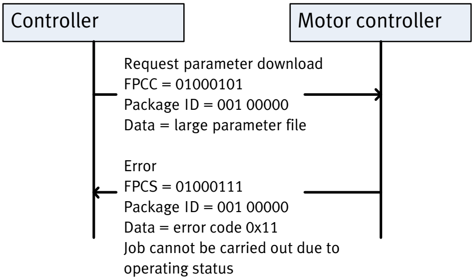


Fig. C.8 Error EFPC is blocked

Certain functions are blocked during active parameter transmission, e.g. switching to download is not allowed during an upload, and vice versa, before transmission is stopped by the controller.

C.4.7 Error codes

In parameter file transmission, errors are reported in the FPCS or in the package ID of the EFPC, depending on the type of error.

Error type 1 – The error is displayed in the FPCS (FPCS = xxxx0111)

This variant exists for all FPC variants, including all EFPC types. In the user data, the error code is reported according to the following table:

Error code		Error
17	0x11	Job cannot be carried out in the operating status. A parameter file transmission is not possible in the current operating status or configuration (e.g. no EFPC parameterised).
101	0x65	Festo: ReqID is not supported.

Tab. C.10 Error codes in the parameter file transmission – error type 1

Error type 2 – Error is displayed in the control bytes (package ID = 011xxxxx)

Byte 3 includes error codes that the motor controller sends to the controller. If the controller sends an error or the motor controller error results in interruption of parameter file transmission, this is stored as information in the diagnostic memory of the motor controller. The controller does not send an error number in the user data. The motor controller answers the controller with the error status without an error text in the user data.

Error code		Error
0	0x00	Error message from the controller
1	0x01	Incorrect sequence of the received packages (sequence number)
2	0x02	Timeout between 2 packages
3	0x03	Formatting of the telegram invalid
4	0x04	Incorrect command sequence, e.g. renewed start without a stop in between
5	0x05	Error when reading (length of the parameter file invalid or incorrect status of transmission)
6	0x06	Error in writing the parameter file
7	0x07	Dataset received or sent does not agree with what was expected
8	0x08	Error in accessing the parameter file, e.g. no master control
9	0x09	Timeout in accessing the parameter file, e.g. error is still present and must be acknowledged

Tab. C.11 Error codes in the parameter file transmission – error type 2

The errors differ regarding the effect on transmission → Tab. C.12:

Transmission	**Effect**	**Error**	
		Error code	Error
... is not interrupted	If the motor controller is again addressed with the valid syntax after the error, the upload or download can be continued. To cancel the transmission, a stop command must be sent.	Error type 1: is reported in FPCS	
		17 (0x11)	Job cannot be carried out in the operating status
		101 (0x65)	Festo: ReqID is not supported
		Error type 2: is reported in package ID	
		1 (0x01)	Incorrect sequence of the received packages
		3 (0x03)	Formatting of the telegram invalid
		4 (0x04)	Incorrect command sequence
... is interrupted	With error messages sent by the controller, transmission is interrupted. In this case, the user data are not evaluated. Differentiation between different error causes is optionally foreseeable on the controller. If transmission is interrupted, the data transmitted to the controller up to then are discarded. This should also be planned in the controller.	Error type 2: is reported in package ID	
		0 (0x00)	Error message from the controller
		2 (0x02)	Timeout between 2 packages
		5 (0x05)	Error during reading
		6 (0x06)	Error during writing
		7 (0x07)	Dataset received does not agree with what was expected
		8 (0x08)	Error of parameter file
		9 (0x09)	Access to parameter file

Tab. C.12 Effect of errors on transmission

In case of errors that result in termination of transmission, error number E27 is output at the 7-segment display of the CMMO-ST and written in the diagnostic memory. The error must then be acknowledged via the control function “Reset Fault” or with the FCT.

D Diagnostic messages

Explanation of the parameterisable classification, error response, diagnostic memory options and acknowledgement option ➔ 7.2.2.

Diagnostic messages and fault clearance		
01h	**Software error**	Definable as: **F**/-/- Diagnostic memory: always
An internal firmware error has been detected. • Contact Festo Service. – Resettable: Cannot be reset; software reset is necessary. Definable error reaction(s): A		
02h	**Default parameter file invalid**	Definable as: **F**/-/- Diagnostic memory: always
An error has been detected when examining the default parameter file. The file is damaged. • Reload the default parameter file into the device via a firmware update. If the error is still present, it means the memory may be faulty and the device needs to be replaced. – Resettable: Cannot be reset; software reset is necessary. Definable error reaction(s): A		
05h	**Zero angle determination**	Definable as: **F**/-/- Diagnostic memory: always
The rotor position could not be clearly identified. The commutation point is invalid. • Is there a motor with encoder and, if yes, is the encoder cable connected? The drive is blocked: Ensure freedom of movement. • Excessively high load: Reduce load. • The axis is not fastened stiffly enough: Stiffen the axis mounting. • The effective load is not fastened stiffly enough on the axis: Stiffen the connection. • Effective load can vibrate: Form a stiffer load; modify the natural frequency of the load. • If several drives are fitted in a system that can vibrate: Carry out commutation point search one after the other. • Controller parameters have been set incorrectly: Determine the controller parameters and set the correct value. To do this, you may have to perform a commutation point search without a load (remove the load, correctly set the tool mass and applied load), start the axis, connect the load (correctly set the tool mass and applied load), determine the new controller parameters (see FCT help on controller parameterisation), reparameterise the drive and then restart the commutation point search with new controller parameters. • This error can also occur if the set motor current is too low to move the shaft and any possible load. Correct the settings for the motor current, if necessary. – Resettable: Error can be reset immediately. Definable error reaction(s): A		

<table>
<tr><th colspan="3">Diagnostic messages and fault clearance</th></tr>
<tr><td>06h</td><td>Encoder</td><td>Definable as: F/-/-
Diagnostic memory: always</td></tr>
<tr><td colspan="3">An error has occurred during evaluation of the encoder. The current position values may be incorrect.
• Check encoder cable and connection for short-circuits, breaks or incorrect pin assignment.
• Conduct a software reset with a commutation angle search and homing procedure.
• If the error is still present, the hardware (encoder) may be defective.
– Resettable: Cannot be reset; software reset is necessary.
Definable error reaction(s): A</td></tr>
<tr><td>09h</td><td>Offset determination for current measurement</td><td>Definable as: F/-/-
Diagnostic memory: always</td></tr>
<tr><td colspan="3">An error has occurred during initialisation of the current measurement.
• Perform a software reset.
– Resettable: Cannot be reset; software reset is necessary.
Definable error reaction(s): A</td></tr>
<tr><td>0Ah</td><td>General error</td><td>Definable as: F/-/-
Diagnostic memory: always</td></tr>
<tr><td colspan="3">An internal error has occurred.
• Restart device. If the error occurs frequently, contact Festo Service.
– Resettable: Error can be reset immediately.
Definable error response(s): B</td></tr>
<tr><td>0Bh</td><td>Parameter file invalid</td><td>Definable as: F/-/-
Diagnostic memory: always</td></tr>
<tr><td colspan="3">No valid parameter set stored. After creation of the parameter file, a firmware update is performed, if necessary: As much data as possible is automatically loaded from the parameter file. Parameters that are not initialised by the parameter file are loaded from the default parameter file.
• Enter a valid parameter set in the device. If the error is still present, the hardware may be defective.
– Resettable: Error can be reset immediately.
Definable error reaction(s): A</td></tr>
<tr><td>0Ch</td><td>Firmware update execution error</td><td>Definable as: F/-/-
Diagnostic memory: optional</td></tr>
<tr><td colspan="3">The firmware update has not been properly executed or completed.
• Check Ethernet connection between device and PC. Restart device and perform the firmware update again. Check whether valid firmware has been selected for the device. The previous firmware version remains active until the firmware update has been successfully completed. If this error is still present, the hardware may be defective.
– Resettable: Cannot be reset; software reset is necessary.
Definable error reaction(s): A</td></tr>
</table>

Diagnostic messages and fault clearance		
0Dh	**Over-current**	Definable as: **F**/-/- Diagnostic memory: always
Short circuit in the motor, lines or brake chopper. Output stage defective. Incorrect parameterisation of the current regulator. • Check parameterisation of the current regulator. An incorrectly parameterised current regulator can generate currents up to the short-circuit limit; as a rule this is clearly noticeable through high frequency whistling. Inspection with the trace function in FCT (active current actual value). • Error message immediately after connecting to the load supply: Short circuit in the output stage. The device must be replaced immediately. • Error message only occurs when setting the output stage enable: Disconnect the motor plug directly at the controller; if the error still occurs, the controller must be replaced. If the error only occurs when the motor cable is connected, check the motor and cable for short circuits, e.g. with a multimeter. – Resettable: Cannot be reset; software reset is necessary. Definable error reaction(s): A		
0Eh	**I²t malfunction motor**	Definable as: **F**/-/- Diagnostic memory: always
The I²t limit for the motor is reached. The motor or the drive system may be insufficient for the required task. • Check the layout of the drive system. • Check the mechanical system for sluggishness. • Reduce load/dynamic response, longer time delays. – Resettable: Error can only be reset after the cause of the error has been eliminated. Definable error response(s): B, **C**		
11h	**Software limit positive**	Definable as: **F**/-/- Diagnostic memory: optional
The position setpoint has reached or exceeded the respective software end position. • Check target data. • Check positioning area. • This error can be reset immediately. Afterwards start a corresponding positioning record or move the drive by using the jogging function. Movements in a positive direction are blocked. – Resettable: Error can be reset immediately. Definable error response(s): A, **B**, C, E, F		

Diagnostic messages and fault clearance

12h	**Software limit negative**	Definable as: **F**/-/- Diagnostic memory: optional

The position setpoint has reached or exceeded the respective software end position.

- Check target data.
- Check positioning area.
- This error can be reset immediately. Afterwards start a corresponding positioning record or move the drive by using the jogging function. Movements in a negative direction are blocked.

– Resettable: Error can be reset immediately.
  Definable error response(s): A, **B**, C, E, F

13h	**Positive direction locked**	Definable as: **F**/-/- Diagnostic memory: optional

A limit switch error or a software limit position error has occurred and subsequent positioning in the blocked direction has been initiated.

- Check target data.
- Check positioning area.
- This error can be reset immediately. Afterwards start a corresponding positioning record or move the drive by using the jogging function. Movements in a positive direction are blocked.

– Resettable: Error can be reset immediately.
  Definable error response(s): A, **B**, C, E, F

14h	**Negative direction locked**	Definable as: **F**/-/- Diagnostic memory: optional

A limit switch error or a software limit position error has occurred and subsequent positioning in the blocked direction has been initiated.

- Check target data.
- Check positioning area.
- This error can be reset immediately. Afterwards start a corresponding positioning record or move the drive by using the jogging function. Movements in a negative direction are blocked.

– Resettable: Error can be reset immediately.
  Definable error response(s): A, **B**, C, E, F

Diagnostic messages and fault clearance		
15h	**Output stage temperature exceeded**	Definable as: **F**/-/- Diagnostic memory: optional
The permissible limit value for the output stage temperature has been exceeded. The output stage is possibly overloaded. • This error can only be acknowledged if the temperature is within the permissible range. • Check cylinder sizing. • Check the mechanical system for sluggishness. • Reduce the ambient temperature, improve heat dissipation. Check motor and cabling for short circuits. – Acknowledgeability: Error can only be acknowledged after eliminating the cause. Definable error response(s): A, B, **C**, D		
16h	**Output stage temperature too low**	Definable as: **F**/-/- Diagnostic memory: optional
The ambient temperature is below the permissible range. • Increase the ambient temperature. This error can only be acknowledged if the temperature is within the permissible range. – Resettable: Error can only be reset after the cause of the error has been eliminated. Definable error response(s): A, B, **C**, D		
17h	**Logic voltage exceeded**	Definable as: **F**/-/- Diagnostic memory: optional
The logic power supply monitor has detected an overvoltage. This is either due to an internal defect or an excessive supply voltage. • Check external supply voltage directly on the device. • If the error is still present after a reset has been conducted, it means there is an internal defect and the device has to be replaced. – Resettable: Error can only be reset after the cause of the error has been eliminated. Definable error response(s): A, **B**		
18h	**Logic voltage too low**	Definable as: **F**/-/- Diagnostic memory: optional
The logic power supply monitor has detected an undervoltage. There is either an internal defect or an overload/short circuit caused by connected peripherals. • Separate device from the entire peripheral equipment and check whether the error is still present after reset. If it is, it means there is an internal defect and the device has to be replaced. – Resettable: Cannot be reset; software reset is necessary. Definable error reaction(s): A		

Diagnostic messages and fault clearance

19h	**Real time error LM-CPU**	Definable as: **F**/-/- Diagnostic memory: optional
The LM-CPU requires more computation time than is available to it. • Check whether multiple connections have been established to the device. If yes, terminate the unneeded connections. Further remedial measures: Do without trace drawings, reduce bus load – Resettable: Error can be reset immediately. Definable error response(s): A, **B**		
1Ah	**Intermediate circuit voltage exceeded**	Definable as: **F**/-/- Diagnostic memory: always
Load voltage not within the permissible range. Braking resistor is overloaded; too much braking energy, which cannot be dissipated quickly enough. Braking resistor is defective. • Check the load voltage; measure voltage directly at the controller input. • Check cylinder sizing: braking resistor overloaded? • In the event of a defective internal braking resistor: Replace the controller. – Acknowledgeability: Error can only be acknowledged after eliminating the cause. Definable error response(s): A, **B**		
1Bh	**Intermediate circuit voltage too low**	Definable as: **F**/W/- Diagnostic memory: optional
The load voltage is too low. • Voltage drops under load: power supply unit too weak, supply line too long, cross section too small? • If you intentionally want to operate the device with a lower voltage, parameterise this malfunction as a warning or information. • Measure load voltage (directly at the controller input). – For parameterisation as an error: The error can only be acknowledged after the cause is eliminated. Definable error reaction(s): **A** – For parameterisation as a warning: The warning disappears if the load voltage is back within the permissible range.		
22h	**Homing**	Definable as: **F**/-/- Diagnostic memory: optional
Homing run to switch unsuccessful. A corresponding switch has not been found. • Check to make sure the correct homing method is set. • Check to see if the homing switch is connected and if it has been parameterised correctly (normally closed contact/normally open contact?). Check the functionality of the switch and check the cable for wire breaks. • If the error is still present, it means there is an internal defect and the device has to be replaced. – Resettable: Error can be reset immediately. Definable error response(s): **B**, C, E, F		

Diagnostic messages and fault clearance

23h	**No index pulse found**	Definable as: **F**/-/- Diagnostic memory: optional

Error during homing: no zero pulse found. Encoder defective or incorrect parameterisation of the encoder resolution.

- Check the output signals of the encoder, in particular the index signal.
- Check the parameterisation of the encoder resolution.

– Resettable: Cannot be reset; software reset is necessary.
  Definable error response(s): **B**, C, E, F

24h	**Drive function is not supported in open-loop operation**	Definable as: **F**/W/- Diagnostic memory: optional

Function is not supported in this mode. The request has been ignored.

- Change the operating mode or select a different drive function.

– If parameterisation as an error: Error can be acknowledged immediately.
  Definable error response(s): **E**, F
– For parameterisation as a warning: The warning disappears if a switch is made to a valid drive function.

25h	**Path calculation**	Definable as: **F**/-/- Diagnostic memory: optional

The positioning target cannot be reached through the positioning options or the edge conditions. During record sequencing: The end speed of the last record was higher than the target speed of the following record.

- Check the parameterisation of the affected records.
- Also check the actual values of the previous positioning process at the switching point, if necessary, by using the trace function. The error may be caused by the actual velocity or the actual acceleration being too high at the switching point.

– Resettable: Error can be reset immediately.
  Definable error reaction(s): A

27h	**Save parameters**	Definable as: **F**/-/- Diagnostic memory: optional

Error during writing of the internal permanent memory.

- Execute the last operation again.
- Check the following: Is an error present that can be reset first? When downloading a parameter file, check if the version of the parameter file fits the firmware. If the error occurs again, please contact Festo Service.

– Resettable: Error can be reset immediately.
  Definable error response(s): **F**, G

Diagnostic messages and fault clearance		
28h	**Homing required**	Definable as: **F**/W/- Diagnostic memory: optional
A valid reference travel has not yet been conducted. The drive is no longer referenced (e.g. as a result of a logic power failure or because the homing method or axis zero point has been changed). • Perform homing or repeat the last homing if it was not completed successfully. – If defined as an error: Error can be reset immediately. Definable error reaction(s): **B**, C, D, E, F, G – For parameterisation as a warning: The warning disappears if the homing run has been completed successfully.		
29h	**Target position behind negative software limit**	Definable as: **F**/-/- Diagnostic memory: optional
The start of a positioning task was suppressed because the target lies behind the negative software limit position. • Check target data. • Check positioning area. • Check position set type (absolute/relative?). – Resettable: Error can be reset immediately. Definable error response(s): **B**, C, E, F		
2Ah	**Target position behind positive software limit**	Definable as: **F**/-/- Diagnostic memory: optional
The start of a positioning task was suppressed because the target lies behind the positive software limit position. • Check target data. • Check positioning area. • Check position set type (absolute/relative?). – Resettable: Error can be reset immediately. Definable error response(s): **B**, C, E, F		
2Bh	**Firmware update, invalid firmware**	Definable as: **F**/W/- Diagnostic memory: optional
The firmware update process could not be performed. The firmware version is incompatible with the hardware used. • Determine the version of the hardware. You can ascertain the compatible firmware designs and download the appropriate firmware from the Festo website. – If defined as an error: Error can be reset immediately. Definable error reaction(s): **A** – For parameterisation as a warning: The warning disappears if a new FW download process is started.		

<table>
<tr><th colspan="3">Diagnostic messages and fault clearance</th></tr>
<tr><td>2Dh</td><td>I^2t warning motor</td><td>Definable as: -/W/I
Diagnostic memory: optional</td></tr>
<tr><td colspan="3">The I^2t warning limit for the motor is reached.
• Parameterise message as a warning or suppress completely as information.
– For parameterisation as a warning: The warning disappears if the I^2t integral is below 80 %.</td></tr>
<tr><td>2Eh</td><td>Index pulse too close on proximity sensor</td><td>Definable as: F/-/-
Diagnostic memory: optional</td></tr>
<tr><td colspan="3">The switching point of the proximity sensor is too close to the index pulse. This can in some cases mean that no reproducible reference position can be determined.
• Move reference switches on the axis. You can display the distance between the switch and index pulse in the FCT.
– Resettable: Error can be reset immediately.
Definable error response(s): B, C, E, F</td></tr>
<tr><td>2Fh</td><td>Following error</td><td>Definable as: F/W/I
Diagnostic memory: optional</td></tr>
<tr><td colspan="3">The following error has become too great. This error can occur during positioning and speed modes.
• Enlarge error window.
• Acceleration, speed, jerk or load too great? Mechanics stiff?
• Motor overloaded (current limitation from I^2t monitoring active?)
– If defined as an error: The error can only be reset after the cause is eliminated.
Definable error response(s): B, C, E, F
– For parameterisation as a warning: The warning disappears if the following error is back within the permissible range.</td></tr>
<tr><td>32h</td><td>FCT connection with master control</td><td>Definable as: F/-/-
Diagnostic memory: optional</td></tr>
<tr><td colspan="3">Connection to the FCT has been interrupted.
• Check the connection and perform a reset, if necessary.
– Resettable: Error can be reset immediately.
Definable error response(s): B, C, D, E, F, G</td></tr>
<tr><td>33h</td><td>Output stage temperature warning</td><td>Definable as: -/W/I
Diagnostic memory: optional</td></tr>
<tr><td colspan="3">Temperature of output stage increased.
• Check cylinder sizing.
• Check motor and cabling for short circuits.
• Check the mechanical system for sluggishness.
• Reduce the ambient temperature; take output derating into account; improve heat dissipation.
– For parameterisation as a warning: The warning disappears if the temperature is back below the danger threshold.</td></tr>
</table>

Diagnostic messages and fault clearance		
34h	**Safe Torque Off (STO)**	Definable as: F/**W**/I Diagnostic memory: optional
The “Safe Torque Off” safety function has been requested. • Observe the separate documentation for the STO function. – If defined as an error: The error can only be reset after the cause is eliminated. Definable error reaction(s): 0 – For parameterisation as a warning: The warning disappears if the STO is no longer requested.		
37h	**Standstill monitoring**	Definable as: -/**W**/I Diagnostic memory: optional
The actual position is outside the downtime window. Parameterisation of the window may be too narrow. • Check parameterisation of the downtime window. – If defined as a warning: The warning is no longer active when the actual position is within the standstill window again or a new record has been started.		
38h	**Parameter file access**	Definable as: **F**/-/- Diagnostic memory: optional
During a parameter file procedure all other reading and writing routines for the parameter file are blocked. • Wait until the process is complete. The time between 2 parameter file downloads should not be less than 3 s. – Acknowledgement option: Error can only be acknowledged after eliminating the cause. Definable error response(s): **F**, G		
39h	**Trace warning**	Definable as: -/**W**/- Diagnostic memory: optional
An error has occurred during trace recording. • Start a new trace recording. – For parameterisation as a warning: The warning disappears if a new trace has been started.		
3Ah	**Homing timeout**	Definable as: **F**/-/- Diagnostic memory: optional
Error during homing process in controlled operation. The switch has not been found within a certain time. • Check the switch configuration and the electric connection of the switch(es). – Resettable: Error can be reset immediately. Definable error response(s): **B**, C, E, F		
3Bh	**Homing method invalid**	Definable as: **F**/-/- Diagnostic memory: optional
Homing error. A homing method block has been set, for example, in open-loop operation. • Select permitted reference travel method. – Resettable: Error can be reset immediately. Definable error response(s): **E**, F		

<table>
<tr><th colspan="3">Diagnostic messages and fault clearance</th></tr>
<tr><td>3Ch</td><td>Two edges in one pulse</td><td>Definable as: F/-/-
Diagnostic memory: optional</td></tr>
<tr><td colspan="3">Two input signals have been set in the valve type in one input read cycle.
• Program the PLC so that two records (or a record and homing run) are not started in the same pulse. In the event of manual control, only operate one switch after the other.
– Resettable: Error can be reset immediately.
Definable error response(s): B, C, E, F</td></tr>
<tr><td>3Dh</td><td>Start-up event</td><td>Definable as: -/-/
Diagnostic memory: always</td></tr>
<tr><td colspan="3">The device has been switched on or was switched on for longer than 48 days. This event also occurs when deleting the diagnostic memory. The start-up event does not occur if the preceding entry in the diagnostic memory has already been a start-up event.
• This event is used only for better documentation of the diagnostic messages that occurred.</td></tr>
<tr><td>3Eh</td><td>Diagnostic memory</td><td>Definable as: F/-/-
Diagnostic memory: always</td></tr>
<tr><td colspan="3">An error has occurred when writing or reading from the diagnostic memory.
• Reset error. If the error is still present, it means a memory module is probably defective or an incorrect entry has been stored.
• Clear diagnostic memory. If the error is still present, the device needs to be replaced.
– Resettable: Error can be reset immediately.
Definable error response(s): F, G</td></tr>
</table>

<table>
<tr><th colspan="3">Diagnostic messages and fault clearance</th></tr>
<tr><td>3Fh</td><td>Record invalid</td><td>Definable as: F/-/-
Diagnostic memory: optional</td></tr>
<tr><td colspan="3">The started record is invalid. The record data is implausible or the record type is invalid.
• Check parameters of the record.
– Resettable: Error can be reset immediately.
Definable error response(s): B, C, D, E, F, G</td></tr>
<tr><td>40h</td><td>Last teaching not successful</td><td>Definable as: -/W/I
Diagnostic memory: optional</td></tr>
<tr><td colspan="3">Teaching of the current positioning record is not possible.
• The current positioning record must be of the type 'position record'.
– If defined as a warning: The warning is no longer active when the following TEACH attempt is successful or a switch takes place from the Teach mode (mode 1) to normal operation (mode 0).</td></tr>
<tr><td>41h</td><td>System reset</td><td>Definable as: F/-/-
Diagnostic memory: always</td></tr>
<tr><td colspan="3">An internal firmware error has been detected.
• Contact Festo Service.
– Resettable: Error can be reset immediately.
Definable error reaction(s): A</td></tr>
<tr><td>43h</td><td>FCT connection without master control</td><td>Definable as: -/W/I
Diagnostic memory: optional</td></tr>
<tr><td colspan="3">There is no longer a connection to the FCT, e.g. the cable was disconnected.
• Check the connection and perform a reset, if necessary.
– For parameterisation as a warning: The warning disappears if the connection to the FCT is re-established.</td></tr>
<tr><td>44h</td><td>Parameter file not compatible with firmware</td><td>Definable as: -/W/I
Diagnostic memory: always</td></tr>
<tr><td colspan="3">The parameter file that was just written to the device is not suitable for the firmware of that device. As much data as possible is automatically taken over from the parameter file. Parameters that are not initialised through the parameter file are imported from the default parameter file. If new firmware software is required, all parameters might not be written.
• Load a valid parameter file into the device.
– If defined as a warning: The warning disappears when a new parameter file is successfully written.</td></tr>
<tr><td>45h</td><td>IO-Link system error</td><td>Definable as: F/-/-
Diagnostic memory: always</td></tr>
<tr><td colspan="3">Error during initialization of the IO-Link protocol stack
• Check the FHPP configuration with FCT.
– Acknowledgeability: Error can only be acknowledged after eliminating the cause.
Definable error response(s): A, B, C, D</td></tr>
</table>

Diagnostic messages and fault clearance		
46h	**IO-Link communication error**	Definable as: **F**/-/- Diagnostic memory: optional
Error during transmission of an IO-Link telegram • Repeat communication. Perform a software reset. If this error occurs frequently, check the IO-Link network. – Acknowledgeability: Error can only be acknowledged after eliminating the cause. Definable error response(s): **B**, C, E, F		
47h	**Modbus connection with master control**	Definable as: **F**/-/- Diagnostic memory: optional
The Modbus connection to the controller has been interrupted. • Check the connection and perform a reset. – Resettable: Error can be reset immediately. Definable error reaction(s): **B**, C, D, E, F, G – For parameterisation as a warning: The warning disappears if the connection to the controller is re-established.		
48h	**Modbus connection without master control**	Definable as: -/**W**/I Diagnostic memory: optional
There is no longer a connection to the controller, e.g. the cable was disconnected. • Check the connection and perform a reset. – For parameterisation as a warning: The warning disappears if the connection to the controller is re-established.		
4Ch	**Value is out of range**	Definable as: **F**/-/- Diagnostic memory: optional
The object value could not be written because the value lies outside the permitted range of values. • Write the object again, taking due account of the permitted range of values. – Resettable: Error can be reset immediately. Definable error response(s): B, C, D, **E**, F, G		
4Dh	**Bootloader memory error**	Definable as: **F**/-/- Diagnostic memory: always
In the boot procedure, a defective memory cell was detected. • Perform a firmware update. If the error is still present, the memory might be faulty. Then the device must be replaced. – Resettable: Cannot be reset; software reset is necessary. Definable error reaction(s): A		

Diagnostic messages and fault clearance		
4Eh	**Overload 24 V Outputs**	Definable as: **F**/-/- Diagnostic memory: always
A short circuit or overload has occurred to an external 24 V supply voltage of the device. • Check wiring of the STO interface, reference switches and digital inputs and outputs. – Acknowledgement option: Error can only be acknowledged after the cause is eliminated. Definable error response(s): A, **B**		
4Fh	**System information**	Definable as: -/-/- Diagnostic memory: always
A device-specific system event has occurred. • This event is used for extended diagnostics.		

E Terms and abbreviations

The following terms and abbreviations are used in this description.
You can find fieldbus-specific terms and abbreviations in the respective chapter.

Term/abbreviation	Significance
Axis	Mechanical component of a drive that transfers the drive force for the movement. An axis enables the attachment and guiding of the effective load and the attachment of a reference switch.
Axis zero point (AZ)	Point of reference of the software end positions and project zero point. The axis zero point AZ is defined by a preset distance (offset) from the reference point REF.
Controller	Programmable logic controller; short: controller (also IPC: industrial PC).
Drive	Complete actuator, consisting of motor, encoder and axis, optionally with gear unit, if necessary with motor controller.
Encoder	Electrical pulse generator (usually a rotor position encoder). The motor controller evaluates the generated electrical signals and calculates from this the position and speed.
Enhanced Festo Parameter Channel (EFPC)	Extended function of the Festo Parameter Channel (FPC), e.g. for transmission of parameter files.
Festo Configuration Tool (FCT)	Software with uniform project and data management for supported types of equipment. The special requirements of a device type are supported with the necessary descriptions and dialogues by means of plug-ins.
Festo Handling and Positioning Profile (FHPP)	Uniform fieldbus data profile for position controllers from Festo
Festo Parameter Channel (FPC)	Parameter access according to the "Festo Handling and Positioning Profile" (I/O messaging, optionally additional 8 bytes I/O)
FHPP standard	Defines the sequence control in accordance with the "Festo Handling and Positioning Profile" (I/O messaging 8 bytes I/O)
Force/torque mode (Profile Torque Mode)	Operating mode for execution of a direct positioning task with force control (open loop transmission control) through regulation of the motor current.
Homing run	Positioning procedure in which the reference point and therefore the origin of the measuring reference system of the axis are defined.
Homing (Homing mode)	Operating mode for determining the measurement reference system of the axis.
Jogging	Manual travel in a positive or negative direction. Function for setting positions by approaching the target position, e.g. when teaching position sets (Teach mode).

Term/abbreviation	Significance
Load voltage, logic voltage	The load voltage supplies the power electronics of the motor controllers and thereby the motor. The logic voltage supplies the evaluation and control logic of the motor controllers.
Logic 0	There is a 0 V signal present at the input or output (positive logic, corresponds to LOW).
Logic 1	There is a 24 V signal present at the input or output (positive logic, corresponds to HIGH).
Motor controller	Includes power electronics + regulator + position controller, evaluates sensor signals, calculates movements and forces and provides the power supply for the motor via the power electronics.
Operating mode	Type of control or internal operating mode of the motor controller. – Type of control: record selection, direct mode – Operating mode of the controller: position profile mode, profile torque mode, profile velocity mode – Predefined sequences: homing mode...
Position mode (Profile Position mode)	Operating mode for executing a position set or a direct positioning task with position control (closed loop position control).
Project zero point (PZ) (Project Zero point)	Point of reference for all positions in positioning tasks. The project zero point PZ forms the basis for all absolute position specifications (e.g. in the position set table or with direct control via control interface). The PZ is defined by an adjustable distance (offset) from the axis zero point.
Reference point (REF)	Point of reference for the incremental measuring system. The reference point defines a known orientation or position within the travel distance of the drive.
Software end position	Programmable stroke limit (point of reference = axis zero point) – Software end position, positive: max. limit position of the stroke in positive direction; must not be exceeded during positioning. – Software end position, negative: min. limit position in negative direction; must not be fallen below during positioning.
Speed adjustment (Profile Velocity mode)	Operating mode for executing a position set or a direct positioning task with control of the speed or rotational speed.
Teaching	Operating mode for setting positions by approaching the target position, e.g. when creating positioning sets.

Tab. E.1 Index of terms and abbreviations

Index

A
Axis zero point 140, 178

D
Diagnostic memory 78
Direct application 29
Drive 178

E
EINC 81
Electric axis 178
Encoder 178
Encoder increments 81
Enhanced Festo Parameter Channel (EFPC) . 178
Error responses 80
Extended Festo Parameter Channel (EFPC) . . 150

F
Festo Configuration Tool (FCT) 178
Festo Parameter Channel (FPC) 150, 178
FHPP operating mode
– Direct application 29
– Record selection 29

H
Homing 178
– Reference point 179

I
I-Port 16
Interface increments 81
IO-Link 11
IODD files 13

J
Jogging 178

M
Master 18
Message
– Following error 71
– Motion Complete 70
– Standstill monitoring 72
Motor controller 179

N
Notes on the documentation 7

O
Operating mode 179
– Homing 178
– Positioning mode 179
– Profile torque mode (see force mode) 178
– Speed adjustment 179
– Teaching 179
Operating mode (FHPP operating mode)
– Direct application 29
– Record selection 29

P
Parameter file transmission 154
PLC 178
Positioning mode 179
Profile position mode 179
Profile torque mode (see force mode) 178
Profile velocity mode 179
Project zero point 125, 179

R
Record selection 29

S
SINC 81
Software end position 125, 179
– Negative (lower) 179
– Positive (upper) 179
Speed regulation 179
Standstill monitoring 72

T
Teaching 179

Copyright:
Festo AG & Co. KG
Ruiter Straße 82
73734 Esslingen
Germany

Phone:
+49 711 347-0

Fax:
+49 711 347-2144

E-mail:
service_international@festo.com

Internet:
www.festo.com