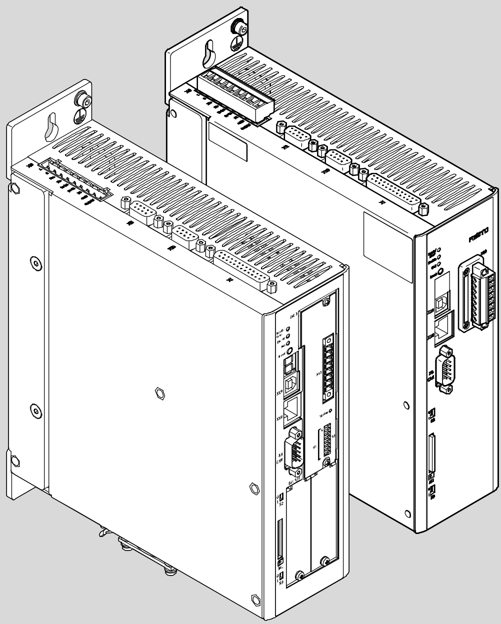
FHPP for motor controller

CMMP-AS-...-M3/-M0



FESTO

Description

Festo handling and positioning profile

for motor controller CMMP-AS-...-M3 via fieldbus:
– CANopen
– Modbus TCP
– PROFINET
– PROFIBUS
– EtherNet/IP
– DeviceNet
– EtherCAT

with interface:
– CAMC-F-PN
– CAMC-PB
– CAMC-F-EP
– CAMC-DN
– CAMC-EC

for motor controller CMMP-AS-...-M0 via fieldbus:
– CANopen
– Modbus TCP

8046788
1512b

Translation of the original instructions
GDCP-CMMP-M3/-M0-C-HP-EN

Identification of hazards and instructions on how to prevent them:

Danger
Immediate dangers which can lead to death or serious injuries

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

Table of Contents – CMMP-AS-...-M3/-M0

Instructions on this documentation

This documentation includes the Festo Handling and Position Profile (FHPP) for the motor controller CMMP-AS-...-M3 and CMMP-AS-...-M0 corresponding to the section "Information on the version".
This provides you with supplementary information about control, diagnostics and parameterisation of the motor controllers via the fieldbus.

- Unconditionally observe the general safety regulations for the CMMP-AS-...-M3/-M0.

i You will find the general safety regulations in the hardware documentation, GDCP-CMMP-M3-HW-... and GDCP-CMMP-M0-HW-...➔ Tab. 2.

M3 Sections that are marked "M3", as illustrated here, are only valid for the controller family CMMP-AS-...**-M3**. This also applies to the marking "M0" accordingly.

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.

Service

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

Information on the version

This documentation refers to the following versions:

Motor controller	Version
CMMP-AS-...-M3	Motor controller CMMP-AS-...-M3 from Rev 01
	FCT plug-in CMMP-AS from Version 2.0.x.
CMMP-AS-...-M0	Motor controller CMMP-AS-...-M0 from Rev 01
	FCT plug-in CMMP-AS from Version 2.0.x.

Tab. 1 Versions

i This description does not apply to the older variants CMMP-AS-... (without -M3/-M0). Use the assigned FHPP description for these variants.

➔ **Note**
With newer revisions, check whether there is a newer version of this documentation available ➔ www.festo.com/sp

Documentation

You will find additional information on the motor controller in the following documentation:

User documentation on the motor controller CMMP-AS-...-M3/-M0	
Name, type	**Contents**
Hardware description, GDCP-CMMP-M3-HW-...	Mounting and installation of the motor controller CMMP-AS-...-**M3** for all variants/output classes (1-phase, 3-phase), pin assignments, error messages, maintenance.
Description of functions, GDCP-CMMP-M3-FW-...	Functional description (firmware) CMMP-AS-...-**M3**, instructions on commissioning.
Hardware description, GDCP-CMMP-M0-HW-...	Mounting and installation of the motor controller CMMP-AS-...-**M0** for all variants/output classes (1-phase, 3-phase), pin assignments, error messages, maintenance.
Description of functions, GDCP-CMMP-M0-FW-...	Functional description (firmware) CMMP-AS-...-**M0**, instructions on commissioning.
Description of FHPP, GDCP-CMMP-M3/-M0-C-HP-...	Control and parameterisation of the motor controller via the FHPP Festo profile. – Motor controller CMMP-AS-...-**M3** with the following fieldbuses: CANopen, Modbus TCP, PROFINET, PROFIBUS, EtherNet/IP, DeviceNet, EtherCAT. – Motor controller CMMP-AS-...-**M0** with fieldbuses CANopen, Modbus TCP.
Description of CiA 402 (DS 402), GDCP-CMMP-M3/-M0-C-CO-...	Control and parameterisation of the motor controller via the device profile CiA 402 (DS402) – Motor controller CMMP-AS-...-**M3** with the following fieldbuses: CANopen and EtherCAT. – Motor controller CMMP-AS-...-**M0** with fieldbus CANopen.
Description of CAM editor, P.BE-CMMP-CAM-SW-...	Cam disc function (CAM) of the motor controller CMMP-AS-...-**M3/-M0**.
Description of the safety module, GDCP-CAMC-G-S1-...	Functional safety engineering for the motor controller CMMP-AS-...-**M3** with the safety function STO.
Description of the safety module, GDCP-CAMC-G-S3-...	Functional safety engineering for the motor controller CMMP-AS-...-**M3** with the safety functions STO, SS1, SS2, SOS, SLS, SSR, SSM, SBC.
Description of the safety function STO, GDCP-CMMP-AS-M0-S1-...	Functional safety engineering for the motor controller CMMP-AS-...-**M0** with the integrated safety function STO.
Description for exchange and project conversion GDCP-CMMP-M3/-M0-RP-...	Motor controller CMMP-AS-...-**M3/-M0** as a replacement device for previous motor controller CMMP-AS. Changes to the electrical installation and description of project conversion.
Help for the FCT plug-in CMMP-AS	User interface and functions of the CMMP-AS plug-in for the Festo Configuration Tool. ➔ www.festo.com/sp

Tab. 2 Documentation on the motor controller CMMP-AS-...-M3/-M0

1 Overview of FHPP for motor controller CMMP-AS

1.1 Overview of Festo Handling and Positioning Profile (FHPP)

Festo has developed an optimised data profile especially tailored to the target applications for handling and positioning tasks, the “Festo Handling and Positioning Profile (FHPP)”.
The FHPP enables uniform control and parameterisation for the various fieldbus systems and controllers from Festo.
In addition, it defines for the user in a largely uniform way
– Operating modes,
– I/O data structure,
– parameter objects,
– sequence control.

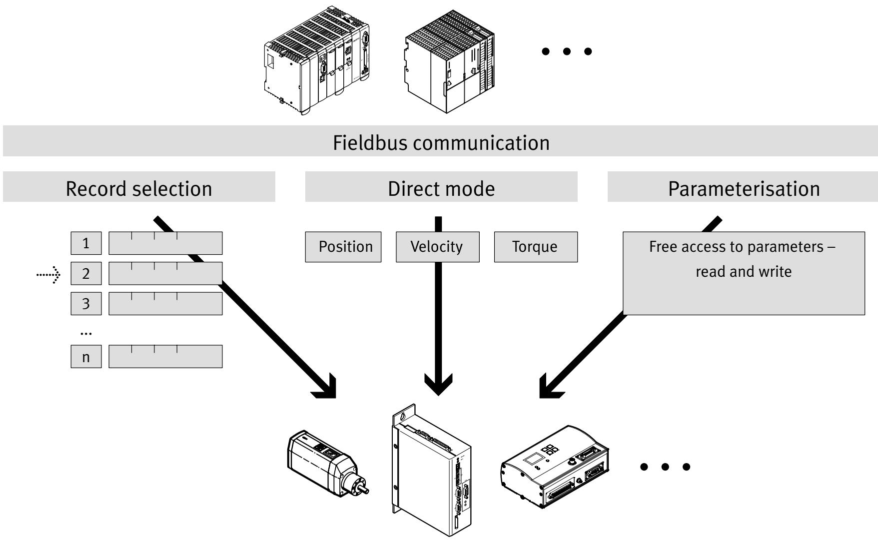


Fig. 1.1 Principle of FHPP

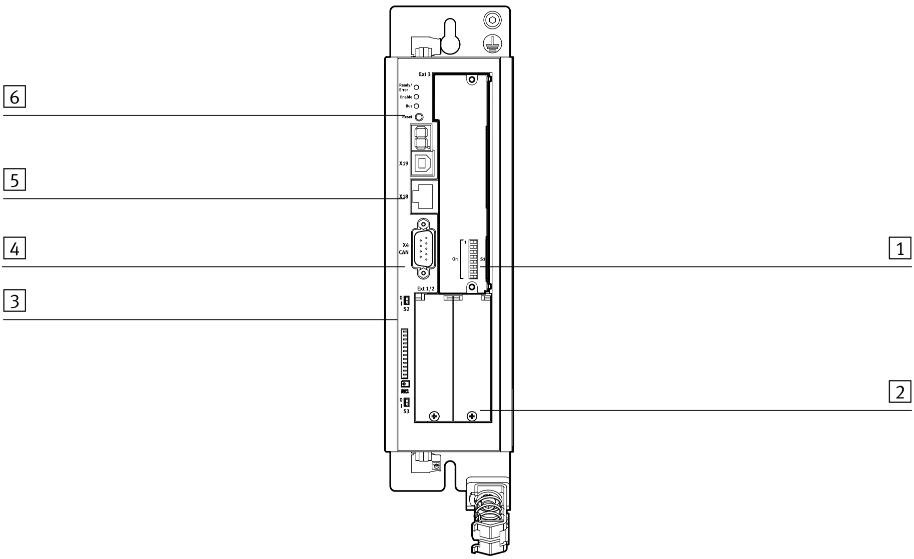
Control and status data (FHPP Standard)
Communication over the fieldbus is effected by way of 8-byte control and status data. Functions and status messages required in operation can be written and read directly.
Parameterisation (FPC)
The controller can access all parameter values of the controller via the fieldbus through the parameter channel. A further 8 bytes of I/O data are used for this purpose.
Parameterisation (FHPP+)
The I/O expansion FHPP+ allows additional PNUs configured by the user to be transmitted via the cyclic telegram in addition to the control and status bytes and the optional parameter channel (FPC).

1.2 Fieldbus interfaces

Control and parameterisation through FHPP is supported in the CMMP-AS-...-M3 through various field-bus interfaces conforming to Tab. 1.1. The CANopen interface is integrated into the motor controller; through interfaces, the motor controller can be extended with one of the following fieldbus interfaces. The fieldbus is configured with the DIP switches [S1].

Fieldbus	Interface	Slot	Description
CANopen	[X4] – integrated CANopen interface	–	➔ Chapter 2
Modbus TCP	[X18] – integrated Ethernet interface	–	➔ Chapter 3
PROFINET	Interface CAMC-F-PN	Ext2	➔ Chapter 4
PROFIBUS	Interface CAMC-PB	Ext2	➔ Chapter 5
EtherNet/IP	Interface CAMC-F-EP	Ext2	➔ Chapter 6
DeviceNet	Interface CAMC-DN	Ext1	➔ Chapter 7
EtherCAT	Interface CAMC-EC	Ext2	➔ Chapter 8

Tab. 1.1 Fieldbus interfaces for FHPP



1 DIP switches [S1] for fieldbus settings on the switch or safety module in slot Ext3
2 Slots Ext1/Ext2 for interfaces
3 CANopen terminating resistor [S2]
4 CANopen interface [X4]
5 CAN-LED

Fig. 1.2 Example of motor controller CMMP-AS-...-M3: Front view, with micro switch module in Ext3

The motor controllers CMMP-AS-...**-M0** are only equipped with the CANopen and Modbus TCP fieldbus interface and do not feature any slots for interfaces, switches or safety modules.

1.2.1 Mounting interface CAMC-...

M3

The CAMC-... interfaces are only available for the motor controllers CMMP-AS-...**-M3**.

Note
Before performing mounting and installation work, observe the safety instructions in the hardware description GDCP-CMMP-M3-HW-... and the accompanying assembly instructions.

The motor controllers CMMP-AS-...-M3 are shipped without interfaces in the slots Ext1 and Ext2; the slots are sealed with covers.
Through the interfaces, the motor controller can be extended by digital I/Os and/or fieldbus interfaces. Tab. 1.1 shows the permissible slots for the interfaces.

Mount interface
1. Unscrew screw [1].
2. Pry out cover [2] to the side. Use a small screwdriver.
3. Slide interface [3] into the guides.
4. Tighten screw [1]. Observe tightening torque 0.4 Nm ± 20 %.
   Result: Front plate has conducting contact with the housing.

Dismantle interface
1. Unscrew screw [1].
2. Pry out interface [2] to the side. Use a small screwdriver.
3. Pull interface [3] out of the slot.
4. Mount other interface or cover.

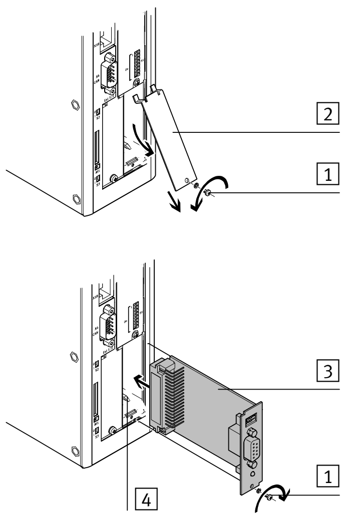


Fig. 1.3 Mounting or dismantling (example CAMC-PB)

2 CANopen with FHPP

2.1 Overview

This part of the documentation describes the connection and configuration of the motor controller CMMP-AS in a CANopen network. It is directed at people who are already familiar with this bus protocol.

CANopen is a standard worked out by the "CAN in Automation" association. Numerous device manufacturers are organised in this network. This standard has largely replaced the current manufacturer-specific CAN protocols. As a result, the end user has a non-proprietary communication interface.
The following manuals, among others, can be obtained from this association:

CiA 201 … 207:
These documents cover the general basic principles and embedding of CANopen into the OSI layered architecture. The relevant points of this book are presented in this CANopen manual, so procurement of DS201 … 207 is generally not necessary.

CiA 301:
This book describes the fundamental design of the object directory of a CANopen device and access to it. The statements of DS201 … 207 are also made concrete. The elements of the object directory needed for the CMMP motor controller families and the related access methods are described in this manual. Procurement of CiA 301 is recommended but not unconditionally necessary.
Source address: ➔ www.can-cia.org

2.2 CAN interface

The CAN interface is already integrated into the motor controller CMMP-AS and thus is always available. The CAN bus connection is designed as a 9-pin D-SUB plug in accordance with standards.

2.2.1 Connection and display components

The following components can be found on the front plate of the CMMP-AS:

– Status LED “CAN”
– a 9-pin D-SUB plug [X4]
– a DIP switch for activation of the terminating resistor.

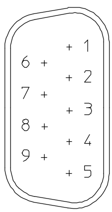
2.2.2 CAN LED

The LED CAN on the motor controller displays the following:

LED	Status
Off	No telegrams are sent
Flickers yellow	Acyclic communication (telegrams are send only when data change)
Lights up yellow	Cyclic communication (telegrams are sent permanently)

Tab. 2.1 CAN LED

2.2.3 Pin assignments of CAN-interface



[X4]	Pin no.		Designation	Value	Description
		1	-	-	Not assigned
	6		CAN-GND	-	Ground
		2	CAN-L	-	Negative CAN signal (dominant low)
	7		CAN-H	-	Positive CAN signal (dominant high)
		3	CAN-GND	-	Ground
	8		-	-	Not assigned
		4	-	-	Not assigned
	9		-	-	Not assigned
		5	CAN-Shield	-	Screening

Tab. 2.2 Pin assignment for CAN-interface

CAN bus cabling

When cabling the motor controller via the CAN bus, you should unconditionally observe the subsequent information and instructions to obtain a stable, trouble-free system.
If cabling is improperly done, malfunctions can occur on the CAN bus during operation. These can cause the motor controller to shut off with an error for safety reasons.

Termination

A terminating resistor (120 Ω) can, if required, be switched by means of DIP switch S2 = 1 (CAN Term) on the basic unit.

2.2.4 Cabling instructions

The CAN bus offers a simple, fail-safe ability to network all the components of a system together. But a requirement for this is that all of the following instructions on cabling are observed.

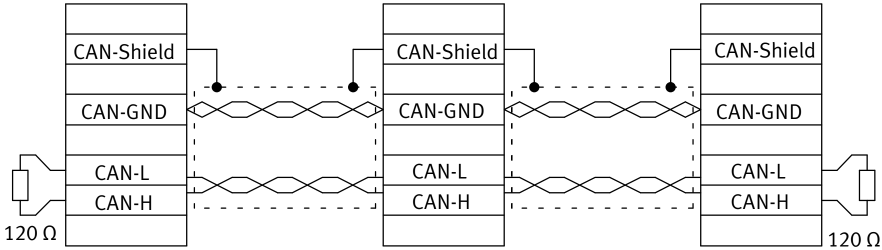


Fig. 2.1 Cabling example

- The individual nodes of the network are connected point-to-point to each other, so the CAN cable is looped from controller to controller (➔ Fig. 2.1).
- A terminating resistor of exactly 120 Ω +/-5 % must be available at both ends of the CAN cable. Such a terminating resistor is often already integrated into CAN cards or PLCs, which must be taken into account correspondingly.
- Screened cables with exactly two twisted conductor pairs must be used.
  One twisted conductor pair is used for connecting CAN-H and CAN-L. The conductors of the other pair are used together for CAN-GND. The cable screening is connected to the CAN-Shield connection at all nodes. (A table with the technical data of usable cables is located at the end of this chapter.)
- The use of adapters is not recommended for CAN bus cabling. If this is unavoidable, then metallic plug housings should be used to connect the cable screening.
- To keep the disturbance coupling as low as possible, motor cables should always be laid in accordance with the specification, not parallel to signal lines, and properly screened and earthed.
- For additional information on design of trouble-free CAN bus cabling, refer to the Controller Area Network protocol specification, Version 2.0 from Robert Bosch GmbH, 1991.

Characteristic		Value
Wire pairs	–	2
Wire cross section	[mm^2]	≥ 0.22
Screening	–	Yes
Loop resistance	[Ω / m]	< 0.2
Surge impedance	[Ω]	100 … 120

Tab. 2.3 Technical data, CAN bus cable

2.3 Configuration of CANopen stations on the CMMP-AS-...-M3

M3

This section is only applicable for the motor controller CMMP-AS-...**-M3**.

Several steps are required in order to produce an operational CANopen interface. Some of these settings should or must be carried out before the CANopen communication is activated. This section provides an overview of the steps required by the slave for parameterisation and configuration. As some parameters are only effective after saving and reset of the controller, we recommend that commissioning with the FCT should be carried out first without connection to the CANopen bus.

Instructions on commissioning with the Festo Configuration Tool can be found in the Help for the device-specific FCT plug-in.

When designing the CANopen interface, the user must therefore make these determinations. Only then should parameterisation of the fieldbus connection take place on both pages. We recommend that parameterisation of the slave should be undertaken first. Then the master should be configured.

We recommend the following procedure:

1. Setting of the offset of the node number, bit rate and activation of the bus communication via DIP switches.

   The status of the DIP switches is read once at Power- ON / RESET.
   The CMMP-AS takes over changes to the switch setting in ongoing operation only at the next RESET or restart

2. Parameterisation and commissioning with the Festo Configuration Tool (FCT).
   In particular on the Application Data page:
   – CANopen control interface (Mode Selection tab)

   In addition, the following settings on the fieldbus page:
   – Basic address of the node number
   – Festo FHPP protocol (Operation Parameters tab)
   – Physical units (Factor Group tab)
   – Optional use of FHPP+ (FHPP+ Editor tab)

Observe that the parameterisation of the CANopen function only remains intact after a reset if the parameter set of the motor controller was saved.
While the FCT device control is active, CAN communication is automatically deactivated.

3. Configuration of the CANopen master ➔ sections 2.5 and 2.6.

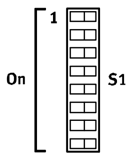
2.3.1 Setting of the node number with DIP switches and FCT

Each device in the network must be assigned a unique node number.
The node number can be set via the DIP switches 1 … 5 on the module in slot Ext3 and in the program FCT.

The resulting node number consists of the base address (FCT) and the offset (DIP switches).
Permissible values for the node number lie in the range 1 … 127.

Setting of the offset of the node number with DIP switches

The node number can be set with DIP switches 1 … 5. The offset of the node number set via DIP switches 1 … 5 is displayed in the program FCT on the Fieldbus page in the Operating Parameters tab.



DIP switch		**Value**			**Example**	
		ON	OFF			Value
	1	1	0		ON	1
	2	2	0		ON	2
	3	4	0		OFF	0
	4	8	0		ON	8
	5	16	0		ON	16
Sum of 1 … 5 = offset		1 … 31 [1]				**27**

1) The value 0 for the offset is interpreted in connection with a base address 0 as node number 1.
A node number larger than 31 must be set with the FCT.

Tab. 2.4 Setting of the offset of the node number

Setting the base address of the node number with FCT

With the Festo Configuration Tool (FCT), the node number is set as base address on the Fieldbus page in the Operating Parameters tab.
Default setting = 0 (that means offset = node number).

If a node number is assigned simultaneously via DIP switches 1...5 and in the FCT program, the resulting node number consists of the sum of the base address and the offset.
If this sum is greater than 127, the value is automatically limited to 127.

2.3.2 Setting of the transmission rate with DIP switches

The transmission rate must be set with DIP switches 6 and 7 on the module in slot Ext3. The status of the DIP switches is read one time at Power On/Reset. The CMMP-AS-...-M3 takes over changes to the switch setting in ongoing operation only at the next RESET.

Transmission rate		DIP switch 6	DIP switch 7
125	[Kbit/s]	OFF	OFF
250	[Kbit/s]	ON	OFF
500	[Kbit/s]	OFF	ON
1	[Mbps]	ON	ON

Tab. 2.5 Setting of the transmission rate

2.3.3 Activation of CANopen communication with DIP switches

When the node number und transmission rate have been set, CANopen communication can be activated. Please note that the above-mentioned parameters can only be revised when the protocol is deactivated.

CANopen communication	DIP switch 8
Disabled	OFF
Enabled	ON

Tab. 2.6 Activation of CANopen communication

Please observe that CANopen communication can only be activated after the parameter set (the FCT project) has been saved and a Reset carried out.

If another fieldbus interface is plugged into Ext1 or Ext2 (➔ section 1.2), CANopen communication is activated with DIP switch 8 instead of via [X4] of the corresponding fieldbus.

2.3.4 Setting the physical units (factor group)

In order for a fieldbus master to exchange position, velocity and acceleration data in physical units (e.g. mm, mm/s, mm/s^2) with the motor controller, it must be parameterised via the factor group ➔ section A.1.

Parameterisation can be carried out via FCT or the fieldbus.

2.3.5 Setting for optional use of FHPP+

Besides the control or status bytes and the FPC, additional I/O data can also be transmitted ➔ section C.2.

This is set via the FCT (Fieldbus panel, tab FHPP+ Editor).

2.4 Configuration of CANopen stations on the CMMP-AS-...-M0

M0

This section is only applicable for the motor controller CMMP-AS-...**-M0**.

Several steps are required in order to produce an operational CANopen interface. Some of these settings should or must be carried out before the CANopen communication is activated. This section provides an overview of the steps required by the slave for parametrisation and configuration.

Instructions on commissioning with the Festo Configuration Tool can be found in the Help for the device-specific FCT plug-in.

When designing the CANopen interface, the user must therefore make these determinations. Only then should parametrisation of the fieldbus connection take place on both sides. We recommend that parameterisation of the slave should be executed first. Then the master should be configured.

The CAN bus-specific parameters can be set in two ways. These ways are separated from one another and are accessed via the option “Fieldbus parameterisation via DINs” on the “Application data” panel in the FCT.
The option “Fieldbus parameterisation via DINs” is active in a delivery status and after a reset to the factory settings. Parameterisation with FCT for activation of the CAN bus is thus not necessary.

The following parameters can be set via the DINs or FCT:

<table>
<tr><th rowspan="2">Parameters</th><th colspan="2">Setting via</th></tr>
<tr><th>DIN</th><th>FCT</th></tr>
<tr><td>Node number</td><td>0...3[1]</td><td rowspan="4">“Fieldbus” panel, operating parameters.
Activation of the CAN bus is performed automatically by FCT (dependent on device control):
– Device control by FCT → CAN deactivated
– Device control released → CAN activated</td></tr>
<tr><td>Transmission rate (bit rate)</td><td>12, 13[1]</td></tr>
<tr><td>Input/activation</td><td>8</td></tr>
<tr><td>Protocol (data profile)</td><td>9[2]</td></tr>
</table>

1) Only transferred in the event of inactive CAN communication
2) Only transferred after a device RESET

Tab. 2.7 Overview of settings for CAN parameters via DINs or FCT

2.4.1 Setting the node number via DINs and FCT

Each device in the network must be assigned a unique node number.
The node number can be set via the digital inputs DIN0 .... DIN3 **and** in the FCT programme.

Permissible values for the node number lie in the range 1...127.

Setting the offset of the node number via DINs

The node number can be set via the circuitry of the digital inputs DIN0 .... DIN3. The offset of the node number set via the digital inputs is displayed in the FCT programme on the “Fieldbus” panel in the “Operating parameters” tab.

DINs	Value			Example	
	High	Low			Value
0	1	0		High	1
1	2	0		High	2
2	4	0		Low	0
3	8	0		High	8
Total 0...3 = node number 0...15					**11**

Tab. 2.8 Setting the node number

Setting the base address of the node number via FCT

The base address of the node number can be set via FCT on the “Fieldbus” panel in the “Operating parameters” tab.
The resulting node number is dependent on the option “Fieldbus parameterisation via DINs” on the “Application data” panel. If this option is activated, the node number is determined by adding the base address in the FCT to the offset via the digital inputs DIN0...3.
If the option is deactivated, the base address in the FCT corresponds to the resulting node number.

2.4.2 Setting the transmission rate via DINs or FCT

The transmission rate can be set via the digital inputs DIN12 and DIN13 **or** in the FCT.

Setting the transmission rate via DINs

Transmission rate		DIN 12	DIN 13
125	[Kbit/s]	Low	Low
250	[Kbit/s]	High	Low
500	[Kbit/s]	Low	High
1	[Mbps]	High	High

Tab. 2.9 Setting the transmission rate

Setting the transmission rate via FCT

The transmission rate can be set via FCT on the “Fieldbus” panel in the “Operating parameters” tab. The option “Fieldbus parameterisation via DINs” must be deactivated beforehand on the “Application data” panel. When this option is deactivated the inputs automatically become active again as DIN12 and DIN13.

2.4.3 Setting the protocol (data profile) via DINs or FCT

The protocol (data profile) can be set via the digital input DIN9 **or** the FCT.

Setting the protocol (data profile) via DINs

Protocol (data profile)	DIN 9
CiA 402 (DS 402)	Low
FHPP	High

Tab. 2.10 Activating the protocol (data profile)

Setting the protocol (data profile) via FCT

The protocol is set via FCT on the “Fieldbus” panel in the “Operating parameters” tab.

2.4.4 Activation of CANopen communication via DINs or FCT

When the node number, transmission rate and protocol (data profile) have been set, CANopen communication can be activated.

Activation of CANopen communication via DIN

CANopen communication	DIN 8
Deactivated	Low
Activated	High

Tab. 2.11 Activation of CANopen communication

The device does not need to be reset again for activation via digital input. The CAN bus is activated immediately after a level change (Low → High) at DIN8.

Activation of CANopen communication via FCT

CANopen communication is automatically activated by the FCT if the option “Fieldbus parameterisation via DINs” is deactivated.

The CAN bus is switched off for as long as the device control remains with FCT.

2.4.5 Setting the physical units (factor group)

In order for a fieldbus master to exchange position, velocity and acceleration data in physical units (e.g. mm, mm/s, mm/s^2) with the motor controller, it must be parameterised via the factor group
➔ section A.1.
Parameterisation can be carried out via FCT or the fieldbus.

2.4.6 Setting for optional use of FHPP+

Besides the control or status bytes and the FPC, additional I/O data can also be transmitted
➔ section C.2.
This is set via the FCT (Fieldbus panel, tab FHPP+ Editor).

2.5 Configuration CANopen master

You can use an EDS file to configure the CANopen master.
The EDS file is included on the CD-ROM supplied with the motor controller.

You will find the most current version under ➔ www.festo.com/sp

Electronic data sheet (EDS) files	Description
CMMP-AS-...-M3_FHPP.eds	Motor controller CMMP-AS-...**-M3** with protocol "FHPP"
CMMP-AS-...-M0_FHPP.eds	Motor controller CMMP-AS-...-**M0** with protocol "FHPP"

Tab. 2.12 EDS files for FHPP with CANopen

To simplify commissioning of the CMMP-AS-...-M3/-M0 with CODESYS controllers from various manufacturers, you will find corresponding modules and application notes at
➔ www.festo.com/sp

2.6 Access procedure

2.6.1 Introduction

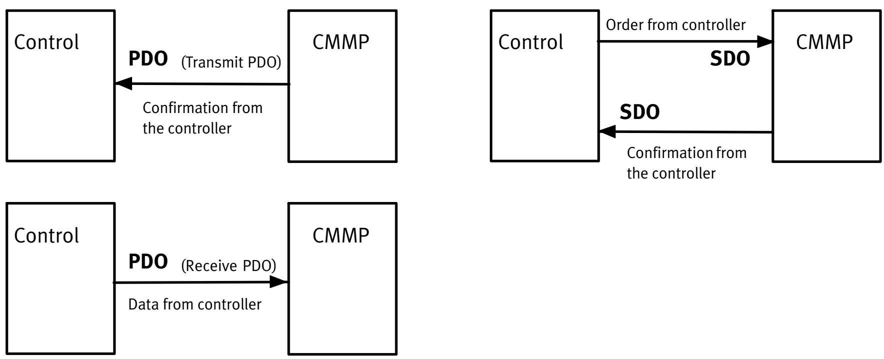
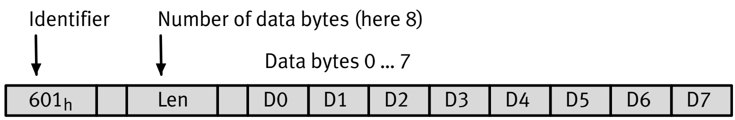


Fig. 2.2 Access procedure PDO and SDO

Overview of communication objects		
PDO	Process Data Object.	The FHPP I/O data are transferred in the PDOs ➔ chapter 9. Mapping is automatically determined in parameterisation with FCT ➔ section 2.6.2.
SDO	Service Data Object	Parallel to the FHPP I/O data, parameters can be transferred via SDOs corresponding to CiA 402.
SYNC	Synchronisation Message	Synchronisation of multiple CAN nodes
EMCY	Emergency Message	Transmission of error messages
NMT	Network management	Network service: All CAN nodes can be worked on simultaneously, for example.
HEART-BEAT	Error Control Protocol	Monitoring of the communications participants through regular messages.

Tab. 2.13 Communication objects

Every message sent on the CAN bus contains a type of address which is used to determine the bus participant for which the message is meant and from which bus participant the message is sent. This number is designated the identifier. The lower the identifier, the greater the priority of the message. Identifiers are established for the above-named communication objects ➔ section 2.6.10. The following sketch shows the basic design of a CANopen message:



2.6.2 PDO Message

A distinction is made between the following types of PDOs:

Type	Path	Remark
Transmit PDO	Motor controller ➔ Host	Motor controller sends PDO when a certain event occurs.
Receive PDO	Host ➔ motor controller	Motor controller evaluates PDO when a certain event occurs.

Tab. 2.14 PDO types

The FHPP I/O data are divided among several process data objects for CANopen communication. This assignment is established through the parameterisation during commissioning with the FCT. The mapping is thereby automatically created.

Supported process data objects	Data mapping of the FHPP data
TxPDO 1	FHPP Standard 8 bytes status data
TxPDO 2	FPC parameter channel Transmission of requested FHPP parameter values
TxPDO 3 (optional)	FHPP+ data[1)] Mapping = 8 bytes of FHPP+ data
TxPDO 4 (optional)	FHPP+ data[1)] Mapping = 8 bytes of FHPP+ data
RxPDO 1	FHPP Standard 8 byte control data
RxPDO 2	FPC parameter channel Read/write FHPP parameter values
RxPDO 3 (optional)	FHPP+ data[1)] Mapping = 8 bytes of FHPP+ data
RxPDO 4 (optional)	FHPP+ data[1)] Mapping = 8 bytes of FHPP+ data

1) Optional if parameterised through the FCT (page Fieldbus – tab FHPP+ Editor)

Tab. 2.15 Overview of supported PDOs

i You can find the allocation of the FHPP I/O data in ➔ chapter 9.

2.6.3 SDO Access

Through the service data objects (SDO), the CiA 402 object directory of the motor controller can be accessed.

> **i** Observe that the contents of FHPP parameters (PNUs) can differ from the CiA objects. In addition, not all objects are available in an active FHPP protocol.
>
> You will find documentation of the objects in the ➔ description CiA 402.

SDO access always starts from the higher-order controller (Host). This either sends the motor controller a write command to modify a parameter in the object directory or a read command to read a parameter. For each command, the host receives an answer that either contains the read-out value or – in the case of a write command – serves as an acknowledgement.
For the motor controller to recognise that the command is meant for it, the host must send the command with a specific identifier. This identifier is made up of the base 600_h + node number of the motor controller. The motor controller answers with the identifier 580_h + node number.
The design of the commands or answers depends on the data type of the object to be read or written, since either 1, 2 or 4 data bytes must be sent or received.

SDO Sequences for Reading and Writing

To read out or describe objects of these number types, the following listed sequences are used. The commands for writing a value into the motor controller begin with a different identifier, depending on the data type. The answer identifier, in contrast, is always the same. Read commands always start with the same identifier, and the motor controller answers differently, depending on the data type returned.

Identifier	8 bits	16 bits	32 bits
Task identifier	$2F_h$	$2B_h$	23_h
Response identifier	$4F_h$	$4B_h$	43_h
Response identifier in case of error	–	–	80_h

Tab. 2.16 SDO – response/task identifier

EXAMPLE

UINT8/INT8	Reading of Obj. 6061_00_h Return data: 01_h	Writing of Obj. 1401_02_h Data: EF_h
Command	40_h 61_h 60_h 00_h	$2F_h$ 01_h 14_h 02_h EF_h
Answer:	$4F_h$ 61_h 60_h 00_h 01_h	60_h 01_h 14_h 02_h
UINT16/INT16	Reading of Obj. 6041_00_h Return data: 1234_h	Writing of Obj. 6040_00_h Data: $03E8_h$
Command	40_h 41_h 60_h 00_h	$2B_h$ 40_h 60_h 00_h $E8_h$ 03_h
Answer:	4Bh 41_h 60_h 00_h 34_h 12_h	60_h 40_h 60_h 00_h
UINT32/INT32	Reading of Obj. 6093_01_h Return data: 12345678_h	Writing of Obj. 6093_01_h Data: 12345678_h
Command	40_h 93_h 60_h 01_h	23_h 93_h 60_h 01_h 78_h 56_h 34_h 12_h
Answer:	43_h 93_h 60_h 01_h 78_h 56_h 34_h 12_h	60_h 93_h 60_h 01_h

Note
The acknowledgement from the motor controller must always be waited for!
Only when the motor controller has acknowledged the request may additional requests be sent.

SDO Error Messages

In case of an error when reading or writing (for example, because the written value is too large), the motor controller answers with an error message instead of the acknowledgement:

Command	23_h	41_h	60_h	00_h	...	...	...	...
Answer:	80_h	41_h	60_h	00_h	02_h	00_h	01_h	06_h
	↑				↑	↑	↑	↑
	Error identifier				Error code (4 byte)			

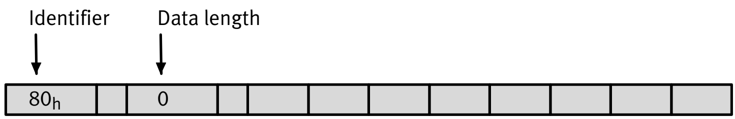
Error code	Significance
05 03 00 00_h	Protocol error: Toggle bit was not revised
05 04 00 01_h	Protocol error: Client / server command specifier invalid or unknown
06 06 00 00_h	Access faulty due to a hardware problem[1]
06 01 00 00_h	Access type is not supported.
06 01 00 01_h	Read access to an object that can only be written
06 01 00 02_h	Write access to an object that can only be read
06 02 00 00_h	The addressed object does not exist in the object directory
06 04 00 41_h	The object must not be entered into a PDO (e.g. ro-object in RPDO)
06 04 00 42_h	The length of the objects entered in the PDO exceeds the PDO length
06 04 00 43_h	General parameter error
06 04 00 47_h	Overflow of an internal variable / general error
06 07 00 10_h	Protocol error: Length of the service parameter does not agree
06 07 00 12_h	Protocol error: Length of the service parameter is too large
06 07 00 13_h	Protocol error: Length of the service parameter is too small
06 09 00 11_h	The addressed subindex does not exist
06 09 00 30_h	The data exceed the range of values of the object
06 09 00 31_h	The data are too large for the object
06 09 00 32_h	The data are too small for the object
06 09 00 36_h	Upper limit is less than lower limit
08 00 00 20_h	Data cannot be transmitted or stored[1]
08 00 00 21_h	Data cannot be transmitted/stored; motor controller is working locally
08 00 00 22_h	Data cannot be transmitted/stored, since the motor controller is not in the correct status for this[2]
08 00 00 23_h	There is no object dictionary available[3]

1) Returned in accordance with CiA 301 in case of incorrect access to store_parameters / restore_parameters.
2) “Status” here generally: for example, incorrect operating mode, module not on hand, or the like.
3) Returned, for example, if another bus system controls the motor controller or the parameter access is not permitted.

Tab. 2.17 Error codes SDO access

2.6.4 SYNC message

Several devices of a system can be synchronised with each other. To do this, one of the devices (usually the higher-order controller) periodically sends out synchronisation messages. All connected controllers receive these messages and use them for treatment of the PDOs (➔ chapter 2.6.2).



The identifier on which the motor controller receives the SYNC message is set permanently to 080_h. The identifier can be read via the object cob_id_sync.

Index	**1005_h**
Name	**cob_id_sync**
Object Code	VAR
Data Type	UINT32

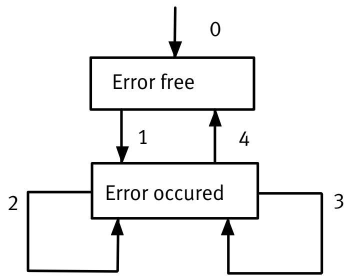
Access	rw
PDO mapping	no
Units	--
Value Range	80000080_h, 00000080_h
Default Value	00000080_h

2.6.5 EMERGENCY Message

The motor controller monitors the function of its major assemblies. These include the power supply, output stage, angle encoder evaluation, etc. In addition, the motor (temperature, angle encoder) and limit switch are also checked. Incorrect parameter setting can also result in error messages (division by zero, etc.).
When an error occurs, the error number is shown in the motor controller's display. If several error messages occur simultaneously, the message with the highest priority (lowest number) is always shown in the display.

Overview

When an error occurs or an error acknowledgment is carried out, the controller transmits an EMERGENCY message. The identifier of this message is made up of the identifier 80_h and the node number of the relevant controller.



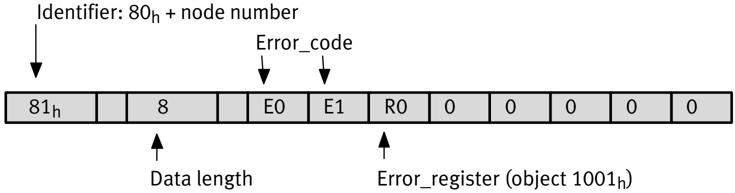
After a reset, the controller is in the status Error free (which it might leave again immediately, because an error is on hand from the beginning). The following status transitions are possible:

No.	Cause	Significance
0	Initialisation completed	
1	Error occurs	No error is present and an error occurs. An EMERGENCY telegram with the error code of the occurring error is sent.
2	Error acknowledgment	An error acknowledgment is attempted, but not all causes have been eliminated.
3	Error occurs	An error is present and an additional error occurs. An EMERGENCY telegram with the error code of the new error is sent.
4	Error acknowledgment	An error acknowledgment is attempted, and all causes are eliminated. An EMERGENCY telegram with the error code 0000 is sent.

Tab. 2.18 Possible status transitions

Structure of the EMERGENCY Message

When an error occurs, the motor controller transmits an EMERGENCY message. The identifier of this message is made up of the identifier 80_h and the node number of the relevant motor controller.
The EMERGENCY message consists of eight data bytes, whereby the first two bytes contain an error_code ➔ D.1, Tab. D.1. An additional error code is in the third byte (object 1001_h). The remaining five bytes contain zeros.



error_register (R0)		
Bit	M/O[1)]	Significance
0	M	generic error: Error is present (Or-link of the bits 1 … 7)
1	O	current: I^2t error
2	O	voltage: voltage monitoring error
3	O	temperature: motor overtemperature
4	O	communication error: (overrun, error state)
5	O	–
6	O	reserved, fix = 0
7	O	reserved, fix = 0
Values: 0 = no error; 1 = error present		

1) M = required / O =

Tab. 2.19 Bit assignment error_register

The error codes as well as the cause and remedial measures can be found in ➔ section D.

Description of the objects

Object 1003_h: pre_defined_error_field

The respective error_code of the error messages is also stored in a four-stage error memory. This is structured like a shift register, so that the last occurring error is always stored in the object 1003_h_01_h (standard_error_field_0). Through read access on the object 1003_h_00_h (pre_defined_error_field_0), it can be determined how many error messages are currently stored in the error memory. The error memory is cleared by writing the value 00_h into the object 1003_h_00_h (pre_defined_error_field_0). To be able to reactivate the output stage of the motor controller after an error, an error acknowledgement must also be performed.

Index	**1003_h**
Name	**pre_defined_error_field**
Object Code	ARRAY
No. of Elements	4
Data Type	UINT32

Sub-Index	**01_h**
Description	**standard_error_field_0**
Access	ro
PDO mapping	no
Units	–
Value Range	–
Default Value	–

Sub-Index	**02_h**
Description	**standard_error_field_1**
Access	ro
PDO mapping	no
Units	–
Value Range	–
Default Value	–

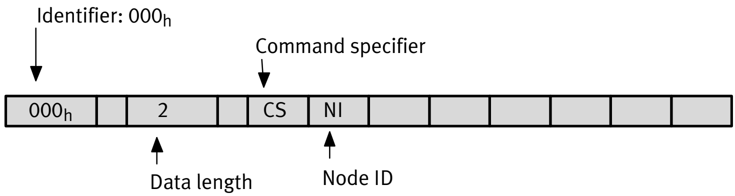
Sub-Index	**03_h**
Description	**standard_error_field_2**
Access	ro
PDO mapping	no
Units	–
Value Range	–
Default Value	–

Sub-Index	**04_h**
Description	**standard_error_field_3**
Access	ro
PDO mapping	no
Units	–
Value Range	–
Default Value	–

2.6.6 Network Management (NMT Service)

All CANopen equipment can be triggered via the Network Management. Reserved for this is the identifier with the top priority (000_h). By means of NMT, commands can be sent to one or all controllers. Each command consists of two bytes, whereby the first byte contains the command code (command specifier, CS) and the second byte the node address (node id, NI) of the addressed controller. Through the node id zero, all nodes in the network can be addressed simultaneously. It is thus possible, for example, that a reset is triggered in all devices simultaneously. The controllers do not acknowledge the NMT commands. Successful completion can only be determined indirectly (e.g. through the switch-on message after a reset).

Structure of the NMT Message:



For the NMT status of the CANopen node, statuses are established in a status diagram. Changes in statuses can be triggered via the CS byte in the NMT message. These are largely oriented on the target status.

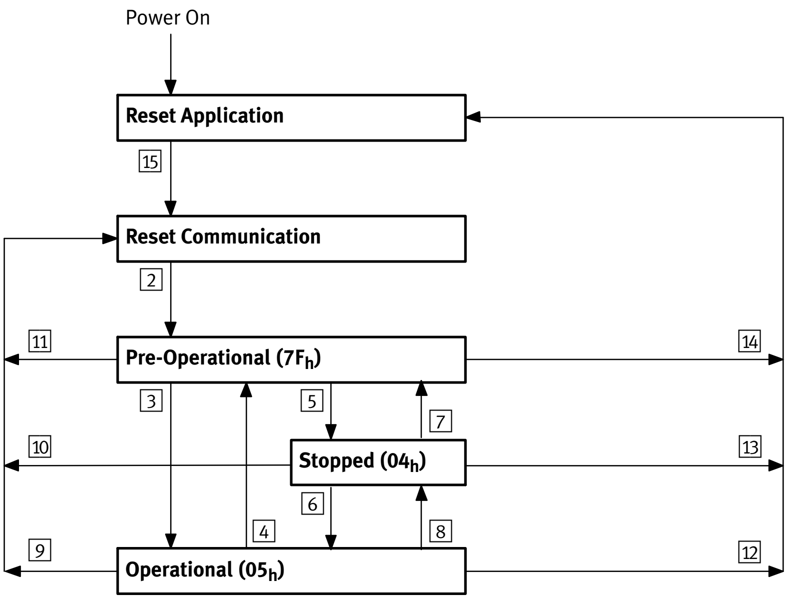


Fig. 2.3 Status diagram

Transition	Significance	CS	Target status	
2	Bootup	--	Pre-Operational	$7F_h$
3	Start Remote Node	01_h	Operational	05_h
4	Enter Pre-Operational	80_h	Pre-Operational	$7F_h$
5	Stop Remote Node	02_h	Stopped	04_h
6	Start Remote Node	01_h	Operational	05_h
7	Enter Pre-Operational	80_h	Pre-Operational	$7F_h$
8	Stop Remote Node	02_h	Stopped	04_h
9	Reset Communication	82_h	Reset Communication 1)	
10	Reset Communication	82_h	Reset Communication 1)	
11	Reset Communication	82_h	Reset Communication 1)	
12	Reset Application	81_h	Reset Application 1)	
13	Reset Application	81_h	Reset Application 1)	
14	Reset Application	81_h	Reset Application 1)	

1) The final target status is pre-operational ($7F_h$), since the transitions 15 and 2 are automatically performed by the controller.

Tab. 2.20 NMT state machine

All other status transitions are performed automatically by the controller, e.g. because the initialisation is completed.
In the NI parameter, the node number of the controller must be specified, or zero if all nodes in the network are to be addressed (broadcast). Depending on the NMT status, certain communication objects cannot be used: For example, it is absolutely necessary to place the NMT status to operational so that the controller sends PDOs.

Name	Significance	SDO	PDO	NMT
Reset Application	No Communication. All CAN objects are reset to their reset values (application parameter set)	–	–	–
Reset Communication	No communication: The CAN controller is newly initialised.	–	–	–
Initialising	Status after hardware reset. Resetting of the CAN node, Sending of the bootup message	–	–	–
Pre-Operational	Communication via SDOs possible; PDOs not active (no sending/evaluating)	X	–	X
Operational	Communication via SDOs possible; all PDOs active (sending/evaluating)	X	X	X
Stopped	No communication except for heartbeating	–	–	X

Tab. 2.21 NMT state machine

NMT telegrams must not be sent in a burst (one immediately after another)!
At least twice the position controller cycle time must lie between two consecutive NMT messages on the bus (also for different nodes!) for the controller to process the NMT messages correctly.

If necessary, the NMT command “Reset Application” is delayed until an ongoing saving procedure is completed, since otherwise the saving procedure would remain incomplete (defective parameter set).
The delay can be in the range of a few seconds.

The communication status must be set to operational for the controller to transmit and receive PDOs.

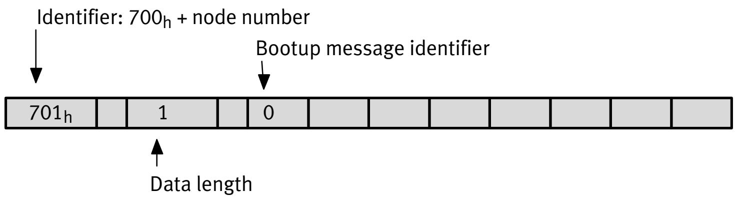
2.6.7 Bootup

Overview

After the power supply is switched on or after a reset, the controller reports via a Bootup message that the initialisation phase is ended. The controller is then in the NMT status preoperational
(➔ chapter 2.6.6, Network Management (NMT Service))

Structure of the Bootup Message

The Bootup message is structured almost identically to the following Heartbeat message.
Only a zero is sent instead of the NMT status.



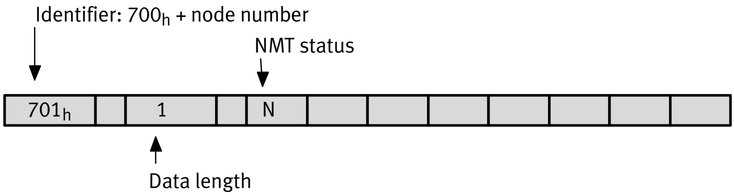
2.6.8 Heartbeat (Error Control Protocol)

Overview

The so-called Heartbeat protocol can be activated to monitor communication between slave (drive) and master: Here, the drive sends messages cyclically to the master. The master can check whether these messages occur cyclically and introduce corresponding measures if they do not. Since both Heartbeat and Nodeguarding telegrams (➔ chap. 2.6.9) are sent with the identifier 700h + node number, both protocols can be active at the same time. If both protocols are activated simultaneously, only the Heartbeat protocol is active.

Structure of the Heartbeat Message

The Heartbeat telegram is transmitted with the identifier 700_h + node number. It contains only 1 byte of user data, the NMT status of the controller (➔ chapter 2.6.6, Network Management (NMT Service)).



N	Significance
04_h	Stopped
05_h	Operational
$7F_h$	Pre-Operational

Description of the objects

Object 1017_h: producer_heartbeat_time

To activate the Heartbeat function, the time between two Heartbeat telegrams can be established via the object producer_heartbeat_time.

Index	**1017_h**
Name	**producer_heartbeat_time**
Object Code	VAR
Data Type	UINT16

Access	rw
PDO	no
Units	ms
Value Range	0 ... 65535
Default Value	0

The producer_heartbeat_time can be stored in the parameter record. If the controller starts with a producer_heartbeat_time not equal to zero, the bootup message is considered to be the first Heartbeat.
The controller can only be used as a so-called Heartbeat producer. The object 1016_h (consumer_heartbeat_time) is therefore implemented only for compatibility reasons and always returns 0.

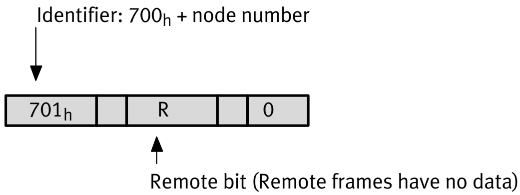
2.6.9 Nodeguarding (Error Control Protocol)

Overview

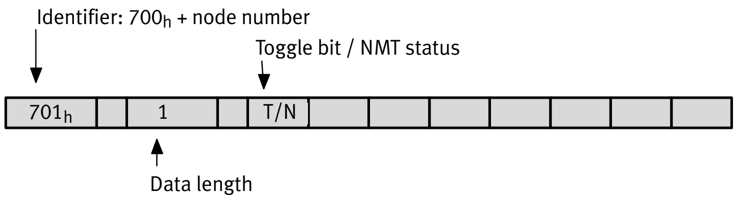
The so-called Nodeguarding protocol can also be used to monitor communication between slave (drive) and master. In contrast to the Heartbeat protocol, master and slave monitor each other: The master queries the drive cyclically about its NMT status. In every response of the controller, a specific bit is inverted (toggled). If these responses are not made or the controller always responds with the same toggle bit, the master can react correspondingly. Likewise, the drive monitors the regular arrival of the Nodeguarding requests from the master: If messages are not received for a certain time period, the controller triggers error 12-4. Since both Heartbeat and Nodeguarding telegrams (➔ chapter 2.6.8) are sent with the identifier 700_h + node number, both protocols cannot be active simultaneously. If both protocols are activated simultaneously, only the Heartbeat protocol is active.

Structure of the Nodeguarding Messages

The master's request must be sent as a so-called remote frame with the identifier 700_h + node number. In the case of a remote frame, a special bit is also set in the telegram, the remote bit. Remote frames have no data.



The response of the controller is built up analogously to the Heartbeat message. It contains only 1 byte of user data, the toggle bit and the NMT status of the controller (➔ chapter 2.6.6).



The first data byte (T/N) is constructed in the following way:

Bit	Value	Name	Significance
7	80_h	toggle_bit	Changes with every telegram
0 … 6	$7F_h$	nmt_state	04_h Stopped 05_h Operational $7F_h$ Pre-Operational

The monitoring time for the master's requests can be parameterised. Monitoring begins with the first received remote request of the master. From this time on, the remote requests must arrive before the monitoring time has passed, since otherwise error 12-4 is triggered.
The toggle bit is reset through the NMT command Reset Communication. It is therefore deleted in the first response of the controller.

Description of the objects

Object $100C_h$: guard_time

To activate the Nodeguarding monitoring, the maximum time between two remote requests of the master is parameterised. This time is established in the controller from the product of guard_time ($100C_h$) and life_time_factor ($100D_h$). It is therefore recommended to write the life_time_factor with 1 and then specify the time directly via the guard_time in milliseconds.

Index	**$100C_h$**
Name	**guard_time**
Object Code	VAR
Data Type	UINT16

Access	rw
PDO mapping	no
Units	ms
Value Range	0 … 65535
Default Value	0

Object $100D_h$: life_time_factor

The life_time_factor should be written with 1 in order to specify the guard_time directly.

Index	**$100D_h$**
Name	**life_time_factor**
Object Code	VAR
Data Type	UINT8

Access	rw
PDO mapping	no
Units	–
Value Range	0.1
Default Value	0

2.6.10 Table of Identifiers

The following table gives an overview of the identifiers used:

Object type	Identifier (hexadecimal)	Remark
SDO (Host to controller)	600_h + node number	
SDO (Controller to host)	580_h + node number	
TPDO1	180_h + node number	Standard values. Can be revised if needed.
TPDO2	280_h + node number	
TPDO3	380_h + node number	
TPDO4	480_h + node number	
RPDO1	200_h + node number	
RPDO2	300_h + node number	
RPDO3	400_h + node number	
RPDO4	500_h + node number	
SYNC	080_h	
EMCY	080_h + node number	
HEARTBEAT	700_h + node number	
NODEGUARDING	700_h + node number	
BOOTUP	700_h + node number	
NMT	000_h	

3 Modbus TCP with FHPP

Requirement: Modbus TCP is supported in CMMP-AS-...-M3 and CMMO-AS-...-M0 from Firmware Version: 4.0.1501.2.1 and FCT PlugIn 2.3.0.

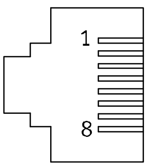
3.1 Overview

This part of the documentation describes connection and configuration of the motor controller within a Modbus network. It is targeted at people who are already familiar with this bus protocol.
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.

3.2 Modbus-TCP interface

Modbus connection is established via the integrated interface [X18] included with the basic device as an RJ45 socket. This can be used simultaneously with the 2 UDP connections (for FCT parameterisation software). As a Modbus participant, the motor controller can be reached via the same IP address as is also used by the FCT.

3.2.1 Pin allocation and cable specifications

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

Tab. 3.1 Assignment [X18]

Type and design of the cable
Shielded twisted-pair STP, Cat.5 cables must be used for cabling.

3.3 Configuration of Modbus participant

Several steps are required in order to establish an operational Modbus interface. This section provides an overview of the steps required for parameterisation and configuration of the slave. As some parameters are only effective after saving and reset, we recommend that commissioning with the FCT be carried out first without connection to the Modbus TCP.

> **i** Notes on commissioning with the Festo Configuration Tool can be found in the Help for the device-specific FCT plug-in.

When designing the Modbus/TCP interface, you must make these determinations. Only then should parameterisation of the fieldbus interface take place at both ends. We recommend that the slave parameters should be set first. The master should be configured thereafter. With correct parameterisation, the application is ready immediately without communication faults.

We recommend the following procedure:

1. Deactivation of the CAN interface (CMMP-AS-...-M3 via DIL switches, CMMP-AS-...-M0 via FCT).

> **i** The status of the DIL switches is read one time at Power On / Reset.
> The CMMP-AS-...-M3 takes over changes to the switch setting in ongoing operation only at the next RESET or restart

2. Parameterisation and commissioning with the Festo Configuration Tool (FCT).
   On the “Application data” page in the “Operating mode selection” tab:
   – Select “Modbus/TCP” as the control interface (activation of communication)
   Enter the following settings on the “Fieldbus” page as well:
   – TCP port (“Operating parameters” tab)
   – Timeout (“Operating parameters” tab)
   – Physical units of measure (“Factor group” tab)
   – Optional use of FHPP+ (“FHPP+ editor” tab)

> **i** Note that parameterisation of the Modbus/TCP function only remains intact after a reset if the motor controller’s parameter set has been saved.

3. Configuration of the Modbus master ➔ section 3.4.

3.3.1 Deactivation of CANopen communication with DIL switches

All DIL switches on the module in slot [Ext 3] must be set to OFF, because otherwise the CAN Bus would be activated with corresponding settings.

3.3.2 Activation of Modbus TCP

To activate, select “Modbus TCP” as the control interface on the Application Data page in the “Operating Mode Settings” tab.

3.3.3 TCP port setting and Timeout

If necessary, you can set the TCP port and the communication “Timeout” value in FCT on the “Fieldbus” page in the “Operating parameters” tab.
Presetting in the FCT:
- TCP port 502 (Standard port for Modbus TCP/IP)
- Timeout 2000 ms (connection timeout, to detect an interruption of the Modbus and change to a corresponding status).

3.3.4 Setting of the physical units of measure (factor group)

In order for a master to exchange position, speed and acceleration data in physical units (e.g. mm, mm/s, mm/s^2) with the motor controller, it must be parameterised via the factor group ➔ section A.1. Parameterisation can be carried out via either FCT or the fieldbus.

3.3.5 Setting the optional use of FPC and FHPP+

Besides the control or status bytes and the FPC, additional I/O data can be transmitted ➔ section C.2. This is set via the FCT (page Fieldbus, tab FHPP+ Editor).

3.4 Modbus master configuration

The IP address of the motor controller as a Modbus/TCP participant is identical to the FCT interface address set in the FCT.

3.4.1 Address assignment and Modbus commands

The start address is always “0”; the byte sequence is always “Big endian”.
Tab. 3.2 shows the supported Modbus commands.

Modbus command	Function code	Significance
read holding registers	3	Read the process data
Write multiple registers	16	Write the process data
Read/write multiple registers	23	From FW 4.0.1501.2.3: combined reading/writing of the process data
Read device identification	43	See ➔ section 3.4.2.

Tab. 3.2 Overview of Modbus function codes

3.4.2 Data objects

Tab. 3.3 shows the supported data objects.

Object ID		Object Name	Value
Basic	0x00	VendorName	“Festo AG & Co KG”
	0x01	ProductCode	Controller-specific (e.g. “0x00002045”)
	0x02	MajorMinorRevision	Firmware-specific (e. B. “004.000.101501.001.004”)
Regular	0x03	VendorURL	“www.festo.com”
	0x04	ProductName	Controller-specific (e.g. “CMMP-AS-C5-3A-M3”)
	0x05	ModelName	“ ” (space)
	0x06	UserApplicationName	Name of the component in the FCT project

Tab. 3.3 Data objects

3.4.3 Monitoring functions

The motor controller supports TCP/IP connection monitoring, and timeout duration is adjustable
➔ section 3.3.3.
In the event of a timeout, error message E67-0 is generated – the error response for error group 67 can be parameterised (“Error management page” in FCT).

Node guard monitoring is not supported.

The CMMP-AS always sends its user data in segmented Ethernet frames. The first segment thereby includes N-1 bytes of user data, the second segment 1 byte of user data. In addition, the user data are filled to the 16 bit limit with padding (zero) bytes.

4 PROFINET-IO with FHPP

M3

This chapter is only applicable for the motor controller CMMP-AS-...**-M3**.

4.1 Overview

This part of the documentation describes the connection and configuration of the motor controller CMMP-AS-...-M3 in a PROFINET IO network. It is directed at people who are already familiar with this bus protocol.

PROFINET (**PRO**cess **Fi**eld **Net**work) is the open Industrial Ethernet standard from PROFIBUS & PROFINET International. PROFINET is standardised in IEC 61158 and IEC 61784.
In PROFINET, there are the two perspectives, PROFINET CBA and PROFINET IO.
PROFINET CBA (Component Based Automation) is the original variant, which is based on a component model for communication of intelligent automation devices with each other.
Profinet IO was created for real-time (RT) and synchronous communication IRT (IRT= Isochronous Real-Time) between a controller and the decentralised peripherals.

To better scale the communication options and thus also the determinism in PROFINET IO, real-time classes (RT_CLASS) have been defined for data exchange.

RT Class	Comment	Is supported by CAMC-F-PN
RTC 1	Based on an unsynchronised RT communication within a subnet.	Yes, as active participant.
RTC2 not synchronised	Permits both synchronised and unsynchronised communication.	Compatible (only passive)
RTC 2 synchronised		No
RTC 3	Permits only synchronised communication.	Compatible (only passive)
RTC via UDP		No

Tab. 4.1 Real-time classes

PROFINET IO is a network system optimised on performance. Since the complete function range is not always needed in each automation system, PROFINET IO is cascadeable with regard to the supported function. The Profibus user organisation has therefore divided the PROFINET function range into conformance classes. The target is to simplify use of PROFINET IO and make things easier for the system operator through a simple selection of field devices and bus components with uniquely defined minimum characteristics.
The minimum requirements for 3 conformance classes (CC-A, CC-B, CC-C) have been defined.

Class A lists all devices according to the PROFINET IO standard. Class B specifies that the network infrastructure must also be constructed in accordance with the guidelines of PROFINET IO. Class C permits synchronous applications.

Additional information, contact addresses etc. can be found under:
➔ http://www.profinet.com
➔ http://www.profibus.com/download
Observe the available documents on planning, mounting and commissioning.

4.2 PROFINET interface CAMC-F-PN

The PROFINET interface is implemented for the motor controllers CMMP-AS-...-M3 through the optional interface CAMC-F-PN. The interface is mounted in slot Ext2. The PROFINET connection is designed as a 2-port Ethernet switch with 8-pin RJ sockets at the interface CAMC-F-PN.

With the help of the CAMC-F-PN, it is possible to integrate the CMMP-AS-...-M3 into a PROFINET network. The CAMC-F-PN permits the exchange of process data between a PROFINET controller and the CMMP-AS-...-M3.

Note
The PROFINET interface of the CAMC-F-PN is intended exclusively for connection to local, industrial fieldbus networks.
Direct connection to a public telecommunications network is not permissible.

4.2.1 Supported protocols and profiles

The interface CAMC-F-PN supports the following protocols:

Protocol/profile	Description
Profile	
PROFIenergy	Profile for energy management
Protocol	
MRP	The interface behaves MRP-compatibly at the bus and supports the general function of MRP as an MRP slave. The interface is able to communicate with a redundancy manager (RM) and pass on the MRP packages in accordance with the MRP specification. In case of a string failure, the interface receives the new path specifications of the RM and uses them.
LLDP	The protocol permits information exchange between neighbouring devices.
SNMP	Monitoring and control through a central component

Tab. 4.2 Supported protocols and profiles

4.2.2 Connection and display components at the interface CAMC-F-PN

1 ACT-LED (orange)
2 LNK-LED (green)
3 SF-LED
4 BF-LED
5 PROFINET interface (RJ-45 socket, 8-pin)

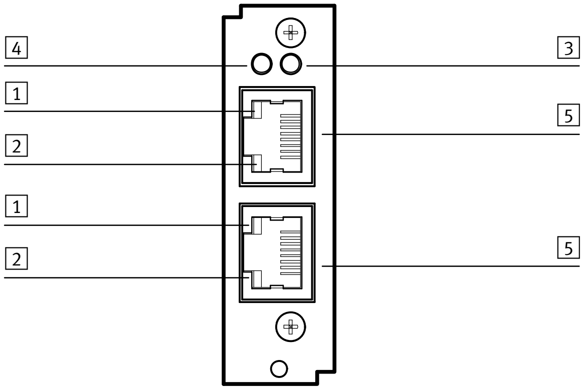


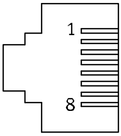
Fig. 4.1 Connection and display components at the PROFINET-IO interface

4.2.3 PROFINET LEDs

LED	Status:	Significance:
SF	Off	No system error
	Lights up red	Watchdog timeout
		Channel diagnostics
		General or extended diagnostics
		System fault
	Flashes red (2 Hz for 3 s)	PROFINET equipment identification
BF	Off	No bus error
	Lights up red	No configuration
		Error at the physical link
		No physical link
	flashes red (2 Hz)	No data are transmitted
LNK	Off	No link present
	Lights up green	Link present
ACT	Off	No Ethernet communication present
	Lights up orange	Ethernet communication present
	Flashes orange	Ethernet communication active

Tab. 4.3 PROFINET LEDs

4.2.4 Pin allocation for PROFINET interface

Socket	Pin no.	Designation	Description
	1	RX–	Receiver signal-
	2	RX+	Receiver signal+
	3	TX-	Transmission signal-
	4	-	Not assigned
	5	-	Not assigned
	6	TX+	Transmission signal+
	7	-	Not assigned
	8	-	Not assigned

Tab. 4.4 Pin allocation: PROFINET interface

4.2.5 PROFINET copper cabling

PROFINET cables are 4-wire, screened copper cables. The wires are marked by colour. The maximum bridgeable distance for copper cabling is 100 m between communication end points. This transmission distance is defined as PROFINET end-to-end link.

Use only PROFINET-specific cabling corresponding to conformance class B
➔ EN 61784-5-3.

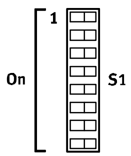
4.3 Configuration PROFINET-IO participants

Several steps are required in order to produce an operational PROFINET interface.
We recommend the following procedure:

1. Activation of the bus communication via DIP switches.
2. Parameterisation and commissioning with the Festo Configuration Tool (FCT).
   The following settings on the Fieldbus page:
   – IP address
   – Issue of the PROFINET-IO device name
   – Physical units (Factor Group tab)
   – Optional use of FPC and FHPP+ (FHPP+ editor tab)
3. Linking of the GSDML file into the project planning software

4.3.1 Activation of PROFINET communication with DIP switches

The PROFINET interface can be activated with switch 8 through DIP switch S1 on the module in slot Ext3. The remaining switches 1…7 have no significance for PROFINET.

DIP switch	DIP switch 8	PROFINET interface
	OFF	Disabled
	ON	Enabled

Tab. 4.5 Activation of PROFINET communication

4.3.2 Parameterisation of the PROFINET interface

With the help of the FCT, settings of the PROFINET interface can be read and parameterised. The target is to configure the PROFINET interface through the FCT in such a way that the motor controller CMMP-AS-...-M3 can build up PROFINET communication with a PROFINET controller. Parameterisation can take place even if no PROFINET interface CAMC-F-PN has yet been installed in the motor controller CMMP-AS-...-M3. If a PROFINET interface CAMC-F-PN is plugged into the controller, the interface is automatically recognised after the motor controller is switched on and is placed in operation with the stored information. This ensures that the motor controller CMMP-AS-...-M3 remains addressable through the same network configuration if the CAMC-F-PN is replaced.

The configuration and status of the DIP switches is read once at Power ON/RESET. The CMMP-AS-...-M3 takes over changes to the configuration and switch settings in ongoing operation only at the next RESET or restart. In order to activate the settings made, proceed as follows:

– Save all parameters in the flash with the help of the FCT
– Carry out a reset or restart of the CMMP-AS-...-M3.

4.3.3 Commissioning with the Festo Configuration Tool (FCT)

Instructions on commissioning with the Festo Configuration Tool can be found in the Help for the device-specific FCT plug-in.

To be able to make the subsequent settings, select "PROFINET IO" as the control interface in the FCT program on the Application Data page in the Operating Mode Selection tab.
Then change to the Fieldbus page.

4.3.4 Setting the interface parameters

Fieldbus device name

For a controller to communicate with the interface CAMC-F-PN, a unique name must be assigned to the interface. The name must be unique in the network.

Follow the PROFINET name conventions when assigning fieldbus device names.

PROFIenergy

The PROFIenergy profile can be activated or deactivated through a corresponding selection.
In the PROFIenergy status, the CMMP-AS-...-M3 engages the holding brake and switches off the output stage.

Note
PROFIenergy should not be used with vertically mounted axes, since it can not be ensured that the holding brake will hold the load if the load is large.

4.3.5 IP address allocation

A unique IP address must be assigned to each device in the network.

Static address allocation

A static IP address, such as the related subnet mask and the gateway, can be set in the FCT.

Assignment of already used IP addresses can result in temporary overloading of your network.
You may need to contact your network administrator for manual assignment of a permissible IP address.

Dynamic address allocation
With dynamic address allocation, IP addresses, like the related subnet mask and the gateway, are set through the DCP protocol. A previously assigned static IP address is hereby overwritten.

4.3.6 Setting of the physical units (factor group)

In order for a fieldbus master to exchange position, velocity and acceleration data in physical units (e.g. mm, mm/s, mm/s^2) with the motor controller, it must be parameterised via the factor group ➔ section A.1.
Parameterisation can be carried out via FCT or the fieldbus.

4.3.7 Setting of the optional use of FPC and FHPP+

Besides the control and status bytes, additional I/O data can be transmitted ➔ sections C.1 and C.2. This is set via the FCT (Fieldbus page, tab FHPP+ Editor).

4.4 Identification & service function (I&M)

The PROFINET interface CAMC-F-PN supports the device-specific base information of the I&M0.

Byte	Designation	Contents	Description	Data type
00...09	Header	Reserved	-	-
10...11	MANUFACTURER_ID	0x014D	Manufacturer's code (333 = FESTO)	UINT16
12...31	ORDER_ID	CMMP-AS-...-M3	Order code	STRING
32...47	SERIAL_NUMBER	e.g. "10234"	Serial number	STRING
48...49	HARDWARE_REVISION	e.g. 0x0202	Hardware issue status	UINT16
50...53	SOFTWARE_REVISION	e.g. V1.4.0	Software issue status	UINT16
54...55	REVISION_COUNTER	0x0000	Software Revisions	UINT16
56...57	IM_PROFILE_ID	0x0000	"Non-profile device"	UINT16
58...59	IM_PROFILE_SPECIFIC_TYPE	0x0000	No profiles are supported	UINT16
60...61	IM_VERSION	0 x 01, 0 x 02	I&M Version V1.2	UINT8 UINT8
62...63	IM_SUPPORTED	0x0000	Only I&M0 is supported	16 bit array

Tab. 4.6 PROFINET I&M 0 Block

4.5 Configuration PROFINET master

A GSDML file is available to you for project planning of the PROFINET IO interface. This file is read in with the help of the project planning software of the used PROFINET IO controller and is then available for project planning. The GSDML file describes the motor controller as a modular device. In it are described all possible device structure variants in a PROFINET-conforming manner.
You can obtain the detailed procedure for linking from the documentation of your corresponding project planning software

The GSDML file and the related symbol files are included on a CD-ROM supplied with the motor controller.

GSDML file	Description
GSDML...-CMMP-AS-M3-*.xml	Motor controller CMMP-AS-...-M3 with protocol "FHPP"

Tab. 4.7 GSDML file

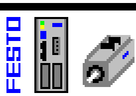
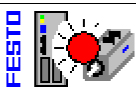
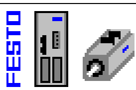
i You can find the most current versions under: ➔ www.festo.com/sp

The following languages are supported in the GSDML file:

Language	XML tag
English	PrimaryLanguage
German	Language xml:lang="de"

Tab. 4.8 Supported languages

The following symbol files are available to represent the motor controller CMMP-AS-...-M3 in your configuration software (for example, STEP 7):

Operating status	Symbol	Symbol file
Normal operating status		GSDML-014D-0202-CMMP-AS-M3_N.bmp
Diagnostic case		GSDML-014D-0202-CMMP-AS-M3_D.bmp
Special operating status		GSDML-014D-0202-CMMP-AS-M3_S.bmp

Tab. 4.9 Symbol file CMMP-AS-...-M3

i To simplify commissioning of the CMMP-AS-...-M3 with controllers from various manufacturers, you will find corresponding modules and application notes at
➔ www.festo.com/sp

4.6 Channel diagnostics – extended channel diagnostics

The malfunction number (➔ chapter D) is made up of a main index (MI) and a subindex (S).
The main index of the malfunction number is transferred in the manufacturer-specific range of channel diagnostics (ChannelErrorType) 0x0100 ... 0x7FFF.
The subindex of the malfunction number is transferred in the manufacturer-specific range of the extended channel diagnostics (ExtChannelErrorType) 0x1000 ... 0x100F.

Example

Malfunction Number	**ChannelErrorType**	**ExtChannelErrorType**
72-4	$HH_{h\,+}\,1000_h$ = 0x1048	$S_{h\,+}\,1000_h$ = 0x1004

Tab. 4.10 Channel diagnostics – extended channel diagnostics

5 PROFIBUS DP with FHPP

M3

This chapter is only applicable for the motor controller CMMP-AS-...**-M3**.

5.1 Overview

This part of the documentation describes the connection and configuration of the motor controller CMMP-AS-...-M3 in a PROFIBUS-DP network. It is directed at people who are already familiar with this bus protocol.

PROFIBUS (**PRO**cess **FI**eld**BUS**) is a standard developed by the PROFIBUS User Organisation. A complete description of the fieldbus system can be found in the following standard:
IEC 61158 “Digital data communication for measurement and control – Fieldbus for use in industrial control systems”. This standard contains several parts and defines 10 “field bus protocol types”. Among these, PROFIBUS is specified as “Type 3”. PROFIBUS exists in two designs. PROFIBUS-DP is used for fast data exchange in manufacturing engineering and building automation (DP = decentralised periphery). The incorporation into the ISO/OSI layer model is also described in this standard.

Additional information, contact addresses etc. can be found under:
➔ http://www.profibus.com

5.2 Profibus interface CAMC-PB

The PROFIBUS interface is implemented for the motor controllers CMMP-AS-...-M3 through the optional interface CAMC-PB. The interface is mounted in slot Ext2. The PROFIBUS connection is designed as a 9-pin DSUB socket on the CAMC-PB interface.

5.2.1 Connection and display components at the interface CAMC-PB

1. DIP switches for termination
2. PROFIBUS interface (DSUB socket, 9-pin)
3. PROFIBUS LED (green)

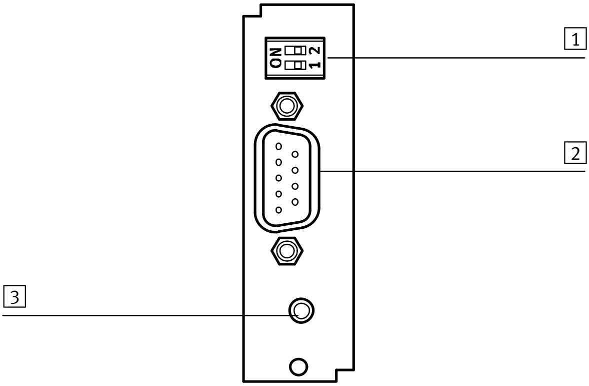


Fig. 5.1 Connection and display components on the PROFIBUS-DP interface

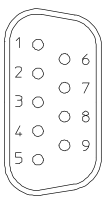
5.2.2 PROFIBUS LED

The PROFIBUS LED displays the communication status.

LED	Status
Off	No communication via PROFIBUS.
Lights up green	Communication active over PROFIBUS.

Tab. 5.1 PROFIBUS LED

5.2.3 Pin assignment of PROFIBUS interface

Plug	Pin no.		Designation	Value	Description
	1		Screened	–	Cable screening
		6	+5 V	+5 V	+5 V – output (potential isolated)[1)]
	2		–	–	Not assigned
		7	–	–	Not assigned
	3		RxD / TxD-P	–	Received / transmitted data B cable
		8	RxD / TxD-N	–	Received / transmitted data A cable
	4		RTS / FOC	–	Request to Send[2)]
		9	–	–	Not assigned
	5		GND5V	0 V	Reference potential GND 5V[1)]

1) Use for external bus termination or for supplying transmitter / receiver of an external fibre-optic-cable module.
2) Signal is optional, serves direction control when used with an external FOC module.

Tab. 5.2 Pin assignment: PROFIBUS DP interface

5.2.4 Termination and bus terminating resistors

Each bus segment of a PROFIBUS network must be fitted with terminating resistors in order to minimise cable reflections and set a defined rest potential on the cable. The bus termination is made at the beginning and end of a bus segment.

A defective or incorrect bus termination is often the cause of malfunctions

The terminating resistors are already integrated in most commercially available PROFIBUS plug connectors. The PROFIBUS interface CAMC-PB has its own integrated terminating resistors for coupling to buses with plug connectors without their own terminating resistors. These can be switched on via the two-pin DIP switches on the PROFIBUS interface CAMC-PB (**both** switches ON). To switch off the terminating resistors, **both** switches must be set to OFF.

To guarantee reliable operation of the network, only one bus termination may be used, internal (via DIL switch) **or** external.

The external circuitry can also be constructed discretely (➔ Fig. 5.2, page 56). The 5 V supply voltage required for the externally switched terminating resistors is provided at the 9-pin SUB-D socket of the PROFIBUS interface CAMP-PB (➔ pin assignment in Tab. 5.2).

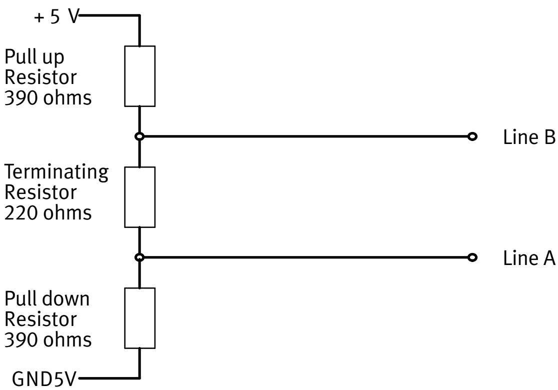


Fig. 5.2 External bus termination

PROFIBUS cabling

Due to the very high possible baud rates, we recommend that you use only the standardised cables and plug connectors. These are in some cases provided with additional diagnostic possibilities and in the event of a malfunction they facilitate the fast analysis of the fieldbus hardware.

If the set baud rate > 1.5 Mbit/s, plugs with integrated series inductance (110 nH) must be used due to the capacitive load of the station and the cable reflection thereby created.

When setting up the PROFIBUS network, it is essential that you follow the advice in the relevant literature or the following information and instructions in order to maintain a stable, trouble-free system. If the cabling is not correct, malfunctions may occur on the PROFIBUS which cause the motor controller to switch off with an error for safety reasons.

5.3 PROFIBUS station configuration

Several steps are required in order to produce a functioning PROFIBUS interface. Some of these settings should or must be carried out before the PROFIBUS communication is activated. This section provides an overview of the steps required by the slave for parameterisation and configuration. As some parameters are only effective after saving and reset, we recommend that commissioning with the FCT be carried out first without connection to the PROFIBUS.

Instructions on commissioning with the Festo Configuration Tool can be found in the Help for the device-specific FCT plug-in.

When planning the PROFIBUS interface, the user must make these determinations. Only then should parametrisation of the fieldbus connection take place on both pages. We recommend that parameterisation of the slave should be undertaken first. Then the master should be configured. With correct parameterisation the application is ready immediately without communication faults.

We recommend the following procedure:

1. Set the offset of the bus address and activate the bus communication via DIP switches.

The status of the DIP switches is read once at Power- ON / RESET.
The CMMP-AS-...-M3 takes over changes to the switch setting in ongoing operation only at the next RESET or restart

2. Parameterisation and commissioning with the Festo Configuration Tool (FCT).
   In addition, the following settings on the fieldbus page:
   – Base address of the bus address
   – Physical units (Factor Group tab)
   – Optional use of FPC and FHPP+ (FHPP+ Editor tab)

Observe that parameterisation of the CANopen function remains intact after a reset only if the parameter set of the motor controller was saved.

3. Configuration of the PROFIBUS master ➔ section 5.4.

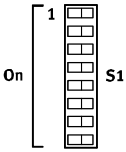
5.3.1 Setting the bus address with DIP switches and FCT

The inserted PROFIBUS interface is automatically detected after the motor controller is switched on.
A unique node address must be assigned to each device in the network.
The bus address can be set via the DIP switches 1 ... 7 on the interface in slot Ext3 and in the program FCT. Assignment of the address by the master is not possible, since the “Set_Slave_Address” service is not supported.

The resulting bus address consists of the base address (FCT) and the offset (DIP switches).
Permissible values for the bus address lie in the range 3 ... 125.

Setting the offset of the bus address with DIP switches

The bus address can be set via the DIP switches 1 … 7 on the module in slot Ext3. The offset of the bus address set via DIP switches 1 … 7 is displayed in the program FCT on the Fieldbus page in the Operating Parameters tab.

DIP switch		Value			Example	
		ON	OFF			Value
	1	1	0		ON	1
	2	2	0		ON	2
	3	4	0		OFF	0
	4	8	0		ON	8
	5	16	0		ON	16
	6	32	0		OFF	0
	7	64	0		ON	64
Sum of 1 … 7= bus address		0 … 127 1)				**91**

1) The resulting bus address is limited to a maximum of 125.

Tab. 5.3 Setting of the offset of the bus address

Changes to the DIP switches are not effective until Power On or RESET.

Setting the base address of the bus address with FCT

In the FCT program, the bus address is set on the Fieldbus page in the Operating Parameters tab as base address.
Default setting = 0 (that means offset = bus address).

If a bus address is assigned simultaneously via DIP switches 1 … 7 and in the FCT program, the resulting bus address consists of the sum of the base address and the offset. If this sum is greater than 125, the value is automatically limited to 125.

5.3.2 Activation of PROFIBUS communication with DIP switches

After setting the bus address, PROFIBUS communication can be activated. Please note that the above-mentioned parameters can only be revised when the protocol is deactivated.

PROFIBUS communication	DIP switch 8
Disabled	OFF
Enabled	ON

Tab. 5.4 Activation of CANopen communication

5.3.3 Setting of the physical units (factor group)

In order for a fieldbus master to exchange position, velocity and acceleration data in physical units (e.g. mm, mm/s, mm/s^2) with the motor controller, it must be parameterised via the factor group ➔ section A.1.

Parameterisation can be carried out via FCT or the fieldbus.

5.3.4 Setting of the optional use of FPC and FHPP+

Besides the control and status bytes, additional I/O data can be transmitted ➔ sections C.1 and C.2.

This is set via the FCT (Fieldbus page, tab FHPP+ Editor).

5.3.5 Storing the configuration

After configuration with subsequent download and saving, the PROFIBUS configuration is adopted after a reset of the controller.

Please observe that the PROFIBUS configuration can only be activated when the parameter records have been saved and a reset has been carried out.

5.4 PROFIBUS I/O configuration

Name	Cyclical I/O update		DP identifier
FHPP standard	1 x 8 bytes of I/O data, consistent data transmission	Cyclically transmitted 8 control and status bytes	0xB7
FHPP Standard + FPC	2 x 8 bytes of I/O data, consistent data transmission	As FHPP standard, additional 8 bytes of I/O data for parameterisation	0xB7, 0xB7
FHPP+ 8 bytes input	1 x 8 bytes of input data, consistent data transmission	Additional 1 x 8 bytes of input data for parameterisation	0x40, 0x87
FHPP+ 16 bytes input	+ 2 x 8 bytes of input data, consistent data transmission	Additional 2 x 8 bytes of input data for parameterisation	0x40, 0x8F
FHPP+ 24 bytes input	+ 3 x 8 bytes of input data, consistent data transmission	Additional 3 x 8 bytes of input data for parameterisation	0x40, 0x97
FHPP+ 8 bytes output	+ 1 x 8 bytes of output data, consistent data transmission	Additional 1 x 8 bytes of output data for parameterisation	0x80, 0x87
FHPP+ 16 bytes output	+ 2 x 8 bytes of output data, consistent data transmission	Additional 2 x 8 bytes of output data for parameterisation	0x80, 0x8F
FHPP+ 24 bytes output	+ 3 x 8 bytes of output data, consistent data transmission	Additional 3 x 8 bytes of output data for parameterisation	0x80, 0x97

Tab. 5.5 PROFIBUS I/O configuration

You can find information on the I/O allocation here:
- FHPP standard ➔ section 9.2.
- FPC ➔ section C.1.
- FHPP+ ➔ section C.2.

5.5 PROFIBUS master configuration

This section provides an overview of the steps required by the master for parametrisation and configuration. We recommend the following procedure:

1. Installation of the GSD file (device master data file)
2. Specification of the node address (slave address)
3. Configuration of the input and output data
   On the side of the master, the motor controller must be incorporated in the PROFIBUS in a way corresponding to the I/O configuration ➔ section 5.4.
4. When the configuration is concluded, transfer the data to the master.

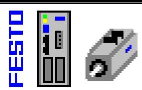
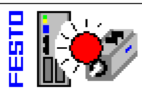
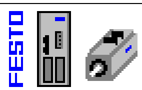
The GSD file and the related symbol files are included on a CD-ROM supplied with the motor controller.

GSD file	Description
P-M30D56.gsd	motor controller CMMP-AS-...-M3

Tab. 5.6 GSD file

You will find the most current version under ➔ www.festo.com/sp

The following symbol files are available to represent the motor controller CMMP-AS-...-M3 in your configuration software (for example, STEP 7):

Operating status	Symbol	Symbol files
Normal operating status		cmmpas_n.bmp cmmpas_n.dib
Diagnostic case		cmmpas_d.bmp cmmpas_d.dib
Special operating status		cmmpas_s.bmp cmmpas_s.dib

Tab. 5.7 Symbol files CMMP-AS-...-M3

To simplify commissioning of the CMMP-AS-...-M3 with controllers from various manufacturers, you will find corresponding modules and application notes at
➔ www.festo.com/sp

6 EtherNet/IP with FHPP

M3

This chapter is only applicable for the motor controller CMMP-AS-...**-M3**.

6.1 Overview

This part of the documentation describes the connection and configuration of the motor controller CMMP-AS-...-M3 in an EtherNet/IP network. It is directed at people who are already familiar with the bus protocol and motor controller.

The Ethernet Industrial Protocol (EtherNet/IP) is an open standard for industrial networks. EtherNet/IP is used to transmit cyclical I/O data as well as acyclic parameter data.
EtherNet/IP was developed by Rockwell Automation and the ODVA (Open DeviceNet Vendor Association) and standardised in the international standards series IEC 61158.

EtherNet/IP is the implementation of CIP over TCP/IP and Ethernet (IEEE 802.3). Standard Ethernet twisted-pair cables are used as the transmission medium.

Additional information, contact addresses etc. can be found under:
➔ http://www.odva.com
➔ http://www.ethernetip.de
Observe the available documents on planning, mounting and commissioning.

6.2 EtherNet/IP-Interface CAMC-F-EP

The EtherNet/IP interface is implemented for the motor controllers CMMP-AS-...-M3 through the optional interface CAMC-F-EP. The interface is mounted in slot Ext2. The EtherNet/IP connection is designed as a 2-port Ethernet switch with 8-pin RJ sockets at the interface CAMC-F-EP.

With the help of the CAMC-F-EP, it is possible to integrate the motor controllers CMMP-AS-...-M3 into an EtherNet/IP network. The CMMP-AS-...-M3 is a pure EtherNet/IP adapter and requires an EtherNet/IP controller (scanner) in order to be controlled via EtherNet/IP.

The CAMC-F-EP supports the Device Level Ring function (DLR). The CAMC-F-EP is able to communicate with an EtherNet/IP Ring Supervisor. In case of a string failure, the CAMC-F-EP receives the new path specifications of the Ring Supervisor and uses them.

Note
The EtherNet/IP interface of the CAMC-F-EP is intended exclusively for connection to local, industrial fieldbus networks.
Direct connection to a public telecommunications network is not permissible.

6.2.1 Connection and display components at the interface CAMC-F-EP

1 ACT-LED (Ethernet communication activity)
2 LNK-LED (Ethernet line monitoring)
3 MS-LED (module status)
4 NS-LED (network status)
5 EtherNet/IP interface port 2 (RJ-45 socket, 8-pin)
6 EtherNet/IP interface port 1 (RJ-45 socket, 8-pin)

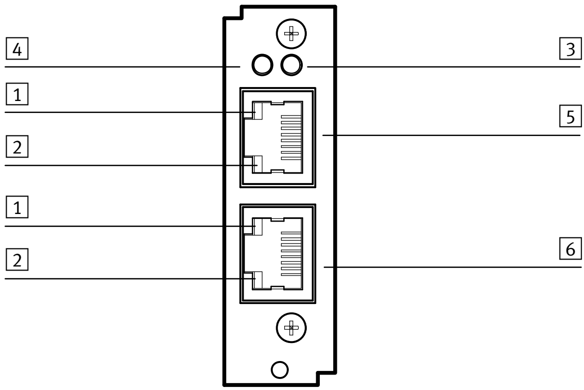


Fig. 6.1 Connection and display components at the EtherNet/IP interface

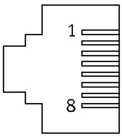
6.2.2 EtherNet/IP LEDs

Diagnostic messages generated by the CAMC-F-EP are recorded and evaluated by the CMMP-AS-...-M3. If the conditions for an error status are recognised, an error message is generated. The generated error message is signalled via the LEDs at the front side of the CAMC-F-EP.

LED	Function	Status:	Significance:
ACT	Ethernet communication activity	Off	No bus activity
		Flashes orange	Bus activity present
LNK	Ethernet line monitoring	Off	No link present
		Lights up green	Link present
MS	EtherNet/IP module status	Off	No supply voltage
		Lights up green	Interface ready for operation
		Flashes green	Standby
		Lights up red	Major fault
		Flashes red	Minor Fault
		Flashes red/ green	Self test
NS	EtherNet/IP network status	Off	No supply voltage
			No IP address
		Lights up green	Connection present
		Flashes green	No connection
		Lights up red	Duplicate IP address
		Flashes red	Connection timeout
		Flashes green	No connection
		Flashes red/green	Self test

Tab. 6.1 EtherNet/IP interface display elements LED

6.2.3 Pin allocation Ethernet/IP interface

Socket	Pin no.	Designation	Description
	1	RX–	Receiver signal-
	2	RX+	Receiver signal+
	3	TX–	Transmission signal-
	4	-	Not assigned
	5	-	Not assigned
	6	TX+	Transmission signal+
	7	-	Not assigned
	8	-	Not assigned

Tab. 6.2 Pin allocation: Ethernet/IP interface

6.2.4 EtherNet/IP copper cabling

EtherNet/IP cables are 4-wire, screened copper cables. The maximum permissible segment length for copper cabling is 100 m.

Use only EtherNet/IP specific cabling for the industrial environment corresponding to
➔ EN 61784-5-3

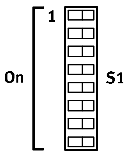
6.3 Configuration EtherNet/IP stations

Several steps are required in order to produce an operational EtherNet/IP interface.
We recommend the following procedure:

1. Activation of the bus communication via DIP switches.
2. Parameterisation and commissioning with the Festo Configuration Tool (FCT).
   In addition, the following settings on the fieldbus page:
   – IP address
   – Physical units (Factor Group tab)
   – Optional use of FPC and FHPP+ (FHPP+ editor tab)
3. Linking of the electronic data sheet (EDS) file into the project planning software.

6.3.1 Activation of the EtherNet/IP communication

The EtherNet/IP interface can be activated with switch 8 through DIP switch S1 on the module in slot Ext3.

DIP switch	DIP switch 8	Ethernet/IP interface
	OFF	Disabled
	ON	Enabled

Tab. 6.3 Activation of the EtherNet/IP communication

6.3.2 Parameterisation of the Ethernet/IP interface

With the help of the FCT, settings of the EtherNet/IP interface can be read and parameterised. The goal is to configure the EtherNet/IP interface through the FCT in such a way that the motor controller CMMP-AS-...-M3 can build up EtherNet/IP communication with an EtherNet/IP controller.
The settings of the EtherNet/IP interface can be parameterised in the FCT even if no EtherNet/IP interface CAMC-F-EP is integrated into the motor controller CMMP-AS-...-M3. If an EtherNet/IP interface CAMC-F-EP is plugged into the controller, the interface is placed in operation with the stored information. This ensures that the CMMP-AS-...-M3 remains addressable through the same network configuration if the CAMC-F-EP is replaced.
The inserted EtherNet/IP interface is automatically detected after the motor controller is switched on.

The configuration and status of the DIP switches is read once at Power ON/RESET. The CMMP-AS-...-M3 takes over changes to the configuration and switch settings in ongoing operation only at the next RESET or restart. In order to activate the settings made, proceed as follows:

– Save all parameters in the flash with the help of the FCT
– Carry out a reset or restart of the CMMP-AS-...-M3.

6.3.3 Commissioning with the Festo Configuration Tool (FCT)

Instructions on commissioning with the Festo Configuration Tool can be found in the Help for the device-specific FCT plug-in.

To be able to make the subsequent settings, select EtherNet/IP as the control interface in the FCT on the Application Data page in the Operating Mode Selection tab.
Then change to the Fieldbus page.

6.3.4 Setting the IP address

A unique IP address must be assigned to each device in the network.

Assignment of already used IP addresses can result in temporary overloading of your network.
You may need to contact your network administrator for manual assignment of a permissible IP address.

There are several options for addressing the CAMC-F-EP interface.

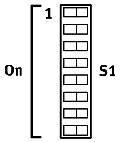
Static addressing with DIP switches

The first three bytes of the IP address are preset with 192.168.1.xxx. The fourth byte of the IP address can be set in the range 0 … 127 with DIP switches 1 … 7 at the module in slot Ext3. The address is thus freely selectable in the range 192.168.1.1 to 192.168.1.127.

If the 4th byte is set to zero (DIP switches 1 … 7 = OFF), the IP address parameterised in the FCT is used.

If the IP address is set via the DIP switches, the subsequent standard values are assigned for the subnet mask and gateway address:

– Subnet mask: 255.255.255.0
– Gateway address: 0.0.0.0

DIP switch		**Value**			**Example**	
		ON	OFF			Value
	1	1	0		ON	1
	2	2	0		OFF	0
	3	4	0		OFF	0
	4	8	0		ON	8
	5	16	0		ON	16
	6	32	0		OFF	0
	7	64	0		OFF	0
Sum of 1 ... 7 = 4th byte of IP address		0 [1] ... 127 [2]				**25**

1) If the fourth byte is zero, dynamic address allocation takes place via DHCP/BOOTP

2) For values larger than 127, the IP address must be set with the FCT.

Tab. 6.4 Setting the IP address with DIP switch

Static addressing with FCT (Festo Configuration Tool)

With the Festo Configuration Tool (FCT), the values for IP address, subnet mask and gateway address can be assigned on the Fieldbus page in the Operating Parameters tab.

Dynamic addressing

The dynamic addressing parameterised in the FCT is only used if:
– the DIP switches 1 ... 7 on the module in the slot Ext3 = OFF.
– Obtain IP address automatically has been selected in the FCT on the Fieldbus page in the Operating parameters tab.

For dynamic addressing, there is the option of addressing either through DHCP or BOOTP. Both protocols are standard and are supported by the CAMC-F-EP. If dynamic addressing is set at device start or reset (DIP switches 1 ... 7 = OFF, on the module in slot Ext3), an IP address is assigned to the device either through DHCP and an available DHCP server or through the BOOTP protocol.

6.3.5 Setting of the physical units (factor group)

In order for a fieldbus master to exchange position, velocity and acceleration data in physical units (e.g. mm, mm/s, mm/s^2) with the motor controller, they must be parameterised via the factor group
➔ section A.1.
Parameterisation can be carried out via FCT or the fieldbus.

6.3.6 Setting of the optional use of FPC and FHPP+

Besides the control and status bytes, additional I/O data can be transmitted ➔ sections C.1 and C.2.
This is set via the FCT (Fieldbus page, tab FHPP+ Editor).

6.4 Electronic data sheet (EDS)

In order to permit fast and simple commissioning, the abilities of the EtherNet/IP interface of the motor controller are described in an EDS file.

Type	File
CMMP-AS-...-M3_FHPP.eds	Motor controller CMMP-AS-...-**M3** with protocol “FHPP”

Tab. 6.5 EDS files

By using an appropriate configuration tool, you can configure a device within a network. The EDS files for EtherNet/IP are included on a CD-ROM supplied with the motor controller.

You can find the most current version of the EDS under ➔ www.festo.com/sp

The way in which you configure your network depends on the configuration software used. Follow the instructions of the controller manufacturer for registering the EDS file of the motor controller CMMP-AS-...-M3.

To simplify commissioning of the CMMP-AS-...-M3 with controllers from various manufacturers, you will find corresponding modules and application notes at
➔ www.festo.com/sp

Data types

The following data types corresponding to the EtherNet/IP specification are used:

Type	Signed	Unsigned
8 bit	SINT	USINT
16 bit	INT	UINT
32 bit	DINT	UDINT

Tab. 6.6 Data types

Identity Object (Class Code: 0x01)

The identity object includes identification and general information about the motor controller.
Instance 1 identifies the total motor controller. This object is used to identify the motor controller in the network.

Instance		Attribute	Name	Description
0	Class	1	Revision	Revision of this object
		2	Max. Instance	Maximum instance number of an object currently created in this class level of the device.
		6	Max. Class Attribute	The attribute ID number of the last class attribute of the class definition implemented in the device.
		7	Max. Instance Attribute	The attribute ID number of the last instance attribute of the class definition implemented in the device.
1	Instance Attributes	1	Vendor ID	Device manufacturer’s Vendor ID.
		2	Device Type	Device Type of product.
		3	Product code	Product Code assigned with respect to device type.
		4	Major Revision	Major device revision.
			MinorRevision	Minor device revision.
		5	Status	Current status of device.
		6	Serial number	Serial number of device.
		7	Product name	Human readable description of device.
		8	State	Current state of device.
		9	Configuration Consistency Value	Contents identify configuration of device.

Tab. 6.7 Identity object

Message Router Object (Class Code: 0x02)

The Message Router Object offers a message connection with which a client can address a service to an object class or instance within the device. No services are offered from the Message Route Object.

Assembly Object (Class Code: 0x04)

The Assembly Object links attributes or several objects that allow sending or receiving data from an object. Assembly Objects can be used to link input or output data. The terms “Input” and “Output” are defined from the network perspective.

Instance		Attribute	Name	Description
0	Class	1	Revision	Revision of this object.
		2	Max. Instance	Maximum instance number of an object currently created in this class level of the device.
1-x	Instance Attributes	3	Data	Data
		4	Size	Number of bytes in Attribute 3.

Tab. 6.8 Assembly Object

Connection Manager Object (Class Code: 0x06)

The Connection Manager Object is used to set up a connection and must always be supported. The Connection Manager Object is instanced only once.

TCP/IP Interface Object (Class Code: 0xF5)

The TCP/IP Object is used to configure a TCP/IP network. For example, IP address, subnet mask and gateway address

Instance		Attribute	Name		Description
0	Class	1	Revision		Revision of this object.
		2	Max. Instance		Maximum instance number of an object currently created in this class level of the device.
1	Instance Attributes	1	Status		Interface status.
		2	Configuration Capacity		Interface capability flags.
		3	Configuration Control		Interface control flags.
		4	Physical Link Object		Path to physical link object.
		5	Interface Configuration		TCP/IP network interface configuration.
				IP Address	The device’s IP address.
				Network Mask	The device’s network mask.
				Gateway Address	Default gateway address.
				Name Server	Primary name server.
				Name Server 2	Secondary name server.
				Domain Name	Default domain name.
		6	Host Name		Host Name

Tab. 6.9 TCP/IP Interface Object

Ethernet Link Object (Class Code: 0xF6)

The Ethernet Link Object includes link-specific counters and status information for an Ethernet IEEE 802.3 communication interface. Each instance of an Ethernet Link Object corresponds exactly to an Ethernet IEEE 802.3 communication interface.

Instance		Attribute	Name	Description
0	Class	1	Revision	Revision of this object.
		2	Max. Instance	Maximum instance number of an object currently created in this class level of the device.
		3	Number of Instances	Number of object instances currently created at this class level of the device.
1-x	Instance Attributes	1	Interface Speed	Interface speed currently in use; speed in Mbps (e. g. 0, 10, 100, 1000, usw.).
		2	Interface Flags	Interface status flags
		3	Physical Address	MAC layer address.
		4	Interface Counters	Contains counters relevant to the receipt of packets on the interface.
		5	Media Counters	Media-specific counters.
		6	Interface Control	Configuration for physical interface.

Tab. 6.10 Ethernet Link Object

Device Level Ring Object (Class Code: 0x47)

The DLR object is used to configure a network with the ring topology corresponding to the DLR (Device Level Ring) specification of EtherNet/IP.

Instance		Attribute	Name	Description
0	Class	1	Revision	Revision of this object.
1	Instance Attributes	1	Network Topology	Current network topology mode 0 indicates “Linear” 1 indicates “Ring”
		2	Network Status	Current status of network 0 indicates “Normal” 1 indicates “Ring Fault” 2 indicates “Unexpected Loop Detected” 3 indicates “Partial Network Fault” 4 indicates “Rapid Fault/Restore Cycle”
		10	Active Supervisor Address	IP and/or MAC address of the active ring supervisor.
		12	Capability Flags	Describes the DLR capabilities of the device.

Tab. 6.11 Device Level Ring Object

QOS Object (Class Code: 0x48)

The Quality of Service Object offers mechanisms that can occupy the transmission stream with various priorities.

Instance		Attribute	Name	Description
0	Class	1	Revision	Revision of this object.
		2	Max. Instance	Maximum instance number of an object currently created in this class level of the device.
1-x	Instance Attributes	1	802.1Q Tag Enable	Enables or disables sending 802.1Q frames on CIP and IEEE 1588 messages.
		4	DCCP Urgent	DSCP value for CIP transport class 0/1 Urgent priority messages.
		5	DCSP Scheduled	DSCP value for CIP transport class 0/1 Scheduled priority messages.
		6	High	DSCP value for CIP transport class 0/1 High priority messages.
		7	Low	DSCP value for CIP transport class 0/1 low priority messages.
		8	Explicit	DSCP value for CIP explicit messages (transport class 2/3 and UCMM).

Tab. 6.12 QOS Object

6.5 CIP objects

Supported CIP objects → section 7.5.

7 DeviceNet with FHPP

M3

This chapter is only applicable for the motor controller CMMP-AS-...**-M3**.

7.1 Overview

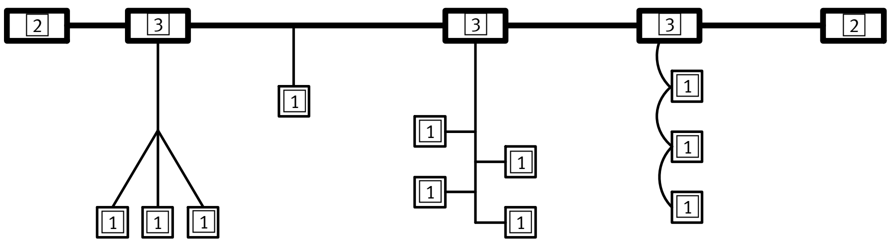
This part of the documentation describes the connection and configuration of the motor controller CMMP-AS-...-M3 in a DeviceNet network. It is directed at people who are already familiar with this bus protocol.

DeviceNet was developed by Rockwell Automation and the ODVA (Open DeviceNet Vendor Association) as an open fieldbus standard based on the CAN protocol. DeviceNet belongs to the CIP-based networks. CIP (Common Industrial Protocol) forms the application layer of DeviceNet and defines the exchange of

– explicit messages with low priority, e.g. for configuration or diagnostics
– I/O messages, e.g. time-critical process data

The Open DeviceNet Vendor Association (ODVA) is the user organisation for DeviceNet. Publications concerning the DeviceNet/CIP specification are available at ODVA (Open DeviceNet Vendor Association) → http://www.odva.org

DeviceNet is a machine-oriented network which enables connections between simple industrial devices (sensors, actuators) and higher-order devices (controllers). DeviceNet is based on the CIP protocol (Common Industrial Protocol) and shares all common aspects of CIP with adaptations enabling the frame size of messages to be adapted to that of DeviceNet. Fig. 7.1 shows an example of a typical DeviceNet network.



1 DeviceNet stations or nodes
2 Terminating resistor 121 Ohm
3 Multiple-port tap

Fig. 7.1 DeviceNet network

DeviceNet offers:
- a low-cost solution for networks at the device level
- Access to information in devices at a lower level
- Possibility for master/slave and peer-to-peer

DeviceNet pursues two main objectives:
- Transporting control-orientated information, which is in connection with devices of the lower level (I/O connection).
- Transporting further information which is indirectly connected with the closed-loop system, such as configuration parameters (Explicit Messaging Connection).

7.1.1 I/O connection

Some types of I/O connection are defined by DeviceNet. At present only Poll Command /Response Message with 16 bytes of input data and 16 bytes of output data are supported with FHPP. This means that the master periodically sends 16 bytes of data to the slave and the slave also replies with 16 bytes.

7.1.2 Optional use of FHPP+

Besides the control or status bytes and the FPC, additional I/O data can be transmitted ➔ section C.2.
This is set via the FCT (page fieldbus, tab FHPP+ editor).
The meaning of the data is determined by the FHPP user protocol.

7.1.3 Explicit Messaging

The Explicit Messaging protocol is used for transporting configuration data and for configuring a system. Explicit Messaging is also used for setting up an I/O connection. Explicit Messaging connections are always point-to-point connections. An end point sends a request, the other end point replies with an answer. The answer may be a success message or an error message.

Explicit messaging makes various services possible. The most common services are:
- opening the explicit messaging connection,
- closing the explicit messaging connection,
- get single attribute (read parameter),
- get single attribute (save parameter).

7.2 DeviceNet interface CAMC-DN

The DeviceNet interface for the motor controllers CMMP-AS-...-M3 is implemented through the CAMC-DN interface. The interface is mounted in the Ext1 slot. The DeviceNet connection is designed as a 5-pin open connector.

7.2.1 Display and control elements at the CAMC-DN interface

1 Open connector (5-pin)
2 DeviceNet LED (green/red)

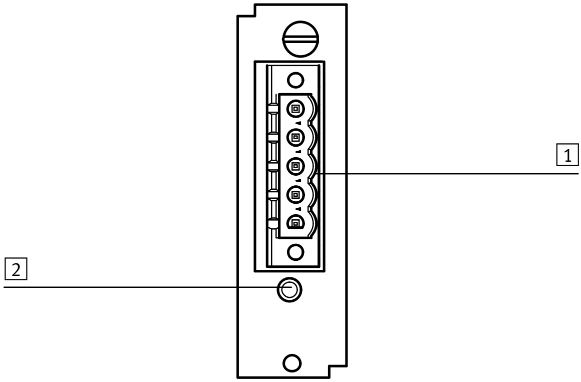


Fig. 7.2 Connection and display elements at the DeviceNet interface

7.2.2 DeviceNet LED

A two-colour LED shows information about the device and the communication status. It has been designed as a combined module/network status (MSN) LED. The combined module and network status LED supplies limited information on the device and the communication status.

LED	Status	Shows:
is off	Device is not online.	The device has not yet finished initialisation or has no power supply.
Flashes green	Ready for operation and online, Not connected or Online and requires commissioning	The device works in a normal status and is online without established connection.
Lights up green	Ready to operate and online, connected	The device works in a normal status and is online with established connections.

LED	Status	Shows:
Flashes red-green	Communication failed and receives an Identify Comm Fault Request	The device has ascertained a network access error and is in the status “Communication Faulted”. The device then received and accepted an “Identify Communication Faulted Request”. Normal behaviour during commissioning.
Flashes red	Minor error or connection interrupted (time-out)	Correctable error and / or at least one I/O connection is in the time-out status.
Lights up red	Critical error or critical connection error	The device has an error which cannot be corrected. The device has ascertained an error which makes communication in the network impossible (e.g. bus off, double MAC-ID).

Tab. 7.1 DeviceNet LED

7.2.3 Pin allocation

Plug	Pin no.	Designation	Value	Description
	5	V +	24 V	CAN transceiver supply voltage
	4	CAN-H	-	Positive CAN signal (dominant high)
	3	Drain/Shield	-	Screening
	2	CAN-L	-	Negative CAN signal (dominant low)
	1	V –	0 V	Reference potential CAN transceiver

Tab. 7.2 Pin assignment: DeviceNet interface

Next to the contacts CAN_L and CAN_H for the network connection, 24 V DC must be connected to V+ and V- in order to supply the CAN transceiver.
The cable screening is connected to the Drain/Shield contact.
In order to connect the DeviceNet interface correctly to the network, consult the very detailed “Planning and Installation Manual” on the ODVA homepage. The different types of network supply are also represented in detail there.

7.3 Configuration DeviceNet participants

Several steps are required in order to produce an operational DeviceNet interface. Some of these settings should or must be carried out before the DeviceNet communication is activated. This section provides an overview of the steps required by the slave for parameterisation and configuration. As some parameters are only effective after saving and reset of the controller, we recommend that commissioning with the FCT should be carried out first without connection to the DeviceNet.

Instructions on commissioning with the Festo Configuration Tool can be found in the Help for the device-specific FCT plug-in.

When designing the DeviceNet interface, the user must therefore make these determinations. Only then should parameterisation of the fieldbus connection take place on both pages. We recommend that parameterisation of the slave should be executed first. Then the master should be configured. With correct parameterisation, the application is ready immediately without communication errors.

We recommend the following procedure:

1. Set the offset of the MAC ID and activate the bus communication via DIP switches.

The status of the DIP switches is read once at Power- ON / RESET.
The CMMP-AS-...-M3 takes over changes to the switch setting in ongoing operation only at the next RESET or restart

2. Parameterisation and commissioning with the Festo Configuration Tool (FCT).
   In addition, the following settings on the fieldbus page:
   – For MAC IDs > 31: base address of the MAC ID
   – Physical units (Factor Group tab)
   – Optional use of FPC and FHPP+ (FHPP+ editor tab)

Observe that parameterisation of the DeviceNet function remains intact after a reset only if the parameter set of the motor controller was saved.

3. Configuration of the DeviceNet master ➔ section 7.4.

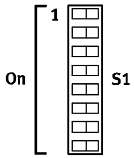
7.3.1 Setting the MAC ID with DIP switches and FCT

A unique MAC ID must be assigned to each device in the network. The MAC ID can be set via the DIP switches 1 ... 5 on the module in slot Ext3 or in the FCT.

The resulting MAC ID consists of the base address (FCT) and the offset (DIP switches). Permissible values for the MAC ID lie in the range 0 ... 63.

Setting the offset of the MAC ID with DIP switches

A MAC ID in the range 0 ... 31 can be set using the DIP switches 1 ... 5. The offset of the MAC ID set via DIP switches 1...5 is displayed in the program FCT on the fieldbus page in the operating parameters tab.

DIP switch		Value			Example	
		ON	OFF			Value
	1	1	0		ON	1
	2	2	0		OFF	0
	3	4	0		OFF	0
	4	8	0		ON	8
	5	16	0		ON	16
Total of 1 ... 5 = MAC ID		0 ... 31 [1]				**25**

1) A MAC ID larger than 31 must be set with the FCT.

Tab. 7.3 Setting the offset of the MAC ID

Setting the base address of the MAC ID with FCT

With the Festo Configuration Tool (FCT), the MAC ID is set as base address on the fieldbus page in the operating parameters tab.
Default setting = 0 (that means offset = MAC ID).

If a MAC-ID greater than 63 is set, the value is set automatically to 63.

7.3.2 Setting of the transmission rate using DIP switches

The transmission rate must be set with DIP switches 6 and 7 on the module in slot Ext3. The status of the DIP switches is read one time at Power On / Reset. The CMMP-AS-...-M3 takes over changes to the switch setting in ongoing operation only at the next RESET.

Transmission rate		DIP switch 6	DIP switch 7
125	[Kbit/s]	OFF	OFF
250	[Kbit/s]	ON	OFF
500	[Kbit/s]	OFF	ON
500	[Kbit/s]	ON	ON

Tab. 7.4 Setting of the transmission rate

7.3.3 Activation of DeviceNet communication

After the MAC-ID und the transmission rate have been set, DeviceNet communication can be activated. Please note that the above-mentioned parameters can only be revised when the protocol is deactivated.

DeviceNet communication	DIP switch 8
Disabled	OFF
Enabled	ON

Tab. 7.5 Activation of DeviceNet communication

Please observe that DeviceNet communication can only be activated after the parameter set (the FCT project) has been saved and a Reset carried out.

7.3.4 Setting of the physical units (factor group)

In order for a fieldbus master to exchange position, velocity and acceleration data in physical units (e.g. mm, mm/s, mm/s^2) with the motor controller, they must be parameterised via the factor group ➔ section A.1.
Parameterisation can be carried out via FCT or the fieldbus.

7.3.5 Setting of the optional use of FPC and FHPP+

Besides the control or status bytes and the FPC, additional I/O data can be transmitted ➔ sections C.1 and C.2.
This is set via the FCT (page fieldbus, tab FHPP+ editor).

7.4 Electronic data sheet (EDS)

You can use an EDS file to configure the DeviceNet master.
The EDS file is included on the CD-ROM supplied with the motor controller.

You will find the most current version under ➔ www.festo.com/sp

EDS files	Description
CMMP-AS-...-M3_*.eds	Motor controller CMMP-AS-...-M3 with protocol “FHPP” (static for Beckhoff PLC)
CMMP-AS-...-M3_*.eds	Motor controller CMMP-AS-...-M3 with protocol “FHPP” (modular for Rockwell PLC)

Tab. 7.6 EDS files for FHPP with DeviceNet

The way in which you configure your network depends on the configuration software used. Follow the instructions of the controller manufacturer for registering the EDS file of the motor controller.

To simplify commissioning of the CMMP-AS-...-M3 with controllers from various manufacturers, you will find corresponding modules and application notes at
➔ www.festo.com/sp

7.5 CIP objects

This chapter describes only the implemented DeviceNet object model, i.e. how you can access the FHPP parameters via DeviceNet.

Data types
The following data types corresponding to the DeviceNet specification are used:

Type	Signed	Unsigned
8 bit	SINT	USINT
16 bit	INT	UINT
32 bit	DINT	UDINT

Tab. 7.7 Data types

Device Data Object (Object Class ID , Number of Instances)
This object supplies information to identify a device.
Object class ID: 100
Number of Instances: 1

Allocation	Name	Attribute	FHPP-PNU	Type
Version	Manufacturer hardware version	0x01	100.1	UINT
	Firmware version	0x02	101.1	UINT
	Version FHPP	0x03	102.1	UINT
Identification	Project identifier	0x07	113.1	UDINT
	Serial number controller	0x08	114.1	UDINT
	Manufacturer device name	0x09	120.1	SHORT_STRING
	User device name	0x0A	121.1	SHORT_STRING
	Drive manufacturer	0x0B	122.1	SHORT_STRING
	http address manufacturer	0x0C	123.1	SHORT_STRING
	Festo order number	0x0D	124.1	SHORT_STRING
	I/O Control + FCT Control	0x0E	125.1	USINT
Data Memory Control	Data Memory Control: Load default	0x14	127.1	USINT
	Data Memory Control: Save	0x15	127.2	USINT
	Data Memory Control: SW reset	0x16	127.3	USINT
	Encoder Data Memory Control	0x19	127.6	USINT

Tab. 7.8 Device Data Object

Process Data Object.

This object supplies demand and actual values for position, velocity and torque. The digital inputs and outputs can also be controlled.

Object Class ID: 103

Number of Instances: 1

Allocation	Name	Attribute	FHPP-PNU	Type
Position	Position: Actual value	0x01	300.1	DINT
	Position: Setpoint	0x02	300.2	DINT
	Position: Actual deviation	0x03	300.3	DINT
Torque	Torque: Actual value, “mNm”	0x04	301.1	DINT
	Torque: Setpoint, “mNm”	0x05	301.2	DINT
	Torque: Actual deviation	0x05	301.3	DINT
Digital Inputs/outputs	Digital Inputs: DIN 0 … 7	0x0A	303.1	USINT
	Digital Inputs: DIN 8 … 11	0x0B	303.2	USINT
	Dig. inputs: EA88_1: DIN1 … 8	0x0C	303.4	USINT
	Digital Outputs: DOUT 0 … 3	0x14	304.1	USINT
	Dig. outputs: EA88_1: DOUT1…8	0x15	304.3	USINT
Record control	Demand record number	0x20	400.1	USINT
	Actual record number	0x21	400.2	USINT
	Record status byte	0x22	400.3	USINT
Operating hour counter	Operating hours meter, “s”	0x23	305.3	UDINT
Velocity	Velocity: Actual value	0x24	310.1	DINT
	Velocity: Demand value	0x25	310.2	DINT
	Velocity: Actual deviation	0x26	310.3	DINT
Remaining Distance	Remaining distance for remaining distance message	0x38	1230.1	UDINT
Status Signal outputs	State signal outputs	0x3A	311.1	UDINT
	Trigger state	0x3B	311.2	UDINT
Other axis parameters	Torque feed forward	0x64	1080.1	DINT
	Setup velocity	0x65	1081.1	USINT
	Velocity override	0x65	1082.1	USINT

Tab. 7.9 Process Data Object

Project Data Object

This object supplies project information, i.e. common parameters for all devices of a machine.

Object Class ID: 105

Number of Instances: 1

Allocation	Name	Attribute	FHPP-PNU	Type
General project data	Project zero point	0x01	500.1	DINT
	Negative position limit	0x02	501.1	DINT
	Positive position limit	0x03	501.2	DINT
	Max. velocity	0x04	502.1	UDINT
	Max. acceleration	0x05	503.1	UDINT
	Max. jerk-free filter time, “ms”	0x07	505.1	UDINT
Teach	Teach target	0x14	520.1	USINT

Tab. 7.10 Project Data Object

Jog Mode Object

This object supplies information on the jog mode.

Object Class ID: 105

Number of Instances: 1

Allocation	Name	Attribute	FHPP-PNU	Type
Jog mode	Jog mode: Crawling velocity (phase 1)	0x1E	530.1	DINT
	Jog mode: Max. velocity (phase 2)	0x1F	531.1	DINT
	Jog mode: Acceleration	0x20	532.1	UDINT
	Jog mode: Deceleration	0x21	533.1	UDINT
	Jog mode: Slow motion time, “ms”	0x22	534.1	UDINT

Tab. 7.11 Jog Mode Object

Direct Mode Position Object

This object supplies information on the project via the direct mode position control.

Object Class ID: 105

Number of Instances: 1

Allocation	Name	Attribute	FHPP-PNU	Type
Direct mode position	Direct mode pos: Base velocity	0x28	540.1	DINT
	Direct mode pos: Acceleration	0x29	541.1	UDINT
	Direct mode pos: Deceleration	0x2A	542.1	UDINT
	Direct mode pos: Jerk-free filter time, “ms”	0x2E	546.1	UDINT

Tab. 7.12 Direct Mode Position Object

Direct Mode Torque Object

This object supplies information on the project via the direct mode torque object.

Object Class ID: 105

Number of Instances: 1

Allocation	Name	Attribute	FHPP-PNU	Type
Direct mode torque	Direct mode torque: Base torque ramp, “mNm/s”	0x32	550.1	UDINT
	Direct mode torque: Force target window, “mNm”	0x34	552.1	UINT
	Direct mode torque: Time window, “ms”	0x35	553.1	UINT
	Direct mode torque: Velocity limit	0x36	554.1	UDINT

Tab. 7.13 Direct Mode Torque Object

Direct Mode Velocity Object

This object supplies information on the project via the direct mode velocity control.

Object Class ID: 105

Number of Instances: 1

Allocation	Name	Attribute	FHPP-PNU	Type
Direct mode velocity:	Direct mode velocity: Base velocity ramp	0x3C	560.1	UDINT
	Direct mode velocity: Velocity window	0x3D	561.1	UINT
	Direct mode velocity: Velocity window time, “ms”	0x3E	562.1	UINT
	Direct mode velocity: Velocity threshold	0x3F	563.1	UINT
	Direct mode velocity: Velocity threshold time, “ms”	0x40	564.1	UINT
	Direct mode velocity: Torque limit, “mNm”	0x41	565.1	UDINT

Tab. 7.14 Direct Mode velocity Object

Direct Mode General Object

This object supplies general information on the project through the direct mode.

Object Class ID: 105

Number of Instances: 1

Allocation	Name	Attribute	FHPP-PNU	Type
Direct mode general	Direct mode general: Torque limit selector	0x50	580.1	SINT
	Direct mode general: Torque limit, “mNm”	0x51	581.1	UDINT

Tab. 7.15 Direct Mode General Object

Axis Parameter Object

This object supplies axis information, i.e. parameters for an individual device in a machine.

Object Class ID: 107

Number of Instances: 1

Allocation	Name	Attribute	FHPP-PNU	Type
Mechanics	Polarity	0x01	1000.1	USINT
	Encoder resolution: Increments	0x02	1001.1	UDINT
	Encoder resolution: Motor revolutions	0x03	1001.2	UDINT
	Gear ratio: Motor revolutions	0x04	1002.1	UDINT
	Gear ratio: Shaft revolutions	0x05	1002.2	UDINT
	Feed constant: Feed	0x06	1003.1	UDINT
	Feed constant: Shaft revolutions	0x07	1003.2	UDINT
	Position factor: Numerator	0x08	1004.1	UDINT
	Position factor: Divisor	0x09	1004.2	UDINT
	Axis parameter: X2A gear numerator	0x0B	1005.2	DINT
	Axis parameter: X2A gear divisor	0x0C	1005.3	DINT
	Velocity encoder factor: Numerator	0x0F	1006.1	UDINT
	Velocity encoder factor: Divisor	0x10	1006.2	UDINT
	Acceleration factor: Numerator	0x11	1007.1	UDINT
	Acceleration factor: Divisor	0x12	1007.2	UDINT

Tab. 7.16 Axis Parameter Object

Homing Object

This object supplies information on the project via homing.

Object Class ID: 107

Number of Instances: 1

Allocation	Name	Attribute	FHPP-PNU	Type
Homing	Offset axis zero point	0x14	1010.1	DINT
	Homing method	0x15	1011.1	SINT
	Homing: velocity (search for switch)	0x16	1012.1	UDINT
	Homing: velocity (search for zero)	0x17	1012.2	UDINT
	Homing: acceleration	0x18	1013.1	UDINT
	Homing required	0x19	1014.1	USINT
	Homing max. torque, “%”	0x1A	1015.1	USINT

Tab. 7.17 Homing Object

Controller Parameters Object

This object supplies information on the project via the controller.

Object Class ID: 107

Number of Instances: 1

Allocation	Name	Attribute	FHPP-PNU	Type
Controller parameters	Halt option code	0x1E	1020.1	UINT
	Position window	0x20	1022.1	UDINT
	Position window time, “ms”	0x21	1023.1	UINT
	Gain position controller	0x22	1024.18	UINT
	Gain velocity controller	0x23	1024.19	UINT
	Time velocity controller, “µs”	0x24	1024.20	UINT
	Gain current controller	0x25	1024.21	UINT
	Time current controller “µs”	0x26	1024.22	UINT
	Save position	0x28	1024.32	UINT
Motor data	Festo serial number + motor’s serial number	0x2C	1025.1	UDINT
	I^2t time motor, “ms”	0x2D	1025.3	UINT
Drive data	Power stage temperature	0x31	1026.1	UDINT
	Max. power stage temperature	0x32	1026.2	UDINT
	Nominal motor current, “mA”	0x33	1026.3	UDINT
	Current limit (thousandths of nominal motor current)	0x34	1026.4	UDINT
	Controller serial number	0x37	1026.7	UDINT

Tab. 7.18 Controller Parameters Object

Electronic Identification Plate Object

This object supplies information on the project via the electronic type plate.

Object Class ID: 107

Number of Instances: 1

Allocation	Name	Attribute	FHPP-PNU	Type
Type plate data	Max. current	0x40	1034.1	UINT
	Motor rated current, “mA”	0x41	1035.1	UDINT
	Motor rated torque, “mNm”	0x42	1036.1	UDINT
	Torque constant, “mNm/A”	0x43	1037.1	UDINT
Axis parameter, following error monitoring	Following error window	0x48	1044.1	UDINT
	as from FW 4.0.1501.2.3: Shutdown following error	0x4D	1044,2	UDINT
	Following error message delay, “ms”	0x49	1045.1	UINT

Tab. 7.19 Electronic Identification Plate Object

Standstill Object

This object supplies information on the project via the standstill monitoring.

Object Class ID: 107

Number of Instances: 1

Allocation	Name	Attribute	FHPP-PNU	Type
Standstill monitoring	Position demand value	0x44	1040.1	DINT
	Position actual value	0x45	1041.1	DINT
	Standstill position window	0x46	1042.1	UDINT
	Standstill timeout, “ms”	0x47	1043.1	UINT

Tab. 7.20 Standstill Object

Fault Buffer Administration Parameters Object

This object supplies information on the project via the diagnostic memory.

Object Class ID: 102

Number of Instances: 1

Allocation	Name	Attribute	FHPP-PNU	Type
Error	Error buffer: Incoming/outgoing error	0x01	204.1	USINT
	Error buffer: Resolution time stamp	0x02	204.2	USINT
	Error buffer: Number of entries	0x04	204.4	USINT
Warnings	Warning buffer: Incoming/outgoing warning	0x05	214.1	USINT
	Warning buffer: Resolution time stamp	0x06	214.2	USINT
	Warning buffer: Number of entries	0x08	214.4	USINT

Tab. 7.21 Fault Buffer Administration Parameters Object

Error Record List Object

This object represents the error memory.

An individual object group is available for each sub-Index (x) from 1 ... 32.

Object Class ID: 101

Number of Instances: 32

Allocation	Name	Attribute	FHPP-PNU	Type
Diagnostic memory	Diagnosis	0x01	200 x	USINT
	Error number	0x02	201.x	UINT
	Time stamp “s”	0x03	202 x	UDINT
	Additional information	0x04	203 x	UDINT

Tab. 7.22 Error Record List Object

Warning Record List Object

This object represents the warning memory.

An individual object group is available for each sub-index (x) from 1 … 16.

Object Class ID: 108

Number of Instances: 16

Allocation	Name	Attribute	FHPP-PNU	Type
Warning memory	Diagnosis	0x01	210.x	USINT
	Warning number	0x02	211.x	UINT
	Time stamp “s”	0x03	212.x	UDINT
	Additional information	0x04	213.x	UDINT

Tab. 7.23 Warning Record List Object

Record List Object

This object represents the data record list. Data records can be processed automatically and also linked to each other.

An individual object group is available for each sub-index (x) from 1 … 250.

Object Class ID: 104

Number of Instances: 250

Allocation	Name	Attribute	FHPP-PNU	Type
Record data	Record Control Byte 1	0x01	401.x	USINT
	Record Control Byte 2	0x02	402.x	USINT
	Setpoint	0x04	404.x	DINT
	Velocity	0x06	406.x	UDINT
	Acceleration	0x07	407.x	UDINT
	Deceleration	0x08	408.x	UDINT
	velocity limit (in torque control)	0x0C	412.x	UDINT
	Jerk-free filter time, “ms”	0x0D	413.x	UDINT
	Following Position	0x10	416.x	USINT
	Torque limitation “mNm”	0x12	418.x	UDINT
	CAM disc number	0x13	419.x	USINT
	Remaining distance for message	0x14	420.x	UDINT
	Record Control Byte 3	0x15	421.x	USINT

Tab. 7.24 Record List Object

FHPP+ Data

This object represents the output and input data of the controller.

An individual object group is available for each sub-index (x) from 1 ... 10.

Object Class ID: 115

Number of Instances: 16

Allocation	Name	Attribute	FHPP-PNU	Type
FHPP+ Data	FHPP_Receive_Telegram	0x01	40.x	UDINT
	FHPP_Respond_Telegram	0x02	41.x	UDINT

Tab. 7.25 FHPP+ Data List Object

FHPP+ Status

This object represents the status of the FHPP+ data.

Object Class ID: 116

Number of Instances: 1

Allocation	Name	Attribute	FHPP-PNU	Type
FHPP+ Status	FHPP_Rec_Telegram_State	0x01	42.1	UDINT
	FHPP_Resp_Telegram_State	0x01	43.1	UDINT

Tab. 7.26 FHPP+ Status List Object

Safety

This object represents the safety status of the motor controller.

Object Class ID: 107

Number of Instances: 1

Allocation	Name	Attribute	FHPP-PNU	Type
Safety Status	safety state	0x01	280.0	UDINT
Safety VOUT	from FW 4.0.1501.2.1: FSM_VOUT_0_31	0x02	281.1	UDINT
	from FW 4.0.1501.2.1: FSM_VOUT_32_63	0x03	281.2	UDINT
Safety LOUT	from FW 4.0.1501.2.1: FSM_IO	0x04	282.1	UDINT

Tab. 7.27 Safety Status List Object

Operation Data

This object represents the function data of the cam disc function.

Object Class ID: 113

Number of Instances: 1

Allocation	Name	Attribute	FHPP-PNU	Type
Cam disc	Cam disc number	0x01	700.1	USINT
	Master start position	0x02	701.1	DINT
	Position: Setpoint virtual master	0x03	300.4	DINT
Synchronisation	Sync.: Input configuration	0x0B	710.1	UDINT
	Sync.: Gear ratio (Motor Revolutions)	0x0C	711.1	UDINT
	Sync.: Gear ratio (Shaft Revolutions)	0x0D	711.2	UDINT
Encoder:	Encoder emulation: Output configuration	0x15	720.1	UDINT
Trigger	Position trigger control	0x1F	730.1	UDINT

Tab. 7.28 Operation Data List Object

Trigger Parameters

This object represents the trigger information.

An individual object group is available for each sub-index (x) from 1 … 4.

Object Class ID: 114

Number of Instances: 4

Allocation	Name	Attribute	FHPP-PNU	Type
Trigger Parameter	Position trigger low	0x20	731.x	DINT
	Position trigger high	0x21	732.x	DINT
	Rotor Position trigger low	0x22	733.x	DINT
	Rotor Position trigger high	0x23	734.x	DINT

Tab. 7.29 Trigger Parameters List Object

8 EtherCAT with FHPP

M3

This chapter is only applicable for the motor controller CMMP-AS-...**-M3**.

8.1 Overview

This part of the documentation describes the connection and configuration of the motor controller CMMP-AS-...-M3 in an EtherCAT network. It is directed at people who are already familiar with this bus protocol.

The EtherCAT fieldbus system means "Ethernet for Controller and Automation Technology" and was developed by Beckhof Industrie. It is managed by the international EtherCAT Technology Group (ETG) organisation and supports and is designed as an open technology, which is standardised by the International Electrotechnical Commission (IEC).
EtherCAT is a fieldbus system based on Ethernet, which sets new speed standards and can be handled like a fieldbus, thanks to flexible topology (line, tree, star) and simple configuration.
The EtherCAT protocol is transported with a special standardised Ethernet type directly in the Ethernet frame in accordance with IEEE802.3. The slaves can broadcast, multicast and communicate laterally.

Abbreviation	Significance
CoE	CANopen over EtherCAT protocol
ESC	EtherCAT Slave Controller
PDI	Process Data Interface

Tab. 8.1 EtherCAT-specific abbreviations

Festo supports the CoE protocol (CANopen over EtherCAT) in the CMMP with the Beckhoff FPGA ESC20. CiA402 and FHPP are supported as data profiles.

EtherCAT CAMC-EC interface characteristics

The EtherCAT interface has the following performance characteristics:

- Can be mechanically fully integrated into the CMMP-AS-...-M3 series motor controllers
- EtherCAT conforming to IEEE-802.3u (100Base-TX) with 100Mbps (full-duplex)
- Star and line topology
- Plug connector: RJ45
- Electrically isolated EtherCAT interface
- Communication cycle : min. 1 ms
- Max. 127 slaves
- EtherCAT slave implementation based on the Beckhoff FPGA ESC20
- Support of the "Distributed Clocks" feature for time-synchronous setpoint value transfer
- LED displays for ready status and link detect
- SDO communication corresponding to CANopen CiA 402 ➔ description CiA 402

8.2 EtherCAT CAMC-EC interface

The EtherCAT interface is implemented for the motor controllers CMMP-AS-...-M3 through the optional interface CAMC-EC. The interface is mounted in slot Ext2. The EtherCAT connection is designed in the form of two RJ45 sockets at the interface CAMC-EC.

8.2.1 Connection and display components

1 LED 1 (Port 1, Run)
2 LED 2 (Port 2)
3 Interface X1 (Port 1)
4 Interface X2 (Port 2)

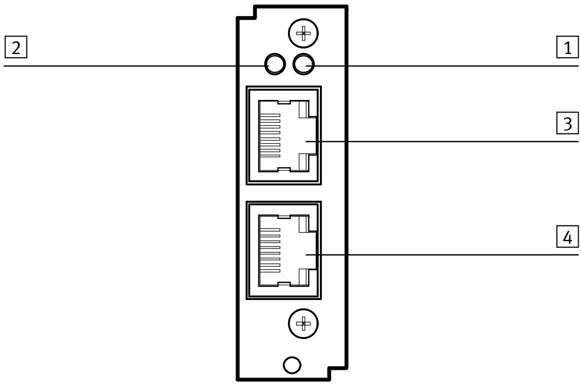


Fig. 8.1 Connection and display components at the EtherCAT interface

The EtherCAT CAMC-EC interface allows the CMMP motor controller to be connected to the EtherCAT fieldbus system. Communication over the EtherCAT interface (IEEE 802.3u) takes place with an EtherCAT standard cabling.

8.2.2 EtherCAT LEDs

The EtherCAT LEDs display the communication status.

LED	Status:	Meaning:
LED 1	Off	No connection to Port 1
	Lights up red	Connection active at Port 1
	Lights up green	Run
LED 2	Off	No connection at Port 2
	Lights up red	Connection active at Port 2

Tab. 8.2 EtherCAT LEDs

8.2.3 Pin allocation and cable specifications

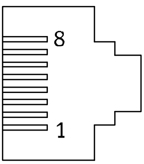
Design of plug connectors X1 and X2

RJ45 sockets	Function
X1 (RJ45 socket on top)	Uplink to the master or a previous station of a series connection (e.g. multiple motor controllers)
X2 (RJ45 socket underneath)	Uplink to the master, end of a series connection or connection of additional downstream stations

Tab. 8.3 RJ45 sockets

With several motor controllers, attention must be paid to the wiring, since trouble-free operation with DC (distributed clocks) cannot be ensured otherwise.

Allocation of the plug connectors X1 and X2

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

Tab. 8.4 Allocation of the plug connectors X1 and X2

EtherCAT interface specification

Value	Function
EtherCAT interface, signal level	0 ... 2.5 V DC
EtherCAT interface, differential voltage	1.9 ... 2.1 V DC

Tab. 8.5 RJ45 sockets

Type and design of cable

Shielded twisted-pair STP, Cat.5 cables must be used for cabling.
The listed cable names refer to cables made by LAPP and Lützе. They have proven themselves in practice and are successfully in use in many applications. However, comparable cables by other manufacturers can also be used.

Cable length	Order number
EtherCAT cable from LAPP	
0.5 m	90PCLC50000
1 m	90PCLC500010
2 m	90PCLC500020G
5 m	90PCLC500050G
EtherCAT cable from Lütze	
0.5 m	192000
1 m	19201
5 m	19204

Tab. 8.6 EtherCAT cable

Errors due to inappropriate bus cable

As very high baud rates can occur, we recommend that you use only the standardised cables and plug connectors. In some cases, they have additional diagnostics options and allow the fieldbus interface to be analysed rapidly in the event of errors.
When setting up the EtherCAT network, you must unconditionally follow the advice in the relevant literature or the subsequent information and instructions in order to maintain a stable, trouble-free system. If the system is not cabled properly, EtherCAT bus malfunctions can occur during operation. These can cause the CMMP motor controller to shut off with an error for safety reasons.

Bus termination

No external bus terminations are required. The EtherCAT interface monitors its two ports and terminates the bus automatically (loop-back function).

8.3 Configuration of EtherCAT stations

Several steps are required in order to produce an operational EtherCAT interface. This section provides an overview of the steps required by the slave for parameterisation and configuration. As some parameters are only effective after saving and reset of the controller, we recommend that commissioning with the FCT should be carried out first without connection to the EtherCAT bus.

Note: Parameterisation and commissioning of the motor controller is possible with EtherCAT control interface only with connected master.

Instructions on commissioning with the Festo Configuration Tool can be found in the Help for the device-specific FCT plug-in.

When designing the EtherCAT interface, the user must therefore make these determinations. Only then should parameterisation of the fieldbus connection take place on both pages. We recommend that parameterisation of the slave should be undertaken first. Then the master should be configured. With correct parameterisation, the application is ready immediately without communication errors.

We recommend the following procedure:

1. Activation of the bus communication.
   EtherCAT communication is automatically started through the CMMP-AS-..-M3 if it detects after switch-on that an EtherCAT interface is plugged in.
   Communication cannot be deactivated by flipping DIL switch 8.
2. Parameterisation and commissioning with the Festo Configuration Tool (FCT).
   In addition, the following settings on the fieldbus page:
   – Festo FHPP cycle time (Operation Parameters tab)
   – Festo FHPP protocol (Operation Parameters tab)
   – Physical units (Factor Group tab)
   – Optional use of FHPP+ (FHPP+ Editor tab)

   Observe that the parameterisation of the EtherCAT function only remains intact after a reset if the parameter set of the motor controller was saved.

3. Configuration of the EtherCAT master ➔ section 8.4.

8.3.1 Setting of the physical units (factor group)

In order for a fieldbus master to exchange position, velocity and acceleration data in physical units (e.g. mm, mm/s, mm/s^2) with the motor controller, it must be parameterised via the factor group ➔ section A.1.
Parameterisation can be carried out via FCT or the fieldbus.

8.3.2 Setting of the optional use of FPC and FHPP+

Besides the control or status bytes and the FPC, additional I/O data can be transmitted ➔ section C.2. This is set via the FCT (page Fieldbus, tab FHPP+ Editor).

8.4 FHPP with EtherCAT

The FHPP data are divided among several process data objects for CANopen communication. Mapping is automatically determined through parameterisation with the FCT (page Fieldbus, tab FHPP+ Editor).

Supported process data objects	Paramet-erisation1)	PDO as-signment	Data mapping of the FHPP data
TxPDO 1	Standard	0x0001	FHPP Standard 8 bytes status data
TxPDO 2	Optional or	0x0002	FPC parameter channel Transmission of requested FHPP parameter values
	Optional	0x0003	FHPP+ data Mapping = 8 bytes of FHPP+ data
TxPDO 3	Optional	0x0004	FHPP+ data Mapping = 8 bytes of FHPP+ data
TxPDO 4	Optional	0x0005	FHPP+ data Mapping = 8 bytes of FHPP+ data
RxPDO 1	Standard	0x0010	FHPP Standard 8 byte control data
RxPDO 2	Optional or	0x0011	FPC parameter channel Read/write FHPP parameter values
	Optional	0x0012	FHPP+ data Mapping = 8 bytes of FHPP+ data
RxPDO 3	Optional	0x0013	FHPP+ data Mapping = 8 bytes of FHPP+ data
RxPDO 4	Optional	0x0014	FHPP+ data Mapping = 8 bytes of FHPP+ data

1) Optional if parameterised through the FCT (page Fieldbus – tab FHPP+ Editor)

Tab. 8.7 Cyclical process data objects

8.5 Configuration EtherCAT Master

In order to connect EtherCAT slave devices easily to an EtherCAT master, there must be a description file for every EtherCAT slave device. This description file is comparable to the EDS files for the CANopen fieldbus system or the GSD files for Profibus. In contrast to the latter, the EtherCAT description file is in the XML format, as is often used for internet and web applications, and contains information on the following features of the EtherCAT slave devices:

- Information on the device manufacturer
- Name, type and version number of the device
- Type and version number of the protocol to be used for this device (e.g. CANopen over Ethernet, ...)
- Parameterisation of the device and configuration of the process data

This file contains the complete parameterisation of the slave, including the parameterisation of the Sync Manager and the PDOs.

The XML file is included on a CD-ROM supplied with the motor controller.

XML file	**Description**
Festo_CMMP-AS_V4p0_FHPP.xml	Motor controller CMMP-AS-...-M3 with protocol “FHPP”
Festo_CMMP-AS_V4p0_CIA402_IP7.xml	Motor controller CMMP-AS-...-M3 with protocol “CiA 402”

Tab. 8.8 XML file

You can find the most current version under: ➔ www.festo.com/sp

To simplify commissioning of the CMMP-AS-...-M3 with controllers from various manufacturers, you will find corresponding modules and application notes at
➔ www.festo.com/sp

8.6 CANopen communication interface

User protocols are tunnelled via EtherCAT. For the CANopen over EtherCAT protocol (CoE) supported by the CMMP-AS-...-M3, most objects for the communication layer are supported by EtherCAT in accordance with CiA 301. This primarily involves objects for setting up communication between masters and slaves.
In general, the following services and object groups are supported by the EtherCAT CoE implementation in the motor controller CMMP-AS-...-M3:

Services/object groups		**Function**
SDO	Service Data Object	Used for normal parameterisation of the motor controller.
PDO	Process Data Object.	Fast exchange of process data (e.g. actual velocity) possible.
EMCY	Emergency Message	Transmission of error messages.

Tab. 8.9 Supported services and object groups

The individual objects which can be addressed via the CoE protocol in the motor controller CMMP-AS-...-M3 are internally forwarded to the existing CANopen implementation and processed there. However, some new CANopen objects are added under the CoE implementation under EtherCAT, which are required for special connection via CoE. This is the result of the revised communication interface between the EtherCAT protocol and the CANopen protocol. A so-called Sync Manager is used to control the transmission of PDOs and SDOs via the two EtherCAT transfer types (mailbox and process data protocol).
This Sync Manager and the necessary configuration steps for operation of the CMMP-AS-...-M3 under EtherCAT-CoE are described in chapter 8.6.1 “Configuration of the Communication Interface”. The additional objects are described in chapter 8.6.2 “New and revised objects under CoE”.
Also, some CANopen objects of the CMMP-AS-...-M3, which are available under a normal CANopen connection, are not supported via a CoE connection over EtherCAT.
A list of the CANopen objects not supported under CoE is provided in chapter 8.6.3 “Objects not supported under CoE”.

8.6.1 Configuration of the Communication Interface

As already described in the previous chapter, the EtherCAT protocol uses two different transfer types for transmission of the device and user protocols, such as the CANopen-over-EtherCAT protocol (CoE) used by the CMMP-AS-...-M3. These two transfer types are the mailbox telegram protocol for non-cyclic data and the process data telegram protocol for transmission of cyclic data.

These two transfer types are used for the different CANopen transfer types for the CoE protocol. They are used as follows:

Telegram protocol	Description	Reference
Mailbox	This transfer type is used to transmit the Service Data Objects (SDOs) defined under CANopen. They are transmitted to EtherCAT in SDO frames.	➔ chapter 8.8 “SDO Frame”
Process Data	This transfer type is used to transmit the Process Data Objects (PDOs) defined under CANopen, which are used to exchange cyclic data. They are transmitted to EtherCAT in PDO frames.	➔ chapter 8.9 “PDO Frame”

Tab. 8.10 Telegram protocol – description

In general, these two transfer types allow all PDOs and SDOs to be used exactly as they are defined for the CANopen protocol for CMMP-AS-...-M3.
However, parameterisation of PDOs and SDOs for sending objects via EtherCAT is different from the settings which must be made under CANopen. In order to link the CANopen objects to be exchanged via PDO or SDO transfers between masters and slaves into the EtherCAT protocol, a so-called Sync Manager is implemented under EtherCAT.
This Sync Manager is used to link the data of the PDOs and SDOs to be sent to the EtherCAT telegrams. To accomplish this, the Sync Manager provides multiple Sync channels which can each implement a CANopen data channel (Receive SDO, Transmit SDO, Receive PDO or Transmit PDO) on the EtherCAT telegram.
The figure shows how the Sync Manager is linked to the system:

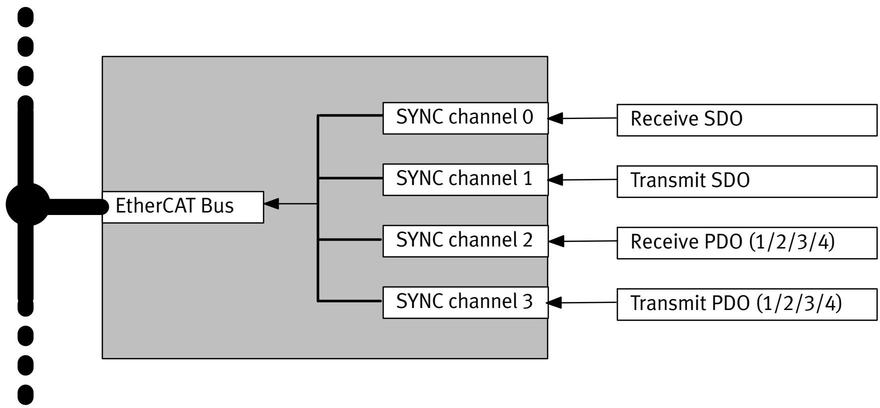


Fig. 8.2 Sample mapping of the SDOs and PDOs to the Sync channels

All objects are sent via so-called Sync channels. The data from these channels is automatically linked to the EtherCAT data flow and transmitted. The EtherCAT implementation in the motor controller CMMP-AS-...-M3 supports four such Sync channels.
For this reason, additional mapping of the SDOs and PDOs to the Sync channels is required compared with CANopen. This occurs via the so-called Sync Manager objects (objects $1C00_h$ and $1C10_h$ ... $1C13_h$ ➔ chapter 8.6.2). These objects are described in more detail below.

These Sync channels are permanently allocated to the individual transfer types and cannot be changed by the user. The allocation is as follows:

- Sync channel 0: Mailbox telegram protocol for incoming SDOs (Master => Slave)
- Sync channel 1: Mailbox telegram protocol for outgoing SDOs (Master <= Slave)
- Sync channel 2: Process data telegram protocol for incoming PDOs (Master => Slave).
  The object $1C12_h$ must be observed here.
- Sync channel 3: Process data telegram protocol for outgoing PDOs (Master <= Slave).
  The object $1C13_h$ must be observed here.

The parameterisation of the individual PDOs is set via objects 1600_h to 1603_h (Receive PDOs) and $1A00_h$ to $1A03_h$ (Transmit PDOs). Parameterisation of the PDOs is carried out as described in chapter 2.6 “Access procedure”.
Fundamentally, the Sync channels can only be set and the PDOs only configured in the “Pre-Operational” status.

It is not intended to parameterise the slave under EtherCAT. The device description files are available for this purpose. They prescribe the total parameterisation, including PDO parameterisation, which is used by the master during initialisation.
All changes to the parameterisation should therefore not be made by hand, but in the device description files. For this purpose, sections of the device description files that are important for the user are described in more detail in section 8.5.

The Sync channels described here are NOT the same as the Sync telegrams familiar from CANopen. CANopen Sync telegrams can still be transmitted as SDOs via the SDO interface implemented under CoE, but do not directly influence the Sync channels described above.

8.6.2 New and revised objects under CoE

The following table contains an overview of the indices and subindices used for CANopen-compatible communication objects, which are inserted in the range from 1000_h to $1FFF_h$ for the EtherCAT fieldbus system. These primarily replace the communication parameters in accordance with CiA 301.

Object	Significance	Permitted with
1000_h	Device type	Device control identifier
1018_h	Identity object	Vendor ID, product code, revision, serial number
1100_h	EtherCAT fixed station address	Fixed address assigned to the slave during initialisation by the master
1600_h	1. RxPDO Mapping	Identifier of the 1th Receive PDO
1601_h	2. RxPDO Mapping	Identifier of the 2th Receive PDO
1602_h	3. RxPDO Mapping	Identifier of the 3th Receive PDO
1603_h	4. RxPDO Mapping	Identifier of the 4th Receive PDO
$1A00_h$	1. TxPDO Mapping	Identifier of the 1th Transmit PDO
$1A01_h$	2. TxPDO Mapping	Identifier of the 2th Transmit PDO
$1A02_h$	3. TxPDO Mapping	Identifier of the 3th Transmit PDO
$1A03_h$	4. TxPDO Mapping	Identifier of the 4th Transmit PDO
$1C00_h$	Sync Manager Communication Type	Object for configuring the individual Sync channels (SDO or PDO Transfer)
$1C10_h$	Sync Manager PDO Mapping for Sync Channel 0	Assignment of the Sync channel 0 to a PDO/SDO (Channel 0 is always reserved for Mailbox Receive SDO Transfer)
$1C11_h$	Sync Manager PDO Mapping for Sync Channel 1	Assignment of the Sync channel 1 to a PDO/SDO (Channel 1 is always reserved for Mailbox Send SDO Transfer)
$1C12_h$	Sync Manager PDO Mapping for Sync Channel 2	Assignment of the Sync channel 2 to a PDO (Channel 2 is reserved for Receive PDOs)
$1C13_h$	Sync Manager PDO Mapping for Sync Channel 3	Assignment of the Sync channel 3 to a PDO (Channel 3 is reserved for Transmit PDOs)

Tab. 8.11 New and revised communication objects

The subsequent chapters describe the objects $1C00_h$ and $1C10_h$...$1C13_h$ in more detail, as they are only defined and implemented under the EtherCAT CoE protocol and therefore are not documented in the CANopen manual for the motor controller CMMP-AS-...-M3.

The motor controller CMMP-AS-...-M3 with the EtherCAT interface supports four Receive PDOs (RxPDO) and four Transmit PDOs (TxPDO).
Objects 1008_h, 1009_h and $100A_h$ are not supported by the CMMP-AS-...-M3, as plain text strings cannot be read from the motor controller.

Object 1100$_h$ - EtherCAT fixed station address

This object allocates a unique address to the slave during the initialisation phase. The object has the following significance:

Index	**1100$_h$**
Name	EtherCAT fixed station address
Object Code	Var
Data Type	uint16
Access	ro
PDO mapping	no
Value Range	0 … FFFF$_h$
Default Value	0

Object 1C00$_h$ - Sync Manager Communication Type

This object allows the transfer type for the various channels of the EtherCAT Sync Manager to be read. As the CMMP-AS-...-M3 only supports the first four Sync channels under the EtherCAT CoE protocol, the following objects are "read only".

The Sync Manager for the CMMP-AS-...-M3 is configured as a result. The objects have the following significance:

Index	**1C00$_h$**
Name	Sync Manager Communication Type
Object Code	Array
Data Type	uint8

Sub-Index	**00$_h$**
Description	Number of Used Sync Manager Channels
Access	ro
PDO mapping	no
Value Range	4
Default Value	4

Sub-Index	**01$_h$**
Description	Communication Type Sync Channel 0
Access	ro
PDO mapping	no
Value Range	2: Mailbox Transmit (Master => Slave)
Default Value	2: Mailbox Transmit (Master => Slave)

Sub-Index	**02_h**
Description	Communication Type Sync Channel 1
Access	ro
PDO mapping	no
Value Range	2: Mailbox Transmit (Master <= Slave)
Default Value	2: Mailbox Transmit (Master <= Slave)

Index	**03_h**
Description	Communication Type Sync Channel 2
Access	ro
PDO mapping	no
Value Range	0: unused 3: Process Data Output (RxPDO / Master => Slave)
Default Value	3

Sub-Index	**04_h**
Description	Communication Type Sync Channel 3
Access	ro
PDO mapping	no
Value Range	0: unused 4: Process Data Input (TxPDO/Master <= Slave)
Default Value	4

Object $1C10_h$ - Sync Manager Channel 0 (Mailbox Receive)

This object allows a PDO to be configured for Sync channel 0. As Sync channel 0 is always allocated to the mailbox telegram protocol, the user cannot change this object. The object therefore always has the following values:

Index	**$1C10_h$**
Name	Sync Manager Channel 0 (Mailbox Receive)
Object Code	Array
Data Type	uint8

Sub-Index	**00_h**
Description	Number of assigned PDOs
Access	ro
PDO mapping	no
Value Range	0 (no PDO assigned to this channel)
Default Value	0 (no PDO assigned to this channel)

The name “Number of assigned PDOs” assigned by the EtherCAT specification for Sub-index 0 of these objects is confusing here, as Sync Manager channels 0 and 1 are always allocated through the mailbox telegram. SDOs are always transmitted in this telegram type under EtherCAT CoE. Sub-index 0 of these two objects is therefore unused.

Object 1C11$_h$ - Sync Manager Channel 1 (Mailbox Send)

This object allows a PDO to be configured for Sync channel 1. As Sync channel 1 is always allocated to the mailbox telegram protocol, the user cannot change this object. The object therefore always has the following values:

Index	**1C11$_h$**
Name	Sync Manager Channel 1 (Mailbox Send)
Object Code	Array
Data Type	uint8

Sub-Index	**00$_h$**
Description	Number of assigned PDOs
Access	ro
PDO mapping	no
Value Range	0 (no PDO assigned to this channel)
Default Value	0 (no PDO assigned to this channel)

Object 1C12$_h$ - Sync Manager Channel 2 (Process Data Output)

This object allows a PDO to be configured for Sync channel 2. Sync channel 2 is permanently assigned for the reception of Receive PDOs (Master => Slave). In this object, the number of PDOs assigned to this Sync channel must be set under sub-index 0.
The object number of the PDO to be allocated to the channel is subsequently entered in sub-indices 1 to 4. Only the object numbers of the previously configured Receive PDOs can be used for this (object 1600$_h$ ... 1603$_h$).
In the current implementation, the data of the objects below is not evaluated further by the firmware of the motor controller.
The CANopen configuration of the PDOs is used for evaluation under EtherCAT.

Index	**1C12$_h$**
Name	Sync Manager Channel 2 (Process Data Output)
Object Code	Array
Data Type	uint8

Sub-Index	**00_h**
Description	Number of assigned PDOs
Access	rw
PDO mapping	no
Value Range	0: no PDO assigned to this channel 1: one PDO assigned to this channel 2: two PDOs assigned to this channel 3: three PDOs assigned to this channel 4: four PDOs assigned to this channel
Default Value	0: no PDO assigned to this channel

Sub-Index	**01_h**
Description	PDO mapping object number of assigned RxPDO
Access	rw
PDO mapping	no
Value Range	1600_h: first Receive PDO
Default Value	1600_h: first Receive PDO

Sub-Index	**02_h**
Description	PDO mapping object number of assigned RxPDO
Access	rw
PDO mapping	no
Value Range	1601_h: second Receive PDO
Default Value	1601_h: second Receive PDO

Sub-Index	**03_h**
Description	PDO mapping object number of assigned RxPDO
Access	rw
PDO mapping	no
Value Range	1602_h: third Receive PDO
Default Value	1602_h: third Receive PDO

Sub-Index	**04_h**
Description	PDO mapping object number of assigned RxPDO
Access	rw
PDO mapping	no
Value Range	1603_h: fourth Receive PDO
Default Value	1603_h: fourth Receive PDO

Object $1C13_h$ - Sync Manager Channel 3 (Process Data Input)

This object allows a PDO to be configured for Sync channel 3. Sync channel 3 is permanently assigned for sending Transmit PDOs (Master <= Slave). In this object, the number of PDOs assigned to this Sync channel must be set under sub-index 0.

The object number of the PDO to be allocated to the channel is subsequently entered in sub-indices 1 to 4. Only the object numbers of the previously configured Transmit PDOs can be used for this ($1A00_h$ to $1A03_h$).

Index	**$1C13_h$**
Name	Sync Manager Channel 3 (Process Data Input)
Object Code	Array
Data Type	uint8

Sub-Index	**00_h**
Description	Number of assigned PDOs
Access	rw
PDO mapping	no
Value Range	0: no PDO assigned to this channel 1: one PDO assigned to this channel 2: two PDOs assigned to this channel 3: three PDOs assigned to this channel 4: four PDOs assigned to this channel
Default Value	0: no PDO assigned to this channel

Sub-Index	**01_h**
Description	PDO mapping object number of assigned TxPDO
Access	rw
PDO mapping	no
Value Range	$1A00_h$: first Transmit PDO
Default Value	$1A00_h$: first Transmit PDO

Sub-Index	**02_h**
Description	PDO mapping object number of assigned TxPDO
Access	rw
PDO mapping	no
Value Range	$1A01_h$: second Transmit PDO
Default Value	$1A01_h$: second Transmit PDO

Sub-Index	**03_h**
Description	PDO mapping object number of assigned TxPDO
Access	rw
PDO mapping	no
Value Range	$1A02_h$: third Transmit PDO
Default Value	$1A02_h$: third Transmit PDO

Sub-Index	**04_h**
Description	PDO mapping object number of assigned TxPDO
Access	rw
PDO mapping	no
Value Range	$1A03_h$: fourth Transmit PDO
Default Value	$1A03_h$: fourth Transmit PDO

8.6.3 Objects not supported under CoE

When connecting the CMMP-AS-...-M3 under “CANopen over EtherCAT”, some CANopen objects, which are available under a direct connection of the CMMP-AS-...-M3 via CiA 402, are not supported. These objects are listed in the table below:

Identifier	Name	Significance
1008_h	Manufacturer Device Name (String)	Device name (object is not available)
1009_h	Manufacturer Hardware Version (String)	HW version (object is not available)
$100A_h$	Manufacturer Software Version (String)	SW version (object is not available)
6089_h	position_notation_index	Specifies the number of decimal places for displaying the position values in the controller. The object is only available as a data container. The firmware is not evaluated further.
$608A_h$	position_dimension_index	Specifies the unit for displaying the position values in the controller. The object is only available as a data container. The firmware is not evaluated further.
$608B_h$	velocity_notation_index	Specifies the number of decimal places for displaying the velocity values in the controller. The object is only available as a data container. The firmware is not evaluated further.
$608C_h$	velocity_dimension_index	Specifies the unit for displaying the velocity values in the controller. The object is only available as a data container. The firmware is not evaluated further.
$608D_h$	acceleration_notation_index	Specifies the number of decimal places for displaying the acceleration values in the controller. The object is only available as a data container. The firmware is not evaluated further.
$608E_h$	acceleration_dimension_index	Specifies the unit for displaying the acceleration values in the controller. The object is only available as a data container. The firmware is not evaluated further.

Tab. 8.12 Unsupported communication objects

8.7 Communication finite state machine

As in almost all fieldbus interfaces for motor controllers, the connected slave (in this case the motor controller CMMP-AS-...-M3) must first be initialised by the master before it can be used by the master in an application. For this purpose, a finite state machine is defined for communication, to specify a fixed sequence of actions for this initialisation process.
A finite state machine is also defined for the EtherCAT interface. Changes between the individual statuses of the finite state machine may only occur between specific statuses, and are always initiated by the master. Slaves may not implement status changes independently. The individual statuses and the permitted status changes are described in the following tables and figures.

Status	Description
Power ON	The device has been switched on. It initialises itself and switches directly to the “Init” status.
Init	In this status, the EtherCAT fieldbus is synchronised by the master. This includes setting up the asynchronous communication between master and slave (mailbox telegram protocol). There is no direct communication between the master and slave yet. The configuration starts, saved values are loaded. When all devices are connected to the bus and configured, the status switches to “Pre-Operational”.
Pre-Operational	In this status, asynchronous communication between the master and slave is active. The master uses this status to set up possible cyclic communication via PDOs and use acyclic communication for necessary parameterisation. If this status runs without errors, the master switches to the “Safe-Operational” status.
Safe-Operational	This status is used to set all equipment connected to the EtherCAT bus to a safe status. The slave sends up-to-date actual values to the master but ignores new setpoint values from the master and uses safe default values instead. If this status runs without errors, the master switches to the “Operational” status.
Operational	In this status, both acyclic and cyclic communication are active. Masters and slaves exchange target and actual value data. In this status, the CMMP-AS-...-M3 can be enabled and travel via the CoE protocol.

Tab. 8.13 Statuses of communication finite state machine

Only transitions in accordance with Fig. 8.3 are permitted between the individual statuses of the communication finite state machine:

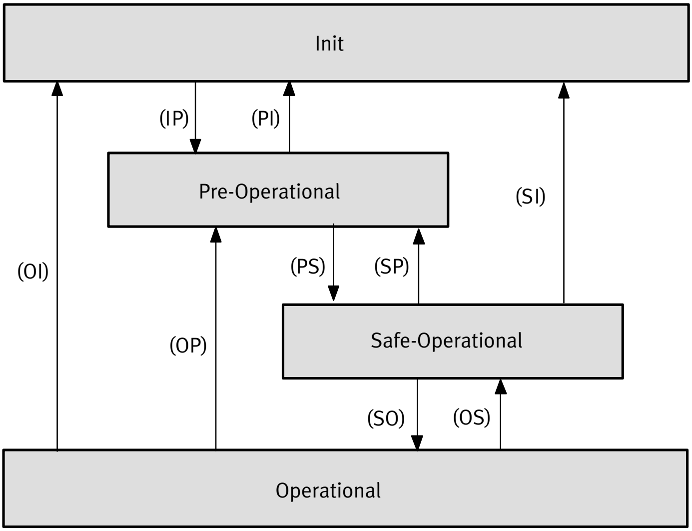


Fig. 8.3 Communication finite state machine

The transitions are described individually in the following table.

Status transition	Status
IP	Start of acyclic communication (mailbox telegram protocol)
PI	Stop of acyclic communication (mailbox telegram protocol)
PS	Start Inputs Update: start of cyclic communication (process data telegram protocol) Slave sends actual values to master. The slave ignores setpoint values from the master and uses internal default values.
SP	Stop Input Update: stop of cyclic communication (process data telegram protocol). The slave no longer sends actual values to the master.
SO	Start Output Update: The slave evaluates up-to-date setpoint specifications from the master.
OS	Stop Output Update: The slave ignores setpoint values from the master and uses internal default values.
OP	Stop Output Update, Stop Input Update: stop of cyclic communication (process data telegram protocol). The slave no longer sends actual values to the master, and the master no longer sends setpoint values to the slave.

Status transition	Status
SI	Stop Input Update, Stop Mailbox Communication: Stop of cyclic communication (process data telegram protocol) and stop of acyclic communication (mailbox telegram protocol). The slave no longer sends actual values to the master, and the master no longer sends setpoint values to the slave.
OI	Stop Output Update, Stop Input Update, Stop Mailbox Communication: Stop of cyclic communication (process data telegram protocol) and stop of acyclic communication (mailbox telegram protocol). The slave no longer sends actual values to the master, and the master no longer sends setpoint values to the slave.

Tab. 8.14 Status transitions

In the EtherCAT finite state machine, the “Bootstrap” status is also specified in addition to the statuses listed here. This status is not implemented for the motor controller CMMP-AS-...-M3.

8.7.1 Differences between the finite state machines of CANopen and EtherCAT

When operating the CMMP-AS-...-M3 via the EtherCAT-CoE protocol, the EtherCAT finite state machine is used instead of the CANopen NMT finite state machine. This differs from the CANopen finite state machine in several aspects. These different characteristics are listed below:

- No direct transition from Pre-Operational after Power On
- No Stopped status, direct transition to the INIT status
- Additional status: Safe-Operational

The following table compares the different statuses:

EtherCAT State	CANopen NMT State
Power ON	Power-On (initialisation)
Init	Stopped
Safe-Operational	–
Operational	Operational

Tab. 8.15 Comparison of the statuses for EtherCAT and CANopen

8.8 SDO Frame

All data of an SDO transfer are transmitted via SDO frames in CoE. These frames have the following structure:

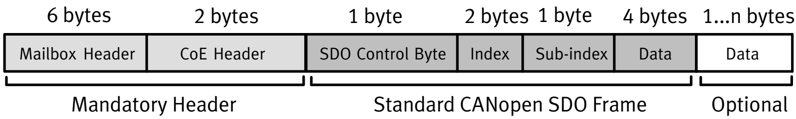


Fig. 8.4 SDO Frame: telegram structure

Element	Description
Mailbox Header	Data for mailbox communication (length, address and type)
CoE Header	Identifier of the CoE service
SDO Control Byte	Identifier for a read or write command
Index	Main index of the CANopen communication object
Sub-index	Sub-index of the CANopen communication object
Data	Data content of the CANopen communication object
Data (optional)	Additional optional data. This option is not supported by the motor controller CMMP-AS-...-M3, as only standard CANopen objects can be addressed. The maximum size of these objects is 32 bits.

Tab. 8.16 SDO Frame: elements

In order to transmit a standard CANopen object via one of these SDO frames, the actual CANopen SDO frame is packaged in an EtherCAT SDO frame and transmitted.
Standard CANopen SDO frames can be used for:
– Initialisation of the SDO download
– Download of the SDO segment
– Initialisation of the SDO upload
– Upload of the SDO segment
– Abort of the SDO transfer
– SDO upload expedited request
– SDO upload expedited response
– SDO upload segmented request (max. 1 segment with 4 bytes of user data)
– SDO upload segmented response (max. 1 segment with 4 bytes of user data)

All above-mentioned transfer types are supported by the motor controller CMMP-AS-...-M3.
As the use of the CoE implementation of the CMMP-AS-...-M3 only allows the standard CANopen objects to be addressed, whose size is restricted to 32 bits (4 bytes), only transfer types with a maximum data length of up to 32 bits (4 bytes) are supported.

8.9 PDO Frame

Process Data Objects (PDO) are used for cyclic transmission of setpoint values and actual values between master and slave. They must be configured in the “Pre-Operational” status by the master before the slave is operated. They are then transmitted in PDO frames. These PDO frames have the following structure:
All data of a PDO transfer are transmitted via PDO frames in CoE. These frames have the following structure:

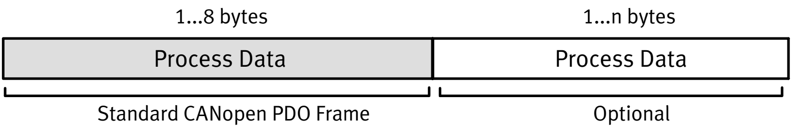


Fig. 8.5 PDO Frame: telegram structure

Element	Description
Process Data	Data content of the PDO (Process Data Object)
Process Data (optional)	Optional data content of additional PDOs

Tab. 8.17 PDO Frame: elements

To transmit a PDO via the EtherCAT-CoE protocol, in addition to the PDO configuration (PDO Mapping), the Transmit and Receive PDOs must be assigned to a transmission channel of the Sync Manager (➔ chapter 8.6.1 “Configuration of the Communication Interface”). The data exchange of PDOs for the motor controller CMMP-AS-...-M3 takes place exclusively via the EtherCAT process data telegram protocol.

The transfer of CANopen process data (PDOs) via acyclic communication (mailbox telegram protocol) is not supported by the motor controller CMMP-AS-...-M3.

As all data exchanged via the EtherCAT CoE protocol is forwarded directly to the internal CANopen implementation in the motor controller CMMP-AS-...-M3, the PDO mapping is also implemented as described in chapter 2.6.2 “PDO Message”. The figure below depicts this process:

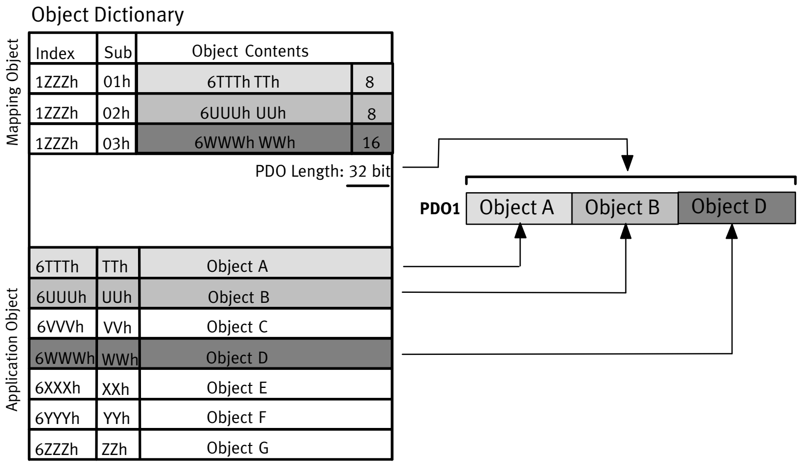


Fig. 8.6 PDO Mapping

The simple forwarding of the data received via CoE to the CANopen protocol implemented in CMMP-AS-...-M3 means that the "Transmission Types" of the PDOs available for the CANopen protocol for the CMMP-AS-...-M3 can be used in addition to CANopen object mapping for the PDOs to be parametrised. The motor controller CMMP-AS-...-M3 also supports the "Sync Message" transmission type. However, the Sync Message does not have to be sent via EtherCAT.
It is used either for the arrival of the telegram or the hardware synchronisation pulse of the "Distributed Clocks" mechanism (see below) for data transfer.
The EtherCAT interface for CMMP-AS-...-M3 supports synchronisation via the "Distributed Clocks" mechanism specified under EtherCAT by means of the use of FPGA module ESC20. The current regulator of the motor controller CMMP-AS-...-M3 is synchronised to this pulse, and the PDOs configured accordingly are evaluated or sent.
The motor controller CMMP-AS-...-M3 with the EtherCAT interface supports the following functions:
– Cyclic PDO frame telegram via the process data telegram protocol.
– Synchronous PDO frame telegram via the process data telegram protocol.
The motor controller CMMP-AS-...-M3 with the EtherCAT interface supports four Receive PDOs (RxPDO) and four Transmit PDOs (TxPDO).

8.10 Error Control

The EtherCAT CoE implementation for the motor controller CMMP-AS-...-M3 monitors the following error statuses of the EtherCAT fieldbus:

- FPGA is not ready when the system is started.
- A bus error has occurred.
- An error has occurred on the mailbox channel. The following errors are monitored here:
  - An unknown service is requested.
  - A protocol other than CANopen over EtherCAT (CoE) is to be used.
  - An unknown Sync Manager is addressed.

All of these errors are defined as corresponding error codes for the motor controller CMMP-AS-...-M3. If one of the above-mentioned errors occurs, it is transmitted to the controller via a “Standard Emergency Frame”. See also Chapter 8.11 “Emergency Frame” and Chapter D “ Diagnostic messages”.
The motor controller CMMP-AS-...-M3 with EtherCAT interface supports the following function:

- Application Controller determines a defined error message number as a result of an event (Error Control Frame telegram from the controller).

8.11 Emergency Frame

The master and slaves exchange error messages via the EtherCAT CoE emergency frame. The CoE emergency frames are used for direct transfer of the “Emergency Messages” defined under CANopen. The CANopen telegrams are simply tunnelled through the CoE emergency frames, as is the case for SDO and PDO transmission.

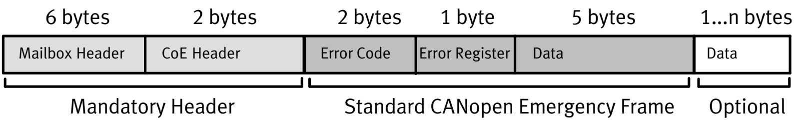


Fig. 8.7 Emergency Frame: telegram structure

Element	Description
Mailbox Header	Data for mailbox communication (length, address and type)
CoE Header	Identifier of the CoE service
ErrorCode	Error Code of the CANopen EMERGENCY Message ➔ Chapter 2.6.5
Error Register	Error Register of the CANopen EMERGENCY Message ➔ Tab. 2.19
Data	Data content of the CANopen EMERGENCY Message
Data (optional)	Additional optional data. As only the standard CANopen emergency frames are supported in the CoE implementation for the motor controller CMMP-AS-...-M3, the “Data (optional)” field is not supported.

Tab. 8.18 Emergency Frame: elements

As the “Emergency Messages” received and sent via CoE are simply forwarded to the CANopen protocol implemented in the motor controller, all error messages can be looked up in the chapter D.

8.12 Synchronisation (Distributed Clocks)

Time synchronisation is implemented via so-called “Distributed Clocks” in EtherCAT. Each EtherCAT slave receives a real-time clock, which is synchronised in all slaves by the clock master during the initialisation phase. The clocks in all slaves are then adjusted during operation. The clock master is the first slave in the network.
This provides a uniform time base in the entire system with which the individual slaves can synchronise. The Sync telegrams provided for this purpose under CANopen are unnecessary under CoE.
The FPGA ESC20 used in the motor controller CMMP-AS-...-M3 supports Distributed Clocks. This facilitates extremely precise time synchronisation. The cycle time of the EtherCAT Frame must exactly match the cycle time tp of the controller-internal interpolator. If necessary, the interpolator time must be adjusted via the object included in the device description file.
In the present implementation, synchronous transfer of PDO data and synchronisation of the controller-internal PLL to the synchronous data framework of the EtherCAT Frame can be implemented even without Distributed Clocks. For this purpose, the firmware uses the arrival of the EtherCAT Frame as a time base.
The following restrictions apply:

- The master must be able to send the EtherCAT Frames with an extremely low jitter.
- The cycle time of the EtherCAT Frame must exactly match the cycle time of the internal interpolator. The internal cycle time must be set in the FCT under “Fieldbus” – “Operating parameters” – “Cycle time”.
- The Ethernet must be available exclusively for the EtherCAT Frame. It may be necessary to synchronise other telegrams to the grid, as they may not block the bus.

9 I/O data and sequence control

9.1 Setpoint specification (FHPP operation modes)

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

Operating mode	Description
Record selection	A specific number of positioning records can be saved in the 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 nominal or actual value.
Direct mode	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. 9.1 Overview of FHPP operating modes in CMM...

9.1.1 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 mode is only permitted in the “ready” status ➔ section 9.6, Fig. 9.1.

9.1.2 Record selection

Each controller has a specific number of records, which contain all the information needed for one positioning job. The record number that the controller is to process at the next start is transferred into the output data of the PLC. The input data contains the record number that was processed last. The positioning job itself does not need to be active.
The controller does not support any automatic mode, i.e. no user program. The controller cannot accomplish any useful tasks in a stand alone situation - close coupling to the PLC is always necessary. However, depending on the controller, it is also possible to concatenate various records and execute them one after the other with the help of a start command. It is also possible - dependent on the controller - to define record chaining before the target position is reached.

Complete parameterisation of record chaining (“path program”), such as of the subsequent record, is only possible through the FCT.
In this way, positioning profiles can be created without the inactive times (which arise from the transfer in the fieldbus and the PLC's cycle time) having an effect.

9.1.3 Direct mode

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

9.2 Configuration of the I/O data

9.2.1 Concept

The FHPP protocol essentially provides 8 bytes of input data and 8 bytes of output data. Of these, the first byte is fixed (the first 2 bytes in the FHPP operating modes record selection and direct mode). It is retained in each operating mode and controls the enabling of the 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 according to the FPC protocol or FHPP+.
A PLC exchanges the following data with the FHPP:

- 8-byte control and status data:
  - 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, an additional 8 bytes of input and 8 bytes of output data for FPC parameterisation, ➔ section C.1.
- If supported, up to 24 (without FPC) or 16 (with FPC) additional bytes of I/O data for parameter transfer via FHPP+, if required, ➔ section C.2.

If applicable, observe the specification in the bus master for the representation of words and double words (Intel/Motorola). For example, when sending via CANopen, in the “little endian” representation (lower-value byte first).

9.2.2 I/O data in the various FHPP operating modes (control view)

Record selection

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Output data	CCON	CPOS	Record no.	Reserved	Reserved			
Input data	SCON	SPOS	Record no.	RSB	Actual position			

Direct mode

	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			

Additional 8 bytes of I/O data for parameterisation as per FPC (➔ section C.1):

<table>
<tr><th colspan="9">Festo FPC</th></tr>
<tr><td></td><td>Byte 1</td><td>Byte 2</td><td>Byte 3</td><td>Byte 4</td><td>Byte 5</td><td>Byte 6</td><td>Byte 7</td><td>Byte 8</td></tr>
<tr><td>Output data</td><td>Reserved</td><td>Sub-index</td><td colspan="2">Task identifier + parameter number</td><td colspan="4">Parameter value</td></tr>
<tr><td>Input data</td><td>Reserved</td><td>Sub-index</td><td colspan="2">Reply identifier + parameter number</td><td colspan="4">Parameter value</td></tr>
</table>

Additional bytes of I/O data for FHPP+ (➔ section C.2):

<table>
<tr><th colspan="16">FHPP with FPC</th><th colspan="16">FHPP+</th></tr>
<tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td><td>32</td></tr>
<tr><td colspan="8">Output data FHPP</td><td colspan="8">Output data FPC</td><td colspan="16">Output data FHPP+ (8 or 16 bytes)</td></tr>
<tr><td colspan="8">Input data FHPP</td><td colspan="8">Input data FPC</td><td colspan="16">Input data FHPP+ (8 or 16 bytes)</td></tr>
</table>

<table>
<tr><th colspan="8">FHPP</th><th colspan="24">FHPP+</th></tr>
<tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td><td>32</td></tr>
<tr><td colspan="8">Output data FHPP</td><td colspan="24">Output data FHPP+ (8, 16 or max. 24 bytes)</td></tr>
<tr><td colspan="8">Input data FHPP</td><td colspan="24">Input data FHPP+ (8, 16 or max. 24 bytes)</td></tr>
</table>

9.3 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 mode)	**B7** **FUNC**	**B6** **FGRP2**	**B5** **FGRP1**	**B4** **FNUM2**	**B3** **FNUM1**	**B2** **COM2**	**B1** **COM1**	**B0** **ABS**
	Execute function	Function group		Function number		Control mode (position, torque, ve-locity, ...)		Absolute/ relative

Tab. 9.2 Overview, assignment of the control bytes

Assignment of the status bytes (overview)								
SCON (all)	**B7** **OPM2**	**B6** **OPM1**	**B5** **FCT/MMI**	**B4** **RDYEN**[1)]	**B3** **FAULT**	**B2** **WARN**	**B1** **OPEN**	**B0** **ENABLED**
	Feedback on FHPP operating mode		FCT device control	Ready for enable	Malfunc-tion	Warning	Opera-tion en-abled	Drive en-abled
SPOS (all)	**B7** **REF**	**B6** **STILL**	**B5** **DEV**	**B4** **MOV**	**B3** **TEACH**	**B2** **MC**	**B1** **ACK**	**B0** **HALT**
	Drive ref-erenced	Standstill monitor-ing	Following error	Axis is moving	Acknow-ledge teach or sample	Motion Com-plete	Acknow-ledge start	Halt
SDIR (Direct mode)	**B7** **FUNC**	**B6** **FGRP2**	**B5** **FGRP1**	**B4** **FNUM2**	**B3** **FNUM1**	**B2** **COM2**	**B1** **COM1**	**B0** **ABS**
	Function is executed	Function group acknowledgment		Function number acknowledgment		Control mode acknowledgment (position, torque, velocity)		Absolute/ relative

1) From FW 4.0.1501.2.3 ➔ Tab. 9.11

Tab. 9.3 Overview, assignment of the status bytes

9.4 Description of the control bytes

9.4.1 Control byte 1 (CCON)

<table>
<tr><th colspan="6">Control byte 1 (CCON)</th></tr>
<tr><th>Bit</th><th>EN</th><th colspan="4">Description</th></tr>
<tr><td rowspan="2">B0
ENABLE</td><td rowspan="2">Enable Drive</td><td colspan="4">= 1: Enable drive (controller).</td></tr>
<tr><td colspan="4">= 0: Drive (controller) blocked.</td></tr>
<tr><td rowspan="2">B1
STOP</td><td rowspan="2">Stop</td><td colspan="4">= 1: Enable operation.</td></tr>
<tr><td colspan="4">= 0: STOP active (cancel positioning job + stop with emergency ramp). The drive stops with maximum braking ramp, the positioning job is reset.</td></tr>
<tr><td rowspan="3">B2
BRAKE</td><td rowspan="3">Open Brake</td><td colspan="4">= 1: Release brake.</td></tr>
<tr><td colspan="4">= 0: Activate brake.</td></tr>
<tr><td colspan="4">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>Reset Fault</td><td colspan="4">A malfunction is acknowledged with a rising edge and the malfunction value is deleted.</td></tr>
<tr><td>B4
–</td><td>–</td><td colspan="4">Reserved, must be at 0.</td></tr>
<tr><td rowspan="3">B5
LOCK</td><td rowspan="3">Lock Software Access</td><td colspan="4">Controls access to the local (integrated) parameterisation interface of the controller.</td></tr>
<tr><td colspan="4">= 1: The software can only observe the controller; the software cannot take over device control (HMI control) from the software.</td></tr>
<tr><td colspan="4">= 0: The software may take over the device control (in order to modify parameters or to control inputs).</td></tr>
<tr><td rowspan="2">B6
OPM1</td><td rowspan="6">Select Operating Mode</td><td colspan="4">Determining the 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 mode</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. 9.4 Control byte 1

CCON controls statuses in all FHPP operation modes. For more information, ➔ description of the drive functions, chapter 11.

9.4.2 Control byte 2 (CPOS)

Control byte 2 (CPOS)			
Bit	EN	Description	
B0 **HALT**	**Halt**	= 1:	Halt is not requested.
		= 0:	Halt activated (cancel positioning job + halt with braking ramp). The axis stops with a defined braking ramp; the positioning job remains active (with CPOS.CLEAR, the remaining path can be deleted).
B1 **START**	**Start** Positioning Task	A **rising edge** transfers the current nominal data and starts a positioning process (also, for example, record 0 = homing!).	
B2 **HOM**	Start **Homing**	A **rising edge** starts homing with the set parameters.	
B3 **JOGP**	**Jog p**ositive	The drive moves at the specified velocity 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 n**egative	The drive moves at the specified velocity 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** 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. The type is determined by the record status byte (RSB) ➔ section 10.5.	
B6 **CLEAR**	**Clear** Remaining Position	In the “Halt” state, a **rising edge** causes the positioning task to be deleted and a transition to the “Ready” state.	
B7 **–**	–	Reserved, must be at 0.	

Tab. 9.5 Control byte 2

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

9.4.3 Control byte 3 (CDIR) – Direct mode

Control byte 3 (CDIR) – Direct mode					
Bit	EN	Description			
B0 **ABS**	**Abs**olute / Relative	= 1:	Nominal value is relative to the last nominal value.		
		= 0:	Nominal value is absolute.		
B1 **COM1**	**Co**ntrol **M**ode	**No.**	**Bit 2**	**Bit 1**	**Control mode**
		0	0	0	Position control.
B2 **COM2**		1	0	1	Force mode (torque, current).
		2	1	0	Velocity control (rotational speed).
		3	1	1	Reserved.
		Only position code mode is permissible for the cam disc function.			
B3 **FNUM1**	**F**unction **Num**ber	Without cam disc function (CDIR.FUNC = 0): No function, = 0!			
B4 **FNUM2**		With cam disc function (CDIR.FUNC = 1):			
		No.	**Bit 4**	**Bit 3**	**Function number 1)**
		0	0	0	Reserved.
		1	0	1	Synchronisation on external input.
		2	1	0	Synchronisation on external input with cam disc function.
		3	1	1	Synchronisation on virtual master with cam disc function.
B5 **FGRP1**	**F**unction **Gr**oup	Without cam disc function (CDIR.FUNC = 0): No function, = 0!			
B6 **FGRP2**		With cam disc function (CDIR.FUNC = 1):			
		No.	**Bit 6**	**Bit 5**	**Function group**
		0	0	0	Synchronisation with/without cam disc.
		All other values (no. 1 ... 3) are reserved.			
B7 **FUNC**	**Func**tion	= 1:	Execute cam disc function, bit 3 ... 6 = function number and group.		
		= 0:	Normal job.		

1) With function numbers 1 and 2 (synchronisation on an external input), the bits CPOS.ABS and CPOS.COMx are not relevant. With function number 3 (virtual master, internal), the bits CPOS.ABS and CPOS.COMx determine the reference and control mode of the master.

Tab. 9.6 Control byte 3 – direct mode

In direct mode, CDIR specifies the type of positioning job.

9.4.4 Bytes 4 and 5 ... 8 – Direct mode

Control byte 4 (setpoint value 1) – Direct mode		
Bit	EN	Description
B0 ... 7	Preselection depends on the control mode (CDIR.COMx):	
	Preselected value with position control:	
	Velocity	Velocity as % of base value (PNU 540)
	Preset value for force mode from FW 4.0.1501.2.3	
	Torque ramp	Force ramp in % of the base value (PNU 550)
	Preset value for force mode up to FW 4.0.1501.2.2	
	–	No function, = 0!
	Preset value for velocity control	
	Velocity ramp	Velocity ramp as % of base value (PNU 560)

Tab. 9.7 Control byte 4 – direct application

Control bytes 5 ... 8 (setpoint value 2) – Direct mode		
Bit	EN	Description
B0 ... 31	Preselection depends on control mode (CDIR.comX), in each case 32-bit number, low byte first:	
	Preselected value with position control:	
	Position	Position in positioning unit ➔ appendix A.1
	Preset value for force mode	
	Torque	Torque setpoint as % of the nominal torque (PNU 1036)
	Preset value for velocity control	
	Velocity	Velocity in units of velocity ➔ appendix A.1

Tab. 9.8 Control bytes 5 ... 8 – direct application

9.4.5 Bytes 3 and 4 ... 8 – record selection

Control byte 4 (setpoint value 1) – Record selection		
Bit	EN	Description
B0 ... 7	Record number	Preselection of the record number.

Tab. 9.9 Control byte 4 – Record selection

Control byte 5 ... 8 (setpoint value 2) – Record selection		
Bit	EN	Description
B0 ... 31	–	Reserved (= 0)

Tab. 9.10 Control bytes 5 ... 8 – Record selection

9.5 Description of the status bytes

9.5.1 Status byte 1 (SCON)

<table>
<tr><th colspan="6">Status byte 1 (SCON)</th></tr>
<tr><td>Bit</td><td>EN</td><td colspan="4">Description</td></tr>
<tr><td rowspan="2">B0
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 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 colspan="4">= 1: Warning applied.</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 colspan="4">= 1: Malfunction present.</td></tr>
<tr><td colspan="4">= 0: Malfunction not present or malfunction reaction active.</td></tr>
<tr><td rowspan="4">B4
RDYEN</td><td rowspan="4">READY ENABLE</td><td colspan="4">From FW 4.0.1501.2.3:</td></tr>
<tr><td colspan="4">= 1: Ready for enable (ENABLE)</td></tr>
<tr><td colspan="4">= 0: Not ready for enable (ENABLE)</td></tr>
<tr><td colspan="4">Up to FW 4.0.1501.2.2:
bit 4, SCON.VLOAD = 1: load voltage is applied</td></tr>
<tr><td rowspan="3">B5
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">Display Operating 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 mode</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. 9.11 Status byte 1

9.5.2 Status byte 2 (SPOS)

Status byte 2 (SPOS)		
Bit	EN	Description
B0 **HALT**	**Halt**	= 1: Halt is not active; axis can be moved.
		= 0: Halt is active.
B1 **ACK**	**Ack**nowledge Start	= 1: Start executed (homing, jogging, positioning)
		= 0: Ready for start (homing, jogging, positioning)
B2 **MC**	**M**otion **C**omplete	= 1: Positioning job completed, where applicable with error
		= 0: Positioning job active
		Note: MC is set after device is switched on (status “Drive blocked”).
B3 **TEACH**	Acknowledge **Teach**/ Sampling	Depending on the setting in PNU 354:
		PNU 354 = 0: Display of teach status:
		= 1: Teaching carried out, actual value has been transferred
		= 0: Ready for teaching
		PNU 354 = 1: Display of the sampling status: 1)
		= 1: Edge detected. New position value available.
		= 0: Ready for sampling
B4 **MOV**	Axis is **Mov**ing	= 1: Velocity of the axis >= limit value
		= 0: Velocity of the axis < limit value
B5 **DEV**	Drag (**Dev**iation) Error	= 1: Following error active
		= 0: No following error
B6 **STILL**	Stand**still**Control	= 1: Axis has left the tolerance window after MC
		= 0: After MC, axis remains in tolerance window
B7 **REF**	Axis **Ref**erenced	= 1: Homing information available, homing does not need to be carried out
		= 0: Homing must be executed

1) Position sampling ➔ section 10.9.

Tab. 9.12 Status byte 2

9.5.3 Status byte 3 (SDIR) – Direct mode

The SDIR status byte acknowledges positioning mode.

<table>
<tr><th colspan="6">Status byte 3 (SDIR) – Direct mode</th></tr>
<tr><td>Bit</td><td>EN</td><td colspan="4">Description</td></tr>
<tr><td rowspan="2">B0
ABS</td><td rowspan="2">Absolute / Relative</td><td colspan="4">= 1: Nominal value is relative to the last nominal value.</td></tr>
<tr><td colspan="4">= 0: Nominal value is absolute.</td></tr>
<tr><td rowspan="2">B1
COM1</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>Position control.</td></tr>
<tr><td rowspan="3">B2
COM2</td><td>1</td><td>0</td><td>1</td><td>Force mode (torque, current).</td></tr>
<tr><td>2</td><td>1</td><td>0</td><td>Velocity control (rotational speed).</td></tr>
<tr><td>3</td><td>1</td><td>1</td><td>Reserved.</td></tr>
<tr><td>B3
FNUM1</td><td rowspan="7">Function Number Feedback</td><td colspan="4">Without cam disc function (CDIR.FUNC = 0):
No function, = 0.</td></tr>
<tr><td rowspan="6">B4
FNUM2</td><td colspan="4">With cam disc function (CDIR.FUNC = 1):</td></tr>
<tr><td>No.</td><td>Bit 4</td><td>Bit 3</td><td>Function number</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>CAM-IN / CAM-OUT / Change active.</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>Synchronisation on external input.</td></tr>
<tr><td>2</td><td>1</td><td>0</td><td>Synchronisation on external input with cam disc function.</td></tr>
<tr><td>3</td><td>1</td><td>1</td><td>Synchronisation on virtual master with cam disc function.</td></tr>
<tr><td>B5
FGRP1</td><td rowspan="5">Function Group Feedback</td><td colspan="4">Without cam disc function (CDIR.FUNC = 0):
No function, = 0</td></tr>
<tr><td rowspan="4">B6
FGRP2</td><td colspan="4">With cam disc function (CDIR.FUNC = 1):</td></tr>
<tr><td>No.</td><td>Bit 4</td><td>Bit 3</td><td>Function group</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>Synchronisation with/without cam disc.</td></tr>
<tr><td colspan="4">All other values (no. 1 ... 3) are reserved.</td></tr>
<tr><td rowspan="2">B7
FUNC</td><td rowspan="2">Function Feedback</td><td colspan="4">= 1: Cam disc function is executed, bit 3 ... 6 = function number and group.</td></tr>
<tr><td colspan="4">= 0: Normal job</td></tr>
</table>

Tab. 9.13 Status byte 3 – Direct mode

9.5.4 Bytes 4 and 5 ... 8 – Direct mode

Status byte 4 (actual value 1) – Direct mode		
Bit	EN	Description
B0 ... 7	Feedback depends on the control mode (CDIR.COMx):	
	Feedback with position control	
	Velocity	Velocity as % of base value (PNU 540)
	Feedback value for force mode	
	Torque	Torque as % of the rated torque (PNU 1036)
	Feedback value for velocity control	
	–	no function, = 0

Tab. 9.14 Status byte 4 – Direct mode

Status bytes 5 ... 8 (actual value 2) – Direct mode		
Bit	EN	Description
B0 ... 31	Feedback depends on control mode (CDIR.comX), in each case 32-bit number, low byte first:	
	Feedbach value with position control	
	Position	Position in positioning unit ➔ appendix A.1
	Feedback value for force mode	
	Position	Position in positioning unit ➔ appendix A.1
	Feedback value for velocity control	
	Velocity	Velocity as an absolute value in unit of velocity ➔ appendix A.1

Tab. 9.15 Status bytes 5 ... 8 – Direct mode

9.5.5 Bytes 3, 4 and 5 ... 8 – record selection

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

Tab. 9.16 Status byte 4 – Record selection

<table>
<tr><th colspan="6">Status byte 4 (RSB) – record selection</th></tr>
<tr><th>Bit</th><th>EN</th><th colspan="4">Description</th></tr>
<tr><td rowspan="2">B0
RC1</td><td rowspan="2">1st Record Chaining Done</td><td colspan="4">= 1: The first step enabling condition was achieved.</td></tr>
<tr><td colspan="4">= 0: A step enabling condition was not configured or not achieved.</td></tr>
<tr><td rowspan="3">B1
RCC</td><td rowspan="3">Record Chaining Complete</td><td colspan="4">Valid, as soon as MC present.</td></tr>
<tr><td colspan="4">= 1: Record chain was processed to the end of the chain.</td></tr>
<tr><td colspan="4">= 0: Record chaining aborted. At least one step enabling condition has not been met.</td></tr>
<tr><td>B2
–</td><td>–</td><td colspan="4">Reserved, = 0.</td></tr>
<tr><td>B3
FNUM1</td><td rowspan="7">Function Number Feedback</td><td colspan="4">Without cam disc function (CDIR.FUNC = 0):
No function, = 0.</td></tr>
<tr><td rowspan="6">B4
FNUM2</td><td colspan="4">With cam disc function (CDIR.FUNC = 1):</td></tr>
<tr><td>No.</td><td>Bit 4</td><td>Bit 3</td><td>Function number</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>CAM-IN / CAM-OUT / Change active.</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>Synchronisation on external input.</td></tr>
<tr><td>2</td><td>1</td><td>0</td><td>Synchronisation on external input with cam disc function.</td></tr>
<tr><td>3</td><td>1</td><td>1</td><td>Synchronisation on virtual master with cam disc function.</td></tr>
<tr><td>B5
FGRP1</td><td rowspan="5">Function Group Feedback</td><td colspan="4">Without cam disc function (CDIR.FUNC = 0):
No function, = 0</td></tr>
<tr><td rowspan="4">B6
FGRP2</td><td colspan="4">With cam disc function (CDIR.FUNC = 1):</td></tr>
<tr><td>No.</td><td>Bit 4</td><td>Bit 3</td><td>Function group</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>Synchronisation with/without cam disc.</td></tr>
<tr><td colspan="4">All other values (no. 1 ... 3) are reserved.</td></tr>
<tr><td rowspan="2">B7
FUNC</td><td rowspan="2">Function Feedback</td><td colspan="4">= 1: Cam disc function is executed, bit 3 ... 6 = function number and group.</td></tr>
<tr><td colspan="4">= 0: Normal job</td></tr>
</table>

Tab. 9.17 Status byte 4 – record selection

<table>
<tr><th colspan="3">Status bytes 5 ... 8 (position) – record selection</th></tr>
<tr><th>Bit</th><th>EN</th><th>Description</th></tr>
<tr><td>B0 ... 31</td><td>Position</td><td>Feedback on the position in position unit → appendix A.1.
32-bit number, low byte first.</td></tr>
</table>

Tab. 9.18 Status bytes 5 ... 8 – Record selection

9.6 FHPP finite state machine

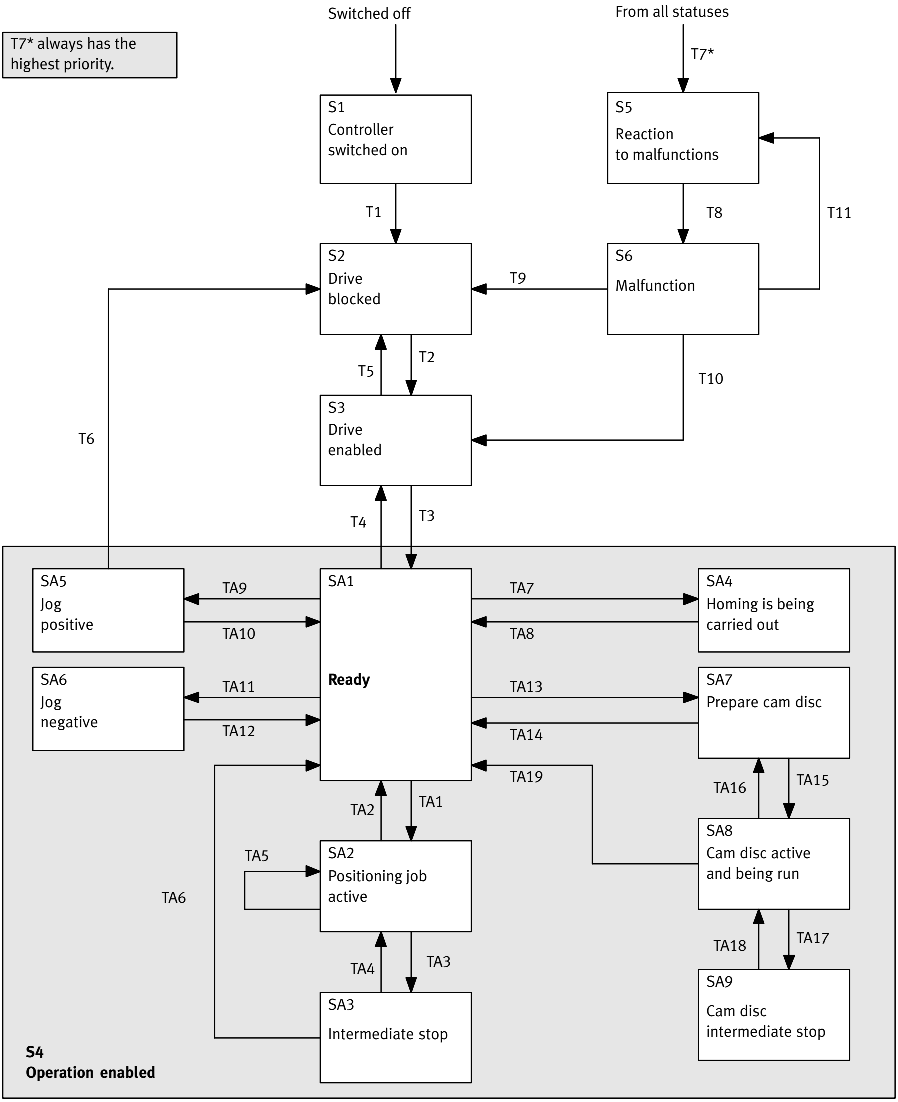


Fig. 9.1 Finite state machine

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. 9.1.
This enables an equivalent circuit diagram (➔ Fig. 9.2) to be used, in which the internal states 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.

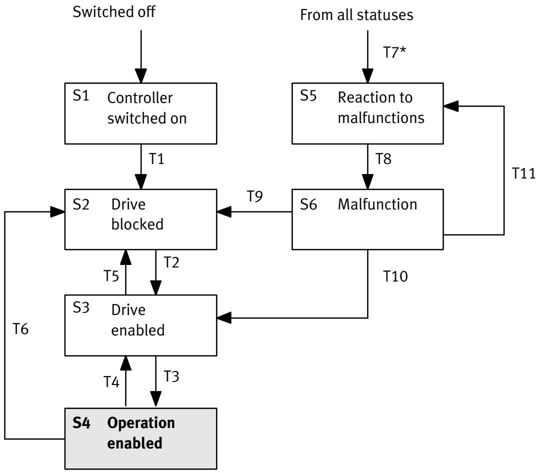


Fig. 9.2 Finite state machine equivalent circuit diagram

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.

9.6.1 Establishing the ready status

To create the ready status, additional input signals may be required, depending on the controller, at DIN4, DIN5, DIN13, etc., for example.
More detailed information can be found in the Hardware description, GDCP-CMMP-M3-HW-...

T	Internal conditions	Actions of the user [1]
T1	Drive is switched on. An error cannot be ascertained.	
T2	Load voltage applied. Higher-order control with PLC.	"Enable drive" = 1 CCON = xxx0.xxx**1**
T3		"Stop" = 1 CCON = xxx0.xx**1**1
T4		"Stop" = 0 CCON = xxx0.xx**0**1
T5		"Enable drive" = 0 CCON = xxx0.xxx**0**
T6		"Enable drive" = 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" = 0 → 1 CCON = xxx0.**P**xxx
T10	There is no longer a malfunction. It was a simple error.	"Acknowledge malfunction" = 0 → 1 CCON = xxx0.**P**xx1
T11	Malfunction still exists.	"Acknowledge malfunction" = 0 → 1 CCON = xxx0.**P**xx1

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

Tab. 9.19 Status transitions while achieving ready status

9.6.2 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 job = 0 → 1 Halt = 1 CCON = xxx0.xx11 CPOS = 0xx0.00**P1**
TA2	Motion Complete = 1 The current record is completed. The next record is not to be carried out automatically	“Halt” status is any CCON = xxx0.xx11 CPOS = 0xxx.xxxx
TA3	Motion Complete = 0	Halt = 1 → 0 CCON = xxx0.xx11 CPOS = 0xxx.xxx**N**
TA4		Halt = 1 Start positioning job = 0 → 1 Delete remaining path = 0 CCON = xxx0.xx11 CPOS = 0**0**xx.xx**P1**
TA5	Record selection: – A single record is finished. – The next record is processed automatically.	CCON = xxx0.xx11 CPOS = 0xxx.xxx**1**
	Direct mode: – A new positioning job has arrived.	CCON = xxx0.xx11 CPOS = 0xxx.xx**11**
TA6		Delete remaining path = 0 → 1 CCON = xxx0.xx11 CPOS = 0**P**xx.xxxx
TA7		Start homing = 0 → 1 Halt = 1 CCON = xxx0.xx11 CPOS = 0xx0.0**P**x**1**
TA8	Referencing finished or stopped.	Halt = 1 → 0 (only for halt) CCON = xxx0.xx11 CPOS = 0xxx.xxx**N**
TA9		Jog positive = 0 → 1 Halt = 1 CCON = xxx0.xx11 CPOS = 0xx0.**P**xx**1**

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

T	Internal conditions	Actions of the user 1)
TA10		Either Jog positive = 1 → 0 – CCON = xxx0.xx11 – CPOS = 0xxx.**N**xx1 or Halt = 1 → 0 – CCON = xxx0.xx11 – CPOS = 0xxx.xxx**N**
TA11		Jog negative = 0 → 1 Halt = 1 CCON = xxx0.xx11 CPOS = 0xx**P**.0xx1
TA12		Either Jog negative = 1 → 0 – CCON = xxx0.xx11 – CPOS = 0xx**N**.xxx1 or Halt = 1 → 0 – CCON = xxx0.xx11 – CPOS = 0xxx.xxx**N**

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

Tab. 9.20 Status transitions at positioning

There are additional transitions if the cam disc function is used
➔ section 9.6.3.

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

Tab. 9.21 Special features dependent on FHPP operating mode

9.6.3 Extended finite state machine with cam disc function

TA	Description	Occurrence with		Secondary condition
		Record selection	Direct mode	
TA13	Prepare cam disk (activate)	"Rising" edge (change) of record number.	–	Old record: FUNC = 0 New record: FUNC = 1
		–	Rising edge at FUNC.	–
		Rising edge at STOP or ENABLE (activation of controller enable).		FUNC = 1
TA14, TA19	Deactivate cam disc	"Rising" edge (change) of record number.	–	Old record: FUNC = 1 New record: FUNC = 0
		–	Falling edge at FUNC.	–
		STOP or withdrawal of ENABLE.		None, FUNC = any
TA15	Cam disc active and being run	Rising edge at START.		Drive is in TA 13.
TA16	Change cam disc	Rising edge at START.	–	Changed cam disc number in PNU 419 or PNU 700. FUNC = 1
		"Rising" edge (change) of record number and rising edge at START.	–	Changed cam disc number in PNU 419 or PNU 700. FUNC = 1
		–	Rising edge at START, starts the virtual master automatically.	PNU 700 has been changed. FUNC = 1
TA17	Intermediate stop	HALT = 0		Intermediate stop with virtual master only.
TA18	End intermediate stop	HALT = 1		

Tab. 9.22

9.6.4 Examples of control and status bytes

On the following pages you will find typical examples of control and status bytes:

1. Establish readiness to operate – Record selection, Tab. 9.23
2. Establish readiness to operate – Direct mode, Tab. 9.24
3. Malfunction handling, Tab. 9.25
4. Homing, Tab. 9.26
5. Positioning record selection, Tab. 9.27
6. Positioning direct mode, Tab. 9.28

Information about the finite state machine ➔ section 9.6.
For all examples: Additional digital I/Os are required for CMM... controller and regulator enabling ➔ Hardware description, GDCP-CMMP-M3-HW-...

1. Establish ready status - Record selection

Step/description	Control bytes (job) 1)		Status bytes (response) 1)	
1.1 Initial status	CCON	= $0000.0x00_b$	SCON	= 0001.0000_b
	CPOS	= 0000.0000_b	SPOS	= 0000.0100_b
1.2 Disable device control for software	CCON.LOCK	= 1	SCON.FCT/MMI	= 0
1.3 Enable drive, enable operation (Record selection)	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

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

Tab. 9.23 Control and status bytes - “Establish ready status – Record selection”

Description of 1. Establish ready status:

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

1.2 Disable device control by software.
Optionally, acceptance of device control by the software can be disabled with CCON.LOCK = 1.
⟶ Step 1.3

1.3 Enable drive in record selection mode. ⟶ Homing: Example 4, Tab. 9.26.

If there are malfunctions after switching on or after setting CCON.ENABLE:
⟶ Malfunction handling: ➔ example 3, Tab. 9.25.

2.Establish ready status – Direct mode

Step/description	Control bytes (job) 1)		Status bytes (response) 1)	
2.1 Initial status	CCON	= 0000.0x00$_b$	SCON	= 0001.0000$_b$
	CPOS	= 0000.0000$_b$	SPOS	= 0000.0100$_b$
2.2 Disable device control for software	CCON.LOCK	= 1	SCON.FCT/MMI	= 0
2.3 Enable drive, enable operation (Record selection)	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) Legend: P = rising edge (positive), N = falling edge (negative), x = any

Tab. 9.24 Control and status bytes “Establish ready status - Direct mode”

Description of 2. Establish ready status:

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

2.2 Disable device control by software. Optionally, acceptance of device control by the software can be disabled with CCON.LOCK = 1. ⟶ Step 2.3

2.3 Enable drive in direct mode. ⟶ Homing: Example 4, Tab. 9.26.

If there are malfunctions after switching on or after setting CCON.ENABLE:
⟶ Malfunction handling: ➔ example 3, Tab. 9.25.
Warnings do not have to be acknowledged; these are automatically deleted after some seconds when their cause has been remedied.

3. Malfunction handling

Step/description	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) Legend: P = rising edge (positive), N = falling edge (negative), x = any

Tab. 9.25 Control and status bytes “Malfunction handling”

Description of 3. Malfunction handling

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.B3 FAULT or SCON.B2 WARN is reset, ⟶ SPOS.MC is set, ⟶ drive is ready for operation

Errors and warnings can be also acknowledged with a falling edge at DIN5 (controller enable) ➔ Hardware description, GDCP-CMMP-M3-HW-...

4. Homing (requires status 1.3 or 2.3)

Step/description	Control bytes (job) 1)		Status bytes (response) 1)	
4.1 Start homing	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 Homing is running	CPOS.HOM	= 1	SPOS.MOV	= 1
4.3 Homing ended	CPOS.HOM	= 0	SPOS.MC	= 1
			SPOS.REF	= 1

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

Tab. 9.26 Control and status bytes “Homing”

Description of 4. Homing:

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 (axis moves).

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

5. Positioning record selection (requires status 1.3/2.3 and possibly 4.3)

Step/description	Control bytes (job) 1)		Status bytes (response) 1)	
5.1 Record number preselection (control byte 3)	Record no.	0 ... 250	Previous record no.	0 ... 250
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.	0 ... 250	Current record no.	0 ... 250
5.4 Job ended	CPOS.START	= 0	SPOS.ACK	= 0
			SPOS.MC	= 1
			SPOS.MOV	= 0

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

Tab. 9.27 Control and status bytes “Positioning record selection”

Description of 5. Positioning record selection:

(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 ... 250 = 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 (axis moves).

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

6. Positioning direct mode (requires status 1.3/2.3 and possibly 4.3)

Step/description	Control bytes (job) [1]		Status bytes (response) [1]	
6.1 Preselect position (byte 4) and velocity (bytes 5...8)	Velocity preselection	0 ... 100 (%)	Velocity acknowledgment	0 ... 100 (%)
	Setpoint position	Position units	Actual position	Position units
6.2 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
	CDIR.ABS	= S	SDIR.ABS	= S
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) Legend: P = rising edge (positive), N = falling edge (negative), x = any, S= travel condition: 0= absolute; 1 = relative

Tab. 9.28 Control and status bytes for “Positioning direct mode”

Description of positioning direct mode:

(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 is transferred in positioning units in bytes 5...8 of the output word. The setpoint velocity is transferred in % in byte 4 (0 = no velocity; 100 = max. velocity).

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.

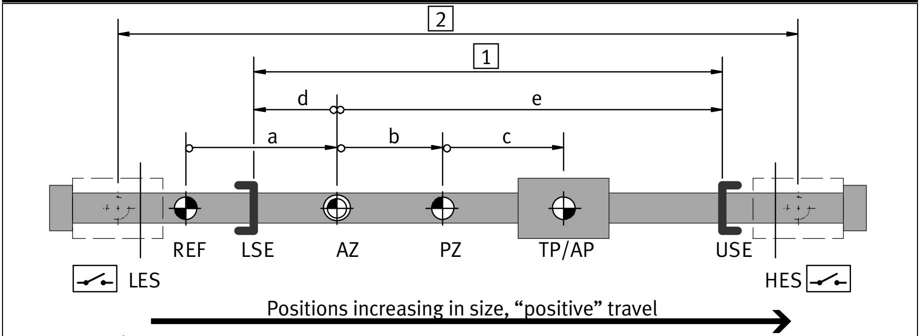
6.3 Movement of the axis is shown with SPOS.MOV.

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

10 Drive functions

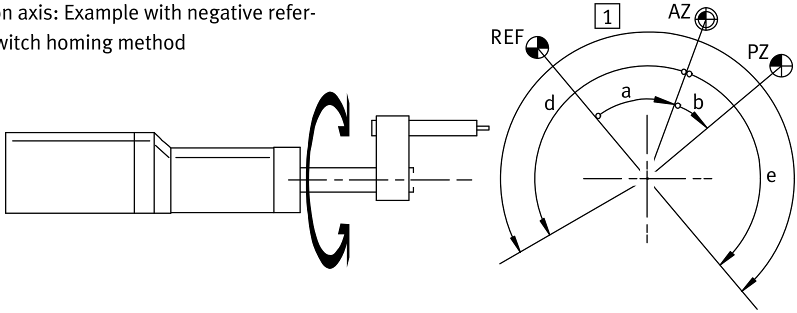
10.1 Reference system for electric drives

Reference system for electric linear drives



REF	Homing point (reference point)
AZ	Axis zero point
PZ	Project zero point
LSE	Lower software end position
USE	Upper software end position
LES	Lower end switch
HES	Higher end switch
TP	Target position
AP	Actual position
a	Offset axis zero point
b	Offset project zero point
c	Offset target/actual position
d, e	Offset software end positions
1	Effective stroke
2	Nominal stroke

Tab. 10.1 Reference system for electric linear drives

Reference system for electric rotary drives	
Rotation axis: Example with negative reference switch homing method 	
REF	Homing point (reference point)
AZ	Axis zero point
PZ	Project zero point
a	Offset axis zero point
b	Offset project zero point
c	Offset target/actual position
d, e	Offset software end positions (optional: endless positioning possible)
1	Effective stroke

Tab. 10.2 Reference system for electric rotary drives

10.2 Calculating specifications for the measuring reference system

Reference point	Calculation rule			
Axis zero point	AZ	= REF + a		
Project zero point	PZ	= AZ + b	= REF + a + b	
Lower software end position.	LSE	= AZ + d	= REF + a + d	
Upper software end position.	USE	= AZ + e	= REF + a + e	
Target/actual position	TP, AP	= PZ + c	= AZ + b + c	= REF + a + b + c

Tab. 10.3 Calculation rules for the measuring reference system with incremental measuring systems

10.3 Homing

In the case of drives with incremental measuring system, homing must always be carried out after the drive is switched on.
This is defined drive-specifically with the parameter “Homing required” (PNU 1014).

For a description of the homing modes, see section 10.3.2.

10.3.1 Homing for electric drives

The drive homes against a stop, a limit switch or a reference switch. An increase in the motor current indicates that a stop has been reached. Since the drive must not continuously home against the stop, it must move at least one millimetre back into the stroke range.

Process:

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

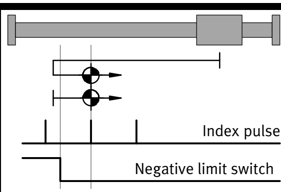
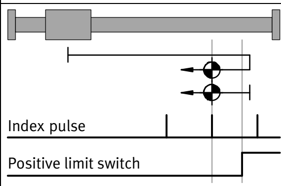
Overview of parameters and I/Os in homing		
Parameters involved	**Parameters**	**PNU**
➔ Section B.4.18	Offset axis zero point	1010
	Homing method	1011
	Homing velocity	1012
	Homing accelerations	1013
	Homing required	1014
	Homing maximum torque	1015
Start (FHPP)	CPOS.HOM = rising edge: start homing	
Acknowledgement (FHPP)	SPOS.ACK = rising edge: Start acknowledgment	
	SPOS.REF = drive homed	
Requirement	Device control by PLC/fieldbus	
	Controller in status “Operation enabled”	
	No command for jogging	

Tab. 10.4 Parameters and I/Os in homing

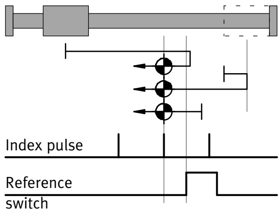
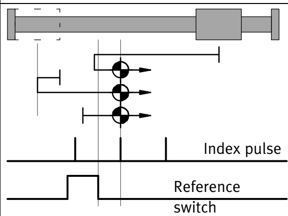
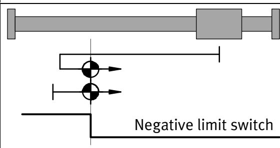
10.3.2 Homing methods

The homing methods are oriented towards CANopen DS402.

With some motors (those with absolute encoders, single/multi-turn) the drive may be permanently referenced. In such cases, methods involving homing to an index pulse (= zero pulse) might not cause homing to be carried out; rather the drive will move directly to the axis zero point (if it has been entered in the parameters).

Homing methods			
hex	dec	Description	
01h	1	**Negative limit switch with index pulse 1)** 1. If negative limit switch inactive: Run at search velocity in negative direction to the negative limit switch. 2. Travel at crawling velocity in positive direction until the limit switch becomes inactive, then continue to the first index pulse. This position is taken as the homing point. 3. If this is parameterised: travel at positioning velocity to the axis zero point.	
02h	2	**Positive limit switch with index pulse 1)** 1. If positive limit switch inactive: Run at search velocity in positive direction to the positive limit switch. 2. Travel at crawling velocity in negative direction until the limit switch becomes inactive, then continue to the first index pulse. This position is taken as the homing point. 3. If this is parameterised: travel at positioning velocity to the axis zero point.	

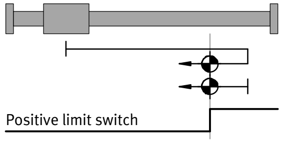
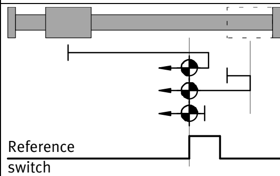
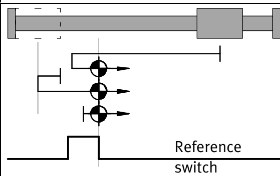
1) Only possible for motors with encoder/resolver with index pulse.
2) Limit switches are ignored during travel to the stop.
3) Since the axis is not to remain at the stop, the travel to the axis zero point must be parameterised and the axis zero point offset must be ≠ 0.

Homing methods			
hex	dec	Description	
07h	7	**Reference switch in positive direction with index pulse [1)]** 1. If reference switch inactive: Travel at search velocity in positive direction to the reference switch. If the stop or limit switch is approached: Travel at search velocity in positive direction to the reference switch. 2. Travel at crawling velocity in negative direction until the reference switch becomes inactive, then continue to the first index pulse. This position is taken as the homing point. 3. If this is parameterised: travel at positioning velocity to the axis zero point.	
0B	11	**Reference switch in negative direction with index pulse [1)]** 1. If reference switch inactive: Travel at search velocity in negative direction to the reference switch. If the stop or limit switch is approached: Travel at search velocity in positive direction to the reference switch. 2. Travel at crawling velocity in positive direction until the reference switch becomes inactive, then continue to the first index pulse. This position is taken as the homing point. 3. If this is parameterised: travel at positioning velocity to the axis zero point.	
11h	17	**Negative limit switch** 1. If negative limit switch inactive: Run at search velocity in negative direction to the negative limit switch. 2. Travel at crawling velocity in positive direction until the limit switch becomes inactive. This position is taken as the homing point. 3. If this is parameterised: travel at positioning velocity to the axis zero point.	

1) Only possible for motors with encoder/resolver with index pulse.

2) Limit switches are ignored during travel to the stop.

3) Since the axis is not to remain at the stop, the travel to the axis zero point must be parameterised and the axis zero point offset must be ≠ 0.

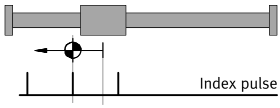
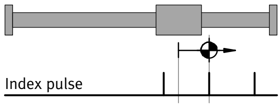
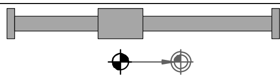
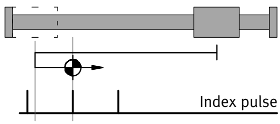
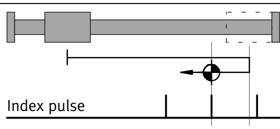
Homing methods			
hex	dec	Description	
12h	18	**Positive limit switch** 1. If positive limit switch inactive: Run at search velocity in positive direction to the positive limit switch. 2. Travel at crawling velocity in negative direction until the limit switch becomes inactive. This position is taken as the homing point. 3. If this is parameterised: travel at positioning velocity to the axis zero point.	
17h	23	**Reference switch in positive direction** 1. If reference switch inactive: Travel at search velocity in positive direction to the reference switch. If the stop or limit switch is approached: Travel at search velocity in positive direction to the reference switch. 2. Travel at crawling velocity in negative direction until the reference switch becomes inactive. This position is taken as the homing point. 3. If this is parameterised: travel at positioning velocity to the axis zero point.	
1Bh	27	**Reference switch in negative direction** 1. If reference switch inactive: Travel at search velocity in negative direction to the reference switch. If the stop or limit switch is approached: Travel at search velocity in positive direction to the reference switch. 2. Travel at crawling velocity in positive direction until the reference switch becomes inactive. This position is taken as the homing point. 3. If this is parameterised: travel at positioning velocity to the axis zero point.	

1) Only possible for motors with encoder/resolver with index pulse.

2) Limit switches are ignored during travel to the stop.

3) Since the axis is not to remain at the stop, the travel to the axis zero point must be parameterised and the axis zero point offset must be ≠ 0.

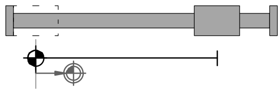
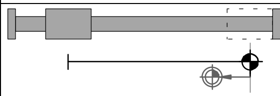
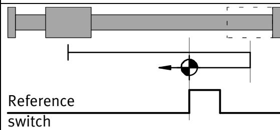
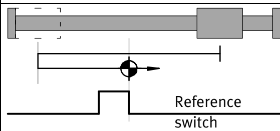
Homing methods

hex	dec	Description	
21h	33	**Index pulse in a negative direction 1)** 1. Travel at crawling velocity in negative direction until the index pulse. This position is taken as the homing point. 2. If this is parameterised: travel at positioning velocity to the axis zero point.	
22h	34	**Index pulse in a positive direction 1)** 1. Travel at crawling velocity in positive direction up to the index pulse. This position is taken as the homing point. 2. If this is parameterised: travel at positioning velocity to the axis zero point.	
23h	35	**Current position** 1. The current position is taken as the reference position. 2. If this is parameterised: travel at positioning velocity to the axis zero point. Note: Through shifting of the reference system, travel to the limit switch or fixed stop is possible. For that reason this method is mostly used for axes of rotation.	
FFh	-1	**Negative stop with index pulse 1) 2)** 1. Travel at search velocity in negative direction to the stop. 2. Travel at crawling velocity in positive direction until the next index pulse. This position is taken as the homing point. 3. If this is parameterised: travel at positioning velocity to the axis zero point.	
FEh	-2	**Positive stop with index pulse 1) 2)** 1. Travel at search velocity in positive direction to the stop. 2. Travel at crawling velocity in negative direction until the next index pulse. This position is taken as the homing point. 3. If this is parameterised: travel at positioning velocity to the axis zero point.	

1) Only possible for motors with encoder/resolver with index pulse.

2) Limit switches are ignored during travel to the stop.

3) Since the axis is not to remain at the stop, the travel to the axis zero point must be parameterised and the axis zero point offset must be ≠ 0.

Homing methods			
hex	dec	Description	
EFh	-17	**Negative stop** 1) 2) 3) 1. Travel at search velocity in negative direction to the stop. This position is taken as the homing point. 2. If this is parameterised: travel at positioning velocity to the axis zero point.	
EEh	-18	**Positive stop** 1) 2) 3) 1. Travel at search velocity in positive direction to the stop. This position is taken as the homing point. 2. If this is parameterised: travel at positioning velocity to the axis zero point.	
E9h	-23	**Reference switch in positive direction with travel to stop or limit switch.** 1. Run at search velocity in positive direction to stop **or limit switch**. 2. Travel at search velocity in negative direction to the reference switch. 3. Travel at crawling velocity in negative direction until the reference switch becomes inactive. This position is taken as the homing point. 4. If this is parameterised: travel at positioning velocity to the axis zero point.	
E5h	-27	**Reference switch in negative direction with travel to stop or limit switch** 1. Run at search velocity in negative direction to stop **or limit switch**. 2. Travel at search velocity in positive direction to the reference switch. 3. Run at crawling velocity in positive direction until reference switch becomes inactive. This position is taken as the homing point. 4. If this is parameterised: travel at positioning velocity to the axis zero point.	

1) Only possible for motors with encoder/resolver with index pulse.
2) Limit switches are ignored during travel to the stop.
3) Since the axis is not to remain at the stop, the travel to the axis zero point must be parameterised and the axis zero point offset must be ≠ 0.

Tab. 10.5 Overview of homing methods

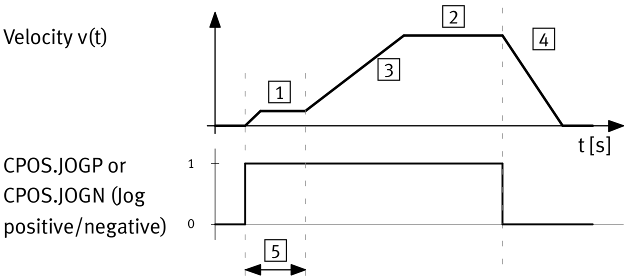
10.4 Jog mode

In the “Operation enabled” state, the drive can be traversed by jogging in the positive/negative directions. This function is usually used for:

- Approaching teach positions,
- Running the drive out of the way (e.g. after a system malfunction),
- Manual traversing as a normal operating mode (manually operated feed).

Process

1. When one of the signals “Jog positive / Jog negative” is set, the drive starts to move slowly. Due to the slow velocity, a position can be defined very accurately.
2. If the signal remains set for longer than the configured “phase 1 period” the velocity is increased until the configured maximum velocity is reached. In this way large strokes can be traversed quickly.
3. If the signal changes to 0, the drive is braked with the pre-set maximum 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 exceeded; the path for stopping is taken into account according to the ramp set. The jog mode can be exited here with Jogging = 0.



1	Low velocity phase 1 (slow travel)
2	Maximum velocity for phase 2
3	Acceleration
4	Deceleration
5	Time period in phase 1

Fig. 10.1 Sequence chart for jog mode

Overview of parameters and I/Os in jog mode		
Parameters involved	**Parameters**	**PNU**
➔ Section B.4.9	Jog mode crawling velocity – phase 1	530
	Jog mode max. speed – phase 2	531
	Jog mode acceleration	532
	Jog mode deceleration	533
	Jog mode slow motion time (T1)	534
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 by PLC/fieldbus	
	Controller in status “Operation enabled”	

Tab. 10.6 Parameters and I/Os during jog mode

10.5 Teaching via fieldbus

Position values can be taught via the fieldbus. Previously taught position values will then be overwritten.

Note: The drive must not stand still for teaching. However, with the typical cycle times of the PLC + fieldbus + controller, there will be inaccuracies of several millimetres even at a velocity of only 100 mm/s.

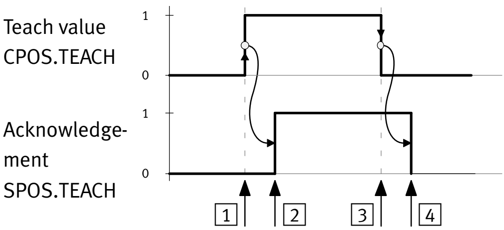
Process

1. The drive will be moved to the desired position by 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. The user must make sure that the desired parameter is selected. For this, the parameter “Teach target” and, if applicable, the correct record address must be entered.

Teach target (PNU 520)	Is taught	
= 1 (specification)	Setpoint position in the positioning record.	Record selection: Positioning record after control byte 3
		Direct mode: Positioning record after PNU=400
= 2	Axis zero point	
= 3	Project zero point	
= 4	Lower software end position.	
= 5	Upper software end position.	

Tab. 10.7 Overview of teach targets

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



1 PLC: Prepare teaching
2 Controller: Ready for teaching
3 PLC: Teach now
4 Controller: Value transferred

Fig. 10.2 Handshake during teaching

Taught parameters must be saved securely against power outages with PNU 127.

Overview of parameters and I/Os when teaching		
Parameters involved	**Parameters**	**PNU**
➔ Sections B.4.8, B.4.9	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 = Falling edge: Teach value	
Acknowledgement (FHPP)	SPOS.TEACH = 1: Value transferred	
Requirement	Device control by PLC/fieldbus	
	Controller in status “Operation enabled”	

Tab. 10.8 Parameters and I/Os when teaching

10.6 Carry out record (Record selection)

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 PLC,
– processing of a positioning profile by linking records,
– known target positions that seldom change (recipe change).

Process

1. Set the desired record number in the output data of the PLC. Until the start, the controller replies 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 controller signals with the rising edge at Start Acknowledgment that the PLC output data has been accepted and that the positioning job 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:
– No homing was carried out (where necessary, see PNU 1014).
– the target position and/or the preselect position cannot be reached.
– Invalid record number.
– Record not initialised.

With conditional record switching/record chaining (see section 10.6.3):
If a new velocity and/or a new target position is specified in the movement, the remaining path to the target position must be large enough to reach a standstill with the braking ramp that was set.

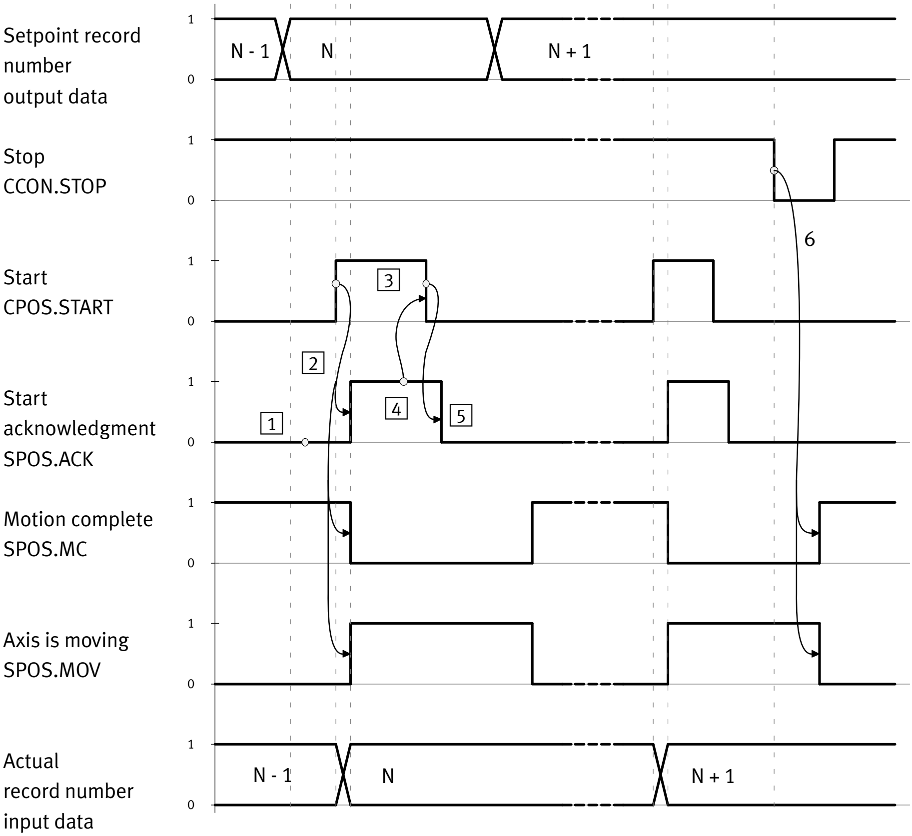
Overview of parameters and I/Os in record selection		
Parameters involved	**Parameters**	**PNU**
➔ Section B.4.8	Record number	400
	All parameters of the record data, see section 10.6.2, Tab. 10.10	401 ... 421
Start (FHPP)	CPOS.START = rising edge: Start Jogging and referencing have priority.	
Acknowledgement (FHPP)	SPOS.MC = 0: Motion Complete	
	SPOS.ACK = rising edge: Start acknowledgment	
	SPOS.MOV = 1: Drive moves	
Requirement	Device control by PLC/fieldbus	
	Controller in status “Operation enabled”	
	Record number must be valid	

Tab. 10.9 Parameters and I/Os with record selection

10.6.1 Record selection flow diagrams

Fig. 10.3, Fig. 10.4 and Fig. 10.5 show typical flow diagrams for starting and stopping a record.

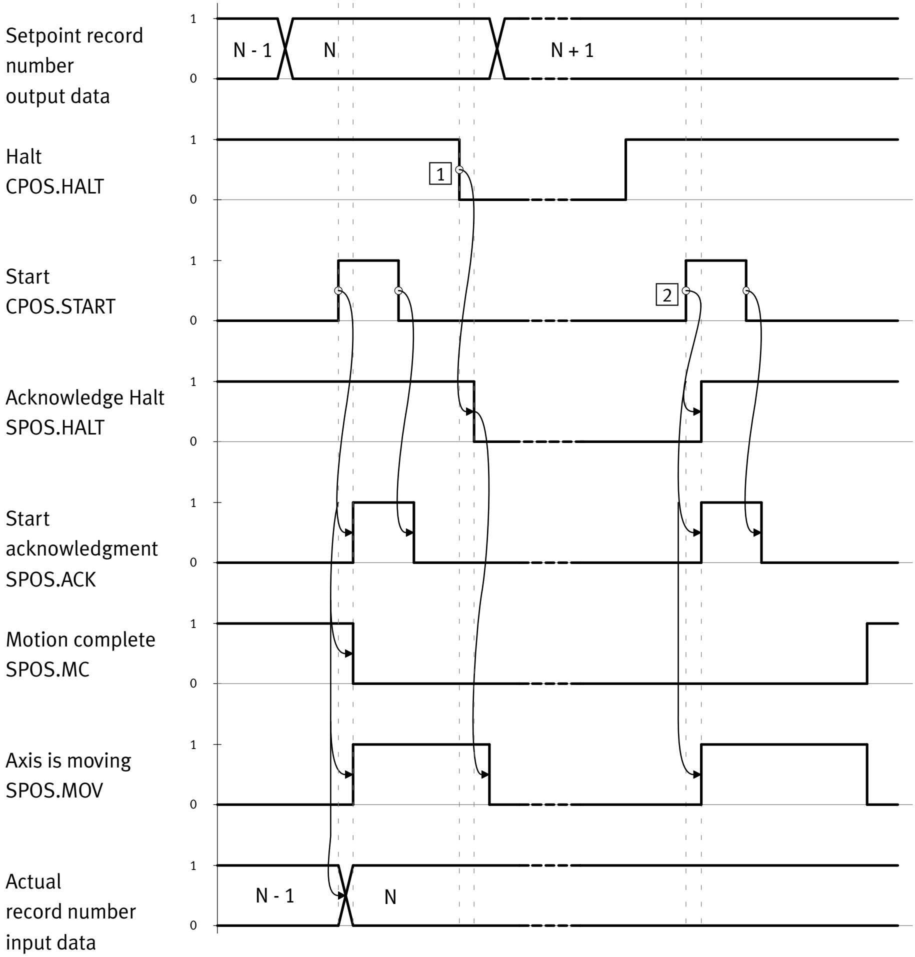
Record start / stop



1 Requirement: “Start acknowledgement” = 0
2 A 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 PLC, “Start” may be set to 0 again
4 The controller reacts with a falling edge at “Start acknowledgment”
5 As soon as “Start acknowledgment” is recognized by the PLC, it can create the next record number
6 A currently running positioning task can be stopped with “Stop”.

Fig. 10.3 Flow diagram Record start/stop

Stop record with halt and continue

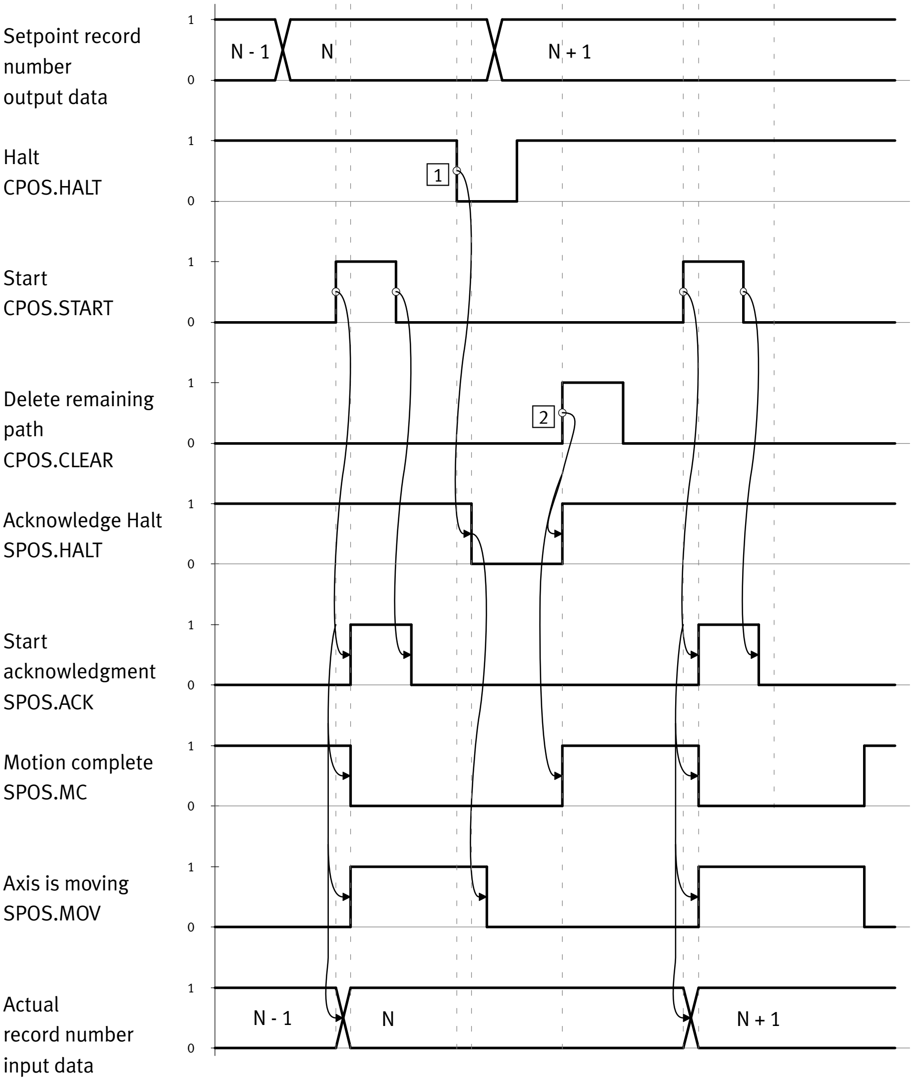


1 Record is stopped with "Halt", actual record number N is retained, "Motion Complete" remains reset

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

Fig. 10.4 Flow diagram for Stop record with halt and continue

Stop record with halt and delete remaining path



1 Stop record
2 Delete remaining path

Fig. 10.5 Flow diagram for stop record with halt and delete remaining path

10.6.2 Record structure

A positioning task in record select mode is described by a record made up of setpoint values. Every setpoint value is addressed by its own 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, position/torque control, ...
402	Record control byte 2	Record control: Settings for conditional record switching and record chaining.
404	Setpoint value	Setpoint value corresponding to record control byte 1.
406	Velocity	Setpoint velocity.
407	Acceleration	Setpoint acceleration during start up.
408	Deceleration	Setpoint acceleration during braking.
413	Jerk-free filter time	Filter time for smoothing the profile ramps.
416	Record following position/record control	Record number that is jumped to if the step enabling condition is met.
418	Torque limitation	limitation of the maximum torque.
419	Cam disc number	Number of the cam disc to be executed with this record. Requires configuration of PNU 401 (virtual master).
420	Remaining path message	Path in front of the target position where a display can be triggered via a digital output to show it has been reached.
421	Record control byte 3	Settings for specific behaviour of the record.

Tab. 10.10 Parameters for positioning record

10.6.3 Conditional record switching / record chaining (PNU 402)

Record selection mode allows multiple positioning jobs to be concatenated. 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 velocity after a position is reached.
To do this, the user sets a (decimal) condition in RCB2 to define that the subsequent record is automatically executed after the current record.

Complete parameterisation of record chaining ("path program"), such as of the subsequent record, is only possible through the FCT.

If a condition was defined, it is possible to prohibit automatic continuation by setting the B7 bit. This function should be used for debugging using FCT and not for normal control purposes.

Record control byte 2 (PNU 402)		
Bit 0 ... 6	Numerical value 0...128: step enabling condition as a list, see Tab. 10.12	
Bit 7	= 0:	Record switching (bit 0 ... 6) is not blocked (default)
	= 1:	Record switching blocked

Tab. 10.11 Settings for conditional record switching and record chaining

Step enabling conditions		
Value	Condition	Description
0	–	No automatic continuation
4	Rest	Continuation occurs once the drive comes to rest and the time T1 specified as the preselected value has expired. (Run to block!).
6	Input Pos. edge	Continuation occurs to the next record if a rising edge is identified at the local input. The preselected value includes the bit address of the input. Preselected value = 1: NEXT1 Preselected value = 2: NEXT2
7	Input Neg. edge	Continuation occurs to the next record if a falling edge is identified at the local input. The preselected value includes the bit address of the input. Preselected value = 1: NEXT1 Preselected value = 2: NEXT2
9	Input Pos. edge waiting	Continuation occurs to the next record after the current record ends if a rising edge is identified at the local input. The preselected value includes the number of the input: Preselected value = 1: NEXT1 Preselected value = 2: NEXT2
10	Input Neg. edge waiting	Continuation occurs to the next record after the current record ends if a falling edge is identified at the local input. The preselected value includes the number of the input: Preselected value = 1: NEXT1 Preselected value = 2: NEXT2

Tab. 10.12 Step enabling conditions

10.7 Direct mode

In the status “Operation enabled” (Direct mode) a task is formulated directly in the I/O data and is transmitted via the fieldbus. Some of the setpoint values for the position are reserved in the PLC.
The function is used in the following situations:
– Selection-free approach to positions within the effective stroke.
– The target positions are unknown during designing or change frequently (e.g. several different workpiece positions).
– A positioning profile through linking of records (G25 function) is not necessary.
– The drive should follow a nominal value continuously.

If short wait times are not critical, it is possible to implement a positioning profile externally PLC-controlled by linking records.

Causes of errors in application
– No homing was carried out (where necessary, see PNU 1014).
– Target position cannot be reached or lies outside the software end positions.
– Load torque is too large.

Overview of parameters and I/Os in direct mode			
Parameters involved		**Parameters**	**PNU**
	Position specifications ➔ B.4.12	Basic value velocity 1)	540
		Direct mode acceleration	541
		Direct mode deceleration	542
		Jerk-free filter time	546
	Torque specifications ➔ B.4.13	Base value torque ramp 1)	550
		Torque target window	552
		Damping time	553
		Permissible velocity during torque control	554
	Rotational velocity specifications ➔ B.4.14	Base value acceleration ramp 1)	560
		Velocity target window	561
		Damping time target window	562
		Standstill target window	563
		Standstill target window damping time	563
		Torque limitation	565
Start (FHPP)		CPOS.START = rising edge: Start	
		CDIR.ABS = setpoint position absolute/relative	
		CDIR.B1/B2 = control mode (see section 9.4.3)	
Acknowledgement (FHPP)		SPOS.MC = 0: Motion Complete	
		SPOS.ACK = rising edge: Start acknowledgment	
		SPOS.MOV = 1: Drive moves	
Requirement		Device control by PLC/fieldbus	
		Controller in status “Operation enabled”	

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

Tab. 10.13 Parameters and I/Os in direct mode

10.7.1 Position control process

1. The user sets the desired setpoint value (position) and the positioning condition (absolute/relative, percentage velocity) in his or her output data.
2. With a rising edge at Start (CPOS.START), the controller accepts the setpoint values and starts the positioning job. After the start, a new setpoint value can be started at any time. There is no need to wait for MC.
3. Once the last setpoint position is reached, MC (SPOS.MC) is set.

Starting the positioning job

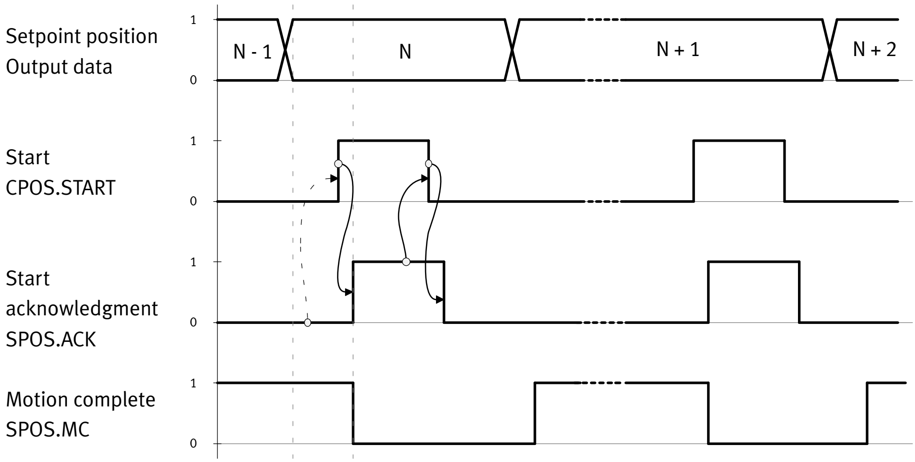


Fig. 10.6 Start the positioning task

The sequence of the remaining control and status bits as well as the functions Hold and Stop react corresponding to the record select function, see Fig. 10.3, Fig. 10.4 and Fig. 10.5.

10.7.2 Sequence for force mode (torque, current control)

Force mode is prepared by switching over the control mode with the bits CDIR - COM1/2. The drive stands with the position controlled.
After the setpoint specification, the start signal (start bit) creates the torque/moment using the torque ramp in the direction indicated by the prefix of the setpoint value and the active torque control mode is displayed via the SDIR - COM1/2 bits.
The velocity is limited to the value in the parameter “Maximum velocity”.
Once the setpoint value has been reached, taking into account the target window and the time window, the “MC” signal is set. Torque/moment continue to be controlled.

Causes of errors in application

– No homing was carried out (where necessary, see PNU 1014).

Setpoint specification / actual value query in direct mode in force mode:

CCON.OPM1 = 1, CCON.OPM2 = 0

CDIR.COM1 = 1, CDIR.COM2 = 0

Direct mode								
	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Output data	CCON	CPOS	CDIR	Setpoint value 1 (Force ramp[1])	Setpoint value 2 (torque)			
Input data	SCON	SPOS	SDIR	Actual value 1 (actual torque)	Actual value 2 (Actual position)			

1) From FW 4.0.1501.2.3 ➔ 9.4.4

Tab. 10.14 Control and status bytes for force mode direct mode

Data	Significance	Unit
Setpoint value 1	Force ramp[2]	Force ramp in % of the base value (PNU 550)
Setpoint value 2	Setpoint torque	Percentage of nominal torque (PNU 1036)
Actual value 1	Actual torque	Percentage of nominal value (PNU 1036)
Actual value 2	Actual position	Positioning unit, see appendix A.1

2) From FW 4.0.1501.2.3 ➔ 9.4.4

Tab. 10.15 Setpoint and actual values for force mode direct mode

10.7.3 Velocity adjustment process

Velocity adjustment is requested by switching the control mode. The drive remains in the operation mode that was set previously. After setpoint specification, the start signal (start bit) switches the system to the velocity adjustment operating mode and the velocity setpoint value comes into effect. The torque is limited here to the value set in the “torque limiting” parameter (PNU 565).

The signal “MC” (Motion Complete) is used in this control mode to mean “target velocity reached”:

Motion Complete / standstill notification

The same comparator type is used to determine “velocity reached” and “velocity 0” and it behaves in a manner corresponding to Fig. 10.7, see Tab. 10.16.

Setpoint value	Specifications for reaching MC (Motion Complete)	
≠ 0	Target velocity:	Setpoint value in accordance with input data
	Tolerance:	Velocity target window (PNU 561)
	Settling time	Damping time velocity target window (PNU 562)
= 0	Target velocity:	Setpoint value in accordance with input data
	Tolerance:	Standstill target window (PNU 563)
	Settling time	Standstill target window damping time (PNU 564)

Tab. 10.16 Motion Complete / standstill notification specifications

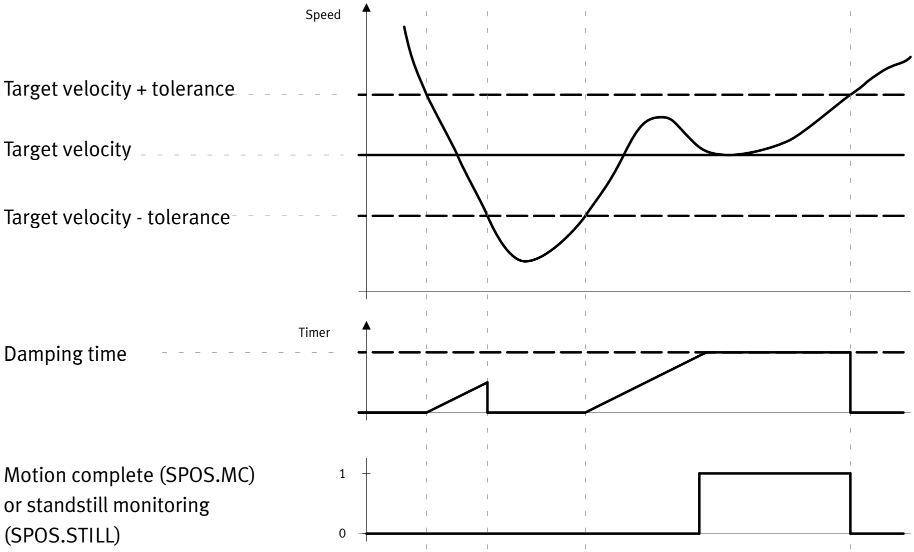
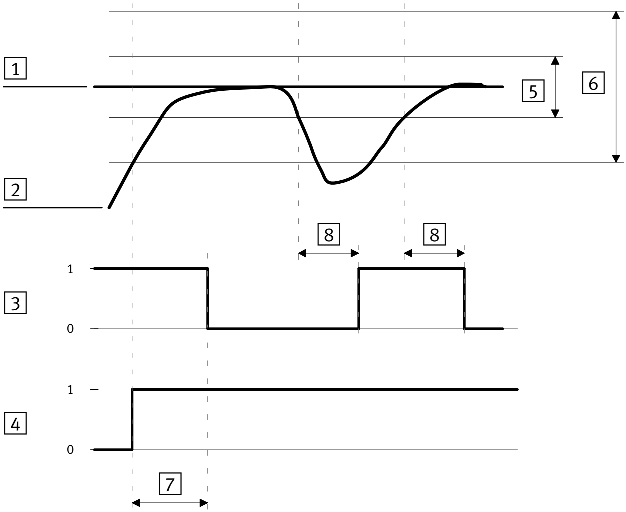


Fig. 10.7 Motion complete / standstill notification

10.8 Standstill monitoring

Standstill monitoring responds when the drive leaves the target position window when at a standstill. Standstill monitoring is based on position control only.
When the target position has been reached and MC is signaled in the status word, the drive switches to the "standstill" state and bit SPOS.STILL (standstill monitor) is reset. If, in this status, the drive is removed from the standstill position window for a defined time due to external forces or other influences, the bit SPOS.STILL will be set.
As soon as the drive is in the standstill position window again for the standstill monitoring time, the bit SPOS.STILL will be reset.
The standstill monitoring cannot be switched on or off explicitly. It becomes inactive when the standstill position window is set to "0".



1 Target position
2 Actual position
3 Standstill monitoring (SPOS.STILL)
4 Motion complete (SPOS.MC)
5 Standstill position window
6 Target position window
7 Monitoring time (position window time)
8 Standstill monitoring time

Fig. 10.8 Standstill monitoring

Overview of parameters and I/Os in standstill monitoring		
Parameters involved	**Parameters**	**PNU**
➔ Section B.4.18	Target position window	1022
	Adjustment time for position	1023
	Setpoint position	1040
	Current position	1041
	Standstill position window	1042
	Standstill monitoring time	1043
Start (FHPP)	SPOS.MC = rising edge: Motion complete	
Acknowledgement (FHPP)	SPOS.STILL = 1: Drive has moved out of standstill position window	
Requirement	Device control by PLC/fieldbus	
	Controller in status “Operation enabled”	

Tab. 10.17 Parameters and I/Os in standstill monitoring

10.9 Flying measurement (position sampling)

To find out whether this function is supported by the controller you are using and its firmware version, see the help for the associated FCT plug-in.

The local digital inputs can be used as fast sample inputs: With every rising and falling edge at the configured sample input (only possible using the FCT), the current position value is written into a register of the controller and can afterwards be read out (PNU 350:01/02) by the higher-order controller (PLC/IPC).

Parameters for position sampling (flying measurement)	PNU
Position value for a rising edge in user-defined units	350:01
Position value for a falling edge in user-defined units	350:02

Tab. 10.18 Parameters for flying measurement

10.10 Operation of cam discs

The CMMP-AS has the option of operating 16 cam disks each with 4 cam tracks assigned to it.

For this function, you will need the software GSPF-CAM-MC-...

The CMMP-AS provides the following functionality for this purpose via FHPP:
- Operation in synchronisation with an external input, slave mode.
- Operation in synchronisation with an external input with cam disc, slave mode.
- Virtual master (internal) with cam disc.

Control is possible in the following operating modes:
- Record selection.
- Direct mode, positioning.

The cam discs are parameterised via the FCT plug-in. For information about parametrisation, see the help for the CMMP-AS plug-in.
For complete information on the cam disc function, see the special cam disc manual.

10.10.1 Cam disc function in direct mode operating mode

Synchronisation with an external master controller with cam disc (slave operation)
Synchronisation operation allows a slave controller to follow a master controller via an additional external input in accordance with parameterised rules.
This can be purely position synchronisation or it can be done with an additional cam disc function, the CAM function.

Activating synchronisation operation in the direct mode:
Synchronised operation can be selected with control byte 3, CDIR by setting CDIR.FUNC, and the desired functionality can be selected in the function group and the function number, CDIR.FNUM1/2 and CDIR.FGRP1/2.
Synchronised operation is then activated with a rising edge at the bit CPOS.START. The bit CCON.STOP stops synchronisation operation. The bit CPOS.HALT has no intermediate stop function (changes to ready with a stop ramp). The negative edge of CPOS.START also stops synchronisation operation.

Setpoint and actual values according to the function numbers

Function number	Allocation of the setpoint/actual values	
FNUM = 0: reserved	–	
FNUM = 1, FNUM = 2: synchronisation operation without/with cam disc	Setpoint value 1:	No importance, since the position setpoint comes via the external input.
	Setpoint value 2:	No importance, since the position setpoint comes via the external input.
	Actual value 1:	Actual velocity of the slave as in position mode (after the cam disc)
	Actual value 2:	Actual position of the slave as in position mode (after the cam disc)
FNUM = 3: Virtual master (internal) with cam disc	Setpoint value 1:	Setpoint velocity of the master, dependent on the operating mode of the master
	Setpoint value 2:	Setpoint position of the master, dependent on the operating mode of the master
	Actual value 1:	Actual velocity of the slave (after the cam disc)
	Actual value 2:	Actual position of the slave (after the cam disc)

Tab. 10.19 Allocation of setpoint/actual values

The cam disc is selected through PNU 700.
FHPP+ can be used to map this selection to the process data.

10.10.2 Cam disc function in record selection mode

In record selection, the type of record is defined with the record control byte in the record list. The expansion to the cam disc operation can be activated as in direct mode with the bit provided for general function expansion, bit 7 (FUNC) in record control byte 1.
The cam disc number is selected with PNU 419. If PNU 419 = 0, the contents of PNU 700 are used.

10.10.3 Parameters for the cam disc function

The parameters for the cam disc function can be found in section B.4.16.

10.10.4 Extended finite state machine with cam disc function

Information on the finite state machine for the cam disc function can be found in section 9.6.3

10.11 Display of drive functions

Additional internal positioning records are used for the various drive functions. This is also shown on the 7-segment display during execution ➔ see functional description GDSP-CMMP-M...-FW-...

Position record	Description	Display
0	Starts homing.	see 256 ... 258
1 ... 250	FHPP positioning records can be started via FHPP in Record Select mode.	P001 ... P250
251 ... 255	Additional positioning records that can be parameterised via FCT can be started via I/O or via record chaining.	P251 ... P255
256 ... 258	Homing, display of the various phases.	
	256: Search for reference point	PH0
	257: Crawl	PH1
	258: Approach zero point	PH2
259	Jog positive	P259
260	Jog negative	P260
262	CAM-IN / CAM-OUT (cam disc).	P262
264	FCT direct record, used for manual travel via FCT.	P264
265	FHPP direct record, used for FHPP direct operation.	P265

Tab. 10.20 Overview of positioning records

11 Malfunction behaviour and diagnostics

11.1 Classification of malfunctions

We differentiate between the following types of malfunctions:
- warnings,
- malfunction type 1 (output stage is not switched off),
- malfunction type 2 (output stage is switched off).

Classification of the possible malfunctions can be partially parameterised ➔ column appendix D.
The controllers signal errors or malfunctions by appropriate error messages or warnings. These can be evaluated via the following options:
- display,
- status bytes (see section 11.4),
- bus-specific diagnostics (see fieldbus-specific chapter),
- diagnostic memory (see section 11.2),
- FCT (see FCT help).

The motor controller has a temporary and a permanent diagnostic memory. Access via FHPP is always to the temporary memory.

The list of diagnostic messages can be found in appendix D.

11.1.1 Warnings

A warning is information for the user, which has no influence on the behaviour of the drive.

Behaviour in the event of warnings
- Controller and output stage remain active.
- The current positioning is not interrupted.
- Dependent on the malfunction number, a new positioning task may be possible.
- The SCON.WARN bit is set.
- If the cause of the warning disappears, the SCON.WARN bit is automatically deleted again.
- The warning numbers are logged in the warning register (PNU 211).

Causes of warnings
- Parameters cannot be written or read (not permissible in the operating status, invalid PNU, ...).
- Following error, drive has exceeded the tolerance after Motion Complete and similar minor control errors.

11.1.2 Malfunction type 1

In the event of an error, the performance that was requested cannot be provided. The drive switches from its current status to the “Fault” status.

Behaviour in the event of type 1 malfunctions

- The output stage is not switched off.
- The current positioning task is interrupted.
- The velocity is reduced on the emergency ramp.
- The sequence control switches to the Fault status. No new positioning task can be carried out.
- The SCON.FAULT bit is set.
- The “Fault” status can be exited through switch-off, through a positive edge at input CCON.RESET or through resetting/setting DIN5 (controller enable).
- Holding brake is activated when the drive is stopped.

Causes of type 1 malfunctions

- Software end positions are violated.
- Motion Complete timeout.
- Following error monitoring.

11.1.3 Fault type 2

In the event of an error, the performance that was requested cannot be provided. The drive switches from its current status to the “Fault” status.

Behaviour in the event of type 2 malfunctions

- The output stage is switched off.
- The current positioning task is interrupted.
- The drive runs down.
- The sequence control switches to the Fault status. No new positioning task can be carried out.
- The SCON.FAULT bit is set.
- The “Fault” status can be exited through switch-off, through a positive edge at input CCON.RESET or through resetting/setting DIN5 (controller enable).
- Holding brake is activated when the drive is stopped.

Causes of type 2 malfunctions

- Load voltage is missing (e.g. if emergency off has been implemented).
- Hardware error:
  - Measuring system error.
  - Bus error.
  - SD card error.
- Impermissible operating mode change.

11.2 Diagnostic memory (malfunctions)

The diagnostic memory for malfunctions contains the codes of the last malfunction messages that occurred. The diagnostic memory is protected against power failure, if possible. If the diagnostic memory is full, the oldest element will be overwritten (FIFO principle).

Structure of the diagnostic memory			
Parameters [1]	200	201	202
Format	uint8	uint16	uint32
Significance	Diagnostic event	Malfunction number	Time
Subindex 1	Most recent/current malfunction		
Subindex 2	2nd stored malfunction		
... [2]	...		
Subindex 32	32nd stored malfunction		

1) See section B.4.5

Tab. 11.1 Structure of diagnostic memory

11.3 Warning memory

The warning memory contains the codes of the last warnings that occurred. It functions in the same way as the diagnostic memory for malfunctions.

Structure of the warning memory			
Parameters [1]	210	211	212
Format	uint8	uint16	uint32
Significance	Warning event	Warning number	Time
Subindex 1	Latest / current warning		
Subindex 2	2nd stored warning		
... [2]	...		
Subindex 32	32nd stored warning		

1) See section B.4.5

Tab. 11.2 Structure of the warning memory

11.4 Diagnosis using FHPP status bytes

The controller supports the following diagnostics options using FHPP status bytes (see section 9.4):
- SCON.WARN – warning
- SCON.FAULT – malfunction
- SPOS.DEV – following error
- SPOS.STILL – standstill monitoring.

In addition, all diagnostic information available as PNU can be read (e.g. the diagnostic memory) through FPC (Festo Parameter Channel ➔ section C.1) or FHPP+ (➔ appendix C.2).

A Technical appendix

A.1 Conversion factors (factor group)

A.1.1 Overview

Motor controllers are used in a wide variety of applications: as direct drives, with downstream gear units, for linear drives, etc.
In order to enable simple parameterisation for all applications, the motor controller can be parameterised with the parameters in the "Factor Group" (PNU 1001 to 1007, see section B.4.18) in such a way that variables such as the rotational velocity can be directly specified or read in the units of measurement required.
The motor controller then uses the factor group to calculate the entries in its internal units of measurement. One conversion factor is available for each of the physical parameters: position, velocity and acceleration. These conversion factors adjust the user's units of measurement to the application in question.
Fig. A.1 clarifies the function of the factor group:

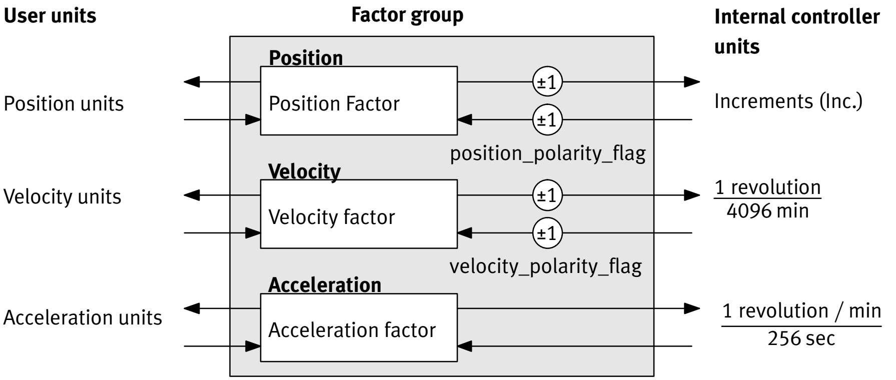


Fig. A.1 Factor group

All parameters are always saved in the motor controller in its internal units of measurement and are only converted (using the factor group) when the parameters are written or read out.
For this reason, the factor group should be set first during parameterisation and should not be changed again during parameterisation.
The factor group is set to the following units by default:

Size	Designation	Unit	Explanation
Length	Position units	Increments	65536 increments per revolution
Velocity	Velocity units	min^{-1}	Revolutions per minute
Acceleration	Acceleration units	$(min^{-1})/s$	Rotational velocity increase per second

Tab. A.1 Factor group presettings

A.1.2 Objects in the factor group

Tab. A.2 shows the parameters in the factor group.

Name	PNU	Object	Type	Access
Polarity (reversal of direction)	1000	Var	uint8	rw
Position Factor	1004	Array	uint32	rw
Velocity factor	1006	Array	uint32	rw
Acceleration factor	1007	Array	uint32	rw

Tab. A.2 Overview of the factor group

Tab. A.3 shows the parameters involved in the conversion.

Name	PNU	Object	Type	Access
Encoder Resolution	1001	Array	uint32	rw
Gear ratio	1002	Array	uint32	rw
Feed constant	1003	Array	uint32	rw
Axis parameter	1005	Array	uint32	rw

Tab. A.3 Overview of parameters involved

A.1.3 Calculation of the position units

The **position factor** (PNU 1004, see section B.4.18) is used to convert all the length values from the user's **positioning units** into the internal unit **increments** (65536 increments are equivalent to one motor revolution). The position factor consists of a numerator and a denominator.

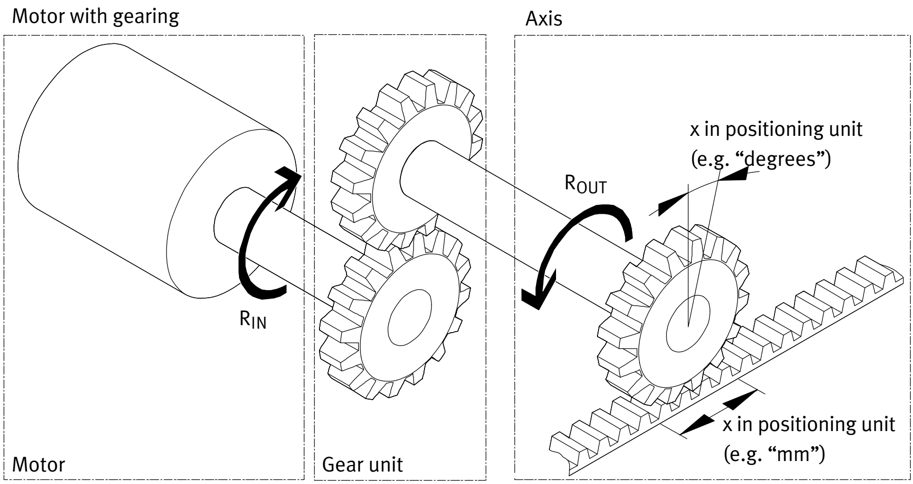


Fig. A.2 Calculation of the position units

The following parameters are involved in the position factor's calculation formula:

Parameters	Description
Gear ratio	Gear ratio between revolutions at the input shaft (R_{IN}) and revolutions at the output shaft (R_{OUT}).
Feed constant	Ratio between movement in position units at the drive and revolutions at the drive-out of the gear unit (R_{OUT}). Example: 1 revolution ≙ 63.15 mm or 1 revolution ≙ 360° degrees.

Tab. A.4 Position factor parameters

The position factor is calculated in accordance with the following formula:

$$\text{Position factor} = \frac{\text{gear ration} \; * \; \text{increments/revolution}}{\text{feed constant}}$$

The position factor must be written to the motor controller separated into numerators and denominators. It can therefore be necessary to interpolate the fraction to integers.

Example

First, the desired unit (column 1) and the desired number of decimal places (dp) have to be specified, along with the application's gear ratio and its feed constant (if applicable). The feed constant is then displayed in the desired positioning units (column 2).
In this way, all the values can be entered into the formula and the fraction can be calculated:

Position factor calculation sequence				
Position units	Feed constant	Gear ratio	Formula	Result shortened
Degree, 1 DP ➔ 1/10 degree $\boxed{(°/_{10})}$	**1** R_{OUT} = $\boxed{\mathbf{3600}\ \frac{°}{10}}$	$\boxed{\mathbf{1/1}}$	$\frac{\frac{1}{1} * 65536\ \text{Inc}}{3600\ \frac{°}{10}} = \frac{\mathbf{65536}\ \text{Inc}}{\mathbf{3600}\ \frac{°}{10}}$	num : 4096 div : 225

Fig. A.3 Position factor calculation sequence

Examples of calculating the position factor				
Position units 1)	Feed constant 2)	Gear ratio 3)	Formula 4)	Result shortened
Increments, 0 DP ➔ Inc.	**1** R_{OUT} = **65536** Inc	**1/1**	$\frac{\frac{1}{1} * 65536\ Inc}{65536\ Inc} = \frac{1\ Inc}{1\ Inc}$	$\frac{num : 1}{div : 1}$
Degree, 1 DP ➔ 1/10 degree (°/10)	**1** R_{OUT} = **3600** $\frac{°}{10}$	**1/1**	$\frac{\frac{1}{1} * 65536\ Inc}{3600\ \frac{°}{10}} = \frac{\mathbf{65536}\ Inc}{\mathbf{3600}\ \frac{°}{10}}$	$\frac{num : 4096}{div : 225}$
Rev., 2 DP ➔ 1/100 Rev. (R/100)	**1** R_{OUT} = **100** $\frac{U}{100}$	**1/1**	$\frac{\frac{1}{1} * 65536\ Inc}{100\ \frac{1}{100}} = \frac{\mathbf{65536}\ Inc}{\mathbf{100}\ \frac{1}{100}}$	$\frac{num : 16384}{div : 25}$
		2/3	$\frac{\frac{2}{3} * 65536\ Inc}{100\ \frac{1}{100}} = \frac{\mathbf{131072}\ Inc}{\mathbf{300}\ \frac{1}{100}}$	$\frac{num : 32768}{div : 75}$
mm, 1 DP ➔ 1/10 mm (mm/10)	**1** R_{OUT} = **631, 5** $\frac{mm}{10}$	**4/5**	$\frac{\frac{4}{5} * 65536\ Inc}{631,5\ \frac{mm}{10}} = \frac{\mathbf{2621440}\ Inc}{\mathbf{31575}\ \frac{mm}{10}}$	$\frac{num: 524288}{div: 6315}$

1) Desired unit at the drive-out

2) Positioning units per revolution at the drive-out (R_{OUT}). Feed constant of the drive (PNU 1003) * 10^{-DP} (points after the decimal)

3) Revolutions at the drive in per revolutions at the drive-out (R_{IN} per R_{OUT})

4) Insert values into equation.

Tab. A.5 Examples of calculating the position factor

A.1.4 Calculating the velocity units

The **velocity factor** (PNU 1006, see section B.4.18) is used to convert all the velocity values from the user’s **units of velocity** into the internal unit **revolutions per 4096 minutes**.
The velocity factor consists of a numerator and a denominator.
Calculation of the velocity factor consists of two parts: a conversion factor from internal length units into the user’s position units and a conversion factor from internal time units into user-defined time units (e.g. from seconds to minutes). The first part corresponds to calculating the position factor, while for the second part an additional factor comes into play:

Parameters	Description
Time factor_v	The ratio between the internal time unit and the user-defined time unit.
Gear ratio	Gear ratio between revolutions at the input shaft (R_{IN}) and revolutions at the output shaft (R_{OUT}).
Feed constant	Ratio between movement in position units at the drive and revolutions at the drive-out of the gear unit (R_{OUT}). Example: 1 revolution ≙ 63.15 mm or 1 revolution ≙ 360° degrees.

Tab. A.6 Velocity factor parameters

The velocity factor is calculated in accordance with the following formula:

$$\text{Speed factor} = \frac{\text{gear ratio} \;*\; \text{time factor\_v}}{\text{feed constant}}$$

Like the position factor, the velocity factor also has to be written to the motor controller separated into numerators and denominators. It can therefore be necessary to interpolate the fraction to integers.

Example

First, the desired unit (column 1) and the desired number of decimal places (dp) have to be specified, along with the application's gear ratio and its feed constant (if applicable). The feed constant is then displayed in the desired positioning units (column 2).
Then, the desired unit of time is converted into the motor controller's unit of time (column 3).
In this way, all the values can be entered into the formula and the fraction can be calculated:

Velocity factor calculation sequence					
Velocity units	Feed const.	Time constant	Gear	Equation	Result shortened
mm/s, 1 DP ➔ 1/10 $^{mm}/_{s}$ $\boxed{(^{mm}/_{10\,s})}$	$63,15\ \frac{mm}{R}$ ⇒ $1\ R_{OUT}$ = $\boxed{631,5\ \frac{mm}{10}}$	$\boxed{1\ \frac{1}{s}}$ = $60\ \frac{1}{min}$ = $\boxed{60 * 4096\ \frac{1}{4096\ min}}$	$\boxed{4/5}$	$\dfrac{\frac{4}{5} * \dfrac{60 * 4096\ \frac{1}{4096\,min}}{1\ \frac{1}{s}}}{631,5\ \frac{mm}{10}} = \dfrac{1966080\ \frac{1}{4096\,min}}{6315\ \frac{mm}{10s}}$	num: 131072 div: 421

Fig. A.4 Velocity factor calculation sequence

Examples of calculating the velocity factor					
Velocity units 1)	Feed const. 2)	Time constant 3)	Gear 4)	Equation 5)	Result shortened
R/min, 0 DP ➔ R/min	$\mathbf{1}\ R_{OUT} = \mathbf{1}\ R_{OUT}$	$\mathbf{1}\ \frac{1}{min} = \mathbf{4096}\ \frac{1}{4096\ min}$	**1/1**	$\dfrac{\frac{1}{1} * \dfrac{4096 \frac{1}{4096\,min}}{1 \frac{1}{min}}}{1} = \dfrac{\mathbf{4096}\ \frac{1}{4096\ min}}{\mathbf{1}\ \frac{1}{min}}$	num: 4096 div: 1
R/min, 2 DP ➔ 1/100 R/min (R/100 min)	$\mathbf{1}\ R_{OUT} = \mathbf{100}\ \frac{R}{100}$	$\mathbf{1}\ \frac{1}{min} = \mathbf{4096}\ \frac{1}{4096\ min}$	**2/3**	$\dfrac{\frac{2}{3} * \dfrac{4096 \frac{1}{4096\,min}}{1 \frac{1}{min}}}{\dfrac{100 \frac{1}{100}}{1}} = \dfrac{\mathbf{8192}\ \frac{1}{4096\ min}}{\mathbf{300}\ \frac{1}{100\ min}}$	num: 2048 div: 75
°/s, 1 DP ➔ 1/10 °/s (°/10 s)	$\mathbf{1}\ R_{OUT} = \mathbf{3600}\ \frac{°}{10}$	$\mathbf{1}\ \frac{1}{s} = \mathbf{60}\ \frac{1}{min} = \mathbf{60 * 4096}\ \frac{1}{4096\ min}$	**1/1**	$\dfrac{\frac{1}{1} * \dfrac{60 * 4096 \frac{1}{4096\ min}}{1 \frac{1}{s}}}{\dfrac{3600 \frac{°}{10}}{1}} = \dfrac{\mathbf{245760}\ \frac{1}{4096\,min}}{\mathbf{3600}\ \frac{°}{10\ s}}$	num: 1024 div: 15
mm/s, 1 DP ➔ 1/10 mm/s (mm/10 s)	$\mathbf{63,15}\ \frac{mm}{R} \Rightarrow \mathbf{1}\ R_{OUT} = \mathbf{631,5}\ \frac{mm}{10}$	$\mathbf{1}\ \frac{1}{s} = \mathbf{60}\ \frac{1}{min} = \mathbf{60 * 4096}\ \frac{1}{4096\ min}$	**4/5**	$\dfrac{\frac{4}{5} * \dfrac{60 * 4096 \frac{1}{4096\,min}}{1 \frac{1}{s}}}{\dfrac{631,5 \frac{mm}{10}}{1}} = \dfrac{\mathbf{1966080}\ \frac{1}{4096\,min}}{\mathbf{6315}\ \frac{mm}{10\ s}}$	num: 131072 div: 421

1) Desired unit at the drive-out

2) Positioning units per revolution at the drive-out (R_{OUT}). Feed constant of the drive (PNU 1003) * 10^{-DP} (points after the decimal)

3) Time factor_v: desired time unit per internal time unit

4) Gear factor: R_{IN} per R_{OUT}

5) Insert values into equation.

Tab. A.7 Examples of calculating the velocity factor

A.1.5 Calculating the acceleration units

The **acceleration factor** (PNU 1007, see section B.4.18) is used to convert all the acceleration values from the user's **units of acceleration** into the internal unit **revolutions per minute per 256 seconds**. The velocity factor consists of a numerator and a denominator.

Calculation of the acceleration factor likewise consists of two parts: a conversion factor from internal units of length into the user’s position units and a conversion factor from internal units of time into user-defined units of time squared (e.g. from seconds^2 to minutes^2). The first part corresponds to calculating the position factor, while for the second part an additional factor comes into play:

Parameters	Description
Time factor_a	Ratio between internal times units squared and user-defined time unit squared (e.g. 1 min² = 1 min * 1 min = 60 s * 1 min = $^{60}/_{256}$ min * s).
Gear ratio	Gear ratio between revolutions at the input shaft (R_{IN}) and revolutions at the output shaft (R_{OUT}).
Feed constant	Ratio between movement in position units at the drive and revolutions at the drive-out of the gear unit (R_{OUT}). Example: 1 revolution ≙ 63.15 mm or 1 revolution ≙ 360° degrees.

Tab. A.8 Acceleration factor parameter

The acceleration factor is calculated using the following formula:

$$\text{Acceleration factor} = \frac{\text{gear ratio} \ * \ \text{time factor\_a}}{\text{feed constant}}$$

Like the position and velocity factors, the acceleration factor also has to be written to the motor controller separated into numerators and denominators. It can therefore be necessary to interpolate the fraction to integers.

Example

First, the desired unit (column 1) and the desired number of decimal places (dp) have to be specified, along with the application's gear ratio and its feed constant (if applicable). The feed constant is then displayed in the desired positioning units (column 2).
Then, the desired unit of time² is converted into the motor controller's unit of time² (column 3).
In this way, all the values can be entered into the formula and the fraction can be calculated:

Process of calculating the acceleration factor					
Units of acceleration	Feed const.	Time constant	Gear	Equation	Result shortened
mm/s², 1 DP ➔ 1/10 $^{mm}/_{s^2}$ $\boxed{(^{mm}/_{10\,s^2})}$	$\mathbf{63,15}\ \frac{mm}{R}$ ⇒ $\mathbf{1}\ R_{OUT}$ = $\boxed{\mathbf{631,5}\ \frac{mm}{10}}$	$\boxed{\mathbf{1}\ \frac{1}{s^2}}$ = $\mathbf{60}\ \frac{1}{min * s}$ = $\boxed{\mathbf{60 * 256}\ \frac{\frac{1}{min}}{256 * s}}$	$\boxed{\mathbf{4/5}}$	$\dfrac{\frac{4}{5} * \dfrac{60 * 256 \frac{1}{256\ min * s}}{1 \frac{1}{s^2}}}{631,5\ \frac{mm}{10}} = \dfrac{\mathbf{122880}\ \frac{\frac{1}{min}}{256\ s}}{\mathbf{6315}\ \frac{mm}{10 s^2}}$	num: 8192 div: 421

Fig. A.5 Process of calculating the acceleration factor

Examples of calculating the acceleration factor					
Acceleration units 1)	Feed const. 2)	Time constant 3)	Gear 4)	Equation 5)	Result shortened
R/min, 0 DP ➔ R/min s	**1** R_{OUT} = **1** R_{OUT}	$\mathbf{1}\ \frac{1}{min * s} =$ $\mathbf{256}\ \frac{\frac{1}{min}}{256 * s}$	**1/1**	$\frac{\frac{1}{1} * \frac{256 \frac{1}{256\ min\ s}}{1 \frac{1}{min * s}}}{\frac{1}{1}} = \frac{\mathbf{256}\ \frac{\frac{1}{min}}{256 * s}}{\mathbf{1}\ \frac{\frac{1}{min}}{s}}$	num: 256 div: 1
°/s², 1 DP ➔ 1/10 °/s² (°/10 s²)	**1** R_{OUT} = **3600** $\frac{°}{10}$	$\mathbf{1}\ \frac{1}{s^2} =$ $\mathbf{60}\ \frac{1}{min * s} =$ $\mathbf{60 * 256}\ \frac{\frac{1}{min}}{256 * s}$	**1/1**	$\frac{\frac{1}{1} * \frac{60 * 256 \frac{1}{256\ min * s}}{1 \frac{1}{s^2}}}{\frac{3600 \frac{°}{10}}{1}} = \frac{\mathbf{15360}\ \frac{\frac{1}{min}}{256 * s}}{\mathbf{3600}\ \frac{°}{10\ s^2}}$	num: 64 div: 15
R/min², 2 DP ➔ 1/100 R/min² (R/100 min²)	**1** R_{OUT} = **100** $\frac{R}{100}$	$\mathbf{1}\ \frac{1}{min^2} =$ $\frac{\mathbf{1}}{\mathbf{60}}\ \frac{\frac{1}{min}}{s} =$ $\frac{\mathbf{256}}{\mathbf{60}}\ \frac{\frac{1}{min}}{256 * s}$	**2/3**	$\frac{\frac{2}{3} * \frac{256 \frac{1}{256\ min * s}}{60 \frac{1}{min^2}}}{\frac{100 \frac{1}{100}}{1}} = \frac{\mathbf{512}\ \frac{\frac{1}{min}}{256\ s}}{\mathbf{18000}\ \frac{1}{100\ min^2}}$	num: 32 div: 1125
mm/s², 1 DP ➔ 1/10 mm/s² (mm/10 s²)	**63, 15** $\frac{mm}{R}$ ⇒ **1** R_{OUT} = **631, 5** $\frac{mm}{10}$	$\mathbf{1}\ \frac{1}{s^2} =$ $\mathbf{60}\ \frac{1}{min * s} =$ $\mathbf{60 * 256}\ \frac{\frac{1}{min}}{256 * s}$	**4/5**	$\frac{\frac{4}{5} * \frac{60 * 256 \frac{1}{256\ min * s}}{1 \frac{1}{s^2}}}{\frac{631{,}5 \frac{mm}{10}}{1}} = \frac{\mathbf{122880}\ \frac{\frac{1}{min}}{256\ s}}{\mathbf{6315}\ \frac{mm}{10\ s^2}}$	num: 8192 div: 421

1) Desired unit at the drive-out

2) Positioning units per revolution at the drive-out (R_{OUT}). Feed constant of the drive (PNU 1003) * 10^{-DP} (points after the decimal)

3) Time factor_v: desired time unit per internal time unit

4) Gear factor: R_{IN} per R_{OUT}

5) Insert values into equation.

Tab. A.9 Examples of calculating the acceleration factor

B Reference parameter

B.1 FHPP general parameter structure

A controller contains a parameter set with the following structure for each axis.

Group	Indices	Description
Administrative and configuration data	1 … 99	Special objects, e.g. for FHPP+
Device Data	100 … 199	Device identification and device-specific settings, version numbers, etc.
Diagnostics	200 … 299	Diagnostic events and diagnostic memory. fault numbers, fault time, incoming/outgoing event.
Process Data	300 … 399	Current nominal and actual values, local I/Os, status data, etc.
Record list	400 … 499	A record contains all the setpoint value parameters required for a positioning procedure.
Project data	500 … 599	Basic project settings. Maximum velocity and acceleration, offset project zero point, etc. These parameters are the basis for the record list.
Function data	700 … 799	Parameters for special functions, e.g. for the camming function.
Axis data electric drives 1	1000 … 1099	All axis-specific parameters for electric drives: gear ratio, feed constant, reference parameters …
Function parameters for digital I/Os	1200 … 1239	Specific parameters for control and evaluation of the digital I/Os.

Tab. B.1 Parameter structure

B.2 Access protection

The user can prevent the drive from being operated simultaneously by PLC and FCT. The CCON.LOCK bit (FCT access blocked) and the SCON.FCT/MMI bit (FCT control sovereignty) are used for this.

Prevent operation through FCT: CCON.LOCK
By setting the CCON.LOCK control bit, the PLC prevents the FCT from taking over control sovereignty. So if the LOCK is set, FCT cannot write parameters or control the drive, execute homing, etc.
The PLC is programmed not to issue this release until the user carries out the relevant action. This generally causes exit from automatic operation. This means that the PLC programmer can ensure that the PLC always knows when it has control over the drive.
Important: The lock is active if the CCON.LOCK has a 1-signal. It us therefore not mandatory to set it. A 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 PLC that the drive is controlled by the FCT and that the PLC no longer has any control over the drive. This bit does not need to be evaluated. A possible reaction of the PLC is transitioning to stop or manual operation.

B.3 Overview of FHPP parameters

The following overview (Tab. B.2) shows the FHPP's parameters.
The parameters are described in sections B.4.2 to B.4.22.

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

Group / name	PNU	Sub-index	Type
PNUs for the telegram entries FHPP+ ➔ section B.4.2			
FHPP Receive Telegram (FHPP telegram received by controller)	40	1 … 10	uint32
FHPP Response Telegram (FHPP telegram sent by controller)	41	1 … 10	uint32
FHPP Receive Telegram State (status of FHPP telegram received by controller)	42	1	uint32
FHPP Response Telegram State (status of FHPP telegram sent by controller)	43	1	uint32
Device Data			
Device data – standard parameter ➔ section B.4.3			
Manufacturer Hardware Version (hardware version of the manufacturer)	100	1	uint16
Manufacturer Firmware Version (Firmware version of the manufacturer)	101	1	uint16
Version FHPP (FHPP version)	102	1	uint16
Project Identifier (project identification)	113	1	uint32
Controller Serial Number (serial number of controller)	114	1	uint32

Group / name	PNU	Sub-index	Type
Device data – extended parameters → section B.4.4			
Manufacturer Device Name (Device name of the manufacturer)	120	01 … 30	uint8
User Device Name (Device name of the user)	121	01 … 32	uint8
Drive Manufacturer (manufacturer name)	122	01 … 30	uint8
HTTP Drive Catalog Address (HTTP address of manufacturer)	123	01 … 30	uint8
Festo Order Number (order number of Festo)	124	01 … 30	uint8
Device Control (Device control)	125	01	uint8
Data Memory Control (Control of data storage)	127	01 … 03, 06	uint8
Diagnostics → section B.4.5			
Diagnostic Event (diagnosis event)	200	01 … 32	uint8
Fault Number (malfunction number)	201	01 … 32	uint16
Fault Time Stamp (Time stamp error)	202	01 … 32	uint32
Fault Additional Information (Error additional information)	203	01 … 32	unt32
Diagnostics Memory Parameter (Parameter, diagnostic memory)	204	01, 02, 04	uint8
Field Bus Diagnosis (Feldbus diagnostics)	206	05	uint8
Device Warnings (Device warnings)	210	01 … 16	uint8
Warning Number (Warning number)	211	01 … 16	uint16
Warning Time Stamp (Time stamp, warning)	212	01 … 16	uint32
Warning Additional Information (Additional information for warning, error)	213	01 … 16	unt32
Warning Memory Parameter (Parameter, warning memory)	214	01, 02, 04	uint8

Group / name	PNU	Sub-index	Type
Safety State (Safety status)	280	01	uint32
FSM Status word (FSM status word)	281	01, 02	uint32
FSM IO (FSM IO)	282	01	uint32
Process data → section B.4.6			
Position Values (position values)	300	01 … 04	int32
Torque Values (Torque values)	301	01 … 03	int32
Local Digital Inputs (Local digital inputs)	303	01, 02, 04	uint8
Local Digital Outputs (Local digital outputs)	304	01, 03	uint8
Maintenance Parameter (Service parameter)	305	03	uint32
Velocity Values (velocity values)	310	01 … 03	int32
State Signal Outputs (Status of signal outputs)	311	01, 02	uint32
Flying measurement → section B.4.7			
Position Value Storage (Position value memory)	350	01, 02	int32
Record list → section B.4.8			
Record Status (Record status)	400	01 … 03	uint8
Record Control Byte 1 (Record control byte 1)	401	01 … 250	uint8
Record Control Byte 2 (Record control byte 2)	402	01 … 250	uint8
Record Setpoint Value (Positioning record setpoint value)	404	01 … 250	int32
Record Velocity (Positioning record velocity)	406	01 … 250	uint32
Record Acceleration (Positioning record acceleration)	407	01 … 250	uint32

Group / name	PNU	Sub-index	Type
Record Deceleration (Positioning record deceleration)	408	01 ... 250	uint32
Record Velocity Limit (Positioning record velocity limit)	412	01 ... 250	uint32
Record Jerkfree Filter Time (Positioning record jerk-free filter time)	413	01 ... 250	uint32
Record Following Position (Positioning record for record chaining)	416	01 ... 250	uint8
Record Torque Limitation (Positioning record torque limitation)	418	01 ... 250	uint32
Record CAM ID (positioning record cam disc number)	419	01 ... 250	uint8
Record Remaining Distance Message (Positioning record, remaining distance message)	420	01 ... 250	uint32
Record Record Control Byte 3 (Record control byte 3)	421	01 ... 250	uint8
Project Data			
Project data – General project data ➔ section B.4.9			
Project Zero Point (offset project zero point)	500	01	int32
Software End Positions (Software end positions)	501	01, 02	int32
Max. Velocity (Max. permissible velocity)	502	01	uint32
Max. acceleration (Max. permissible acceleration)	503	01	uint32
Max. jerkfree filter time (Max. jerk-free filter time)	505	01	uint32
Project data – Teach ➔ section B.4.10			
Teach Target (Teach target)	520	01	uint8
Project data – Jog mode ➔ section B.4.11			
Jog Mode Crawling Velocity – Phase 1 (Inching operation velocity slow – phase 1)	530	01	int32
Jog Mode Max. Velocity – Phase 2 (Inching operation velocity fast – phase 2)	531	01	int32
Jog Mode Acceleration (Inching operation acceleration)	532	01	uint32

Group / name	PNU	Sub-index	Type
Jog Mode Deceleration (Inching operation deceleration)	533	01	uint32
Jog Mode Slow Motion Time (Inching operation slow motion time)	534	01	uint32
Project data – Direct mode position control ➔ section B.4.12			
Direct Mode Position Base Velocity (Direct operation mode position base velocity)	540	01	int32
Direct Mode Position Acceleration (Direct operation mode position acceleration)	541	01	uint32
Direct Mode Position Deceleration (Direct operation mode position deceleration)	542	01	uint32
Direct Mode Jerkfree Filter Time (Direct operation mode position jerk-free filter time)	546	01	uint32
Project data – Direct mode torque control ➔ section B.4.13			
Direct Mode Torque Base Torque Ramp (Direct operation mode torque, base value torque ramp)	550	01	uint32
Direct Mode Torque Target Torque Window (Direct operation mode torque, target torque window)	552	01	uint16
Direct Mode Torque Time Window (Direct operation mode torque, time window)	553	01	uint16
Direct Mode Torque Velocity Limit (Direct operation mode torque, velocity limiting)	554	01	uint32
Project data – Direct mode velocity adjustment ➔ section B.4.14			
Direct Mode Velocity Base Velocity Ramp (Direct operation mode, acceleration ramp)	560	01	uint32
Direct Mode Velocity Target Window (Direct operation mode velocity, velocity target window)	561	01	uint16
Direct Mode Velocity Window Time (Direct operation mode velocity, damping time target window)	562	01	uint16
Direct Mode Velocity Threshold (Direct operation mode velocity, standstill target window)	563	01	uint16
Direct Mode Velocity Threshold Time (Direct operation mode, velocity damping time)	564	01	uint16
Direct Mode Velocity Torque Limit (Direct operation mode velocity, torque limit)	565	01	uint32

Group / name	PNU	Sub-index	Type
Project data – Direct mode general ➔ section B.4.15			
Direct Mode General Torque Limit Selector (Direct operation mode general, torque limitation selector)	580	01	int8
Direct Mode General Torque Limit (Direct operation mode general, torque limitation)	581	01	uint32
Function data			
Function data – Cam disc function ➔ section B.4.16			
CAM ID (cam disc number)	700	01	uint8
Master Start Position Direkt Mode (Master start position direct operation mode)	701	01	int32
Input Config Sync. (Input configuration for synchronisation)	710	01	uint32
Gear Sync. (Synchronisation gear ratio)	711	01, 02	uint32
Output Config Encoder Emulation (Output configuration for encoder emulation)	720	01	uint32
Function data – Position and rotor position switch ➔ section B.4.17			
Position Trigger Control (Position trigger selection)	730	01	uint32
Position Switch Low (Position switch low)	731	01 … 04	int32
Position Switch High (Position switch high)	732	01 … 04	int32
Rotor Position Switch Low (Rotor position switch low)	733	01 … 04	int32
Rotor Position Switch High (Rotor position switch high)	734	01 … 04	int32
Axis parameters electrical drives 1 – mechanical parameters			
Axis parameters electric drives 1 – mechanical parameters ➔ section B.4.18			
Polarity (reversal of direction)	1000	01	uint8
Encoder Resolution (Encoder resolution)	1001	01, 02	uint32
Gear Ratio (Gear ratio)	1002	01, 02	uint32

Group / name	PNU	Sub-index	Type
Feed Constant (Feed constant)	1003	01, 02	uint32
Position Factor (Position factor)	1004	01, 02	uint32
Axis Parameter (Axis parameter)	1005	02, 03	int32
Velocity Factor (Velocity factor)	1006	01, 02	uint32
Acceleration Factor (Acceleration factor)	1007	01, 02	uint32
Polarity Slave (Reversal of direction slave)	1008	01	uint8
Axis parameters electric drives 1 – homing parameters ➔ section B.4.19			
Offset Axis Zero Point (Offset axis zero point)	1010	01	int32
Homing Method (Reference travel method)	1011	01	int8
Homing Velocities (Reference travel velocitys)	1012	01, 02	uint32
Homing Acceleration (Reference travel acceleration)	1013	01	uint32
Homing Required (Reference travel required)	1014	01	uint8
Homing Max. Torque (Reference travel max. torque)	1015	01	uint8
Axis parameters electric drives 1 – controller parameters ➔ section B.4.20			
Halt Option Code (Halt option code)	1020	01	uint16
Position Window (Tolerance window position)	1022	01	uint32
Position window time (Adjustment time position)	1023	01	uint16
Control Parameter Set (Parameters of the controller)	1024	18 … 22, 32	uint16
Motor Data (Motor data)	1025	01, 03	uint32/ uint16
Drive Data (Drive data)	1026	01 … 04, 07	uint32

Group / name	PNU	Sub-index	Type
Axis parameters electric drives 1 – electronic rating plate ➔ section B.4.21			
Max. Current (Maximum current)	1034	01	uint16
Motor Rated Current (Motor nominal current)	1035	01	uint32
Motor Rated Torque (Motor nominal torque)	1036	01	uint32
Torque Constant (Torque constant)	1037	01	uint32
Axis parameters electric drives 1 – Standstill monitoring ➔ section B.4.22			
Position Demand Value (Setpoint position)	1040	01	int32
Position Actual Value (Current position)	1041	01	int32
Standstill Position Window (Standstill position window)	1042	01	uint32
Standstill Timeout (Standstill monitoring time)	1043	01	uint16
Axis parameters for electric drives 1 – Following error monitoring ➔ section B.4.23			
Following Error Message Window (Following error message window)	1044	01	uint32
Shutdown Following Error (Following error shutdown limit)		02	uint32
Following Error Message Delay (Following error time window for warning message)	1045	01	uint16
Axis parameters for electric drives 1 – Other parameters ➔ section B.4.24			
Torque Feed Forward Control (Torque pilot control)	1080	01	int32
Setup Velocity (Setup velocity)	1081	01	uint8
Velocity Override (Velocity override)	1082	01	uint8
Function parameters for digital I/Os ➔ section B.4.25			
Remaining Distance for Remaining Distance Message (Remaining path for remaining path message)	1230	01	uint32

Tab. B.2 Overview of FHPP parameters

B.4 Descriptions of FHPP parameters

B.4.1 Representation of the parameter entries

	1	2			
	PNU 1001	**Encoder Resolution**			
3	Subindex 01, 02	Class: Struct	Data type: uint32	all	Access: rw
4	Encoder resolution in encoder increments / motor revolutions. The calculated value is derived from the fraction “encoder-increments/motor revolution”.				
5	Subindex 01	Encoder increments			
	Fix: 0x00010000 (65536)				
5	Subindex 02	Motor Revolutions			
	Fix: 0x00000001 (1)				

1 Parameter number (PNU)
2 Name of the parameter in English
3 General information on the parameter:
– Subindices (01: no subindex, simple variable),
– Class (Var, Array, Struct),
– Data type (int8, int32, uint8, uint32, etc.),
– Applies for firmware version,
– Access (read/write authorisation, ro = read only, rw = read and write).
4 Description of the parameter
5 Name and description of subindices, if present

Fig. B.1 Representation of the parameter entries

B.4.2 PNUs for the telegram entries for FHPP+

PNU 40	FHPP Receive Telegram (FHPP telegram received by controller)			
Subindex 01 ... 10	Class: Array	Data type: uint32	all	Access: ro
This array defines the contents of the received telegrams (the output data of the controller) in the cyclic process data. The array is configured using the FHPP+ editor provided by the FCT plug-in. Gaps between 1-byte PNUs and following 16- or 32-byte PNUs as well as unused subindices are filled with position holder PNUs. Format ➔ Tab. B.5.				
Subindex 01	1st PNU			
1st transmitted PNU:	always PNU 1:01			
Subindex 02	2nd PNU			
2nd transmitted PNU:	– with FPC: Always PNU 2:01 – without FPC: Any PNU			
Subindex 03	3rd PNU			
3rd transmitted PNU:	Any PNU			
Subindex 04 ... 10	4th ... 10th PNU			
4th ... 10th transmitted PNU:	Any PNU			

Tab. B.3 PNU 40

PNU 41	FHPP Response Telegram (FHPP answer telegram)			
Subindex 01 ... 10	Class: Array	Data type: uint32	all	Access: ro
This array defines the contents of the response telegrams (the input data of the control system) in the cyclic process data; ➔ PNU 40. Format ➔ Tab. B.5.				
Subindex 01	1st PNU			
1st transmitted PNU:	Always PNU 1:1			
Subindex 02	2nd PNU			
2nd transmitted PNU:	– with FPC: Always PNU 2:1 – without FPC: Any PNU			
Subindex 03	3rd PNU			
3rd transmitted PNU:	Any PNU			
Subindex 04	4th ... 10th PNU			
4th ... 10th transmitted PNU:	Any PNU			

Tab. B.4 PNU 41

Contents of a subindex for PNU 40 and 41 (uint 32 - 4 bytes)				
Byte	0	1	2	3
Contents	Reserved (= 0)	Sub-index	Transmitted PNU (2-byte value)	

Tab. B.5 Format of the entries in PNU 40 and 41

PNU 42	Receive Telegram State (status of FHPP receive telegram)			
Subindex 01	Class: Var	Data type: uint32	all	Access: rw

Type of error in the telegram editor. Entry and the error location:

Bit	Value	Significance	
0 ... 15		Error location:	Bit-serial, one bit per telegram entry
16 ... 23		Reserved	
24	1	Type of fault:	invalid PNU (with error location in bit 0 ... 15)
25	1	Type of fault:	PNU cannot be written (with error location in bit 0 ... 15)
26	1	Type of fault:	Maximum telegram length exceeded
27	1	Type of fault:	PNU must not be mapped in a telegram
28	1	Type of fault:	Entry cannot be modified in the current status (e.g. during ongoing cyclic communication)
29	1	Type of fault:	16/32-bit entry starts with an uneven address
30 ... 31		Reserved	
Note		If the transmitted telegram is correct, all bits = 0.	

Tab. B.6 PNU 42

PNU 43	Response Telegram State (FHPP response telegram status)			
Subindex 01	Class: Var	Data type: uint32	all	Access: rw

Type of error in the telegram editor. Entry and the error location:

Bit	Value	Significance	
0 ... 15		Error location:	Bit-serial, one bit per telegram entry
16 ... 23		Reserved	
24	1	Type of fault:	invalid PNU (with error location in bit 0 ... 15)
25	1	Type of fault:	PNU not readable (with error location in bit 0 ... 15)
26	1	Type of fault:	Maximum telegram length exceeded
27	1	Type of fault:	PNU must not be mapped in a telegram
28	1	Type of fault:	Entry cannot be modified in the current status (e.g. during ongoing cyclic communication)
29	1	Type of fault:	16/32-bit entry starts with an uneven address
30 ... 31		Reserved	
Note		If the transmitted telegram is correct, all bits = 0.	

Tab. B.7 PNU 43

B.4.3 Device data – Standard parameters

PNU 100	**Manufacturer Hardware Version (hardware version of the manufacturer)**			
Subindex 01	Class: Var	Data type: uint16	all	Access: ro
Coding of the hardware version, specification in BCD: xxyy (xx = main version, yy = secondary version)				

Tab. B.8 PNU 100

PNU 101	**Manufacturer Firmware Version (Firmware design of the manufacturer)**			
Subindex 01	Class: Var	Data type: uint16	all	Access: ro
Coding of the firmware design, specification in BCD: xxyy (xx = main version, yy = secondary version)				

Tab. B.9 PNU 101

PNU 102	**Version FHPP**			
Subindex 01	Class: Var	Data type: uint16	all	Access: ro
Version number of the FHPP, specification in BCD: xxyy (xx = main version, yy = secondary version)				

Tab. B.10 PNU 102

PNU 113	**Project identifier**			
Subindex 01	Class: Var	Data type: uint32	all	Access: rw
32 bit value that can be used together with the FCT plug-in to identify the project. Range of values: 0x00000001 … 0xFFFFFFFF (1 … 2^{32}-1)				

Tab. B.11 PNU 113

PNU 114	**Controller Serial Number**			
Subindex 01	Class: Var	Data type: uint32	all	Access: ro
Serial number for uniquely identifying the controller.				

Tab. B.12 PNU 114

B.4.4 Device data – Extended parameters

PNU 120	**Manufacturer Device Name (Device name of the manufacturer)**			
Subindex 01 … 30	Class: Var	Data type: uint8	all	Access: ro
Designation of the drive or controller (ASCII, 7 bit). Unused characters are filled with zero (00h=’\0’). Example: “CMMP-AS”				

Tab. B.13 PNU 120

PNU 121	**User Device Name (Device name of the user)**			
Subindex 01 … 32	Class: Var	Data type: uint8	all	Access: rw
User's designation of the controller (ASCII, 7 bit). Unused characters are filled with zero (00h='\0').				

Tab. B.14 PNU 121

PNU 122	**Drive manufacturer (manufacturer name)**			
Subindex 01 … 30	Class: Var	Data type: uint8	all	Access: ro
Name of the drive manufacturer (ASCII, 7-bit). Fix: "Festo AG & Co. KG"				

Tab. B.15 PNU 122

PNU 123	**HTTP Drive Catalog Address (HTTP address of manufacturer)**			
Subindex 01 … 30	Class: Var	Data type: uint8	all	Access: ro
Manufacturer's Internet address (ASCII, 7-bit) Fix: "www.festo.com"				

Tab. B.16 PNU 123

PNU 124	**Festo Order Number**			
Subindex 01 … 30	Class: Var	Data type: uint8	all	Access: ro
Festo order number / order code (ASCII, 7-bit).				

Tab. B.17 PNU 124

PNU 125	**Device Control**			
Subindex 01	Class: Var	Data type: uint8	all	Access: rw

Specifies which interface currently has higher-order control over the drive, in other words, which interface can be used to enable and start or stop (control) the drive:

- Fieldbus (e.g. Profibus, CanOpen, Devicenet, …)
- DIN: Digital I/O interface (e.g. multi-pin, I/O interface)
- Parameterisation interface USB/EtherNet (FCT)

The last two interfaces are treated as equals.
The output stage enable (DIN4) and controller enable (DIN5) also have to be set in addition to the respective interface (AND logic operation).

Value	Significance	SCON.FCT/MMI
0x00 (0)	Software has higher-order control (+ DIN)	1
0x01 (1)	Fieldbus has higher-order control (+ DIN) (presetting after power on)	0
0x02 (2)	Only DIN has higher-order control	1

Tab. B.18 PNU 125

PNU 127	**Data Memory Control**			
Subindex 01 … 06	Class: Struct	Data type: uint8	all.1.0	Access: wo
Commands for non-volatile memory (EEPROM, encoder).				
Subindex 01	Delete EEPROM			
Once the object has been written, and after switching power off/on, the data in the EEPROM is reset to the factory settings.				
Value	Significance			
0x10 (16)	Delete data in EEPROM and restore factory settings.			
Note	All user-specific settings will be lost on deletion (factory settings). • After deleting, always carry out the steps for commissioning the device.			
Subindex 02	Save data			
By writing the object, the data in EEPROM will be overwritten with the current user-specific settings.				
Value	Significance			
0x01 (1)	Save user-specific data in EEPROM			
Subindex 03	Reset device			
By writing the object, the data are read from the EEPROM and adopted as the current settings (EEP-ROM is not deleted or cleared; it is in the same status as after switching off and on).				
Value	Significance			
0x10 (16)	Reset device			
0x20 (32)	Auto reset upon incorrect bus cycle (deviating from the configured bus cycle time)			
Subindex 06	Encoder Data Memory Control			
Note: Writing possible only in the status “Drive blocked, controller not active” (SCON.ENABLED = 0)				
Value	Significance			
0x00 (0)	No action (e.g. for test purposes)			
0x01 (1)	Loading of the parameters from the encoder			
0x02 (2)	Saving of the parameters in the encoder without zero offset			
0x03 (3)	Saving of the parameters in the encoder with zero offset			

Tab. B.19 PNU 127

B.4.5 Diagnostics

For a description of how the diagnostic memory functions → section 11.2.

PNU 200	**Diagnostic Event**			
Subindex 01 … 32	Class: Array	Data type: uint8	all	Access: ro
Type of malfunction or diagnostic information saved in the diagnostic memory. Displays whether an incoming or outgoing malfunction is saved.				
Value	Significance			
0x00 (0)	No malfunction (or fault message deleted)			
0x01 (1)	Incoming malfunction			
0x02 (2)	Reserved (outgoing malfunction)			
0x03 (3)	Reserved			
0x04 (4)	Reserved (overrun time stamp)			
Subindex 01	Event 1			
Type of latest / current diagnostic message				
Subindex 02	Event 2			
Type of second saved diagnostic message				
Subindex 03 … 32	Event 03 … 32 (Event 03 … 32)			
Type of 3rd … 32nd saved diagnostic message				

Tab. B.20 PNU 200

PNU 201	**Fault Number (malfunction number)**			
Subindex 01 … 32	Class: Array	Data type: uint16	all	Access: ro
Fault number saved in the diagnostic memory, serves for identifying the fault. Error number, e.g. 402 for main index 40, subindex 2 → section D.				
Subindex 01	Event 1			
Latest / current diagnostic message				
Subindex 02	Event 2			
2nd saved diagnostic message				
Subindex 03 … 32	Event 03 … 32 (Event 03 … 32)			
3rd … 32nd saved diagnostic message				

Tab. B.21 PNU 201

PNU 202	**Fault Time Stamp (error time stamp)**			
Subindex 01 ... 32	Class: Array	Data type: uint32	all	Access: ro
Time of the diagnostic event in seconds after switch-on. In case of overflow, the time stamp jumps from 0xFFFFFFFF to 0.				
Subindex 01	Event 1			
Time of the latest / current diagnostic message				
Subindex 02	Event 2			
Time of the 2nd saved diagnostic message				
Subindex 03 ... 32	Event 03 ... 32 (Event 03 ... 32)			
Time of 3rd ... 32nd saved diagnostic message				

Tab. B.22 PNU 202

PNU 203	**Fault Additional Information (additional information for error)**			
Subindex 01 ... 32	Class: Array	Data type: uint32	all	Access: ro
Additional information for service staff.				
Subindex 01	Event 1			
Additional information for the latest/current diagnostic message				
Subindex 02	Event 2			
Additional information for the 2nd saved diagnostic message				
Subindex 03 ... 32	Event 03 ... 32 (Event 03 ... 32)			
Additional information for the 3rd ... 32nd saved diagnostic message				

Tab. B.23 PNU 203

PNU 204	**Diagnostics Memory Parameter**			
Subindex 01, 02, 04	Class: Struct	Data type: uint8	all	Access: ro
Configuration of the diagnostic memory.				
Subindex 01	Fault type			
Incoming and outgoing faults.				
Value	Significance			
Fix 0x02 (2)	Record only incoming malfunctions			
Subindex 02	Resolution			
Resolution time stamp				
Value	Significance			
Fix 0x03 (3)	1 second			
Subindex 04	Number of entries			
Read out the number of valid entries in the diagnostic memory				
Value	Significance			
0 ... 32	Number			

Tab. B.24 PNU 204

PNU 206	**Fieldbus Diagnosis**			
Subindex 05	Class: Var	Data type: uint8	all	Access: ro
Readout of fieldbus diagnostic data.				
Subindex 05	CANopen diagnosis			
Selected profile (protocol type):				
Value	Significance			
0	DS 402 (not available via FHPP)			
1	FHPP			

Tab. B.25 PNU 206

PNU 210	**Device warnings**			
Subindex 01 ... 16	Class: Array	Data type: uint8	all	Access: ro
Type of warning or diagnostic information saved in the warning memory. Indication of whether an incoming or outgoing warning was saved.				
Value	Significance			
0x00 (0)	No warning (or warning message deleted)			
0x01 (1)	Incoming warning			
0x02 (2)	Reserved (outgoing warning)			
0x03 (3)	Power Down (with valid time stamp)			
0x04 (4)	Reserved (overrun time stamp)			
Subindex 01	Event 1			
Type of latest / current warning message				
Subindex 02	Event 2			
Type of second saved warning message				
Subindex 03 ... 16	Event 03 ... 16 (Event 03 ... 16)			
Type of 3rd ... 16th saved warning message				

Tab. B.26 PNU 210

PNU 211	**Warning number**			
Subindex 01 ... 16	Class: Array	Data type: uint16	all	Access: ro
Warning number saved in the warning memory (e.g. 190 for main index 19, subindex 0), used to identify the warning, ➔ section 11.2 and D.				
Subindex 01	Event 1			
Most recent/current warning message				
Subindex 02	Event 2			
2nd saved warning message				
Subindex 03 ... 16	Event 03 ... 16 (Event 03 ... 16)			
3rd ... 16th saved warning message				

Tab. B.27 PNU 211

PNU 212	**Time Stamp**			
Subindex 01 … 16	Class: Array	Data type: uint32	all	Access: ro
Time of the warning event in seconds after switch-on. In case of overflow, the time stamp jumps from 0xFFFFFFFF to 0.				
Subindex 01	Event 1			
Time of the latest / current warning message				
Subindex 02	Event 2			
Time of the 2nd saved warning message				
Subindex 03 … 16	Event 03 … 16 (Event 03 … 16)			
Time of 3rd … 16th saved warning message				

Tab. B.28 PNU 212

PNU 213	**Warning Additional Information (additional information for warning)**			
Subindex 01 … 16	Class: Array	Data type: uint32	all	Access: ro
Additional information for service staff.				
Subindex 01	Event 1			
Time of the latest / current diagnostic message				
Subindex 02	Event 2			
Time of the 2nd saved diagnostic message				
Subindex 03 … 16	Event 03 … 16 (Event 03 … 16)			
Time of 3rd … 16th saved diagnostic message				

Tab. B.29 PNU 213

PNU 214	**Warning memory parameter**			
Subindex 01, 02, 04	Class: Struct	Data type: uint8	all	Access: ro
Configuration of the warning memory.				
Subindex 01	Warning type			
Incoming and outgoing warnings.				
Value	Significance			
Fix 0x02 (2)	Record only incoming warnings			
Subindex 02	Resolution			
Resolution time stamp				
Value	Significance			
Fix 0x03 (3)	1 second			
Subindex 04	Number of entries			
Read number of valid entries in the warning memory				
Value	Significance			
0 ... 16	Number			

Tab. B.30 PNU 214

PNU 280	**Safety State (Safety status)**			
Subindex 01	Class: Var	Data type: uint32	from FW 4.0.1501.2.1	Access: ro
Status word of the safety function.				

Bit	Name	Value	Meaning
0 … 7	–	0x0000 00FF	Reserved.
8	VOUT_PS_EN	0x0000 0100	Output stage enabling possible.
			CAMC-G-S3: VOUT_PS_EN = NOT (VOUT_SFR).
			CAMC-G-S1: None of the inputs STO-A or STO-B were switched.
9	VOUT_WARN	0x0000 0200	Warning. There is at least one error, whose error response is parameterised as “Warning”.
			CAMC-G-S3: VOUT_WARN (VOUT41).
			CAMC-G-S1: Reserved.
10	VOUT_SCV	0x0000 0400	At least one safety condition was violated.
			CAMC-G-S3: VOUT_SCV (VOUT 42).
			CAMC-G-S1: Reserved.
11	VOUT_ERROR	0x0000 0800	Internal error (common error message) of the safety module.
			CAMC-G-S3: VOUT_ERROR (VOUT 43).
			CAMC-G-S1: Discrepancy time violated.
12	VOUT_SSR	0x0000 1000	Safety state reached (common message).
			CAMC-G-S3: VOUT_SSR (VOUT 44) The bit is set when, in the safety module, the safe state has been reached for all the requested safety functions.
			CAMC-G-S1: STO active.

PNU 280	**Safety State (Safety status)**			
Subindex 01	Class: Var	Data type: uint32	from FW 4.0.1501.2.1	Access: ro

13	VOUT_SFR	0x0000 2000	Safety function requested.
			CAMC-G-S3: VOUT_SFR (VOUT 45): The bit is set when at least one safety function is requested in the safety module. The bit remains set until all the requests have been reset.
			CAMC-G-S1: At least one of the inputs STO-A or STO-B was switched.
14	VOUT_SERVICE	0x0000 4000	Service message.
			CAMC-G-S3: Status is assumed,... ...after a module replacement, ...in delivery status, ...during a parameterisation session.
			CAMC-G-S1: Reserved.
15	VOUT_READY	0x0000 8000	Ready. Normal status, no safety function requested.
			CAMC-G-S3: VOUT_READY= NOT(VOUT_SFR).
			CAMC-G-S1: No STO requested.
16 ... 31	–	0xFFFF 0000	Reserved.

Tab. B.31 PNU 280

PNU 281	**FSM Status word (FSM status word)**			
Subindex 01 ... 02	Class: Array	Data type: uint32	From FW 4.0.1501.2.1	Access: ro
CAMC-G-S3: Content of the status word VOUT (0 ... 63).				
Subindex 01	Lower Bytes (lower bytes)			
Bits 0 ... 31 = VOUT_0 ... 31 of the safety module CAMC-G-S3.				
Subindex 02	Upper Bytes (upper bytes)			
Bits 0 ... 31 = VOUT_32 ... 63 of the safety module CAMC-G-S3.				

Tab. B.32 PNU 281

PNU 282	**FSM IO (FSM IO)**			
Subindex 01	Class: Var	Data type: uint32	From FW 4.0.1501.2.1	Access: ro

CAMC-G-S3: Level at the inputs of the safety module.

Bit	Signal	Significance
0	LOUT48	Logical status DIN40A/B
1	LOUT49	Logical status DIN41A/B
2	LOUT50	Logical status DIN42A/B
3	LOUT51	Logical status DIN43A/B
4	LOUT52	Logical status DIN44
5	LOUT53	Logical status DIN45; mode selector switch (1 of 3)
6	LOUT54	Logical status DIN46; mode selector switch (1 of 3)
7	LOUT55	Logical status DIN47; mode selector switch (1 of 3)
8	LOUT56	Logical status, error acknowledgment via DIN48
9	LOUT57	Logical status, restart via DIN49
10	LOUT58	Logical status, two-handed control device (pair of 2 x DIN4x)
11	LOUT59	Feedback, holding brake
12 . 15	LOUT60 … 63	Unused
16	LOUT64	Logical status of the output DOUT40
17	LOUT65	Logical status of the output DOUT41
18	LOUT66	Logical status of the output DOUT42
19	LOUT67	Logical status of the signal relay
20	LOUT68	Logical status of the brake control
21	LOUT69	Logical status of the SS1 control signal
22 … 31	LOUT70 … 79	Not assigned.

Tab. B.33 PNU 282

B.4.6 Process Data

PNU 300	**Position Values**			
Subindex 01 … 04	Class: Struct	Data type: int32	all	Access: ro
Current values of the position controller in the positioning unit (➔ PNU 1004).				
Subindex 01	Actual position			
Current actual position of the controller				
Subindex 02	Nominal Position (setpoint position)			
Current setpoint position of the controller.				
Subindex 03	Actual Deviation (divergence)			
Current deviation.				
Subindex 04	Nominal Position Virtual Master (setpoint position of virtual master)			
Current setpoint position of the virtual master.				

Tab. B.34 PNU 300

PNU 301	**Torque values**			
Subindex 01 …	Class: Struct	Data type: int32	all	Access: ro
Current values of the torque controller in mNm.				
Subindex 01	Actual Force			
Current actual value of the controller.				
Subindex 02	Nominal Force (setpoint force)			
Current nominal value of the controller.				
Subindex 03	Actual Deviation (divergence)			
Current deviation.				

Tab. B.35 PNU 301

PNU 303	**Local digital inputs**							
Subindex 01, 02, 04	Class: Struct		Data type: uint8		all		Access: ro	
The controller's local digital inputs								
Subindex 01	Input DIN 0 ... 7 (inputs DIN 0 ... 7)							
Digital inputs: standard DIN (DIN 0 ... DIN 7)								
Allocation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	DIN 7 right limit switch	DIN 6 left limit switch	DIN 5 con-troller enable	DIN 4 output stage enable	DIN 3	DIN 2	DIN 1	DIN 0
Subindex 02	Input DIN 8 ... 13 (inputs DIN 8 ... 13)							
Digital inputs: standard DIN (DIN 8 ... DIN 13)								
Allocation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Reserved (= 0)		DIN A13	DIN A12	DIN 11	DIN 10	DIN 9	DIN 8
Subindex 04	Input CAMC DIN 0 ... 7 (inputs CAMC DIN 0 ... 7)							
Digital inputs: CAMC-D-8E8A (DIN 0 ... DIN 7)								
Allocation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	DIN 7	DIN 6	DIN 5	DIN 4	DIN 3	DIN 2	DIN 1	DIN 0

Tab. B.36 PNU 303

PNU 304	**Local digital outputs**							
Subindex 01, 03	Class: Struct		Data type: uint8		all			Access: rw
The controller's local digital outputs.								
Subindex 01	Output DOUT 0 ... 3 (outputs DOUT 0 ... 3)							
Digital outputs: standard DOUT (DOUT 0 ... DOUT 3)								
Allocation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Reserved (= 0)		DOUT: READY LED	DOUT: CAN LED	DOUT 3	DOUT 2	DOUT 1	DOUT 0 Con-troller ready for operation
Subindex 03	Output CAMC DOUT 0 ... 7 (outputs CAMC DOUT 0 ... 7)							
Digital outputs: CAMC-D-8E8A (DOUT 0 ... DOUT 7)								
Allocation	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	DOUT 7	DOUT 6	DOUT5	DOUT 4	DOUT 3	DOUT 2	DOUT 1	DOUT 0

Tab. B.37 PNU 304

PNU 305	**Maintenance Parameter (Service parameter)**			
Subindex 03	Class: Var	Data type: uint32	all	Access: ro
Information about the controller's or the driver's running performance.				
Subindex 03	Operating Hours			
Operating hour counter in s.				

Tab. B.38 PNU 305

PNU 310	**Velocity values**			
Subindex 01 ... 03	Class: Struct	Data type: int32	all	Access: ro
Current values of the velocity regulator.				
Subindex 01	Actual Revolutions (actual velocity)			
Current actual value of the controller.				
Subindex 02	Nominal Revolutions (setpoint velocity)			
Current setpoint value of the controller.				
Subindex 03	Actual Deviation (divergence)			
Velocity deviation.				

Tab. B.39 PNU 310

PNU 311	**State Signal Outputs (status of signal outputs)**			
Subindex 01, 02	Class: Struct	Data type: uint32	all	Access: ro
Parameters for displaying the statuses of the signal outputs				

Subindex 01	Outputs Part 1
Status of the message outputs part 1	

Bit	Value	Significance
0		Reserved (0)
1	0x0000 0002	I^2t motor monitoring active
2	0x0000 0004	Declared velocity reached
3	0x0000 0008	Position Xsetpoint = Xdest
4	0x0000 0010	Position Xact = Xdest
5	0x0000 0020	Remaining Distance
6	0x0000 0040	Homing Active
7	0x0000 0080	Homing Position Valid
8	0x0000 0100	Undervoltage in intermediate circuit
9	0x0000 0200	Following error
10	0x0000 0400	Output Stage Active
11	0x0000 0800	Holding Brake Unlocked
12	0x0000 1000	Linear Motor Identified
13	0x0000 2000	Negative Setpoint Lock Active
14	0x0000 4000	Positive Setpoint Lock Active
15	0x0000 8000	Alternative Target Reached
16	0x0001 0000	Velocity 0
17	0x0002 0000	Declared Torque Reached
18		Reserved (0)
19	0x0008 0000	Cam Disc active
20	0x0010 0000	CAM-IN active
21	0x0020 0000	CAM-CHANGE Active
22	0x0040 0000	CAM-OUT Active
23	0x0080 0000	CAM active without CAM-IN / CAM-CHANGE / CAM-OUT
24	0x0100 0000	Teach Acknowledge (low active)
25	0x0200 0000	Saving process in operation (SAVE!, Save positions)
26	0x0400 0000	FHPP MC (Motion Complete)
27	0x0800 0000	Safe Halt Active
28	0x1000 0000	Safety function: STO active
29	0x2000 0000	Safety function: STO requested
30 ... 31		Reserved (0)

PNU 311	**State Signal Outputs (status of signal outputs)**	
Subindex 02	Outputs Part 2	
Status of the message outputs part 2		
Bit	Value	Significance
0	0x0000 0001	Cam Controller 1
1	0x0000 0002	Cam Controller 2
2	0x0000 0004	Cam Controller 3
3	0x0000 0008	Cam Controller 4
4 … 7		Reserved
8	0x0000 0100	Position Switch 1
9	0x0000 0200	Position Switch 2
10	0x0000 0400	Position Switch 3
11	0x0000 0800	Position Switch 4
12 … 15		Reserved
16	0x0001 0000	Rotor Position Switch 1
17	0x0002 0000	Rotor Position Switch 2
18	0x0004 0000	Rotor Position Switch 3
19	0x0008 0000	Rotor Position Switch 4
20 … 31		Reserved

Tab. B.40 PNU 311

B.4.7 Flying measurement

Flying measurement ➔ section 10.9.

PNU 350	Position Value Storage (position value memory)			
Subindex 01, 02	Class: Array	Data type: int32	all	Access: ro
Sampled positions.				
Subindex 01	Sample Value Rising Edge			
Last sampled position in position units (➔ PNU 1004) with a rising edge.				
Subindex 02	Sample Value Falling Edge			
Last sampled position in position units (➔ PNU 1004) with a falling edge.				

Tab. B.41 PNU 350

B.4.8 Record list

With FHPP, record selection for reading and writing is done via the subindex of the PNUs 401 ... 421. The active record for positioning or teaching is selected via PNU 400.

PNU	Designation	Data type	Sub-index
401	**RCB1 (record control byte 1)**	uint8	1 ... 250
402	**RCB2 (record control byte 2)**	uint8	1 ... 250
404	**Setpoint value**	int32	1 ... 250
406	**Velocity**	uint32	1 ... 250
407	**Acceleration approach**	uint32	1 ... 250
408	**Deceleration**	uint32	1 ... 250
412	**Velocity limit**	uint32	1 ... 250
413	**Jerk-free filter time**	uint32	1 ... 250
416	**Following position**	uint8	1 ... 250
418	**Torque Limitation**	uint32	1 ... 250
419	**Cam disc number**	uint8	1 ... 250
420	**Remaining Distance Message**	int32	1 ... 250
421	**RCB3 (record control byte 3)**	uint8	1 ... 250

Tab. B.42 Structure of FHPP record list

PNU 400	**Record status**			
Subindex 01 … 03	Class: Struct	Data type: uint8	all	Access: rw/ro

Subindex 01	Demand Record Number (setpoint record number)	Access: rw
Setpoint record number. The value can be changed using FHPP. In Record Selection mode, the setpoint record number is always copied from the master's output data with a rising edge at START. Value range: 0x00 … 0xFA (0 … 250)		

Subindex 02	Actual Record Number (current record number)	Access: ro
Current record number		

Subindex 03	Record Status Byte	Access: ro
The record status byte (RSB) includes a feedback code that is transferred to the input data. When a positioning job starts, the RSB is reset.		
Note	this byte is not the same as SDIR, there is only a feedback signal for dynamic states and not absolute/relative, for example. This makes it possible to provide feedback about record chaining, for example.	

Bit	Value	Significance
0 **RC1**	0	A step criterion was not configured/achieved.
	1	The first step criterion was achieved.
1 **RCC**		Valid, as soon as MC present.
	0	Record sequencing aborted. At least one step criterion was not achieved.
	1	Record chain was processed up to the end.
2 … 7		Reserved.

Tab. B.43 PNU 400

PNU 401	**Record Control Byte 1**			
Subindex 01 … 250	Class: Array	Data type: uint8	all	Access: rw
The record control byte 1 (RCB1) controls the most important settings for the positioning task in record selection. The record control byte is bit-orientated. Allocation → Tab. B.45				

Subindex 01	Record 1 (positioning record 1)
Record control byte 1 positioning record 1.	

Subindex 02	Record 2 (positioning record 2)
Record control byte 1 positioning record 2.	

Subindex 03 … 250	Record 3 … 250 (positioning record 3 … 250)
Record control byte 1 positioning record 3 … 250.	

Tab. B.44 PNU 401

Record control byte 1					
Bit	**EN**	**Description**			
B0 **ABS**	**Abs**olute / Relative	= 1: Nominal value is relative to last nominal value.			
		= 0: Nominal value is absolute.			
		Several modes are not available via FHPP, e.g. relative to the actual value, analogue input, ...			
B1 **COM1**	**Co**ntrol **M**ode	**No.**	**Bit 2**	**Bit 1**	**Control mode**
		0	0	0	Position control.
B2 **COM2**		1	0	1	Power mode (torque, current).
		2	1	0	Velocity control (rotational velocity).
		3	1	1	reserved.
		Only Position Code mode is permissible for the camming function.			
B3 **FNUM1**	**F**unction **Num**ber	Without camming function (CDIR.FUNC = 0): No function, = 0!			
B4 **FNUM2**		With camming function (CDIR.FUNC = 1):			
		No.	**Bit 4**	**Bit 3**	**Function number**
		0	0	0	reserved.
		1	0	1	Synchronisation on external input.
		2	1	0	Synchronisation on external input with cam disc function
		3	1	1	Synchronisation on virtual master with cam disc function.
B5 **FGRP1**	**F**unction **Gr**oup	Without camming function (CDIR.FUNC = 0): No function = 0!			
B6 **FGRP2**		With camming function (CDIR.FUNC = 1):			
		No.	**Bit 6**	**Bit 5**	**Function group**
		0	0	0	Synchronisation with/without cam disc.
		All other values (no. 1 ... 3) are reserved.			
B7 **FUNC**	**Func**tion	= 1: Execute cam disc function, bit 3 ... 6 = function number and group.			
		= 0: Normal task.			

Tab. B.45 RCB1 allocation

PNU 402	**Record Control Byte 2**			
Subindex 01 ... 250	Class: Array	Data type: uint8	all	Access: rw

Record control byte 2 (RCB2) controls conditional record chaining.
If a condition was defined, it is possible to prohibit automatic continuation by setting the B7 bit. This function is intended for debugging and not for normal control purposes.

Bit	Value	Significance
0 ... 6	0 ... 128	Step enabling condition as a list, ➔ section 10.6.3, Tab. 10.12.
7	0	Record continuation (bit 0 .... 6) is not blocked
	1	Record continuation blocked

Subindex 01	Record 1
Record control byte 2 positioning record 1.	
Subindex 02	Record 2
Record control byte 2 positioning record 2.	
Subindex 03 ... 250	Record 3 ... 250 (record 3 ... 250)
Record control byte 2 positioning record 3 ... 250.	

Tab. B.46 PNU 402

PNU 404	**Record setpoint value**			
Subindex 01 ... 250	Class: Array	Data type: int32	all	Access: rw

Target position of the positioning record table. Position nominal value correspond to PNU 401 / RCB1 absolute or relative in positioning unit (➔ PNU 1004).

Subindex 01	Record 1 (positioning record 1)
Nominal position value positioning record 1.	
Subindex 02	Record 2 (positioning record 2)
Nominal position value positioning record 2.	
Subindex 03 ... 250	Record 03 ... 250 (positioning record 03 ... 250)
Nominal position value positioning record 03 ... 250.	

Tab. B.47 PNU 404

Regulation	Increment		Default	Minimum		Maximum	
Position [1]	1/100 mm	0	(= 0.0 mm)	-1,000,000	(= -10.0 m)	1,000,000	(= 10.0 m)
	1/1000 inch	0	(= 0.0 inch)	-400,000	(= -400 inch)	400,000	(= 400 inch)
	1/100 °	0	(= 0.0 °)	-36,000	(= -360.0 °)	36,000	(= 360.0 °)
[1] Examples for positioning unit, see (➔ PNU 1004).							

Tab. B.48 Setpoint values for positioning units in PNU 404

PNU 406	**Record Velocity (positioning record velocity)**			
Subindex 01 ... 250	Class: Array	Data type: uint32	all	Access: rw
Nominal velocity in units of velocity (➔ PNU 1006).				
Subindex 01	Record 1 (positioning record 1)			
Nominal velocity value positioning record 1.				
Subindex 02	Record 2 (positioning record 2)			
Nominal velocity value positioning record 2				
Subindex 03 ... 250	Record 03 ... 250 (positioning record 03 ... 250)			
Nominal velocity value positioning record 03 ... 250.				

Tab. B.49 PNU 406

PNU 407	**Record Acceleration (positioning record acceleration)**			
Subindex 01 ... 250	Class: Array	Data type: uint32	all	Access: rw
Nominal acceleration value for start up in acceleration units (➔ PNU 1007).				
Subindex 01	Record 1 (positioning record 1)			
Nominal acceleration value positioning record 1				
Subindex 02	Record 2 (positioning record 2)			
Nominal acceleration value positioning record 2				
Subindex 03 ... 250	Record 03 ... 250 (positioning record 03 ... 250)			
Nominal acceleration value positioning record 03 ... 250.				

Tab. B.50 PNU 407

PNU 408	Record Deceleration (positioning record deceleration)			
Subindex 01 ... 250	Class: Array	Data type: uint32	all	Access: rw
Nominal deceleration value for braking (deceleration) in acceleration units (➔ PNU 1007).				
Subindex 01	Record 1 (positioning record 1)			
Nominal deceleration value positioning record 1				
Subindex 02	Record 2 (positioning record 2)			
Nominal deceleration value positioning record 2				
Subindex 03 ... 250	Record 03 ... 250 (positioning record 03 ... 250)			
Nominal deceleration value positioning record 03 ... 250.				

Tab. B.51 PNU 408

PNU 412	Record Velocity Limit (positioning record velocity limit)			
Subindex 01 ... 250	Class: Array	Data type: uint32	all	Access: rw
Velocity limit for power mode in units of velocity (➔ PNU 1006).				
Subindex 01	Record 1 (positioning record 1)			
Velocity limit for positioning record 1.				
Subindex 02	Record 2 (positioning record 2)			
Velocity limit for positioning record 2.				
Subindex 03 ... 250	Record 03 ... 250 (positioning record 03 ... 250)			
Velocity limit for positioning record 03 ... 250.				

Tab. B.52 PNU 412

PNU 413	**Record jerkfree filter time (positioning record jerk-free filter time)**			
Subindex 01 ... 250	Class: Array	Data type: uint32	all	Access: rw
Jerk-free filter time in ms. Specifies the filter time constant for the output filter that is used to smooth the linear movement profiles. Completely jerk-free movement is achieved if the filter time is the same as the acceleration time.				
Subindex 01	Record 1 (positioning record 1)			
Jerk-free filter time for positioning record 1.				
Subindex 02	Record 2 (positioning record 2)			
Jerk-free filter time for positioning record 2.				
Subindex 03 ... 250	Record 03 ... 250 (positioning record 03 ... 250)			
Jerk-free filter time for positioning record 03 ... 250.				

Tab. B.53 PNU 413

PNU 416	**Record Following Position (positioning record for record chaining)**			
Subindex 01 ... 250	Class: Array	Data type: uint8	all	Access: rw
Record number to which record chaining jumps when the step enabling condition is met. Range of values: 0x01 ... 0x7F (1 ... 250)				
Subindex 01	Record 1 (positioning record 1)			
Following position for positioning record 1.				
Subindex 02	Record 2 (positioning record 2)			
Following position for positioning record 2.				
Subindex 03 ... 250	Record 03 ... 250 (positioning record 03 ... 250)			
Following position for positioning record 03 ... 250.				

Tab. B.54 PNU 416

PNU 418	**Record Torque Limitation (positioning record torque limitation)**			
Subindex 01 ... 250	Class: Array	Data type: uint32	all	Access: rw
Torque/current current limitation in positioning mode in mNm.				
Subindex 01	Record 1 (positioning record 1)			
Torque limitation for positioning record 1.				
Subindex 02	Record 2 (positioning record 2)			
Torque limitation for positioning record 2.				
Subindex 03 ... 250	Record 03 ... 250 (positioning record 03 ... 250)			
Torque limitation for positioning record 03 ... 250.				

Tab. B.55 PNU 418

PNU 419	**Record CAM ID (positioning record cam disc number)**			
Subindex 01 ... 250	Class: Array	Data type: uint8	all	Access: rw
This parameter is used to select the cam disc for the relevant record. Value range: 0 ... 16 (with value 0 the cam disc from PNU 700 is used)				
Subindex 01	Record 1 (positioning record 1)			
Cam disc number for positioning record 1.				
Subindex 02	Record 2 (positioning record 2)			
Cam disc number for positioning record 2.				
Subindex 03 ... 250	Record 03 ... 250 (positioning record 03 ... 250)			
Cam disc number for positioning record 03 ... 250.				

Tab. B.56 PNU 419

PNU 420	**Record Remaining Distance Message (positioning record remaining distance message)**			
Subindex 01 ... 250	Class: Array	Data type: uint32	all	Access: rw
Remaining distance message in the record list in position units (➔ PNU 1004).				
Subindex 01	Record 1 (positioning record 1)			
Remaining distance message for positioning record 1.				
Subindex 02	Record 2 (positioning record 2)			
Remaining distance message for positioning record 2.				
Subindex 03 ... 250	Record 03 ... 250 (positioning record 03 ... 250)			
Remaining distance message for positioning record 03 ... 250.				

Tab. B.57 PNU 420

PNU 421	**Record Control Byte 3**			
Subindex 01 ... 250	Class: Array	Data type: uint8	all	Access: rw
Record control byte 3 (RCB3) controls the specific behaviour of the record when particular events occur. The record control byte is bit-orientated.				

Bit	Bit 1	Bit 0	Significance
B0, B1	0	0	Ignore
	0	1	Interrupt active
	1	0	Append to active positioning (wait)
	1	1	Reserved
B2 ... B9			Reserved (= 0!)

Subindex 01	Record 1 (positioning record 1)
Record control byte 3 positioning record 1.	
Subindex 02	Record 2 (positioning record 2)
Record control byte 3 positioning record 2.	
Subindex 03 ... 250	Record 03 ... 250 (positioning record 03 ... 250)
Record control byte 3 positioning record 03 ... 250.	

Tab. B.58 PNU 421

B.4.9 Project Data – General Project Data

PNU 500	**Project Zero Point (offset project zero point)**			
Subindex 01	Class: Var	Data type: int32	all	Access: rw
Offset of axis zero point to project zero point in positioning unit (➔ PNU 1004). Reference point for position values in the application (➔ PNU 404).				

Tab. B.59 PNU 500

PNU 501	**Software End Positions (Software end positions)**			
Subindex 01, 02	Class: Array	Data type: int32	all	Access: rw
Software end positions in positioning unit (➔ PNU 1004). A setpoint specification (position) outside the end positions is not permissible and will result in an error. The offset to the axis zero point is entered. Plausibility rule: Min-Limit ≤ Max-Limit				
Subindex 01	Lower Limit			
Lower software end position				
Subindex 02	Upper Limit			
Upper software end position.				

Tab. B.60 PNU 501

PNU 502	**Max. Velocity (Max. permissible velocity)**			
Subindex 01	Class: Var	Data type: uint32	all	Access: rw
Max. permissible velocity in units of velocity (➔ PNU 1006). This value limits the velocity in all operation modes except torque mode.				

Tab. B.61 PNU 502

PNU 503	**Max. Acceleration (max. permissible acceleration)**			
Subindex 01	Class: Var	Data type: uint32	all	Access: rw
Max. permissible acceleration in units of acceleration (➔ PNU 1007).				

Tab. B.62 PNU 503

PNU 505	**Max. Jerkfree Filter Time (max. jerk-free filter time)**			
Subindex 01	Class: Var	Data type: uint32	all	Access: rw
Max. permissible jerk-free filter time in ms. Range of values: 0x00000000 ... 0xFFFFFFFF (0 ... 4294967295)				

Tab. B.63 PNU 505

B.4.10 Project Data – Teach

PNU 520	**Teach Target**			
Subindex 01	Class: Var	Data type: uint8	all	Access: rw
The parameter defined is the one written with the actual position at the next Teach command (➔ section 10.5).				

Value		Significance
0x01	1	Nominal position in positioning record (default). – For record selection: Positioning record as per FHPP control bytes – For direct operation: positioning record corresponding to PNU 400/1
0x02	2	Axis zero point (PNU 1010)
0x03	3	Project zero point (PNU 500)
0x04	4	Lower software end position (PNU 501/01)
0x05	5	Upper software end position (PNU 501/02)

Tab. B.64 PNU 520

B.4.11 Project Data – Jog Mode

PNU 530	**Jog Mode Crawling Velocity – Phase 1 (Inching operation velocity slow – phase 1)**			
Subindex 01	Class: Var	Data type: int32	all	Access: rw
Maximum velocity for phase 1 in units of velocity (➔ PNU 1006).				

Tab. B.65 PNU 530

PNU 531	**Jog Mode Max. Velocity – Phase 2 (Inching operation velocity fast – phase 2)**			
Subindex 01	Class: Var	Data type: int32	all	Access: rw
Maximum velocity for phase 2 in units of velocity (➔ PNU 1006).				

Tab. B.66 PNU 531

PNU 532	**Jog Mode Acceleration (inching operation acceleration)**			
Subindex 01	Class: Var	Data type: uint32	all	Access: rw
Acceleration during jogging in units of acceleration (➔ PNU 1007).				

Tab. B.67 PNU 532

PNU 533	Jog Mode Deceleration (inching operation deceleration)			
Subindex 01	Class: Var	Data type: uint32	all	Access: rw
Deceleration during jogging in units of acceleration (➔ PNU 1007).				

Tab. B.68 PNU 533

PNU 534	Jog Mode Slow Motion Time (inching operation slow motion time)			
Subindex 01	Class: Var	Data type: uint32	all	Access: rw
Time duration of phase 1 (T1) in ms				

Tab. B.69 PNU 534

B.4.12 Project Data – Direct Mode Position Control

PNU 540	Direct Mode Position Base Velocity (direct operation mode position base velocity)			
Subindex 01	Class: Var	Data type: int32	all	Access: rw
Base velocity during direct mode position control in units of velocity (➔ PNU 1006).				

Tab. B.70 PNU 540

PNU 541	Direct Mode Position Acceleration			
Subindex 01	Class: Var	Data type: uint32	all	Access: rw
Acceleration during direct mode position control in units of acceleration (➔ PNU 1007).				

Tab. B.71 PNU 541

PNU 542	Direct Mode Position Deceleration			
Subindex 01	Class: Var	Data type: uint32	all	Access: rw
Deceleration during direct mode position control in units of acceleration (➔ PNU 1007).				

Tab. B.72 PNU 542

PNU 546	Direct Mode Position Jerkfree Filter Time (Direct operation mode position jerk-free filter time)			
Subindex 01	Class: Var	Data type: uint32	all	Access: rw
Jerk-free filter time during direct mode position control in ms. Range of values: 0x00000000 ... 0xFFFFFFFF (0 ... 4294967295)				

Tab. B.73 PNU 546

B.4.13 Project Data – Direct Mode, Torque Control

PNU 550	**Direct Mode Torque Base Torque Ramp (direct operation mode torque base torque ramp)**			
Subindex 01	Class: Var	Data type: uint32	all	Access: rw
Base value for torque ramp in direct mode torque control in mNm/s.				

Tab. B.74 PNU 550

PNU 552	**Direct Mode Torque Target Torque Window (direct mode torque target window)**			
Subindex 01	Class: Var	Data type: uint16	all	Access: rw
Torque in mNm, the amount by which the actual torque is permitted to differ from the setpoint torque in order to be interpreted as still being in the target window. The width of the window is twice the value transmitted, with the target torque in the centre of the window.				

Tab. B.75 PNU 552

PNU 553	**Direct Mode Torque Time Window**			
Subindex 01	Class: Var	Data type: uint16	all	Access: rw
Damping time for the torque target window during direct torque mode in ms.				

Tab. B.76 PNU 553

PNU 554	**Direct Mode Torque Velocity Limit (Direct operation mode torque velocity limiting)**			
Subindex 01	Class: Var	Data type: uint32	all	Access: rw
With active torque control, the velocity is limited to this value, stated in units of velocity (PNU 1007).				
Note	PNU 514 allows an absolute velocity limit to be specified, which triggers a malfunction if it is reached. If both functions (limitation and monitoring) are to be active at the same time, PNU 554 must be significantly less than PNU 514.			

Tab. B.77 PNU 554

B.4.14 Project Data – Direct Mode Velocity Adjustment

PNU 560	**Direct Mode Velocity Base Velocity Ramp (Direct operation mode rotational velocity acceleration ramp)**			
Subindex 01	Class: Var	Data type: uint32	all	Access: rw
Base acceleration value (velocity ramp) during direct mode velocity adjustment in units of acceleration (➔ PNU 1007).				

Tab. B.78 PNU 560

PNU 561	**Direct Mode Velocity Target Window (direct operation mode velocity, velocity target window)**			
Subindex 01	Class: Var	Data type: uint16	all	Access: rw
Velocity target window during direct mode velocity adjustment in units of velocity (➔ PNU 1006).				

Tab. B.79 PNU 561

PNU 562	**Direct Mode Velocity Window Time (direct operation mode velocity damping time target window)**			
Subindex 01	Class: Var	Data type: uint16	all	Access: rw
Damping time for velocity target window during direct mode velocity adjustment in ms.				

Tab. B.80 PNU 562

PNU 563	**Direct Mode Velocity Threshold (velocity standstill target window in direct mode)**			
Subindex 01	Class: Var	Data type: uint16	all	Access: rw
Standstill target window during direct mode velocity adjustment in units of velocity (➔ PNU 1006).				

Tab. B.81 PNU 563

PNU 564	**Direct Mode Velocity Threshold Time (direct mode velocity damping time)**			
Subindex 01	Class: Var	Data type: uint16	all	Access: rw
Damping time for standstill target window during direct mode velocity adjustment in ms.				

Tab. B.82 PNU 564

PNU 565	**Direct Mode Velocity Torque Limit (direct operation mode velocity, torque limitation)**			
Subindex 01	Class: Var	Data type: uint32	all	Access: rw
Torque limitation during direct mode velocity adjustment in mNm. PNU 565 is replaced in CMMP-AS-...-M3/-M0 by PNU 581, but remains available for compatibility reasons. Changes to PNU 565 are written directly to PNU 581.				

Tab. B.83 PNU 565

B.4.15 Project Data – Direct Mode General

PNU 580	**Direct Mode General Torque Limit Selector (Direct operation mode general, torque limitation selector)**			
Subindex 01	Class: Var	Data type: int8	all	Access: rw
Activation of torque limitation in direct mode (PNU 581).				

Value		Significance
0x00	0	Torque limitation not active.
0x04	4	Symmetric torque limitation active ➔ PNU 581.

Tab. B.84 PNU 580

PNU 581	**Direct Mode General Torque Limit (Direct operation mode general, torque limitation)**			
Subindex 01	Class: Var	Data type: uint32	all	Access: rw
Torque limiting in direct mode in mNm. The limitation applies for all jobs in direct mode: – Homing (PNU 1015 is “overwritten” through the global setting) – Jogging. – Positioning jobs. Changes to PNU 581 are also written in PNU 565 for compatibility reasons. When changing to record selection, the settings for torque limitation are activated by the selected record at the start. When switching back to direct mode, the last settings for the torque limitation are maintained, since the same selector is used in both operating modes. And so it is recommended to check the torque limitation after shifting to direct mode.				

Tab. B.85 PNU 581

B.4.16 Function Data – Cam Disc Function

Selecting cam disc

PNU 700	**CAM ID (cam disc number)**			
Subindex 01	Class: Var	Data type: uint8	all	Access: rw
This parameter is used to select the number of the cam disc directly. Range of values: 1 … 16				

Tab. B.86 PNU 700

PNU 701	**Master Start Position Direct Mode (master start position in direct mode)**			
Subindex 01	Class: Var	Data type: int32	all	Access: rw
Defines the start position of the master for the cam disc function.				

Tab. B.87 PNU 701

Synchronisation (input, X10)

PNU 710	**Input Config Sync. (input configuration for synchronisation)**			
Subindex 01	Class: Var	Data type: uint32	all	Access: rw
Configuration of the encoder input for synchronisation (physical master on X10, slave operation).				

Bit	Value	Significance
0	0	Evaluate zero pulse
	1	Ignore zero pulse
1	0	Reserved
2	0	Evaluate A/B track
	1	Switch off A/B track
3 … 31		Reserved = 0

Tab. B.88 PNU 710

PNU 711	**Gear Sync. (synchronisation gear ratio)**			
Subindex 01, 02	Class: Var	Data type: uint32	all	Access: rw
Gear ratio for synchronisation with an external input (physical master on X10, slave operation).				
Subindex 01	Motor revolutions			
Motor revolutions (drive). When reversing the direction of rotation is active, the value is negative.				
Subindex 02	Shaft revolutions (spindle rotations)			
Spindle rotations (drive-out).				

Tab. B.89 PNU 711

Encoder emulation (output, X11)

PNU 720	Output Config Encoder Emulation (output configuration for encoder emulation)			
Subindex 01	Class: Var	Data type: uint32	all	Access: rw
Configuration of the encoder for encoder emulation (virtual master).				

Bit	Value	Significance
0	0	Evaluate A/B track
	1	Switch off A/B track
1	0	Evaluate zero pulse
	1	Ignore zero pulse
2	0	Evaluate reversing of direction of rotation
	1	Ignore reversing of direction of rotation
3 ... 31		Reserved = 0

Tab. B.90 PNU 720

B.4.17 Function Data – Position and Rotor Position Switch

PNU 730	Position Trigger Control (position trigger selection)			
Subindex 01	Class: Var	Data type: uint32	all	Access: rw
Bit-by-bit activation of the corresponding triggers. Bit is set = trigger is computed, i.e. the position comparison is carried out. Triggers which are not computed save computing time.				

Value	Bit	Description
0x0000 0001	0	Position Switch (actual position) 0
0x0000 0002	1	Position Switch (actual position) 1
0x0000 0004	2	Position Switch (actual position) 2
0x0000 0005	3	Position Switch (actual position) 3
...	4 ... 15	Reserved
0x0001 0000	16	Rotor Position Switch 0
0x0002 0000	17	Rotor Position Switch 1
0x0004 0000	18	Rotor Position Switch 2
0x0008 0000	19	Rotor Position Switch 3
...	20 ... 31	Reserved

Tab. B.91 PNU 730

PNU 731	**Position Switch Low**			
Subindex 01 … 04	Class: Var	Data type: int32	all	Access: rw
Position values for the low position switch, stated in the positioning unit (➔ PNU 1004).				
Subindex 01	Position Switch 1			
Position values of the 1st low position trigger.				
Subindex 02	Position Switch 2			
Position values of the 2nd low position trigger.				
Subindex 03	Position Switch 3			
Position values of the 3rd low position trigger.				
Subindex 04	Position Switch 4			
Position values of the 4th low position trigger.				

Tab. B.92 PNU 731

PNU 732	**Position Switch High**			
Subindex 01 … 04	Class: Var	Data type: int32	all	Access: rw
Position values for the high position switch, stated in the positioning unit (➔ PNU 1004).				
Subindex 01	Position Switch 1			
Position values of the 1st high position trigger.				
Subindex 02	Position Switch 2			
Position values of the 2nd high position trigger.				
Subindex 03	Position Switch 3			
Position values of the 3rd high position trigger.				
Subindex 04	Position Switch 4			
Position values of the 4th high position trigger.				

Tab. B.93 PNU 732

PNU 733	**Rotor Position Switch Low**			
Subindex 01 … 04	Class: Var	Data type: int32	all	Access: rw
Angle for the rotor position switch low in °. Range of values: -180 … 180				
Subindex 01	Rotor Position Switch 1			
Angle of the 1st rotor position switch low.				
Subindex 02	Rotor Position Switch 2			
Angle of the 2nd rotor position switch low.				
Subindex 03	Rotor Position Switch 3			
Angle of the 3rd rotor position switch low.				
Subindex 04	Rotor Position Switch 4			
Angle of the 4th rotor position switch low.				

Tab. B.94 PNU 733

PNU 734	**Rotor Position Switch High**			
Subindex 01 … 04	Class: Var	Data type: int32	all	Access: rw
Angle for the rotor position switch high in °. Range of values: -180 … 180				
Subindex 01	Rotor Position Switch 1			
Angle of the 1st rotor position switch high.				
Subindex 02	Rotor Position Switch 2			
Angle of the 2nd rotor position switch high.				
Subindex 03	Rotor Position Switch 3			
Angle of the 3rd rotor position switch high.				
Subindex 04	Rotor Position Switch 4			
Angle of the 4th rotor position switch high.				

Tab. B.95 PNU 734

B.4.18 Axis Parameters Electrical Drives 1 – Mechanical Parameters

PNU 1000	**Polarity (reversal of direction)**			
Subindex 01	Class: Var	Data type: uint8	all	Access: rw
Direction of the position values.				

Value	Significance
0x00 (0)	Normal (default)
0x80 (128)	Inverted (multiplied by -1)

Tab. B.96 PNU 1000

PNU 1001	**Encoder Resolution**			
Subindex 01, 02	Class: Struct	Data type: uint32	all	Access: rw
Encoder resolution in encoder increments / motor revolutions. Specified internal conversion factor. The calculated value is derived from the fraction “encoder-increments/motor revolution”.				
Subindex 01	Encoder increments			
Fix: 0x00010000 (65536)				
Subindex 02	Motor Revolutions			
Fix: 0x00000001 (1)				

Tab. B.97 PNU 1001

PNU 1002	**Gear ratio**			
Subindex 01, 02	Class: Struct	Data type: uint32	all	Access: rw
Ratio of motor revolutions to gear unit spindle revolutions (drive-out revolutions) ➔ appendix A.1. Gear transmission = motor revolutions / spindle rotations				
Subindex 01	Motor Revolutions			
Gear ratio – numerator. Range of values: 0x00000000 ... 0x7FFFFFFF (0 ... +(2^{31}-1))				
Subindex 02	Shaft Revolutions (spindle rotations)			
Gear ratio – denominator. Range of values: 0x00000000 ... 0x7FFFFFFF (0 ... +(2^{31}-1))				

Tab. B.98 PNU 1002

PNU 1003	**Feed constant**			
Subindex 01, 02	Class: Struct	Data type: uint32	all	Access: rw
The feed constant specifies the lead of the drive's spindle per revolution, ➔ appendix A.1. Feed constant = feed / spindle rotation				
Subindex 01	Feed			
Feed constant – numerator. Range of values: 0x00000000 ... 0x7FFFFFFF (0 ... +(2^{31}-1))				
Subindex 02	Shaft Revolutions (spindle rotations)			
Feed constant - denominator. Range of values: 0x00000000 ... 0x7FFFFFFF (0 ... +(2^{31}-1))				

Tab. B.99 PNU 1003

PNU 1004	**Position Factor**			
Subindex 01, 02	Class: Struct	Data type: uint32	all	Access: rw
Conversion factor for all position units (converting the user units into internal controller units). Calculation ➔ appendix A.1.				
$\text{Position factor} = \frac{\text{encoder resolution * gear ratio}}{\text{feed constant}}$				
Subindex 01	Numerator			
Position factor - numerator.				
Subindex 02	Denominator			
Position factor – denominator.				

Tab. B.100 PNU 1004

PNU 1005	**Axis Parameter**			
Subindex 02, 03	Class: Struct	Data type: int32	all	Access: rw
Specify and read out axis parameters.				
Subindex 02	Gear Numerator			
Gear ratio – axis gear numerator. Range of values: 0x0 ... 0x7FFFFFFF (0 ... +(2^{31}-1))				
Subindex 03	Gear Denominator			
Gear ratio – axis gear denominator. Range of values: 0x0 ... 0x7FFFFFFF (0 ... +(2^{31}-1))				

Tab. B.101 PNU 1005

PNU 1006	**Velocity Factor (velocity factor)**			
Subindex 01, 02	Class: Struct	Data type: uint32	all	Access: rw
Conversion factor for all velocity units (converting the user units into internal controller units). Calculation ➔ appendix A.1.				
$$\text{Speed factor} = \frac{\text{encoder resolution * time factor\_v}}{\text{feed constant}}$$				
Subindex 01	Numerator			
Velocity factor – numerator.				
Subindex 02	Denominator			
Velocity factor – denominator.				

Tab. B.102 PNU 1006

PNU 1007	**Acceleration factor**			
Subindex 01, 02	Class: Struct	Data type: uint32	all	Access: rw
Conversion factor for all acceleration units. (converting the user units into internal controller units). Calculation ➔ appendix A.1.				
$$\text{Acceleration factor} = \frac{\text{encoder resolution * time factor\_a}}{\text{feed constant}}$$				
Subindex 01	Numerator			
Acceleration factor – numerator.				
Subindex 02	Denominator			
Acceleration factor – denominator.				

Tab. B.103 PNU 1007

PNU 1008	**Polarity Slave (reversal of direction for slave)**			
Subindex 01	Class: Var	Data type: uint8	all	Access: rw
This parameter can be used to reverse the position specification for signals on X10 (slave operation). This applies to the functions “Synchronisation” (including electronic gear units), “Flying saw”, “Cam discs”.				

Value	Significance
0x00	Position value vector normal (default)
0x80	Position value vector inverted

Tab. B.104 PNU 1008

B.4.19 Axis Data Electrical Drives 1 - Homing Parameters

PNU 1010	Offset Axis Zero Point			
Subindex 01	Class: Var	Data type: int32	all	Access: rw
Axis zero point offset in positioning units (➔ PNU 1004). 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 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 (PNU 500). The axis zero point (AZ) is calculated as follows: AZ = REF + offset axis zero point				

Tab. B.105 PNU 1010

PNU 1011	Homing Method			
Subindex 01	Class: Var	Data type: int8	all	Access: rw
Defines the method which the drive uses to carry out the homing ➔ section 10.3 and 10.3.2.				

Tab. B.106 PNU 1011

PNU 1012	Homing Velocities (reference travel velocitys)			
Subindex 01, 02	Class: Struct	Data type: uint32	all	Access: rw
Velocitys during homing in units of velocity (➔ PNU 1006).				
Subindex 01	Search for Switch (search velocity)			
Velocity when searching for the homing point REF or a stop or switch.				
Subindex 02	Running for Zero (travel velocity)			
Velocity of travel to the axis zero point AZ. Range of values: 0x00000000 … 0x7FFFFFFF (0 … +(2^{31}-1))				

Tab. B.107 PNU 1012

PNU 1013	Homing acceleration			
Subindex 01	Class: Var	Data type: uint32	all	Access: rw
Acceleration during the homing in units of acceleration (➔ PNU 1007). Range of values: 0x00000000 … 0x7FFFFFFF (0 … +(2^{31}-1))				

Tab. B.108 PNU 1013

PNU 1014	Homing Required (reference travel required)			
Subindex 01	Class: Var	Data type: uint8	all	Access: rw
Defines whether or not homing must be carried out after switching on in order to carry out positioning tasks.				
Note	**Drives with the multi-turn absolute displacement encoder only need one homing run after installation.**			
Value	Significance			
0x00 (0)	Reserved			
0x01 (1) (Fix)	Homing must be carried out.			

Tab. B.109 PNU 1014

PNU 1015	Homing Max. Torque (reference travel max. torque)			
Subindex 01	Class: Var	Data type: uint8	all	Access: rw
Max. torque during homing. Specified as a multiple of the nominal torque in % (➔ PNU 1036). The maximum permissible torque (via current limiting) during homing. If this value is reached, the drive identifies the stop (REF) and travels to the axis zero point.				

Tab. B.110 PNU 1015

B.4.20 Axis Parameters Electrical Drives 1 – Controller Parameters

PNU 1020	Halt Option Code			
Subindex 01	Class: Var	Data type: uint16	all	Access: rw
Reaction to a hold command (falling edge at SPOS.HALT).				
Value	Significance			
0x00 (0)	Reserved (motor off – coils without current, brake unactuated)			
0x01 (1)	Brake with hold ramp			
0x02 (2)	Reserved (brake with emergency stop ramp)			

Tab. B.111 PNU 1020

PNU 1022	Position window (tolerance window position)			
Subindex 01	Class: Var	Data type: uint32	all	Access: rw
Tolerance window in positioning units (➔ PNU 1004). Amount by which the current position may deviate from the target position, in order that it may still be regarded as being within the target window. The width of the window is twice the value transferred, with the target position in the centre of the window.				

Tab. B.112 PNU 1022

PNU 1023	**Position Window Time (adjustment time position)**			
Subindex 01	Class: Var	Data type: uint16	all	Access: rw
Readjustment time in milliseconds. If the actual position has been in the target position window this amount of time, the bit SPOS.MC is set.				

Tab. B.113 PNU 1023

PNU 1024	**Control Parameter Set (parameters of the controller)**			
Subindex 18 … 22, 32	Class: Struct	Data type: uint16	all	Access: rw
Control parameters as well as parameters for "quasi-absolute position registering".				
Subindex 18	Gain Position (position amplification)			
Gain position controller Range of values: 0x0000 … 0xFFFF (0 … 65535)				
Subindex 19	Gain Velocity (velocity amplification)			
Gain velocity controller Range of values: 0x0000 … 0xFFFF (0 … 65535)				
Subindex 20	Time Velocity (velocity time constant)			
Time constant for the velocity controller. Range of values: 0x0000 … 0xFFFF (0 … 65535)				
Subindex 21	Gain Current (current amplification)			
Gain current controller. Range of values: 0x0000 … 0xFFFF (0 … 65535)				
Subindex 22	Time Current (current time constant)			
Current regulator time constant. Range of values: 0x0000 … 0xFFFF (0 … 65535)				
Subindex 32	Save Position			
Save the current position at power-off, see → PNU 1014.				

Bit	Value	Significance
0x00F0	240	Current position will **not** be saved at power-off (default)
0x000F	15	Reserved

Tab. B.114 PNU 1024

PNU 1025	**Motor data**			
Subindex 01, 03	Class: Struct	Data type: uint32/uint16	all	Access: rw/ro
Motor-specific data.				
Subindex 01	reserved		Data type: uint32	Access: ro
Reserved (= 0)				
Subindex 03	Time Max. Current		Data type: uint16	Access: rw
I^2t-time in ms. When the I^2t time elapses, the current is limited automatically to the motor nominal current in order to protect the motor (Motor Rated Current, PNU 1035).				

Tab. B.115 PNU 1025

PNU 1026	**Drive data**			
Subindex 01 … 04, 07	Class: Struct	Data type: uint32	all	Access: rw/ro
General motor data				
Subindex 01	Power Temp. (temp. output stage)			Access: ro
Current temperature of the output stage in °C.				
Subindex 02	Power Stage Max. Temp. (max. temp. output stage)			Access: ro
Maximum temperature of the output stage in °C.				
Subindex 03	Motor Rated Current (motor nominal current)			Access: rw
Motor nominal current in mA, identical to PNU 1035.				
Subindex 04	Current Limit (max. motor current)			Access: rw
Maximum motor current, identical to PNU 1034.				
Subindex 07	Controller Serial Number			Access: ro
Controller's internal serial number.				

Tab. B.116 PNU 1026

B.4.21 Axis Parameters Electric Drives 1 – Electronic Rating Plate

PNU 1034	**Maximum current**			
Subindex 01	Class: Var	Data type: uint16	all	Access: rw
As a rule, servo motors may be overloaded for a certain time period. With PNU 1034 (identical to PNU 1026/4), the maximum permissible motor current is set. It refers to the nominal motor current (PNU 1035) and is set in thousandths. The range of values is limited upward through the maximum controller current (see technical data, dependent on the controller cycle time and the output stage cycle frequency). PNU 1034 may only be written on if PNU 1035 has already been validly written on.				
Note	Observe that the current limitation also limits the maximum possible velocity and that (higher) setpoint velocitys may therefore not be achieved.			

Tab. B.117 PNU 1034

PNU 1035	**Motor Rated Current (motor nominal current)**			
Subindex 01	Class: Var	Data type: uint32	all	Access: rw
The motor's rated current in mA, identical to PNU 1026/3.				

Tab. B.118 PNU 1035

PNU 1036	**Motor Rated Torque (motor nominal torque)**			
Subindex 01	Class: Var	Data type: uint32	all	Access: rw
The motor's rated torque in 0.001 Nm.				

Tab. B.119 PNU 1036

PNU 1037	**Torque Constant**			
Subindex 01	Class: Var	Data type: uint32	all	Access: rw
Ratio between the current and torque in the motor used in mNM/A.				

Tab. B.120 PNU 1037

B.4.22 Axis Parameters Electric Drives 1 – Standstill Monitoring

PNU 1040	**Position Demand Value**			
Subindex 01	Class: Var	Data type: int32	all	Access: ro
Nominal target position of the last positioning task in positioning units (➔ PNU 1004).				

Tab. B.121 PNU 1040

PNU 1041	Position Actual Value (current position)			
Subindex 01	Class: Var	Data type: int32	all	Access: ro
Current position of the drive in positioning units (➔ PNU 1004).				

Tab. B.122 PNU 1041

PNU 1042	Standstill Position Window			
Subindex 01	Class: Var	Data type: uint32	all	Access: rw
Standstill position window in positioning units (➔ PNU 1004). Amount of the position by which the drive may move after MC until the standstill monitoring responds.				

Tab. B.123 PNU 1042

PNU 1043	Standstill Timeout (standstill monitoring time)			
Subindex 01	Class: Var	Data type: uint16	all	Access: rw
Standstill monitoring time in ms. Time during which the drive must be outside the standstill position window before standstill monitoring responds.				

Tab. B.124 PNU 1043

B.4.23 Axis Parameters for Electric Drives 1 – Following Error Monitoring

PNU 1044	Following Error Window (contouring error window)			
Subindex 01, 02	Class: Array	Data type: uint32	all/from FW 4.0.1501.2.3	Access: rw
Subindex 01	Following Error Message Window		all	
Define or read the permissible range for reporting following errors, in positioning units.				
Subindex 02	Shutdown Following Error		From FW 4.0.1501.2.3	
Define or read the range for the shutdown limit for following errors, in positioning units. 0xFFFFFFFF = following error monitoring OFF				

Tab. B.125 PNU 1044

PNU 1045	Following Error Message Delay (following error time window for warning message)			
Subindex 01	Class: Var	Data type: uint16	all	Access: rw
Define or read a timeout time for following error monitoring in ms. Range of values: 1 ... 60000				

Tab. B.126 PNU 1045

B.4.24 Axis Parameters for Electric Drives 1 – Other Parameters

PNU 1080	Torque Feed Forward Control			
Subindex 01	Class: Var	Data type: int32	all	Access: rw
Torque pilot control in mNm (only effective for direct mode with position control).				

Tab. B.127 PNU 1080

PNU 1081	Setup Velocity (setup velocity)			
Subindex 01	Class: Var	Data type: uint8	all	Access: rw
Setup velocityy as % of whatever velocity is specified. Range of values: 0 ... 100				

Tab. B.128 PNU 1081

PNU 1082	Velocity Override (velocity override)			
Subindex 01	Class: Var	Data type: uint8	all	Access: rw
Velocity override as % of whatever velocity is specified. Range of values: 0 ... 255				

Tab. B.129 PNU 1082

B.4.25 Function Parameters for Digital I/Os

PNU 1230	Remaining Distance for Remaining Distance Message (Remaining path for remaining path message)			
Subindex 01	Class: Var	Data type: uint32	all	Access: rw
The remaining distance is the trigger condition for the remaining distance message, which can be issued on a digital output. With CMMP-AS: effective in Direct mode only.				

Tab. B.130 PNU 1230

C Festo Parameter Channel (FPC) and FHPP+

C.1 Festo parameter channel (FPC) for cyclic data (I/O data)

C.1.1 Overview of FPC

The parameter channel is used for transmitting parameters. The parameter channel is made up of the following:

Components	Description
Parameter identifier (ParID)	Component of the parameter channel which contains the Job and Response identifiers (AK) and the parameter number (PNU). The parameter number is used to identify or address the respective parameters. The Job or Response identifier (AK) describes the job or the reply in the form of an index.
Subindex (IND)	Addresses an element of an array parameter (sub-parameter number).
Parameter value (ParVal)	Value of the parameter. If a parameter processing job cannot be executed, an error number is transmitted in place of the value in the response telegram. The error number describes the cause of the error.

Tab. C.1 Components of the parameter channel (PKW)

The parameter channel consists of 8 bytes. The structure of the parameter channel dependent on the size or type of the parameter value is shown in the following table:

FPC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Output data	0	IND 1)	ParID (PKE) 2)		Value (PWE) 3)			
Input data	0	IND 1)	ParID (PKE) 2)		Value (PWE) 3)			

1) IND Subindex - for addressing an array element

2) ParID (PKE) Parameter Identifier - comprising ReqID or ResID and PNU

3) Value (PWE) Parameter value: for double word: bytes 5...8; for word: bytes 7, 8; for byte: byte 8

Tab. C.2 Structure of parameter channel

Parameter identifier (ParID)

The parameter identifier includes the job or response identifier (AK) and the parameter number (PNU).

ParID	Byte 4								Byte 3							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Job	ReqID (AK) 1)				res.	Parameter number (PNU) 3)										
Response	ResID (AK) 2)				res.	Parameter number (PNU) 3)										

1) ReqID (AK): Request Identifier – job identifier (read, write, ...)

2) ResID (AK): Response Identifier (transferred value, error, ...)

3) Parameter number (PNU) – identifies and addresses the respective parameter ➔ section C.1. The task or response identifier indicates the type of task or reply ➔ section C.1.2.

Tab. C.3 Structure of parameter identifier (ParID)

C.1.2 Task identifiers, response identifiers and error numbers

The task identifiers are shown in the following table. All parameter values are always transmitted as a double word, independent of the data type.

ReqID	Description	Response identifier	
		Positive	Negative
0	No job ("Zero request")	0	–
6	Request parameter value (array, double word)	5	7
8	Modify parameter value (array, double word)	5	7
13	Request lower limit	5	7
14	Request upper limit	5	7

Tab. C.4 Task and response identifiers

If the job cannot be carried out, response identifier 7 as well as the appropriate error number will be transmitted (negative reply).
The following table shows the Response identifiers:

ResID	Description
0	No reply
5	Parameter value transferred (array, double word)
7	Job cannot be carried out (with error number) 1)

1) Error numbers ➔ Tab. C.6

Tab. C.5 Reply identifiers

If the parameter processing job cannot be carried out, a corresponding error number will be transmitted in the response telegram (byte 5 … 8 of the FPC range). The sequence of error checking and the possible error numbers are shown in the following table:

No.	Error numbers		Description
1	0	0x00	Impermissible PNU. The parameter does not exist.
2	3	0x03	Faulty subindex
3	101	0x65	ReqID is not supported
4	1	0x01	Parameter value cannot be changed (read only)
	102	0x66	Parameter is write-only (e.g. with passwords)
5	17	0x11	Task cannot be carried out due to operating status
6	11	0x0B	No supervising access
7	12	0x0C	Incorrect password
8	2	0x02	Lower or upper value limit exceeded

Tab. C.6 Sequence of error checking and error numbers

C.1.3 Rules for job reply processing

Rule	Description
1	If the master transmits the identifier for “No job”, the controller responds with the reply identifier for “No reply”.
2	A job or reply telegram always refers to a single parameter.
3	The master must continue to send a job until it has received the appropriate reply from the controller.
4	The master recognises the reply to the job placed: – By evaluating the Response identifier – By evaluating the parameter number (PNU) – If applicable, by evaluating the subindex (IND) – If applicable, by evaluating the parameter value.
5	The controller supplies the reply until the master sends a new job.
6	a) A write task, even with cyclic repetition of the same job, will only be carried out once by the controller. b) **Important:** Between two successive jobs, the task identifier 0 (no job, “zero request”) must be sent and the response identifier 0 (no reply) must be awaited. This ensures that an “old” response is not interpreted as a “new” response.

Tab. C.7 Rules for job reply processing

Sequence of parameter processing

Note
Observe the following when modifying parameters:
An FHPP control signal (e.g. start of a positioning job), which is to refer to a modified parameter, may only follow when the response identifier “Parameter value transferred” is received for the corresponding parameter.

If, for example, a position value in a position register is to be modified and if a movement is then to be made to this position, the positioning command must not be given until the controller has completed and confirmed the modification of the position register.

Changed parameters must be saved securely against power outages with PNU 127.

Example of parameterisation via FPC

The following tables show an example of parameterisation of a positioning task in the position set table via (FPC – Festo Parameter Channel).

Observe the specification in the bus master for the representation of words and double words (Intel/Motorola). In the example, the representation uses the “little endian” representation (lowest-order byte first).

Step 1

Output status of the 8 bytes of FPC data:

FPC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
	Reserved	Sub-index	ReqID/ResID + PNU		Parameter value			
Output data	0x00	0x00	0x**00**	0x00	0x00	0x00	0x00	0x00
Input data	0x00	0x00	0x**00**	0x00	0x00	0x00	0x00	0x00

Tab. C.8 Example, Step 1

Step 2

Read setpoint value from record number 2:
PNU 404 (0x0194), subindex 2 – Request parameter value (array, double word): ReqID 6.
Received value in the response: 0x64 = 100_d

FPC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
	Reserved	Sub-index	ReqID/ResID + PNU		Parameter value			
Output data	0x00	0x**02**	0x**94**	0x**61**	0x00	0x00	0x00	0x00
Input data	0x00	0x**02**	0x**94**	0x**51**	0x**64**	0x00	0x00	0x00

Tab. C.9 Example, Step 2

Step 3

“Zero request”: After receiving the input data with ResID 5, send output data with ReqID = 0 and wait for input data with ResID = 0:

FPC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
	Reserved	Sub-index	ReqID/ResID + PNU		Parameter value			
Output data	0x00	0x00	0x**0**0	0x00	0x00	0x00	0x00	0x00
Input data	0x00	0x00	0x**00**	0x00	0x64	0x00	0x00	0x00

Tab. C.10 Example, Step 3

Step 4

Write setpoint value 4660_d (0x1234) in record number 2:
PNU 404 (0x0194), subindex 2 – Modify parameter value (array, double word): ReqID 8 – value 0x1234.

FPC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
	Reserved	Sub-index	ReqID/ResID + PNU		Parameter value			
Output data	0x00	0x02	0x**94**	0x**81**	0x**34**	0x**12**	0x**00**	0x**00**
Input data	0x00	0x02	0x**94**	0x**51**	0x**34**	0x**12**	0x**00**	0x**00**

Tab. C.11 Example, Step 4

Step 5

After receiving the input data with ResID 5: “Zero request”, like Step 3 ➔ Tab. C.10.

Step 6

Write velocity 30531_d (0x7743) in record number 2:
PNU 406 (0x0196), subindex 2 – Modify parameter value (array, double word): ReqID 8 – value 0x7743.

FPC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
	Reserved	Sub-index	ReqID/ResID + PNU		Parameter value			
Output data	0x00	0x00	0x**96**	0x**81**	0x**43**	0x**77**	0x**00**	0x**00**
Input data	0x00	0x00	0x**96**	0x**51**	0x**43**	0x**77**	0x**00**	0x**00**

Tab. C.12 Example, Step 6

Step 7

After receiving the input data with ResID 5: “Zero request”, like Step 3 ➔ Tab. C.10.

C.2 FHPP+

C.2.1 FHPP+ overview

FHPP+ is an expansion of the FHPP communication protocol.

To find out whether this function is supported by the controller you are using and from which firmware version, see the help for the associated FCT plug-in.

The FHPP+ expansion allows additional PNUs configured by the user to be transmitted via the cyclic telegram, in addition to the control and status bytes and the optional parameter channel (FPC).
The minimum configuration for each telegram contains the control and status bytes, meaning that 8 bytes are sent and received. If the parameter channel is transmitted as well, it directly follows the I/O channel.
FHPP+ can be used to attach additional setpoint values to the received telegram which are not represented in the control and status bytes or in the FPC. Additional actual values can be forwarded in the response telegram, such as the intermediate circuit voltage or the temperature of the output stage.
The additional data (FHPP+) must always be transmitted in multiples of 8 bytes, up to a total length of 32 bytes.

The data transmitted via FHPP+ is configured using the FHPP+ telegram editor in the controller's FCT plug-in.

Note
Not all PNUs can be configured for the FHPP+ telegram. For example, the PNUs 40 to 43 cannot be transmitted at all; PNUs without write access cannot be configured in the output data; etc.

C.2.2 Structure of the FHPP+ telegram

The first entry in the telegram (address 0) is reserved for the I/O channel.
Optionally, if the parameter channel FPC is required by the application and it has been defined in the bus configuration, it must be selected as the second entry (address 8). The parameter channel must only be configured in this position.
From the third entry onwards in the telegram (address 16), or the second entry if FPC is not used (address 8), all remaining PNUs can be mapped which are required in the application.
With certain control systems (e.g. SIEMENS S7), make sure that PNUs with lengths of 2 or 4 bytes are in suitable addresses. These PNUs should only be inserted in even addresses. Placeholders are defined so that any gaps can be filled. They can be used to ensure that PNUs can be mapped in the addresses desired.
All unused parts of a telegram and especially all unused entries in the telegram editor are filled with the placeholders.

C.2.3 Examples

Example 1: With FPC, maximum 16 bytes for FHPP+

<table>
<tr><th colspan="32">Output data, bytes 1 … 32</th></tr>
<tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td><td>32</td></tr>
<tr><td colspan="8">CCON, CPOS, …</td><td colspan="8">PKW (PNU, SI)</td><td>…</td><td>…</td><td colspan="4">PNU…</td><td colspan="2">…</td><td colspan="4">PNU…</td><td colspan="2">…</td><td colspan="2">…</td></tr>
<tr><td colspan="8">Control bytes</td><td colspan="8">Parameter channel FPC</td><td colspan="16">FHPP+ (max. 16 bytes)</td></tr>
</table>

Tab. C.13 Example 1, output data

<table>
<tr><th colspan="32">Input data, bytes 1 … 32</th></tr>
<tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td><td>32</td></tr>
<tr><td colspan="8">SCON, SPOS, …</td><td colspan="8">PKW (PNU, SI)</td><td colspan="4">PNU…</td><td colspan="4">PNU…</td><td colspan="4">PNU…</td><td colspan="4">PNU…</td></tr>
<tr><td colspan="8">Status bytes</td><td colspan="8">Parameter channel FPC</td><td colspan="16">FHPP+ (max. 16 bytes)</td></tr>
</table>

Tab. C.14 Example 1, input data

Example 2: Without FPC, maximum 24 bytes for FHPP+

<table>
<tr><th colspan="32">Output data, bytes 1 … 32</th></tr>
<tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td><td>32</td></tr>
<tr><td colspan="8">CCON, CPOS, …</td><td colspan="4">PNU…</td><td colspan="2">…</td><td colspan="4">PNU…</td><td colspan="4">PNU…</td><td colspan="2">…</td><td colspan="4">PNU…</td><td colspan="2">…</td><td colspan="2">…</td></tr>
<tr><td colspan="8">Control bytes</td><td colspan="24">FHPP+ (max. 24 bytes)</td></tr>
</table>

Tab. C.15 Example 2, output data

<table>
<tr><th colspan="32">Input data, bytes 1 … 32</th></tr>
<tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td><td>32</td></tr>
<tr><td colspan="8">SCON, SPOS, …</td><td colspan="4">PNU…</td><td colspan="4">PNU…</td><td colspan="4">PNU…</td><td colspan="4">PNU…</td><td>…</td><td>…</td><td colspan="4">PNU…</td><td colspan="2">…</td></tr>
<tr><td colspan="8">Status bytes</td><td colspan="24">FHPP+ (max. 24 bytes)</td></tr>
</table>

Tab. C.16 Example 2, input data

The lengths of the output and input data can deviate from each other.
For example, 8 bytes of output data and 16 bytes of input data are possible.

C.2.4 Telegram editor for FHPP+

The transmitted data is configured solely via the FHPP+ Editor provided by the FCT plug-in. The corresponding PNUs 40 and 41 can only be read ➔ section B.4.2.
The FHPP+ telegram editor assigns the data contents of the cyclic FHPP telegram uniquely to the PNUs. The specifications provide generally for 16 entries per received and sent telegram. The current stage of development permits up to 10 entries for the CMMP-AS controller. The maximum length of a telegram is restricted to 32 bytes.
The PNUs for telegram mapping settings must not be mapped in the FHPP+ telegram.

C.2.5 Configuration of the fieldbuses with FHPP+

The data defined in the Telegram Editor must be configured on the master/scanner specifically for each fieldbus, for example by means of the corresponding GSD or EDS files.

D Diagnostic messages

If an error occurs, the motor controller CMMP-AS-...-M0 displays a diagnostic message cyclically in the 7-segments display. An error message consists of an E (for Error), a main index and a sub-index, e.g.: **- E 0 1 0 -**.
Warnings have the same number as an error message. In contrast to error messages, however, warnings are preceded and followed by hyphens, e.g. **- 1 7 0 -**.

D.1 Explanations of the diagnostic messages

The following table summarises the significance of the diagnostic messages and the actions to be taken in response to them:

Terms	Significance
No.	Main index (fault group) and sub-index of the diagnostic message. Display in the indicator, in FCT or diagnostic memory via FHPP.
Code	The Code column includes the error code (Hex) via CiA 301.
Message	Message that is displayed in the FCT.
Cause	Possible causes for the message.
Action	Action by the user.
Reaction	The Reaction column includes the error response (default setting, partially configurable): – PS off (switch off output stage), – MCStop (fast stop with maximum current), – QStop (fast stop with parameterised ramp), – Warn (warning), – Ignore (No message, only entry in diagnostic memory), – NoLog (No message and no entry in diagnostic memory).

Tab. D.1 Explanations of the diagnostic messages

A complete list of the diagnostic messages corresponding to the firmware statuses at the time of printing of this document can be found in section D.2.

D.2 Diagnostic messages with instructions for fault clearance

Error group 0		Information		
No.	Code	Message		Reaction
0-0	-	**Invalid error**		Ignore
		Cause	Information: An invalid error entry (corrupted) was found in the diagnostic memory marked with this error number. The system time entry is set to 0.	
		Measure	–	
0-1	-	**Invalid error detected and corrected**		Ignore
		Cause	Information: An invalid error entry (corrupted) was found in the diagnostic memory and corrected. The Additional informationrma-tion contains the original error number. The system time entry includes the address of the corrupted error number.	
		Measure	–	
0-2	-	**Error cleared**		Ignore
		Cause	Information: Active errors were acknowledged.	
		Measure	–	
0-4	-	**Serial number / device type changed (change of modules)**		Ignore
		Cause	Information: ➔ Entry in the diagnostic memory.	
		Measure	–	
0-7	-	**Consecutive Entry**		Ignore
		Cause	Information: ➔ Entry in the diagnostic memory.	
		Measure	–	
0-8	-	**Controller switched on**		Ignore
		Cause	Information: ➔ Entry in the diagnostic memory.	
		Measure	–	
0-9	-	**Controller safety parameters changed**		Ignore
		Cause	Information: ➔ Entry in the diagnostic memory.	
		Measure	–	
0-11	-	**Module change: Previous module**		Ignore
		Cause	Information: ➔ Entry in the diagnostic memory.	
		Measure	–	
0-12	-	**Module change: Current module**		Ignore
		Cause	Information: ➔ Entry in the diagnostic memory.	
		Measure	–	
0-21	-	**Log entry of the Safety module**		Ignore
		Cause	Information: ➔ Entry in the diagnostic memory.	
		Measure	–	
0-22	-	**Default parameter set loaded**		Ignore
		Cause	Information: ➔ Entry in the diagnostic memory.	
		Measure	–	

Error group 1		**Stack overflow**		
No.	Code	Message		Reaction
1-0	6180h	**Stack overflow**		PSoff
		Cause	– Incorrect firmware? – Sporadic high processor load due to cycle time being too short and specific processor-intensive processes (save parameter set, etc.).	
		Measure	• Load an approved firmware. • Reduce processor load. • Contact Technical Support.	

Error group 2		**Intermediate circuit**		
No.	Code	Message		Reaction
2-0	3220h	**Intermediate circuit undervoltage**		configurable
		Cause	Intermediate circuit voltage falls below the parameterised threshold (➔ Additional informationrmation). Error priority set too high?	
		Measure	• Quick discharge due to switched-off mains supply. • Check the power supply. • Couple intermediate circuits if technically permissible. • Check intermediate circuit voltage (measure). • Check undervoltage monitoring (threshold value).	
		Additional informa-tion	Additional informationrmation in PNU 203/213: Top 16 bits: Status number of internal state machine Bottom 16 bits: Intermediate circuit voltage (internal scaling ap-prox. 17.1 digit/V).	

Error group 3		**Motor over-temperature**		
No.	Code	Message		Reaction
3-0	4310h	**Analogue motor overtemperature**		QStop
		Cause	Motor overloaded, temperature too high. – Motor too hot? – Incorrect sensor? – Sensor faulty? – Broken cable?	
		Measure	• Check parameterisation (current regulator, current limits). • Check the parameterisation of the sensor or the sensor charac-teristics. If the error persists when the sensor is bypassed: device faulty.	

<table>
<tr><th>Error group 3</th><th colspan="4">Motor over-temperature</th></tr>
<tr><th>No.</th><th>Code</th><th colspan="2">Message</th><th>Reaction</th></tr>
<tr><td rowspan="3">3-1</td><td rowspan="3">4310h</td><td colspan="2">Digital motor overtemperature</td><td>configurable</td></tr>
<tr><td>Cause</td><td colspan="2">– Motor overloaded, temperature too high.
– Suitable sensor or sensor characteristics parameterised?
– Sensor faulty?</td></tr>
<tr><td>Measure</td><td colspan="2">• Check parameterisation (current regulator, current limits).
• Check the parameterisation of the sensor or the sensor characteristics.
If the error persists when the sensor is bypassed: device faulty.</td></tr>
<tr><td rowspan="3">3-2</td><td rowspan="3">4310h</td><td colspan="2">Analogue motor overtemperature: Broken wire</td><td>configurable</td></tr>
<tr><td>Cause</td><td colspan="2">The measured resistance value is above the threshold for wire break detection.</td></tr>
<tr><td>Measure</td><td colspan="2">• Check the connecting cables of the temperature sensor for wire breaks.
• Check the parameterisation (threshold value) for wire break detection.</td></tr>
<tr><td rowspan="3">3-3</td><td rowspan="3">4310h</td><td colspan="2">Analogue motor overtemperature: Short circuit</td><td>configurable</td></tr>
<tr><td>Cause</td><td colspan="2">The measured resistance value is below the threshold for short circuit detection.</td></tr>
<tr><td>Measure</td><td colspan="2">• Check the connecting cables of the temperature sensor for wire breaks.
• Check the parameterisation (threshold value) for short circuit detection.</td></tr>
</table>

<table>
<tr><th>Error group 4</th><th colspan="4">Power section/intermediate circuit over-temperature</th></tr>
<tr><th>No.</th><th>Code</th><th colspan="2">Message</th><th>Reaction</th></tr>
<tr><td rowspan="3">4-0</td><td rowspan="3">4210h</td><td colspan="2">Power section overtemperature</td><td>configurable</td></tr>
<tr><td>Cause</td><td colspan="2">Device is overheated
– Is displayed temperature plausible?
– Device fan faulty?
– Device overloaded?</td></tr>
<tr><td>Measure</td><td colspan="2">• Check installation conditions; control cabinet fan filter dirty?
• Check the cylinder sizing (due to possible overloading in continuous duty).</td></tr>
</table>

Error group 4		Power section/intermediate circuit over-temperature		
No.	Code	Message		Reaction
4-1	4280h	**Intermediate circuit overtemperature**		configurable
		Cause	Device is overheated – Is displayed temperature plausible? – Device fan faulty? – Device overloaded?	
		Measure	• Check installation conditions; control cabinet fan filter dirty? • Check the cylinder sizing (due to possible overloading in continuous duty).	

Error group 5		Internal voltage supply		
No.	Code	Message		Reaction
5-0	5114h	**Failure of internal voltage 1**		PSoff
		Cause	Internal power supply monitor has detected undervoltage. This is either due to an internal defect or an overload/short circuit caused by connected peripherals.	
		Measure	• Check digital outputs and brake output for short circuit or specified load. • Separate device from the entire peripheral equipment and check whether the error is still present after reset. If so, an internal defect is present ➔ Repair by the manufacturer.	
5-1	5115h	**Failure of internal voltage 2**		PSoff
		Cause	Internal power supply monitor has detected undervoltage. This is either due to an internal defect or an overload/short circuit caused by connected peripherals.	
		Measure	• Check digital outputs and brake output for short circuit or specified load. • Separate device from the entire peripheral equipment and check whether the error is still present after reset. If so, an internal defect is present ➔ Repair by the manufacturer.	
5-2	5116h	**Failure of driver supply**		PSoff
		Cause	Internal power supply monitor has detected undervoltage. This is either due to an internal defect or an overload/short circuit caused by connected peripherals.	
		Measure	• Check digital outputs and brake output for short circuit or specified load. • Separate device from the entire peripheral equipment and check whether the error is still present after reset. If so, an internal defect is present ➔ Repair by the manufacturer.	

Error group 5		Internal voltage supply		
No.	Code	Message		Reaction
5-3	5410h	**Undervoltage of digital I/O**		PSoff
		Cause	Overloading of the I/Os? Faulty peripheral device?	
		Measure	• Check connected peripherals for short circuit / rated loads. • Check connection of the brake (connected incorrectly?).	
5-4	5410h	**Overcurrent of digital I/O**		PSoff
		Cause	Overloading of the I/Os? Faulty peripheral device?	
		Measure	• Check connected peripherals for short circuit / rated loads. • Check connection of the brake (connected incorrectly?).	
5-5	-	**Module supply voltage failure**		PSoff
		Cause	Defect on the plugged-in interface.	
		Measure	• Interface replacement ➔ Repair by the manufacturer.	
5-6	-	**X10, [X11] and RS232 supply voltage failure**		PSoff
		Cause	Overloading through connected peripherals.	
		Measure	• Check pin allocation of the connected peripherals. • Short circuit?	
5-7	-	**Safety module internal voltage failure**		PSoff
		Cause	Defect on the safety module.	
		Measure	• Internal defect ➔ Repair by the manufacturer.	
5-8	-	**Failure of Internal voltage 3 (15V)**		PSoff
		Cause	Defect in the motor controller.	
		Measure	• Internal defect ➔ Repair by the manufacturer.	
5-9	-	**Encoder supply defective**		PSoff
		Cause	Back measurement of the encoder voltage not OK.	
		Measure	• Internal defect ➔ Repair by the manufacturer.	

Error group 6		Overload current		
No.	Code	Message		Reaction
6-0	2320h	**Output stage short-circuit**		PSoff
		Cause	– Faulty motor, e.g. winding short circuit due to motor overheating or short to PE inside motor. – Short circuit in the cable or the connecting plugs, i.e. short circuit between motor phases or to the screening/PE. – Output stage faulty (short circuit). – Incorrect parameterisation of the current regulator.	
		Measure	Dependent on the status of the system ➔ Additional informationrmation, cases a) to f).	
		Additional information	Actions: a) Error only with active brake chopper: Check external braking resistor for short circuit or insufficient resistance value. Check circuitry of the brake chopper output at the motor controller (jumper, etc.). b) Error message immediately when the power supply is connected: internal short circuit in the output stage (short circuit of a complete half-jumper). The motor controller can no longer be connected to the power supply; the internal (and possibly external) fuses are tripped. Repair by the manufacturer required. c) Short circuit error message not until the output stage or controller is enabled. d) Disconnection of motor plug [X6] directly at the motor controller. If the error still occurs, there is a fault in the motor controller. Repair by the manufacturer required. e) If the error only occurs when the motor cable is connected: Check the motor and cable for short circuits, e.g. with a multimeter. f) Check parameterisation of the current regulator. Oscillations in an incorrectly parameterised current regulator can generate currents up to the short circuit threshold, usually clearly audible as a high-frequency whistling. Verification, if necessary, with the trace in the FCT (actual active current value).	
6-1	2320h	**Brake chopper overcurrent**		PSoff
		Cause	Overload current at the brake chopper output.	
		Measure	• Check external braking resistor for short circuit or insufficient resistance value. • Check circuitry of the brake chopper output at the motor controller (jumpers, etc.).	

Error group 7		Overvoltage in intermediate circuit		
No.	Code	Message		Reaction
7-0	3210h	**Intermediate circuit overvoltage**		PSoff
		Cause	Braking resistor is overloaded; too much braking energy, which cannot be dissipated quickly enough. – Incorrect level of resistance? – Resistor not connected correctly? – Check design (application).	
		Measure	• Check the design of the braking resistor; resistance value may be too great. • Check the connection to the braking resistor (internal/external).	

Error group 8		Angle encoder error		
No.	Code	Message		Reaction
8-0	7380h	**Resolver angle encoder error**		configurable
		Cause	Resolver signal amplitude is faulty.	
		Measure	Step-by-step procedure ➔ Additional informationrmation, cases a) to c).	
		Additional information	a) If possible, test with a different (error-free) resolver (replace the connecting cable, too). If the error still occurs, there is a fault in the motor controller. Repair by the manufacturer required. b) If the error occurs only with a special resolver and its connecting cable: Check resolver signals (carrier and SIN/COS signal), see specification. If the signals do not comply with the signal specifications, replace the resolver. c) If the error recurs sporadically, check the screen bonding or check whether the resolver simply has an insufficient transmission ratio (standard resolver: A = 0.5).	

Error group 8		Angle encoder error		
No.	Code	Message		Reaction
8-1	-	**Direction of rotation of the serial and incremental position evaluation is not identical**		configurable
		Cause	Only encoders with serial position transmission combined with an analogue SIN/COS signal track: The directions of rotation for position determination in the encoder and for incremental evaluation of the analogue track system in the motor controller are the wrong way round ➔ Additional informationrmation.	
		Measure	Swap the following signals on the [X2B] angle encoder interface (the wires in the connecting plug must be changed around), observing the technical data for the angle encoder where applicable: – Swap SIN / COS track. – Swap the SIN+/SIN- or COS+/COS- signals, as applicable.	
		Additional information	The encoder counts internally, for example positively in clockwise rotation, while the incremental evaluation counts in negative direction with the same mechanical rotation. The interchange of the direction of rotation is detected mechanically at the first movement of over 30°, and the error is triggered.	
8-2	7382h	**Incremental encoder Z0 track signals error**		configurable
		Cause	Signal amplitude of the Z0 track at [X2B] is faulty. – Angle encoder connected? – Angle encoder cable defective? – Angle encoder faulty?	
		Measure	Check configuration of the angle encoder interface: a) Z0 evaluation activated, but no tracking signals connected or on hand ➔ Additional informationrmation. b) Encoder signals faulty? c) Test with another encoder. ➔ Tab. D.2, page 306.	
		Additional information	For example, EnDat 2.2 or EnDat 2.1 without analogue track. Heidenhain encoder: order codes EnDat 22 and EnDat 21. With these encoders there are no incremental signals, even when the cables are connected.	

Error group 8		Angle encoder error		
No.	Code	Message		Reaction
8-3	7383h	**Incremental encoder Z1 track signals error**		configurable
		Cause	Signal amplitude of the Z1 track at X2B is faulty. – Angle encoder connected? – Angle encoder cable defective? – Angle encoder faulty?	
		Measure	Check configuration of the angle encoder interface: a) Z1 evaluation activated but not connected. b) Encoder signals faulty? c) Test with another encoder. ➔ Tab. D.2, page 306.	
8-4	7384h	**Digital incremental encoder track signals error [X2B]**		configurable
		Cause	Faulty A, B or N tracking signals at [X2B]. – Angle encoder connected? – Angle encoder cable defective? – Angle encoder faulty?	
		Measure	Check the configuration of the angle encoder interface. a) Encoder signals faulty? b) Test with another encoder. ➔ Tab. D.2, page 306.	
8-5	7385h	**Incremental encoder Hall generator signals error**		configurable
		Cause	Hall encoder signals of a dig. inc. at [X2B] faulty. – Angle encoder connected? – Angle encoder cable defective? – Angle encoder faulty?	
		Measure	Check the configuration of the angle encoder interface. a) Encoder signals faulty? b) Test with another encoder. ➔ Tab. D.2, page 306.	

Error group 8		Angle encoder error		
No.	Code	Message		Reaction
8-6	7386h	**Faulty angle encoder communication**		configurable
		Cause	Communication to serial angle encoders is disrupted (EnDat encoders, HIPERFACE encoders, BiSS encoders). – Angle encoder connected? – Angle encoder cable defective? – Angle encoder faulty?	
		Measure	Check configuration of the angle encoder interface, procedure corresponding to a) to c): a) Serial encoder parameterised but not connected? Incorrect serial protocol selected? b) Encoder signals faulty? c) Test with another encoder. ➔ Tab. D.2, page 306.	
8-7	7387h	**Signal amplitude of encoder erroneous [X10]**		configurable
		Cause	Faulty A, B, or N tracking signals at [X10]. – Angle encoder connected? – Angle encoder cable defective? – Angle encoder faulty?	
		Measure	Check the configuration of the angle encoder interface. a) Encoder signals faulty? b) Test with another encoder. ➔ Tab. D.2, page 306.	
8-8	7388h	**Internal angle encoder error**		configurable
		Cause	Internal monitoring of the angle encoder [X2B] has detected an error and forwarded it via serial communication to the controller. – Diminishing illumination intensity with visual encoders? – Excess rotational speed? – Angle encoder faulty?	
		Measure	If the error occurs repeatedly, the encoder is faulty. ➔ Replace encoder.	

Error group 8		Angle encoder error		
No.	Code	Message		Reaction
8-9	7389h	**Angle encoder at [X2B] not supported**		configurable
		Cause	Angle encoder type read at [X2B], which is not supported or cannot be used in the desired operating mode. – Incorrect or inappropriate protocol type selected? – Firmware does not support the connected encoder variant?	
		Measure	Depending on the Additional informationrmation of the error message ➔ Additional informationrmation: • Load appropriate firmware. • Check/correct the configuration for encoder analysis. • Connect an appropriate encoder type.	
		Additional information	Additional informationrmation (PNU 203/213): 0001: HIPERFACE: Encoder type is not supported by the firmware ➔ connect another encoder type or load more recent firmware, if applicable. 0002: EnDat: The address space in which the encoder parameters would have to lie does not exist with the connected EnDat encoder ➔ check the encoder type. 0003: EnDat: Encoder type is not supported by the firmware ➔ connect another encoder type or load more recent firmware, if applicable. 0004: EnDat: Encoder rating plate cannot be read from the connected encoder. ➔ Change encoder or load more recent firmware, if applicable. 0005: EnDat: EnDat 2.2 interface parameterised, but connected encoder supports only EnDat2.1. ➔ Replace encoder type or reparameterise to EnDat 2.1. 0006: EnDat: EnDat2.1 interface with analogue track evaluation parameterised, but according to rating plate the connected encoder does not support tracking signals. ➔ Replace encoder or switch off Z0 tracking signal evaluation. 0007: Code length measuring system with EnDat2.1 connected, but parameterised as a purely serial encoder. Purely serial evaluation is not possible due to the long response times of this encoder system. Encoder must be operated with analogue tracking signal evaluation ➔ connect to analogue Z0 tracking signal evaluation.	

Error group 9		Error in the angle encoder parameter set		
No.	Code	Message		Reaction
9-0	73A1h	**Old encoder parameter set**		configurable
		Cause	Warning: An encoder parameter set in an old format was found in the EEP-ROM of the connected encoder. This has been converted and saved in the new format.	
		Measure	No action necessary at this point. The warning should not re-appear when the 24 V supply is switched back on.	
9-1	73A2h	**Encoder parameter set cannot be decoded**		configurable
		Cause	Data in the EEPROM of the angle encoder could not be read completely, or access to it was partly refused.	
		Measure	The EEPROM of the encoder contains data (communication objects) which is not supported by the loaded firmware. The data in question is then discarded. • The parameter set can be adapted to the current firmware by writing the encoder data to the encoder. • Alternatively, load appropriate (more recent) firmware.	
9-2	73A3h	**Unknown encoder parameter set version**		configurable
		Cause	The data saved in the EEPROM is not compatible with the current version. A data structure was found which the loaded firmware is unable to decode.	
		Measure	• Save the encoder parameters again in order to delete the parameter record in the encoder and replace it with a readable record (this will, however, delete the data in the encoder irreversibly). • Alternatively, load appropriate (more recent) firmware.	
9-3	73A4h	**Defective data structure angle encoder parameter set**		configurable
		Cause	Data in EEPROM does not match the stored data structure. The data structure was identified as valid but may be corrupted.	
		Measure	• Save the encoder parameters again in order to delete the parameter record in the encoder and replace it with a readable record. If the error still occurs after that, the encoder may be faulty. • Replace the encoder as a test.	

Error group 9		Error in the angle encoder parameter set		
No.	Code	Message		Reaction
9-4	-	**EEPROM data: User-specific configuration faulty**		configurable
		Cause	Only for special motors: The plausibility check returns an error, e.g. because the motor was repaired or replaced.	
		Measure	• If motor repaired: Carry out homing again and save in the angle encoder, after that (!) save in the motor controller. • If motor replaced: Parameterise the controller again, then carry out homing again and save in the angle encoder, after that (!) save in the motor controller.	
9-5	-	**Read/Write Error EEPROM parameter data**		configurable
		Cause	Error occurred during reading or writing data to the internal encoder parameter set.	
		Measure	Occurs with Hiperface encoders: A data field of the encoder is not suitable to be read from the firmware or data can not be written for unknown reasons. • Send motor to the manufacturer for inspection.	
9-7	73A5h	**Encoder EEPROM is write protected**		configurable
		Cause	Data cannot be saved in the EEPROM of the angle encoder. Occurs with Hiperface encoders.	
		Measure	A data field in the encoder EEPROM is write-protected (e.g. after operation on a motor controller of another manufacturer). No solution possible, encoder memory must be unlocked with a corresponding parameterisation tool (from manufacturer).	
9-9	73A6h	**Memory size of encoder EEPROM too small**		configurable
		Cause	It is not possible to save all the data in the EEPROM of the angle encoder.	
		Measure	• Reduce the number of data records to be saved. Please read the documentation or contact Technical Support.	

Error group 10		Exceeding max. speed		
No.	Code	Message		Reaction
10-0	-	**Overspeed**		configurable
		Cause	– Motor racing ("spinning") because the commutation angle offset is incorrect. – Motor is parameterised correctly, but the limit for spinning protection is set too low.	
		Measure	• Check the commutation angle offset. • Check the parameterisation of the limit value.	

Error group 11		Homing		
No.	Code	Message		Reaction
11-0	8A80h	**Error when homing is started**		configurable
		Cause	Controller enable missing.	
		Measure	Homing can only be started when closed-loop controller enable is active. • Check the condition or sequence.	
11-1	8A81h	**Error during homing**		configurable
		Cause	Homing was interrupted, e.g. by: – Withdrawal of controller release. – Reference switch is beyond the limit switch. – External stop signal (a phase was aborted during homing).	
		Measure	• Check homing sequence. • Check arrangement of the switches. • If applicable, lock the stop input during homing if it is not desired.	
11-2	8A82h	**Homing: No valid zero pulse**		configurable
		Cause	Required zero pulse during homing missing.	
		Measure	• Check the zero pulse signal. • Check the angle encoder settings.	
11-3	8A83h	**Homing: Timeout**		configurable
		Cause	The parameterised maximum time for the homing run was exceeded before homing was completed.	
		Measure	• Check the time setting in the parameters.	
11-4	8A84h	**Homing: Incorrect limit switch**		configurable
		Cause	– Associated limit switch not connected. – Limit switches swapped? – No reference switch found between the two limit switches. – Reference switch is on the limit switch. – Current position with zero pulse method: Limit switch active in the area of the zero pulse (not permissible). – Both limit switches active at the same time.	
		Measure	• Check whether the limit switches are connected in the correct direction of travel or whether the limit switches have an effect on the intended inputs. • Reference switch connected? • Check configuration of the reference switches. • Move limit switch so that it is not in the zero pulse area. • Check limit switch parameterisation (N/C contact/N/O contact.	

Error group 11		Homing		
No.	Code	Message		Reaction
11-5	8A85h	**Homing: I²t / following error**		configurable
		Cause	– Unsuitable acceleration ramp parameters. – Change of direction due to premature triggering of following error; check parameterisation of following error. – No reference switch reached between the end stops. – Zero pulse method: End stop reached (not permissible here).	
		Measure	• Parameterise the acceleration ramps so they are flatter. • Check connection of a reference switch. • Method appropriate for the application?	
11-6	8A86h	**Homing: End of search path**		configurable
		Cause	The maximum permissible path for the homing run has been travelled without reaching the point of reference or the homing run destination.	
		Measure	Fault in switch detection. • Switch for homing faulty?	
11-7	-	**Homing: Error encoder difference monitoring**		configurable
		Cause	Deviation between the actual position value and commutation position is too great. External angle encoder not connected or faulty?	
		Measure	• Deviation fluctuating, e.g. due to gear backlash; increase cut-off threshold if necessary. • Check connection of the actual value encoder.	

Error group 12		CAN communication		
No.	Code	Message		Reaction
12-0	8180h	**CAN: Double node number**		configurable
		Cause	Node number assigned twice.	
		Measure	• Check the configuration of the participants on the CAN bus.	
12-1	8120h	**CAN: Communication error, bus OFF**		configurable
		Cause	The CAN chip has switched off communication due to communication errors (BUS OFF).	
		Measure	• Check cabling: cable specification adhered to, broken cable, maximum cable length exceeded, correct terminating resistors, cable screening earthed, all signals terminated? • If necessary, replace device as a test. If a different device works without errors with the same cabling, send the device to the manufacturer for inspection.	

Error group 12		CAN communication		
No.	Code	Message		Reaction
12-2	8181h	**CAN: Communication error during transmission**		configurable
		Cause	The signals are corrupted when transmitting messages. Device boot-up is so fast that no other nodes on the bus have yet been detected when the boot-up message is sent.	
		Measure	• Check cabling: cable specification adhered to, broken cable, maximum cable length exceeded, correct terminating resistors, cable screening earthed, all signals terminated? • If necessary, replace device as a test. If a different device works without errors with the same cabling, send the device to the manufacturer for inspection.	
12-3	8182h	**CAN: Communication error during reception**		configurable
		Cause	The signals are corrupted when receiving messages.	
		Measure	• Check cabling: cable specification adhered to, broken cable, maximum cable length exceeded, correct terminating resistors, cable screening earthed, all signals terminated? • If necessary, replace device as a test. If a different device works without errors with the same cabling, send the device to the manufacturer for inspection.	
12-4	-	**No Node Guarding-telegram received**		configurable
		Cause	Node guarding telegram not received within the parameterised time. Signals corrupted?	
		Measure	• Compare the cycle time of the remote frames with that of the controller. • Check: failure of the controller?	
12-5	-	**CAN: RPDO too short**		configurable
		Cause	A received RPDO does not contain the parameterised number of bytes.	
		Measure	The number of parameterised bytes does not match the number of bytes received. • Check and correct parameterisation.	
12-9	-	**CAN: Protocol error**		configurable
		Cause	Faulty bus protocol.	
		Measure	• Check the parameterisation of the selected CAN bus protocol.	

Error group 13		CAN- bus timeout		
No.	Code	Message		Reaction
13-0	-	**CAN: Timeout**		configurable
		Cause	Error message from manufacturer-specific protocol.	
		Measure	• Check the CAN parameters.	

Error group 14		Identification		
No.	Code	Message		Reaction
14-0	-	**Automatic current controller identification: Insufficient intermediate circuit voltage**		PSoff
		Cause	Current regulator parameters cannot be determined (insufficient supply).	
		Measure	The available intermediate circuit voltage is too low to carry out the measurement.	
14-1	-	**Automatic current controller identification: Measurement cycle insufficient**		PSoff
		Cause	Too few or too many measurement cycles required for the connected motor.	
		Measure	Automatic parameter definition providing a time constant that is outside the parameterisable value range. • The parameters must be manually optimised.	
14-2	-	**Automatic current controller identification: Power stage could not be enabled**		PSoff
		Cause	The output stage has not been enabled.	
		Measure	• Check the connection of DIN4.	
14-3	-	**Automatic current controller identification: Output stage was switched off prematurely**		PSoff
		Cause	Output stage enable was switched off while identification was in progress.	
		Measure	• Check the sequence control.	
14-5	-	**Automatic angle encoder identification: Zero pulse could not be found**		PSoff
		Cause	The zero pulse could not be found following execution of the maximum permissible number of electrical revolutions.	
		Measure	• Check the zero pulse signal. • Angle encoder parameterised correctly?	
14-6	-	**Automatic angle encoder identification: Faulty Hall signals**		PSoff
		Cause	Hall signals faulty or invalid. The pulse train or segmenting of the Hall signals is inappropriate.	
		Measure	• Check connection. • Refer to the technical data to check whether the encoder shows three Hall signals with 1205 or 605 segments; if necessary, contact Technical Support.	

Error group 14		Identification		
No.	Code	Message		Reaction
14-7	-	**Automatic angle encode identification: Identification not possible**		PSoff
		Cause	Angle encoder at a standstill.	
		Measure	• Ensure sufficient intermediate circuit voltage. • Encoder cable connected to the right motor? • Motor blocked, e.g. holding brake does not release?	
14-8	-	**Automatic angle encoder identification: Invalid number of pairs of poles**		PSoff
		Cause	The calculated number of pole pairs lies outside the parameterisable range.	
		Measure	• Compare result with the technical data specifications for the motor. • Check the parameterised number of lines.	

Error group 15		Invalid operation		
No.	Code	Message		Reaction
15-0	6185h	**Division by zero**		PSoff
		Cause	Internal firmware error. Division by 0 when using the math library.	
		Measure	• Load factory settings. • Check the firmware to make sure that approved firmware has been loaded.	
15-1	6186h	**Mathematical overflow during division**		PSoff
		Cause	Internal firmware error. Overflow when using the math library.	
		Measure	• Load factory settings. • Check the firmware to make sure that approved firmware has been loaded.	
15-2	-	**Mathematical underflow**		PSoff
		Cause	Internal firmware error. Internal correction factors could not be calculated.	
		Measure	• Check the setting of the factor group for extreme values and change, if necessary.	

Error group 16		Internal error		
No.	Code	Message		Reaction
16-0	6181h	**Error during program execution**		PSoff
		Cause	Internal firmware error. Error during program execution. Illegal CPU command found in the program sequence.	
		Measure	• In case of repetition, load firmware again. If the error occurs repeatedly, the hardware is defective.	

Error group 16		Internal error		
No.	Code	Message		Reaction
16-1	6182h	**Illegal interrupt**		PSoff
		Cause	Error during program execution. An unused IRQ vector was used by the CPU.	
		Measure	• In case of repetition, load firmware again. If the error occurs repeatedly, the hardware is defective.	
16-2	6187h	**Initialisation error**		PSoff
		Cause	Error in initialising the default parameters.	
		Measure	• In case of repetition, load firmware again. If the error occurs repeatedly, the hardware is defective.	
16-3	6183h	**Unexpected state**		PSoff
		Cause	Error during periphery access within the CPU or error in the program sequence (illegal branching in case structures).	
		Measure	• In case of repetition, load firmware again. If the error occurs repeatedly, the hardware is defective.	

Error group 17		Following error exceeded		
No.	Code	Message		Reaction
17-0	8611h	**Following error limit exceeded**		configurable
		Cause	Comparison threshold for the limit value of the following error exceeded.	
		Measure	• Enlarge error window. • Parameterise acceleration to be less. • Motor overloaded (current limiter from I^2t monitoring active?).	
17-1	8611h	**Encoder difference monitoring**		configurable
		Cause	Deviation between the actual position value and commutation position is too great. External angle encoder not connected or faulty?	
		Measure	• Deviation fluctuating, e.g. due to gear backlash; increase cut-off threshold if necessary. • Check connection of the actual value encoder.	

Error group 18		Temperature warning thresholds		
No.	Code	Message		Reaction
18-0	-	**Analogue motor temperature**		configurable
		Cause	Motor temperature (analogue) more than 5° below T_max.	
		Measure	• Check parameterisation of current regulator and/or speed regulator. • Motor permanently overloaded?	

Error group 21		Current measurement		
No.	Code	Message		Reaction
21-0	5280h	**Error 1 current measurement U**		PSoff
		Cause	Offset for current measurement 1 phase U is too great. The controller carries out offset compensation of the current measurement every time its controller enable is issued. Tolerances that are too large result in an error.	
		Measure	If the error occurs repeatedly, the hardware is defective.	
21-1	5281h	**Error 1 current measurement V**		PSoff
		Cause	Offset for current measurement 1 phase V is too great.	
		Measure	If the error occurs repeatedly, the hardware is defective.	
21-2	5282h	**Error 2 current measurement U**		PSoff
		Cause	Offset for current measurement 2 phase U is too great.	
		Measure	If the error occurs repeatedly, the hardware is defective.	
21-3	5283h	**Error 2 current measurement V**		PSoff
		Cause	Offset for current measurement 2 phase V is too great.	
		Measure	If the error occurs repeatedly, the hardware is defective.	

Error group 22		PROFIBUS (only CMMP-AS-...-M3)		
No.	Code	Message		Reaction
22-0	-	**PROFIBUS: Initialisation error**		configurable
		Cause	Faulty initialisation of the PROFIBUS interface. Interface faulty?	
		Measure	• Replace interface. Repair by the manufacturer may be an option.	
22-2	-	**PROFIBUS: Faulty communication**		configurable
		Cause	Malfunctions in communication.	
		Measure	• Check the configured slave address. • Check the bus termination. • Check the wiring.	
22-3	-	**PROFIBUS: Invalid slave address**		configurable
		Cause	Communication was started with slave address 126.	
		Measure	• Select a different slave address.	
22-4	-	**PROFIBUS: Conversion error**		configurable
		Cause	During conversion with the factor group, the range of values was exceeded. Mathematical error in the conversion of the physical units.	
		Measure	The value ranges of the data and the physical units do not match. • Check and correct.	

Error group 23		Store/Restore actual position		
No.	Code	Message		Reaction
23-0	-	**Actual position: No valid record available**		configurable
		Cause	– No entry stored after activation. – No position stored, because drive is not referenced. – Hardware reset occurred too early.	
		Measure	Observe activation sequence: 1. Activate function. 2. Save and restart. 3. Execute homing.	
23-1	-	**Actual position: invalid checksum**		configurable
		Cause	Save operation can't be attained.	
		Measure	Repeat activation. Observe activation sequence: 1. Activate function. 2. Save and restart. 3. Execute homing.	
23-2	-	**Actual position: Flash content inconsistent**		configurable
		Cause	Internal error during saving operation.	
		Measure	Repeat activation. Observe activation sequence: 1. Activate function. 2. Save and restart. 3. Execute homing.	

Error group 25		Device type/function		
No.	Code	Message		Reaction
25-0	6080h	**Invalid device type**		PSoff
		Cause	Device coding not recognised or invalid.	
		Measure	This fault cannot be fixed by the user. • Send motor controller to the manufacturer.	
25-1	6081h	**Device type not supported**		PSoff
		Cause	Device coding invalid, is not supported by the loaded firmware.	
		Measure	• Load up-to-date firmware. • If newer firmware is not available, the problem may be a hardware defect. Send motor controller to the manufacturer.	
25-2	6082h	**Invalid hardware revision**		PSoff
		Cause	The controller's hardware version is not supported by the loaded firmware.	
		Measure	• Check the firmware version; update the firmware to a more recent version if necessary.	

Error group 25		Device type/function		
No.	Code	Message		Reaction
25-3	6083h	**Device with restricted functionality: Firmware cannot be executed**		PSoff
		Cause	Device is not enabled for this function.	
		Measure	Device is not enabled for the desired functionality and may need to be enabled by the manufacturer. The device must be sent in for this purpose.	
25-4	-	**Invalid power stage type**		PSoff
		Cause	– Power section area in the EEPROM is unprogrammed. – Power section is not supported by the firmware.	
		Measure	• Load appropriate firmware.	

Error group 26		Internal data error		
No.	Code	Message		Reaction
26-0	5580h	**Missing user parameter set**		PSoff
		Cause	No valid user parameter set in the flash memory.	
		Measure	• Load factory settings. If the error remains, the hardware may be defective.	
26-1	5581h	**Checksum error**		PSoff
		Cause	Checksum error of a parameter set.	
		Measure	• Load factory settings. If the error remains, the hardware may be defective.	
26-2	5582h	**Flash: Error when writing**		PSoff
		Cause	Error when writing the internal flash memory.	
		Measure	• Execute the last operation again. If the error appears again, the hardware may be faulty.	
26-3	5583h	**Flash: Error during deletion**		PSoff
		Cause	Error during deletion of the internal flash memory.	
		Measure	• Execute the last operation again. If the error appears again, the hardware may be faulty.	
26-4	5584h	**Flash: Internal flash error**		PSoff
		Cause	The default parameter set is corrupted / data error in the FLASH area where the default parameter set is located.	
		Measure	• Load firmware again. If the error appears again, the hardware may be faulty.	
26-5	5585h	**Missing calibration data**		PSoff
		Cause	Factory-set calibration parameters incomplete / corrupted.	
		Measure	This fault cannot be fixed by the user.	

Error group 26		Internal data error		
No.	Code	Message		Reaction
26-6	5586h	**Missing position data sets**		PSoff
		Cause	Position data sets incomplete or corrupted.	
		Measure	• Load the factory settings or • save the current parameters again so that the position data is written again.	
26-7	-	**Faulty data tables (CAM)**		PSoff
		Cause	Data for the cam disc is corrupted.	
		Measure	• Load factory settings. • Reload the parameter set if necessary. If the error persists, contact Technical Support.	

Error group 27		Following error monitoring		
No.	Code	Message		Reaction
27-0	8611h	**Following error warning threshold**		configurable
		Cause	– Motor overloaded? Check motor capacity. – Acceleration or braking ramps are set too steep. – Motor blocked? Commutation angle correct?	
		Measure	• Check the parameterisation of the motor data. • Check the parameterisation of the following error.	

Error group 28		Operating hour counter		
No.	Code	Message		Reaction
28-0	FF01h	**Missing operating hour counter**		configurable
		Cause	No record for an operating hour counter could be found in the parameter block. A new operating hour counter was created. Occurs during initial start-up or a processor change.	
		Measure	Warning only, no further action required.	
28-1	FF02h	**Operating hour counter: Write error**		configurable
		Cause	The data block in which the operating hour counter is stored could not be written to. Cause unknown; possibly problems with the hardware.	
		Measure	Warning only, no further action required. If the error occurs again, the hardware may be faulty.	

Error group 28		Operating hour counter		
No.	Code	Message		Reaction
28-2	FF03h	**Operating hour counter corrected**		configurable
		Cause	The operating hour counter has a backup copy. If the controller's 24 V power supply fails precisely when the operating hour counter is being updated, the written record may be corrupted. In such cases, the controller restores the operating hour counter from the intact backup copy when it switches back on.	
		Measure	Warning only, no further action required.	
28-3	FF04h	**Operating hour counter converted**		configurable
		Cause	Firmware was loaded in which the operating hour counter has a different data format. The next time the controller is switched on, the old operating hour counter record is converted to the new format.	
		Measure	Warning only, no further action required.	

Error group 29		Memory card		
No.	Code	Message		Reaction
29-0	-	**Memory card not available**		configurable
		Cause	This error is triggered in the following cases: – If an action should be carried out on the memory card (load or create DCO file, firmware download), but no memory card is plugged in. – The DIP switch S3 is set to ON, but no card is plugged in after the reset/restart.	
		Measure	Insert appropriate memory card in the slot. Only if expressly desired!	
29-1	-	**Memory card: Initialisation error**		configurable
		Cause	This error is triggered in the following cases: – Memory card could not be initialised. Card type may not be supported! – File system not supported. – Error in connection with the shared memory.	
		Measure	• Check card type used. • Connect memory card to a PC and format again.	

Error group 29		Memory card		
No.	Code	Message		Reaction
29-2	-	**Memory card: Data error**		configurable
		Cause	This error is triggered in the following cases: – A load or storage process is already running, but a new load or storage process is requested. DCO file ›› Servo – The DCO file to be loaded has not been found. – The DCO file to be loaded is not appropriate for the device. – The DCO file to be loaded is defective. – Servo ›› DCO file – The memory card is write-protected. – Other error while saving the parameter set as a DCO file. – Error in creating the file INFO.TXT.	
		Measure	• Execute load or storage procedure again after waiting 5 seconds. • Connect memory card to a PC and check the files included. • Remove write protection from the memory card.	
29-3	-	**Memory card: Write error**		configurable
		Cause	– This error is triggered while saving the DCO file or INFO.TXT file if the memory card is discovered to be already full. – The maximum file index (99) already exists. That is, all file indexes are assigned. No file name can be issued!	
		Measure	• Insert another memory card. • Change file names.	
29-4	-	**Memory card: Firmware download error**		configurable
		Cause	This error is triggered in the following cases: – No firmware file on the memory card. – The firmware file is not appropriate for the device. – Other error during firmware download.	
		Measure	• Connect memory card to PC and transfer firmware file.	

Error group 30		Internal conversion error		
No.	Code	Message		Reaction
30-0	6380h	**Internal conversion error**		PSoff
		Cause	Range exceeded for internal scaling factors, which are dependent on the parameterised controller cycle times.	
		Measure	• Check whether extremely short or extremely long cycle times were set in the parameters.	

Error group 31		I²t monitoring		
No.	Code	Message		Reaction
31-0	2312h	**Motor I²t**		configurable
		Cause	I²t monitoring of the controller has been triggered. – Motor/mechanical system blocked or sluggish. – Motor under-sized?	
		Measure	• Check the performance rating of the drive package.	
31-1	2311h	**Power stage I²t**		configurable
		Cause	The I²t monitoring is being triggered frequently. – Motor controller does not have the required capacity? – Mechanical system sluggish?	
		Measure	• Check design of the motor controller, • if necessary use a more powerful type. • Check the mechanical system.	
31-2	2313h	**PFC I²t**		configurable
		Cause	PFC power rating exceeded.	
		Measure	• Parameterise operation without PFC (FCT).	
31-3	2314h	**Braking resistor I²t**		configurable
		Cause	– Overloading of the internal braking resistor.	
		Measure	• Use external braking resistor. • Reduce resistance value or use resistor with higher pulse load.	

Error group 32		Intermediate circuit fault		
No.	Code	Message		Reaction
32-0	3280h	**Intermediate circuit charging time exceeded**		configurable
		Cause	The intermediate circuit could not be charged after the mains voltage was applied. – A fuse may be faulty, or – an internal braking resistor may be faulty, or, – in the case of operation with an external resistor, that resistor is not connected.	
		Measure	• Check interface to the external braking resistor. • Alternatively, check whether the jumper for the internal braking resistor is in place. If the interface is correct, the internal braking resistor or the built-in fuse is probably faulty. On-site repair is not possible.	
32-1	3281h	**Undervoltage for active PFC**		configurable
		Cause	The PFC cannot be activated at all until an intermediate circuit voltage of about 130 V DC is reached.	
		Measure	• Check the power supply.	

Error group 32		Intermediate circuit fault		
No.	Code	Message		Reaction
32-5	3282h	**Brake chopper overload**		configurable
		Cause	The extent of utilisation of the brake chopper when quick discharge began was already in the range above 100%. Quick discharge took the brake chopper to the maximum load limit and was prevented/aborted.	
		Measure	No action required.	
32-6	3283h	**Intermediate circuit discharge time exceeded**		configurable
		Cause	Intermediate circuit could not be quickly discharged. The internal braking resistor may be faulty or, in the case of operation with an external resistor, that resistor is not connected.	
		Measure	• Check interface to the external braking resistor. • Alternatively, check whether the jumper for the internal braking resistor is in place. If the internal resistor has been activated and the jumper has been set correctly, the internal braking resistor is probably faulty.	
32-7	3284h	**Power supply missing for controller enable**		configurable
		Cause	Controller enable was issued when the intermediate circuit was still in its charging phase after mains voltage was applied and the mains relay was not yet activated. The drive cannot be enabled in this phase, because the drive is not yet firmly connected to the mains (through the mains relay).	
		Measure	• In the application, check whether the mains supply and controller enable signals were sent quickly one after the other.	
32-8	3285h	**Power supply failure during controller enable**		QStop
		Cause	Interruptions / failure in the power supply while the controller enable was activated.	
		Measure	• Check the power supply.	
32-9	3286h	**Phase failure**		QStop
		Cause	Failure of one or more phases (only in the case of three-phase supply).	
		Measure	• Check the power supply.	

Error group 33		Encoder emulation following error		
No.	Code	Message		Reaction
33-0	8A87h	**Encoder emulation following error**		configurable
		Cause	The critical frequency for encoder emulation was exceeded (see manual) and the emulated angle at [X11] was no longer able to follow. Can occur if very high numbers of lines are programmed for [X11] and the drive reaches high speeds.	
		Measure	• Check whether the parameterised number of lines may be too high for the speed being represented. • Reduce the number of lines if necessary.	

Error group 34		Fieldbus synchronisation		
No.	Code	Message		Reaction
34-0	8780h	**No synchronisation via field bus**		configurable
		Cause	When activating the interpolated position mode, the controller could not be synchronised to the fieldbus. – The synchronisation messages from the master may have failed or – the IPO interval is not correctly set to the synchronisation interval of the fieldbus.	
		Measure	• Check the settings for the controller cycle times.	
34-1	8781h	**Field bus synchronisation error**		configurable
		Cause	– Synchronisation via fieldbus messages during ongoing operation (interpolated position mode) has failed. – Synchronisation messages from master failed? – Synchronisation interval (IPO interval) parameterised too small/too large?	
		Measure	• Check the settings for the controller cycle times.	

Error group 35		Linear motor		
No.	Code	Message		Reaction
35-0	8480h	**Linear motor spinning protection**		configurable
		Cause	Encoder signals are faulty. The motor may be racing ("spinning") because the commutation position has been shifted by the faulty encoder signals.	
		Measure	• Check that the installation conforms to the EMC recommendations. • In the case of linear motors with inductive/optical encoders with separately mounted measuring tape and measuring head, check the mechanical clearance. • In the case of linear motors with inductive encoders, make sure that the magnetic field of the magnets or the motor winding does not leak into the measuring head (this effect usually occurs when high accelerations = high motor current).	
35-5	-	**Error during the determination of the commutation position**		configurable
		Cause	The rotor position could not be clearly identified. – The selected method may be inappropriate. – The selected motor current for the identification may not be set appropriately.	
		Measure	• Check the method for determining the commutation position ➔ Additional informationrmation.	
		Additional information	Information about determining commutation position: a) The alignment method is inappropriate for locked or sluggish drives or drives capable of low-frequency oscillation. b) The microstep method is appropriate for air-core and iron-core motors. As only very small movements are carried out, it works even when the drive is on elastic stops or is locked but can still be moved elastically to some extent. Due to the high excitation frequency, however, the method is very susceptible to oscillations in the case of poorly damped drives. In such cases, you can attempt to reduce the excitation current (%). c) The saturation method uses local occurrences of saturation in the iron of the motor. Recommended for locked drives. Air-core drives are by definition not suitable for this method. If the (iron-core) drive moves too much when locating the commutation position, the measurement result may be adulterated. If this is the case, reduce the excitation current. In the opposite case, if the drive does not move, the excitation current may not be strong enough, causing the saturation to be insufficient.	

Error group 36		Parameter		
No.	Code	Message		Reaction
36-0	6320h	**Parameter was limited**		configurable
		Cause	An attempt was made to write a value which was outside the permitted limits, so the value was limited.	
		Measure	• Check the user parameter set.	
36-1	6320h	**Parameter was not accepted**		configurable
		Cause	An attempt was made to write to an object which is "read only" or is not write-capable in the current status (e.g. with controller enable active).	
		Measure	• Check the user parameter set.	

Error group 40		Software limits		
No.	Code	Message		Reaction
40-0	8612h	**Negative software limit reached**		configurable
		Cause	The position setpoint has reached or exceeded the negative software limit switch.	
		Measure	• Check target data. • Check the positioning range.	
40-1	8612h	**Positive software limit reached**		configurable
		Cause	The position setpoint has reached or exceeded the positive software limit switch.	
		Measure	• Check target data. • Check the positioning range.	
40-2	8612h	**Positioning beyond negative software limit suppressed**		configurable
		Cause	Start of a positioning task was suppressed because the target lies behind the negative software limit switch.	
		Measure	• Check target data. • Check the positioning range.	
40-3	8612h	**Positioning beyond positive software limit suppressed**		configurable
		Cause	The start of a positioning task was suppressed because the target lies behind the positive software limit switch.	
		Measure	• Check target data. • Check the positioning range.	

Error group 41		Record sequence		
No.	Code	Message		Reaction
41-0	-	**Record sequence: Synchronisation error**		configurable
		Cause	Start of synchronisation without prior sampling pulse.	
		Measure	• Check parameterisation of the lead section.	

Error group 42		Positioning		
No.	Code	Message		Reaction
42-0	8680h	**Positioning: Drive stops automatically because there is no follow-up positioning**		configurable
		Cause	The positioning target cannot be reached through the positioning or edge conditions options.	
		Measure	• Check the parameterisation of the relevant position sets.	
42-1	8681h	**Positioning: Drive stops as rotation reversal is not allowed**		configurable
		Cause	The positioning target cannot be reached through the positioning or edge conditions options.	
		Measure	• Check the parameterisation of the relevant position sets.	
42-2	8682h	**Positioning: Illegal rotation reversal after "stop"**		configurable
		Cause	The positioning target cannot be reached through the positioning or edge conditions options.	
		Measure	• Check the parameterisation of the relevant position sets.	
42-3	-	**Start positioning rejected: Wrong mode of operation**		configurable
		Cause	Switching of the operating mode by means of the position record was not possible.	
		Measure	• Check the parameterisation of the relevant position sets.	
42-4	-	**Please enforce homing run!**		configurable
		Cause	A normal position record was started, but the drive needs a valid reference position before starting.	
		Measure	• Execute new homing.	
42-5	-	**Rotary axis: Direction of rotation is not allowed**		configurable
		Cause	– The positioning target cannot be reached through the positioning or edge conditions options. – The calculated direction of rotation is not permitted for the modulo positioning in the set mode.	
		Measure	• Check the chosen mode.	
42-9	-	**Error at starting the positioning**		configurable
		Cause	– Acceleration limit value exceeded. – Position record blocked.	
		Measure	• Check parameterisation and sequence control, correct if necessary.	

Error group 43		Hardware limit switch		
No.	Code	Message		Reaction
43-0	8081h	**Limit switch: Negative setpoint value blocked**		configurable
		Cause	Negative hardware limit switch reached.	
		Measure	• Check parameterisation, wiring and limit switches.	

Error group 43		Hardware limit switch		
No.	Code	Message		Reaction
43-1	8082h	**Limit switch: Positive setpoint value blocked**		configurable
		Cause	Positive hardware limit switch reached.	
		Measure	• Check parameterisation, wiring and limit switches.	
43-2	8083h	**Limit switch: Positioning suppressed**		configurable
		Cause	– The drive has left the designated range of motion. – Technical defect in the system?	
		Measure	• Check the designated range of motion.	

Error group 44		Cam disc error		
No.	Code	Message		Reaction
44-0	-	**Error in Cam data tables**		configurable
		Cause	The cam disc to be started is not available.	
		Measure	• Check transferred cam disc no. • Correct parameterisation. • Correct programming.	
44-1	-	**Cam Disc: General error homing**		configurable
		Cause	– Start of a cam disc, but the drive is not yet referenced.	
		Measure	• Carry out homing.	
		Cause	– Start homing with active cam disk.	
		Measure	• Deactivate cam disc. Then restart cam disc, if necessary.	

Error group 47		Setting-up		
No.	Code	Message		Reaction
47-0	-	**Timeout setup mode**		configurable
		Cause	Failed to fall below the speed required for setting-up within time allowed.	
		Measure	Check processing of the request on the control side.	

Error group 48		Homing required		
No.	Code	Message		Reaction
48-0	-	**Please enforce homing run!**		QStop
		Cause	An attempt is being made to switch to the speed control or torque control operating mode or to issue the controller enable in one of these operating modes, although the drive requires a valid reference position for this.	
		Measure	• Carry out homing.	

Error group 49		DCO file		
No.	Code	Message		Reaction
49-1	-	**DCO file: wrong password**		QStop
		Cause	– Parameter file with wrong password shall be loaded. – Old parameter file (no password defined) should be loaded in protected motor controller.	
		Measure	Loading only possible with valid password.	

Error group 50		CAN communication		
No.	Code	Message		Reaction
50-0	-	**Too many synchronous PDOs**		configurable
		Cause	More PDOs have been activated than can be processed in the underlying SYNC interval. This message also appears if only one PDO is to be transmitted synchronously, but a high number of other PDOs with a different transmission type have been activated.	
		Measure	• Check the activation of PDOs. If the configuration is appropriate, the warning can be suppressed using error management. • Extend the synchronisation interval.	
50-1	-	**SDO error has occurred**		configurable
		Cause	An SDO transfer has caused an SDO abort. – Data exceed the range of values. – Access to non-existent object.	
		Measure	• Check the command sent.	

Error group 51		Safety module/function		
No.	Code	Message		Reaction
51-0	8091h	**Unknown Safety module or driver supply defective**		PSoff
		Cause	CMMP-AS-...-M0: Internal voltage error of the STO circuit.	
		Measure	• Protection circuit defective. No action possible, please contact Festo. If possible, replace with another motor controller.	
		Cause	CMMP-AS-...-M3: Internal voltage error of the safety module or micro switch module.	
		Measure	• Module presumably defective. If possible, replace with another basic unit.	
		Cause	CMMP-AS-...-M3: No safety module detected or unknown module type.	
		Measure	• Install suitable safety or micro switch module for the firmware and hardware. • Load a firmware suitable for the safety or micro switch module, see type designation on the module.	
51-2	8093h	**Safety module: Dissimilar module type**		PSoff
		Cause	Type or version of the module does not fit the design.	
		Measure	• Check whether correct module type and correct version are being used. • With module replacement: module type not yet designed. Accept currently integrated safety or micro switch module.	
51-3	8094h	**Safety module: Dissimilar module version**		PSoff
		Cause	Module type or revision are not supported.	
		Measure	• Mount a module that is compatible to the given hardware and firmware. • Load firmware that is appropriate for the module, see type designation on the module.	
		Cause	The module type is correct but the module version is not supported by the basic unit.	
		Measure	• Check module version; if possible use module of same version after replacement. Install suitable safety or micro switch module for the firmware and hardware. • If only a module with a more recent version is available: Load firmware that is appropriate for the module, see type designation on the module.	

Error group 51		Safety module/function		
No.	Code	Message		Reaction
51-4	8095h	**Safety module: SSIO communication error**		PSoff
		Cause	Fault in the internal communication connection between the basic unit and the safety module.	
		Measure	• This error may occur if a CAMC-G-S3 was designed into the basic unit but a different module type was plugged in. • Load a firmware suitable for the safety or micro switch module, see type designation on the module.	
51-5	8096h	**Safety module: Brake control error**		PSoff
		Cause	Internal hardware error (brake actuation control signals) of the safety module or micro switch module.	
		Measure	• Module presumably defective. If possible, replace with another module.	
		Cause	Error in brake driver circuit section in the basic unit.	
		Measure	• Module presumably defective. If possible, replace with another basic unit.	
51-6	8097h	**Safety module: Dissimilar serial number**		PSoff
		Cause	Serial number of currently connected safety module is different from the stored serial number.	
		Measure	Error only occurs after replacement of the CAMC-G-S3. • With module replacement: module type not yet designed. Accept currently integrated CAMC-G-S3.	

Error group 52		Safety function		
No.	Code	Message		Reaction
52-1	8099h	**Safety function: Discrepancy time expired**		PSoff
		Cause	– Control ports STO-A and STO-B are not actuated simultaneously.	
		Measure	• Check discrepancy time.	
		Cause	– Control ports STO-A and STO-B are not wired in the same way.	
		Measure	• Check discrepancy time.	
		Cause	Upper and lower switch supply not simultaneously activated (discrepancy time exceeded) – Error in control / external circuitry of safety module. – Error in safety module.	
		Measure	• Check circuitry of the safety module – are the inputs STO-A and STO-B switched off on two channels and simultaneously? • Replace safety module if you suspect it is faulty.	

Error group 52		Safety function			
No.	Code	Message			Reaction
52-2	809Ah	**Safety function: Failure of driver supply with active PWM control**			PSoff
		Cause	This error message does not occur with devices delivered from the factory. It can occur with use of a user-specific device firmware.		
		Measure	• The safe status was requested with enabled power output stage. Check inclusion in the safety-oriented interface.		
52-3	809Bh	**Safety module: Overlapping velocity limits in basic unit**			PSoff
		Cause	– Basic unit reports error if the currently requested direction of movement is not possible because the safety module has blocked the setpoint value in this direction.		
		Measure	Error may occur in connection with the SSF if an asymmetrical speed window is used where one limit is set to zero. In this case, the error occurs when the basic unit moves in the "blocked" direction in the Positioning mode. • Check application and change if necessary.		

Error group 53		Violation of Safety conditions (only CMMP-AS-...-M3)			
No.	Code	Message			Reaction
53-0	80A1h	**USF0: Safety condition violated**			configurable
		Cause	– Violation of monitored speed limits of the SSF0 in operation / when USF0 / SSF0 requested.		
		Measure	Check when the violation of the safety condition occurs: a) During dynamic braking to the safe speed b) After the drive has reached the safe speed. • With a) Critical check of braking ramp – record trace - can the drive follow the ramp? • Change parameters for the braking ramp or start time / delay times for monitoring. • With b) Check how far the current speed is from the monitored limit speed; increase distance if necessary (parameter in safety module) or correct speed specified by controller.		
53-1	80A2h	**USF1: Safety condition violated**			configurable
		Cause	– Violation of monitored speed limits of the SSF1 in operation / when USF1 / SSF1 requested.		
		Measure	• See USF0, error 53-0.		
53-2	80A3h	**USF2: Safety condition violated**			configurable
		Cause	– Violation of monitored speed limits of the SSF2 in operation / when USF2 / SSF2 requested.		
		Measure	• See USF0, error 53-0.		

Error group 53		Violation of Safety conditions (only CMMP-AS-...-M3)		
No.	Code	Message		Reaction
53-3	80A4h	**USF3: Safety condition violated**		configurable
		Cause	– Violation of monitored speed limits of the SSF3 in operation / when USF3 / SSF3 requested.	
		Measure	• See USF0, error 53-0.	

Error group 54		Violation of Safety conditions (only CMMP-AS-...-M3)		
No.	Code	Message		Reaction
54-0	80AAh	**SBC: Safety condition violated**		configurable
		Cause	– Brake should engage; no feedback received within the expected time.	
		Measure	• Check how the feedback signal is configured – was the correct input selected for the feedback signal? • Does the feedback signal have the correct polarity? • Check whether the feedback signal is actually switching. • Is the parameterised delay time for the evaluation of the feedback signal appropriate to the brake used (measure switching time if necessary)?	
54-2	80ACh	**SS2: Safety condition violated**		configurable
		Cause	– Actual speed outside permitted limits for too long.	
		Measure	Check when the violation of the safety condition occurs: a) During dynamic braking to zero. b) After the drive has reached zero speed. • With a) Critical check of braking ramp – record trace - can the drive follow the ramp? Change parameters for the braking ramp or start time / delay times for monitoring. • With a) If the option "Trigger basic unit quick stop" is activated: Critical check of the basic unit's quick stop ramp. • With b) Check whether the drive continues to oscillate after reaching the zero speed or remains still and stable – increase monitoring tolerance time if necessary. • With b) If the actual speed value is very noisy at rest. Check and if necessary adjust expert parameters for speed recording and detection of standstill.	

Error group 54		Violation of Safety conditions (only CMMP-AS-...-M3)		
No.	Code	Message		Reaction
54-3	80ADh	**SOS: Safety condition violated**		configurable
		Cause	– Angle encoder evaluation reports "Motor running" (actual speed exceeds limit). – Drive has rotated out of its position since reaching the safe state.	
		Measure	• Check position tolerance for the SOS monitoring and increase if necessary, if this is permissible. • If the actual speed value is very noisy when at rest: Check and if necessary adjust expert parameters for speed recording and detection of standstill.	
54-4	80AEh	**SS1: Safety condition violated**		configurable
		Cause	– Actual speed outside permitted limits for too long.	
		Measure	Check when the violation of the safety condition occurs: a) During dynamic braking to zero. b) After the drive has reached zero speed. • With a) Critical check of braking ramp – record trace - can the drive follow the ramp? Change parameters for the braking ramp or start time / delay times for monitoring. • With a) If the option "Trigger basic unit quick stop" is activated: Critical check of the basic unit's quick stop ramp. • With b) Check whether the drive continues to oscillate after reaching the zero speed or remains still and stable – increase monitoring tolerance time if necessary. • With b) If the actual speed value is very noisy when at rest: Check and if necessary adjust expert parameters for speed recording and detection of standstill.	
54-5	80AFh	**STO: Safety condition violated**		configurable
		Cause	– Internal hardware error (voltage error) of the safety module.	
		Measure	• Module presumably defective. If possible, replace with another module.	
		Cause	– Error in driver circuit section in the basic unit.	
		Measure	• Module presumably defective. If possible, replace with another basic unit.	
		Cause	– No feedback received from basic unit to indicate that output stage was switched off.	
		Measure	• Check whether the error can be acknowledged and whether it occurs again upon a new STO request – if yes: basic unit is presumably faulty. If possible, replace with another basic unit.	

Error group 54		Violation of Safety conditions (only CMMP-AS-...-M3)		
No.	Code	Message		Reaction
54-6	80B0h	**SBC: Brake not released for › 24h**		configurable
		Cause	– Error occurs when SBC is requested and the brake has not been opened by the basic unit in the last 24 hours.	
		Measure	• If the brake is actuated via the brake driver in the basic unit [X6]: The brake must be energised at least once within 24 V before the SBC request because the circuit breaker check can only be performed when the brake is switched on (energised). • Only if brake control takes place via DOUT4x and an external brake controller: Deactivate 24h monitoring in the SBC parameters if the external brake controller allows this.	
54-7	80B1h	**SOS: SOS requested for › 24 h**		configurable
		Cause	– If SOS is requested for more than 24 hours, the error is triggered.	
		Measure	• Terminate SOS occasionally; move axis once occasionally.	

Error group 55		Measuring of actual value 1 (only CMMP-AS-...-M3)		
No.	Code	Message		Reaction
55-0	80C1h	**No actual speed / position value available or standstill for › 24 h**		configurable
		Cause	– Subsequent error when a position encoder fails. – Safety function SSF, SS1, SS2 or SOS requested and actual speed value is not valid.	
		Measure	• Check the function of the position encoder(s) (see following error).	
55-1	80C2h	**SINCOS encoder [X2B] - signal error**		configurable
		Cause	– Vector length $\sin^2+\cos^2$ is outside the permissible range. – The amplitude of one of the two signals is outside the permissible range. – Offset between analogue and digital signal is greater than 1 quadrant.	
		Measure	Error may occur with SIN/COS and Hiperface encoders. • Check the position encoder. • Check the connection wiring (broken wire, short between two signals or signal / screening). • Check the supply voltage for the position encoder. • Check the motor cable / screening on motor and drive side – EMC problems may trigger the error.	

Error group 55		Measuring of actual value 1 (only CMMP-AS-...-M3)		
No.	Code	Message		Reaction
55-2	80C3h	**SINCOS encoder [X2B] - standstill > 24 h**		configurable
		Cause	– Input signals of the SinCos encoder have not changed by a minimum amount for 24 hours (when safety function is requested).	
		Measure	• Terminate SS1, SS2 or SOS occasionally; move axis once occasionally.	
55-3	80C4h	**Resolver [X2A] - signal error**		configurable
		Cause	– Vector length sin^2+cos^2 is outside the permissible range. – The amplitude of one of the two signals is outside the permissible range. – Input signal is static (same values to right and left of maximum).	
		Measure	• Check the resolver. • Check the connection wiring (broken wire, short between two signals or signal / screening). • Check for failure of the exciter signal • Check the motor cable / screening on motor and drive side – EMC problems may trigger the error.	
55-4	-	**EnDat encoder [X2B] - sensor error**		configurable
		Cause	– Communication error between safety module and the ENDAT encoder. – Error message of the ENDAT encoder present.	
		Measure	• Check the ENDAT encoder. • Check the connection wiring (broken wire, short between two signals or signal / screening). • Check the supply voltage for the ENDAT encoder. • Check of the motor cable / screening on motor and drive side – EMC problems may trigger the error.	
55-5	-	**EnDat encoder [X2B] - wrong sensor / type**		configurable
		Cause	– Number of lines does not correspond to parameterisation. – Serial no. Does not correspond to parameterisation. – Sensor type does not correspond to parameterisation.	
		Measure	• Check the parameterisation. • Use only approved encoders.	
55-6	80C5h	**Incremental encoder X10 - signal error**		configurable
		Cause	– Signal error at incremental encoder.	
		Measure	• Check the connection wiring (broken wire, short between two signals or signal / screening). • Check the motor cable / screening on motor and drive side – EMC problems may trigger the error.	

Error group 55		Measuring of actual value 1 (only CMMP-AS-...-M3)		
No.	Code	Message		Reaction
55-7	80C6h	**Other encoder [X2B] - Faulty angle information**		configurable
		Cause	– "Angle faulty" message is sent from basic unit when status lasts for longer than the allowed time. – Encoder at X2B is evaluated by the basic unit, – encoder is faulty.	
		Measure	• Check the position encoder at X2B. • Check the connection wiring (broken wire, short between two signals or signal / screening). • Check the supply voltage for the ENDAT encoder. • Check the motor cable / screening on motor and drive side – EMC problems may trigger the error?	
55-8	-	**Impermissible acceleration detected**		configurable
		Cause	– Encoder error. – EMC problems may trigger the error. – Too high acceleration values. – Max. acceleration is parameterised too low. – Snap angle after homing in the transmitted data from the base unit to the safety module.	
		Measure	• Check the connection wiring (broken wire, short between two signals or signal / screening). • Check the target values given by PLC for invalid acceleration values (P06.07)? • Check the parameterised max. values for correctness. The upper limit (P06.07) should be at least 30...50% above the max. process values. • With snap angle in the data from the base device: Acknowledge it one times.	

Error group 56		Measuring of actual value 2 (only CMMP-AS-...-M3)		
No.	Code	Message		Reaction
56-8	80D1h	**Speed / angle difference encoder 1 - 2**		configurable
		Cause	– Speed difference between encoders 1 and 2 of one µC for longer than allowed time outside the permissible range. – Angle difference between encoders 1 and 2 of one µC for longer than allowed time outside the permissible range.	
		Measure	• Problem may occur if two position encoders are used in the system and they are not "rigidly coupled". • Check for elasticity or looseness, improve mechanical system. • Adjust the expert parameters for the position comparison if this is acceptable from an application point of view.	

Error group 56		Measuring of actual value 2 (only CMMP-AS-...-M3)		
No.	Code	Message		Reaction
56-9	-	**Error Cross comparison encoder evaluation**		configurable
		Cause	Cross-comparison between µC1 and µC2 has detected an angle difference or speed difference or difference in capture times for the position encoders.	
		Measure	• Timing disrupted. If the error occurs against after a reset, the safety module is presumably faulty.	

Error group 57		Input/output error (only CMMP-AS-...-M3)		
No.	Code	Message		Reaction
57-0	80E1h	**Self test I/O error (internal/external)**		configurable
		Cause	– Error at outputs DOUT40 ... DOUT42 (detection by test pulses). – Internal error of digital inputs DIN40 ... DIN49 (via internal test signals). – Error at brake output at X6 (signalling, detection by test pulses). – Internal error of brake output (via internal test signals). – Internal error of digital outputs DOUT40 – DOUT42 (via internal test signals).	
		Measure	• Check the connection wiring for the digital outputs DOUT40 ... DOUT42 (short circuit, cross circuit, etc.). • Check the connection wiring for the brake (short circuit, cross circuit, etc.). • Brake connection: The error may occur with longer motor cables if: 1. The brake output X6 was configured for the brake (this is the case with factory settings!) and 2. A motor without a holding brake is used and the brake connection lines in the motor cable are terminated at X6. In this case: Disconnect the brake connection lines at X6. • If there is not error in the connection wiring, there may be an internal error in the module (check by swapping the module).	

Error group 57		Input/output error (only CMMP-AS-...-M3)		
No.	Code	Message		Reaction
57-1	80E2h	**Digital inputs - wrong signal level**		configurable
		Cause	Exceeding / violation of discrepancy time with multi-channel inputs (DIN40 ... DIN43, two-handed control device, mode selector switch).	
		Measure	• Check the external active and passive sensors – do they switch on two channels and simultaneously (within the parameterised discrepancy time). • Two-handed control device: Check how the device is operated by the user – are both pushbuttons pressed within the discrepancy time? Give training if necessary. • Check the set discrepancy times – are they sufficient?	
57-2	-	**Digital inputs - missing test pulse**		configurable
		Cause	– One or more inputs (DIN40 ... DIN49) were configured for the evaluation of test pulses from the outputs (DOUT40 ... DOUT 42). The test pulses from DOUTx do not arrive at DIN4x.	
		Measure	• Check the wiring (shorts after 0 V, 24 V, cross circuits). • Check the assignment – correct output selected / configured for test pulse?	
57-6	-	**Electronic temperature too high**		configurable
		Cause	– The safety module's temperature monitor has been triggered; the temperature of µC1 or µC2 was below -20° or above +75°C.	
		Measure	• Check the operating conditions (ambient temperature, control cabinet temperature, installation situation in the control cabinet). • If the motor controller is experiencing high thermal load (high control cabinet temperature, high power consumption / output to motor, large number of occupied slots), a motor controller of the next highest output level should be used.	

Error group 58		Error during communication / parameterisation (only CMMP-AS-...-M3)		
No.	Code	Message		Reaction
58-0	80E9h	**Plausibility check parameters**		configurable
		Cause	The plausibility check in the safety module produced errors, e.g. an invalid angle encoder configuration; the error is triggered when a validation code is requested by the SafetyTool and when parameters are backed up in the safety module.	
		Measure	• Note instructions for SafetyTool for complete validation; critically check parameterisation.	

Error group 58		Error during communication / parameterisation (only CMMP-AS-...-M3)		
No.	Code	Message		Reaction
58-1	-	**General error parameterisation**		configurable
		Cause	Parameterisation session for more than 8 h active. The safety module aborted the parameterisation session. The error message is stored in the diagnostic memory.	
		Measure	• Finish the parameterisation session before the 8 h limit or break and restart the session.	
58-4	80E9h	**Buffer internal communication**		configurable
		Cause	– Communication connection faulty. – Timeout / data error / incorrect sequence (packet counter) in data transmission between the basic unit and safety module. – Too much data traffic, new requests are being sent to safety module before old ones have been responded to.	
		Measure	• Check communication interfaces, wiring, screening, etc. • Check whether other devices have read access to the motor controller and safety module during a parameterisation session - this may overload the communication connection. • Check whether the firmware versions of the safety module and basic unit and the versions of the FCT plugin and SafetyTool are compatible.	
58-5	80EAh	**Communication safety module - base unit**		configurable
		Cause	– Packet counter error during transmission µC1 <-➔ µC2. – Checksum error during transmission µC1 <-➔ µC2.	
		Measure	• Internal malfunction in the motor controller. • Check whether the firmware versions of the safety module and basic unit and the versions of the FCT plugin and SafetyTool are compatible.	

Error group 58		Error during communication / parameterisation (only CMMP-AS-...-M3)		
No.	Code	Message		Reaction
58-6	80EBh	**Cross comparison error processor 1 - 2**		configurable
		Cause	Timeout during cross-comparison (no data) or cross-comparison faulty (data for µC1 and µC2 are different). – Error in cross-comparison for digital IO. – Error in cross-comparison for analogue input. – Error in cross-comparison for internal operating voltage measurement (5 V, 3.3 V, 24 V) and reference voltage (2.5 V). – Error in cross-comparison for SIN/COS angle encoder analogue values. – Error in cross-comparison for programme sequence monitoring. – Error in cross-comparison for interrupt counter. – Error in cross-comparison for input map. – Error in cross-comparison for violation of safety conditions. – Error in cross-comparison for temperature measurement.	
		Measure	This is an internal error in the module that should not occur during operation. • Check the operating conditions (temperature, air humidity, condensation). • Check the EMC – wiring as specified, screening concept, are there any external interference sources? • Safety module may be faulty – is error resolved after replacing the module? • Check whether a new firmware for the motor controller or a new version of the safety module is available from the manufacturer.	

Error group 59		Internal safety module error (only CMMP-AS-...-M3)		
No.	Code	Message		Reaction
59-1	80F1h	**Failsafe supply/safe pulse inhibitor**		configurable
		Cause	– Internal error in module in failsafe supply circuit section or in the driver supply for the upper and lower switches.	
		Measure	• Module faulty, replace.	
59-2	80F2h	**External voltage supply error**		configurable
		Cause	– Reference voltage 2.5V outside tolerance. – Logic supply overvoltage +24 V detected.	
		Measure	• Module faulty, replace.	
59-3	80F3h	**Internal voltage supply error**		configurable
		Cause	– Voltage (internal 3.3 V, 5 V, ADU reference) outside the permissible range.	
		Measure	• Module faulty, replace.	

Error group 59		Internal safety module error (only CMMP-AS-...-M3)		
No.	Code	Message		Reaction
59-4	80F4h	**Error management: Too many errors**		configurable
		Cause	– Too many errors have occurred simultaneously.	
		Measure	• Clarify: What is the status of the installed safety module - does it contain a valid parameter set? • Read out and analyse the log file of the basic unit via FCT. • Remedy causes of error step by step. • Install safety module with "delivery status" and perform commissioning of basic unit. • If this is not available: Set factory settings in the safety module, then copy data from the basic unit and perform complete validation. Check whether the error occurs again.	
59-5	80F5h	**Diagnosis Memory writing error**		configurable
		Cause	Subsequent error if internal communication is disrupted. – Basic unit not ready for operation, faulty or memory error.	
		Measure	• Check the function of the basic unit • Generate an error in the basic unit, e.g. by unplugging the position encoder, and check whether the basic unit writes an entry to the log file. • Module or basic unit faulty; replace.	
59-6	80F6h	**Error on saving parameter set**		configurable
		Cause	– Voltage interruption / power off while parameters were being saved.	
		Measure	• Maintain a voltage supply of 24 V throughout the parameterisation session. • Once the error has occurred, parameterise the module again and validate the parameter set again.	
59-7	80F7h	**FLASH checksum error**		configurable
		Cause	– Voltage interruption / power off while parameters were being saved. – Flash memory in safety module corrupted (e.g. by extreme malfunctions).	
		Measure	Check whether the error recurs after a reset. If it does: • Parameterise the module again and validate the parameter set again. If the error remains: • Module is faulty; replace.	

Error group 59		Internal safety module error (only CMMP-AS-...-M3)		
No.	Code	Message		Reaction
59-8	80F8h	**Internal monitoring processor 1 - 2**		configurable
		Cause	– Serious internal error in the safety module: Error detected while dynamising internal signals – Disrupted programme sequence, stack error or OP code test failed, processor exception / interrupt.	
		Measure	Check whether the error recurs after a reset. If it does: • Module is faulty; replace.	
59-9	80F9h	**Other unexpected error**		configurable
		Cause	Triggering of internal programme sequence monitoring.	
		Measure	• Check the firmware version of the basic unit and the version of the safety module – update available? • Safety module faulty; replace.	

Error group 62		EtherCAT (only CMMP-AS-...-M3)		
No.	Code	Message		Reaction
62-0	-	**EtherCAT: Initialisation error**		configurable
		Cause	No EtherCAT bus present.	
		Measure	• Switch on the EtherCAT master. • Check the wiring.	
62-1	-	**EtherCAT: Initialisation error**		configurable
		Cause	Error in the hardware.	
		Measure	• Replace the interface and send it to the manufacturer for inspection.	
62-2	-	**EtherCAT: Protocol error**		configurable
		Cause	CAN over EtherCAT is not in use.	
		Measure	• Incorrect protocol. • EtherCAT bus wiring fault.	
62-3	-	**EtherCAT: Invalid RPDO length**		configurable
		Cause	Sync manager 2 buffer size is too large.	
		Measure	• Check the RPDO configuration of the motor controller and the higher-level control system.	
62-4	-	**EtherCAT: Invalid TPDO length**		configurable
		Cause	Sync manager 3 buffer size is too large.	
		Measure	• Check the TPDO configuration of the motor controller and the higher-level control system.	
62-5	-	**EtherCAT: Erroneous cyclic communication**		configurable
		Cause	Emergency shut-down due to failure of cyclic data transmission.	
		Measure	• Check the configuration of the master. Synchronous transmission is unstable.	

Error group 63		EtherCAT (only CMMP-AS-...-M3)		
No.	Code	Message		Reaction
63-0	-	**EtherCAT: Defective module**		configurable
		Cause	Error in the hardware.	
		Measure	• Replace the interface and send it to the manufacturer for inspection.	
63-1	-	**EtherCAT: Invalid data**		configurable
		Cause	Faulty telegram type.	
		Measure	• Check the wiring.	
63-2	-	**EtherCAT: TPDO data has not been read**		configurable
		Cause	The buffer for sending the data is full.	
		Measure	The data was sent faster than the motor controller could process it. • Reduce the cycle time on the EtherCAT bus.	
63-3	-	**EtherCAT: No distributed clocks active**		configurable
		Cause	Warning: Firmware is synchronising with the telegram, not with the distributed clocks system. When the EtherCAT was started, no hardware SYNC (distributed clocks) was found. The firmware now synchronises with the EtherCAT frame.	
		Measure	• If necessary, check whether the master supports the distributed clocks feature. • Otherwise: Ensure that the EtherCAT frames are not interrupted by other frames if the Interpolated Position Mode is to be used.	
63-4	-	**EtherCAT: Missing SYNC message in IPO cycle**		configurable
		Cause	Telegrams are not being sent in the time slot pattern of the IPO.	
		Measure	• Check responsible participant for distributed clocks.	

Error group 64		DeviceNet (only CMMP-AS-...-M3)		
No.	Code	Message		Reaction
64-0	-	**DeviceNet: Duplicate MAC ID**		configurable
		Cause	The duplicate MAC-ID check has found two nodes with the same MAC-ID.	
		Measure	• Change the MAC-ID of one node to an unused value.	
64-1	-	**DeviceNet: Bus power lost**		configurable
		Cause	The DeviceNet interface is not supplied with 24 V DC.	
		Measure	• In addition to the motor controller, the DeviceNet interface must also be connected to 24 V DC.	
64-2	-	**DeviceNet: RX queue overflow**		configurable
		Cause	Too many messages received within a short period.	
		Measure	• Reduce the scan rate.	

Error group 64		DeviceNet (only CMMP-AS-...-M3)		
No.	Code	Message		Reaction
64-3	-	**DeviceNet: TX queue overflow**		configurable
		Cause	Insufficient free space on the CAN bus for sending messages.	
		Measure	• Increase the baud rate. • Reduce the number of nodes. • Reduce the scan rate.	
64-4	-	**DeviceNet: IO message not sent**		configurable
		Cause	Error sending I/O data.	
		Measure	• Check that the network is connected correctly and has no faults.	
64-5	-	**DeviceNet: Bus OFF**		configurable
		Cause	The CAN controller is BUS OFF.	
		Measure	• Check that the network is connected correctly and has no faults.	
64-6	-	**DeviceNet: CAN controller overflow**		configurable
		Cause	The CAN controller has an overflow.	
		Measure	• Increase the baud rate. • Reduce the number of nodes. • Reduce the scan rate.	

Error group 65		DeviceNet (only CMMP-AS-...-M3)		
No.	Code	Message		Reaction
65-0	-	**DeviceNet active, but no module**		configurable
		Cause	The DeviceNet communication is activated in the parameter set of the motor controller, but no interface is available.	
		Measure	• Deactivate DeviceNet communication. • Connect an interface.	
65-1	-	**Timeout IO connection**		configurable
		Cause	Interruption of an I/O connection.	
		Measure	• No I/O message was received within the expected time.	

Error group 66		Modbus/TCP		
No.	Code	Message		Reaction
66-0	-	**Modbus/TCP: No free TCP/IP instances**		Warn
		Cause	Ethernet stack can download the requested TCP connection does not provide. Internal device error.	
		Measure	• Restart device or restore factory settings. • If the error occurs lasting effect on the HW is defective. Can not be repaired on site.	

Error group 67		Modbus/TCP		
No.	Code	Message		Reaction
67-0	-	**Modbus/TCP: Timeout TCP/IP**		configurable
		Cause	Existing TCP connection between the host and the controller has been disconnected.	
		Measure	• Ethernet cable connected correctly? Host switched off or not reachable?	
67-1	-	**Modbus/TCP: Timeout Modbus TCP/IP**		configurable
		Cause	TCP connection between host and controller still exists, but the host does not send any more data.	
		Measure	• Crashed host?	
67-2	-	**Modbus/TCP: Buffer overflow**		configurable
		Cause	Internal buffer for editing the data is full. Data sent from the host faster than the controller can process it.	
		Measure	• Reduce update time of the host.	
67-3	-	**Modbus/TCP: Telegram length too short**		configurable
		Cause	The data transmitted from the host data is too long. Host sends less data than expected by the controller.	
		Measure	• Correct data length in the host.	
67-4	-	**Modbus/TCP: Telegram length too long**		configurable
		Cause	The data transmitted from the host data is too long. Host sends more data than expected by the controller.	
		Measure	• Correct data length in the host.	

Error group 68		EtherNet/IP (only CMMP-AS-...-M3)		
No.	Code	Message		Reaction
68-0	-	**EtherNet/IP: Serious fault**		configurable
		Cause	A serious internal error has occurred. It can be triggered by a defective interface, for example.	
		Measure	• Try to acknowledge the error. • Carry out a reset. • Replace the interface. • If the error continues, contact Technical Support.	
68-1	-	**EtherNet/IP: General communication fault**		configurable
		Cause	A serious error was detected in the EtherNet/IP interface.	
		Measure	• Try to acknowledge the error. • Carry out a reset. • Replace the interface. • If the error continues, contact Technical Support.	
68-2	-	**EtherNet/IP: Connection closed**		configurable
		Cause	The connection was closed via the controller.	
		Measure	A new connection to the controller must be established.	

Error group 68		EtherNet/IP (only CMMP-AS-...-M3)		
No.	Code	Message		Reaction
68-3	-	**EtherNet/IP: Connection aborted**		configurable
		Cause	A connection interruption occurred during operation.	
		Measure	• Check the cabling between the motor controller and the higher-level control system. • Establish a new connection to the control system.	
68-4	-	**EtherNet/IP: Duplicate network address**		configurable
		Cause	At least one device with the same IP address exists in the network.	
		Measure	• Use unique IP addresses for all devices in the network.	

Error group 69		EtherNet/IP (only CMMP-AS-...-M3)		
No.	Code	Message		Reaction
69-0	-	**EtherNet/IP: Minor fault**		configurable
		Cause	A minor error was detected in the EtherNet/IP interface.	
		Measure	• Try to acknowledge the error. • Carry out a reset.	
69-1	-	**EtherNet/IP: Incorrect IP configuration**		configurable
		Cause	An incorrect IP configuration has been detected.	
		Measure	• Correct the IP configuration.	
69-2	-	**EtherNet/IP: Field bus module not found**		configurable
		Cause	There is no EtherNet/IP interface in the slot.	
		Measure	• Please check whether an EtherNet/IP interface is in slot Ext2.	
69-3	-	**EtherNet/IP: Module version not supported**		configurable
		Cause	There is an EtherNet/IP interface with incompatible version in the slot.	
		Measure	• Carry out a firmware update to the most up-to-date motor controller firmware.	

Error group 70		FHPP protocol		
No.	Code	Message		Reaction
70-1	-	**FHPP: Mathematical error**		configurable
		Cause	Overrun/underrun or division by zero during calculation of cyclic data.	
		Measure	• Check the cyclic data. • Check the factor group.	
70-2	-	**FHPP: Factor group invalid**		configurable
		Cause	Calculation of the factor group leads to invalid values.	
		Measure	• Check the factor group.	

Error group 70		**FHPP protocol**		
No.	Code	Message		Reaction
70-3	-	**FHPP: Invalid operating mode change**		configurable
		Cause	Changing from the current to the desired operating mode is not permitted. – Error occurs when the OPM bits in the status S5 'Reaction to fault' or S4 'Operation enabled' are changed. – Exception: In the status SA1 'Ready', the change between 'Record select' and 'Direct Mode' is permissible.	
		Measure	• Check your application. It may be that not every change is permissible.	

Error group 71		**FHPP protocol**		
No.	Code	Message		Reaction
71-1	-	**FHPP: Wrong receive telegram length**		configurable
		Cause	Too little data is being transmitted by the control system (data length too small).	
		Measure	• Check the data length parameterised in the control system for the controller's receive telegram. • Check the configured data length in the FHPP+ Editor of the FCT.	
71-2	-	**FHPP: Wrong response telegram length**		configurable
		Cause	Too much data is to be transmitted from the motor controller to the control system (data length too large).	
		Measure	• Check the data length parameterised in the control system for the controller's receive telegram. • Check the configured data length in the FHPP+ Editor of the FCT.	

Error group 72		**PROFINET (only CMMP-AS-...-M3)**		
No.	Code	Message		Reaction
72-0	-	**PROFINET: Initialising error**		configurable
		Cause	Interface presumably includes an incompatible stack version or is faulty.	
		Measure	• Replace interface.	
72-1	-	**PROFINET: Bus error**		configurable
		Cause	No communication possible (e.g. line removed).	
		Measure	• Check the wiring • Restart PROFINET communication.	

Error group 72		PROFINET (only CMMP-AS-...-M3)		
No.	Code	Message		Reaction
72-3	-	**PROFINET: Invalid IP configuration**		configurable
		Cause	An invalid IP configuration was entered in the interface. The interface cannot start with this configuration.	
		Measure	• Parameterise a permissible IP configuration via FCT.	
72-4	-	**PROFINET: Invalid Device name**		configurable
		Cause	A PROFINET device name was assigned with which the controller cannot communicate with the PROFINET (character specification from PROFINET standard).	
		Measure	• Parameterise a permissible PROFINET device name via FCT.	
72-5	-	**PROFINET: Module faulty**		configurable
		Cause	Interface CAMC-F-PN faulty.	
		Measure	• Replace interface.	
72-6	-	**PROFINET: Indication invalid/not supported**		configurable
		Cause	A message was issued by the PROFINET interface that is not supported by the motor controller.	
		Measure	• Please contact Technical Support.	

Error group 73		PROFINET (only CMMP-AS-...-M3)		
No.	Code	Message		Reaction
73-0	-	**PROFIenergy: State not possible**		configurable
		Cause	An attempt was made in a positioning motion to place the controller in the energy-saving status. This is only possible at rest. The drive does not take on the status and continues to travel.	
		Measure	–	

Error group 78		NRT communication (only CMMP-AS-...-M3)		
No.	Code	Message		Reaction
78-0	-	**NRT frame can't be send**		configurable
		Cause	NRT Frame can't be send because of too much bus load.	
		Measure	• Switch off or disconnect other bus devices during parametrisation.	

Error group 80		IRQ overflow		
No.	Code	Message		Reaction
80-0	F080h	**Overflow current controller IRQ**		PSoff
		Cause	The process data could not be calculated in the set current/speed/position interpolator cycle.	
		Measure	• Please contact Technical Support.	

Error group 80		IRQ overflow		
No.	Code	Message		Reaction
80-1	F081h	**Overflow speed controller IRQ**		PSoff
		Cause	The process data could not be calculated in the set current/speed/ position interpolator cycle.	
		Measure	• Please contact Technical Support.	
80-2	F082h	**Overflow position controller IRQ**		PSoff
		Cause	The process data could not be calculated in the set current/speed/ position interpolator cycle.	
		Measure	• Please contact Technical Support.	
80-3	F083h	**Overflow interpolator IRQ**		PSoff
		Cause	The process data could not be calculated in the set current/speed/ position interpolator cycle.	
		Measure	• Please contact Technical Support.	

Error group 81		IRQ overflow		
No.	Code	Message		Reaction
81-4	F084h	**Overflow low-level IRQ**		PSoff
		Cause	The process data could not be calculated in the set current/speed/ position interpolator cycle.	
		Measure	• Please contact Technical Support.	
81-5	F085h	**Overflow MDC IRQ**		PSoff
		Cause	The process data could not be calculated in the set current/speed/ position interpolator cycle.	
		Measure	• Please contact Technical Support.	

Error group 82		Internal sequence control		
No.	Code	Message		Reaction
82-0	-	**Internal sequencing control: Event**		configurable
		Cause	IRQ4 overflow (10 ms low-level IRQ).	
		Measure	• Internal sequence control: Process was interrupted. • For information only - no action required.	
82-1	-	**Multiple-started KO write access**		configurable
		Cause	Parameters in cyclical and acyclical operation are used concur-rently.	
		Measure	• Only one parameterisation interface can be used (USB or Ether-net).	

Error group 83		Modules in Ext1/Ext2 (only CMMP-AS-...-M3)		
No.	Code	Message		Reaction
83-0	-	**Invalid module**		configurable
		Cause	– The plugged-in interface could not be detected. – The loaded firmware is not known. – A supported interface might be plugged into the wrong slot (e.g. SERCOS 2, EtherCAT).	
		Measure	• Check firmware whether interface is supported. If yes: • Check that the interface is in the right place and is plugged in correctly. • Replace interface and/or firmware.	
83-1	-	**Module not supported**		configurable
		Cause	The plugged-in interface could be detected but is not supported by the loaded firmware.	
		Measure	• Check firmware whether interface is supported. • If necessary, replace the firmware.	
83-2	-	**Module: Hardware revision not supported**		configurable
		Cause	The plugged-in interface could be detected and is basically also supported. In this case, however, the current hardware version is not supported (because it is too old).	
		Measure	• The interface must be exchanged. If necessary, contact Technical Support.	

Error group 84		Conditions for controller enabled		
No.	Code	Message		Reaction
84-0	-	**Conditions for controller enable not fulfilled**		Warn
		Cause	One or more conditions for controller enable are not fulfilled. This includes: – DIN4 (output stage enable) is off. – DIN5 (controller enable) is off. – Intermediate circuit not yet loaded. – Encoder is not yet ready for operation. – Angle encoder identification is still active. – Automatic current regulator identification is still active. – Encoder data are invalid. – Status change of the safety function not yet completed. – Firmware or DCO download via Ethernet (TFTP) active. – DCO download onto memory card still active. – Firmware download via Ethernet active.	
		Measure	• Check status of digital inputs. • Check encoder cables. • Wait for automatic identification. • Wait for completion of the firmware or DCO download.	

Error group 90		Internal error		
No.	Code	Message		Reaction
90-0	5080h	**External RAM not recognized**		PSoff
		Cause	External SRAM not detected / not sufficient. Hardware error (SRAM component or board is faulty).	
		Measure	• Please contact Technical Support.	
90-2	5080h	**Error at FPGA boot-up**		PSoff
		Cause	The FPGA (hardware) cannot be booted. The FPGA is booted serially when the device is started, but in this case it could not be loaded with data or it reported a checksum error.	
		Measure	• Switch on the device again (24 V). If the error occurs again, the hardware is faulty.	
90-3	5080h	**Error at SD-ADU start**		PSoff
		Cause	SD-ADUs (hardware) cannot be started. One or more SD-ADUs are not supplying any serial data.	
		Measure	• Switch on the device again (24 V). If the error occurs again, the hardware is faulty.	

Error group 90		Internal error		
No.	Code	Message		Reaction
90-4	5080h	**SD-ADU synchronisation error after start**		PSoff
		Cause	SD-ADU (hardware) not synchronous after starting. During operation, the SD-ADUs for the resolver signals continue running with strict synchronisation once they have been initially started synchronously. The SD-ADUs could not be started at the same time during that initial start phase.	
		Measure	• Switch on the device again (24 V). If the error occurs again, the hardware is faulty.	
90-5	5080h	**SD-ADU not synchronous**		PSoff
		Cause	SD-ADU (hardware) not synchronous after starting. During operation, the SD-ADUs for the resolver signals continue running with strict synchronisation once they have been initially started synchronously. This is checked continually during operation and an error is triggered if appropriate.	
		Measure	• Possibly massive EMC coupling. • Switch on the device again (24 V). If the error occurs again, the hardware is faulty.	
90-6	5080h	**IRQ0 (current controller): Trigger error**		PSoff
		Cause	The output stage is not triggering the software IRQ, which then operates the current regulator. Very likely to be a hardware error on the board or in the processor.	
		Measure	• Switch on the device again (24 V). If the error occurs again, the hardware is faulty.	
90-9	5080h	**Illegal firmware version**		PSoff
		Cause	A beta version compiled for the debugger was loaded regularly.	
		Measure	• Check the firmware version, and update the firmware if necessary.	

Error group 91		Initialisation error		
No.	Code	Message		Reaction
91-0	6000h	**Internal initialising error**		PSoff
		Cause	Internal SRAM too small for the compiled firmware. Can only occur with beta versions.	
		Measure	• Check the firmware version, and update the firmware if necessary.	

Error group 91		Initialisation error		
No.	Code	Message		Reaction
91-1	-	**Memory error when copying**		PSoff
		Cause	Firmware parts were not copied correctly from the external FLASH into the internal RAM upon starting.	
		Measure	• Switch on the device again (24 V). If the error occurs repeatedly, check the firmware version and update the firmware if necessary.	
91-2	-	**Error when reading the controller/power section coding**		PSoff
		Cause	The ID-EEPROM in the controller or power section could either not be addressed at all or does not have consistent data.	
		Measure	• Switch on the device again (24 V). If the error occurs repeatedly, the hardware is faulty. No repair possible.	
91-3	-	**Software initialisation error**		PSoff
		Cause	One of the following components is missing or could not be initialised: a) Shared memory not available or faulty. b) Driver library not available or faulty.	
		Measure	• Check firmware version, update if necessary.	

Error group 92		Boot loader/firmware update		
No.	Code	Message		Reaction
92-0	-	**Error during firmware download**		PSoff
		Cause	Error during requested firmware download.	
		Measure	• Check the firmware file. • Restart firmware download.	
92-1	-	**Error during bootloader update**		PSoff
		Cause	Error during requested bootloader download.	
		Measure	• Restart bootloader download. • Send the device to the manufacturer for inspection.	

Instructions on actions with the error messages 08-2 ... 08-7	
Action	Notes
• Check whether encoder signals are faulty.	– Check the wiring, e.g. are one or more phases of the track signals interrupted or short-circuited? – Check that installation complies with EMC recommendations (cable screening on both sides?). – Only with incremental encoders: With TTL single-ended signals (HALL signals are always TTL single-ended signals): Check whether there might be an excessive voltage drop on the GND line; in this case = signal reference. Check whether there might be an excessive voltage drop on the GND line; in this case = signal reference. – Check the level of supply voltage on the encoder. Sufficient? If not, change the cable diameter (connect unused lines in parallel) or use voltage feedback (SENSE+ and SENSE-).
• Test with other encoders.	– If the error still occurs when the configuration is correct, test with a different (error-free) encoder (replace the connecting cable as well). If the error still occurs, there is a fault in the motor controller. Repair by the manufacturer required.

Tab. D.2 Instructions on error messages 08-2 ... 08-7

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	Meaning
0-signal	Means that there is a 0 V signal present at the input or output (positive logic, corresponds to LOW).
1-signal	Means that there is a 24 V signal present at the input or output (positive logic, corresponds to HIGH).
Axis	Mechanical component of a drive that transfers the drive force for the motion. 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	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.
Drive	Complete actuator, consisting of motor, encoder and axis, optionally with a gear unit, if applicable with controller.
Encoder	Electrical pulse generator (generally a rotor position transducer). The controller evaluates the electrical signals that are generated and uses them to calculate the position and speed.
Festo Configuration Tool (FCT)	Software with standardised project and data management for supported device types. The special requirements of a device type are supported with the necessary descriptions and dialogs by means of plug-ins.
Festo Handling and Positioning Profile (FHPP)	Uniform fieldbus data profile for positioning 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 as per the “Festo Handling and Positioning Profile” (I/O messaging 8 bytes I/O)
Force mode (profile torque mode)	Operating mode for executing a direct positioning task with power control (open loop transmission control) through motor current regulation.
HMI	Human-Machine Interface, e.g. control panel with LC display and operating buttons.
Homing	Positioning procedure in which the reference point and therefore the origin of the measuring reference system of the axis are defined.
Homing method	Method for determination of the reference position: against a fixed stop (overload current/velocity evaluation) or with reference switch.

Term / abbreviation	Meaning
Homing Switch	External sensor used for ascertaining the reference position and connected directly to the controller.
I O I/O	Input. Output. Input and/or output.
Jog mode	Manual travel in a positive or negative direction. Function for setting positions by approaching the target position, e.g. by teaching (teach mode) of positioning records.
Load voltage, logic voltage	The load voltage supplies the power electronics of the controller and thereby the motor. The logic voltage supplies the evaluation and control logic of the controller.
Operating mode	Type of control or internal operating mode of the 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...
PLC	Programmable logic controller; short: controller (also IPC: industrial PC).
Positioning mode (Profile Position mode)	Operating mode for executing a positioning record or a direct positioning task with position control (closed loop position control).
Positioning record	Positioning command defined in the position set table, consisting of target position, positioning mode, travel velocity and acceleration.
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 the control interface). The project zero point PZ is defined by a preset 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.
Referencing (Homing mode)	Definition of the measuring reference system of the axis
Velocity adjustment (Profile Velocity mode)	Operating mode for executing a positioning record or a direct positioning task with control of the velocity or rotational velocity.
Software limit	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 short of during positioning.
Teach mode	Operating mode for setting positions by approaching the target position, e.g. when creating positioning records.

Tab. E.1 Index of terms and abbreviations

Index

Copyright:
Festo AG & Co. KG
Postfach
73726 Esslingen
Germany

Phone:
+49 711 347-0

Fax:
+49 711 347-2144

e-mail:
service_international@festo.com

Internet:
www.festo.com

Original: de
Version: 1512b