Festo Handling and Positioning Profile for Multi-Axis Movements

FESTO

Manual

Multi-Axis Interface
CPX-CMXX

Festo Handling and Positioning Profile for Multi-Axis Movements (FHPP-MAX)

Manual
564 224
en 1110b
[8022111]

Original ..... de

Edition ..... en 1110b

Designation ..... P.BE-CMXX-FHPP-SW-EN

Order no. 564 224

Internet: http://www.festo.com
Email: service_international@festo.com

Contents

Intended use

This description contains the Festo handling and positioning profile for multi-axis movements (FHPP-MAX)

The FHPP-MAX is used to control the CPX-CMXX multi-axis interface.

This provides you with supplementary information about controlling, diagnosing and parameterising the multi-axis interface via the fieldbus.

You will find additional information in the documentation on the CPX-CMXX multi-axis controller and the documentation on the motor controller used.

Note
Always observe the safety instructions in the respective documentation.

Safety instructions

When commissioning and programming positioning systems, you must always observe the safety regulations in the descriptions and operating instructions for the components used.

The user must make sure that nobody is within the sphere of influence of the connected actuators or axis system. Access to the potential danger area must be prevented by suitable measures, such as barriers and warning signs.

Warning
Axes can move with high force and at high speed. Collisions can lead to serious injury to people and damage to components.

- Make sure that nobody can enter the sphere of influence of the axes or other connected actuators and that no items are within the positioning range while the system is connected to energy sources.

Warning
Errors in parametrisation can cause injury to people and damage to property.

- Enable the axis groups of the multi-axis system only if the multi-axis system has been properly installed and parameterised.

Target group

This manual is intended exclusively for technicians trained in control and automation technology, who have experience in installing, commissioning, programming and diagnosing positioning systems.

Service

Please consult your local Festo Service or write to the following e-mail address if you have any technical problems:

service_international@festo.com

Important user instructions

Danger categories

This manual contains instructions on the possible dangers which may occur if the product is not used correctly. These instructions are marked (Warning, Caution, etc.), printed on a shaded background and marked additionally with a pictogram. A distinction is made between the following danger warnings:

Warning
This means that failure to observe this instruction may result in serious personal injury or damage to property.

Caution
This means that failure to observe this instruction may result in personal injury or damage to property.

Note
This means that failure to observe this instruction may result in damage to property.

The following pictogram marks passages in the text which describe activities with electrostatically sensitive components.

Electrostatically sensitive components may be damaged if they are not handled correctly.

Marking special information

The following pictograms mark passages in the text containing special information.

Pictograms

Information:
Recommendations, tips and references to other sources of information.

Accessories:
Information on necessary or sensible accessories for the Festo product.

Environment:
Information on environment-friendly use of Festo products.

Text markings

- The bullet indicates activities which may be carried out in any order.

1. Figures denote activities which must be carried out in the numerical order specified.

– Hyphens indicate general activities.

Information on the version

This document describes FHPP-MAX Revision 1.13.

The basis for FHPP-MAX Revision 1.13 is FHPP Version 1.00 Revision 13.

Terms and abbreviations

The following terms and abbreviations are used in this manual:

Term/abbreviation	Meaning
0xA0 $A0_h$	Hexadecimal numbers are marked by a prefixed “0x” or by a lowered “h.” Example: 0xA0 = $A0_h$ = 160 decimal.
Axis	Complete actuator, consisting of motor, encoder and drive, optionally with gear, if applicable with motor controller.
Control	Control of the CPX terminal and the CPX-CMMX is alternatively carried out through: – a higher-level controller: a controller connected via fieldbus to the CPX terminal – a CPX-FEC/CPX-CEC: a controller that can be integrated into the CPX terminal
CPX module	Collective term for the various modules which can be integrated into a CPX terminal.
CPX-CEC-...	The CoDeSys controller as CPX module. Controller which can be integrated into the CPX terminal.
CPX-CMXX	Multi-axis interface CPX-CMXX
CPX-FEC	Front-end controller as CPX module. Controller which can be integrated into the CPX terminal.
Drive	Mechanical component of an axis, which transfers the driving power for the movement, defines the guide for the positioning run and also enables the work load and the homing switch to be configured.
FCT	Festo Configuration Tool. 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 dialogues by means of plug-ins.
FHPP	Festo Handling and Positioning Profile The FHPP is a fieldbus data profile for positioning controllers from Festo.

Term/abbreviation	Meaning
FHPP-MAX	Festo Handling and Positioning Profile for Multi-Axis Movements The FHPP-MAX is a fieldbus data profile for multi-axis systems from Festo. The basis for the FHPP-MAX is the FHPP.
FHPP-MAX operating mode	How the CPX-CMXX multi-axis interface is triggered via FHPP-MAX: – Record selection – Direct mode – Commissioning – Parametrisation
FPC	Festo Parameter Channel. FHPP- or FHPP-MAX-specific parameter access.
FST	Festo Software Tools for programming in instruction list and ladder diagram for the CPX-FEC and other controllers from Festo.
Functions	Special functions in the different operating modes. – Jog mode – Homing
I O I/O	Input. Output. Input and/or output.
Jog mode	Manual positioning of an axis in positive or negative direction.
Linking	With linking, positioning records can be automatically performed one after another.
Logic 0	Input or output provides 0 V (also LOW, FALSE or logical 0).
Logic 1	Input or output provides 24 V (also HIGH, TRUE or logical 1)
Motor controller	Control electronics which evaluate the control signals and provide the power supply for the motor via the power electronics (power electronics + controller + position controller).
Positioning record	Positioning task defined in the positioning record table, consisting of target position, velocity, acceleration, ...

Tab. 0/1: Index of terms and abbreviations

Overview

Chapter 1

Contents

1.1 Mode of operation

Festo has developed an optimised data profile especially tailored to the target applications for handling and positioning tasks of multi-axis systems, the “Festo Handling and Positioning Profile for Multi-Axis Movements (FHPP-MAX).”
The FHPP-MAX builds on the FHPP profile developed for single-axis controls. The FHPP-MAX permits uniform control and programming for various multi-axis systems from Festo with up to 2 groups of 4 axes each.

It defines for the user

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

Control and status data

Communication over the fieldbus is effected per axis group by way of 8-byte control and status data. Functions and status messages required in operation can be written and read directly.

Record selection mode

Saved positioning sets can be processed in the record selection mode.

Direct mode

In the direct mode, dynamically calculated positioning record parameters are loaded into the selected positioning record of the CPX-CMXX via the record register, and then the position record is executed as in the record selection mode.

In addition, stored positioning records can also be executed as in the record selection mode.

Commissioning

This operating mode is used to place the axis group in operation. The following functions are available:

– Jogging

– Homing

Parametrisation

In this operating mode, the controller can access all parameter values of the CPX-CMXX.

Sequence control

Chapter 2

Contents

2.1 FHPP-MAX Operating Modes

The FHPP-MAX operating modes differ in the content and meaning of the cyclic I/O data and in the functions that can be accessed in the multi-axis interface.

Operating mode	Description
Record selection mode	From a positioning record table stored in the CPX-CMXX, the controller selects a record that contains all parameters required for a positioning task. A record contains parameters for one or more axes of an axis group. The record number is transferred to the cyclic output data as a setpoint value.
Direct mode	Through the record register, dynamic positioning tasks are stored incrementally in the positioning record table of the CPX-CMXX. In addition, records from the positioning record table stored in the CPX-CMXX are executed.
Commissioning	In this operating mode, the following functions can be performed: – Jogging – Homing The functions are carried out only with the selected axis.
Parametrisation	Parameters are transferred in this mode and it is possible to read CAN objects of the individual drives.

Tab. 2/1: Overview of operating modes

2.1.1 Switching the operating mode

The operating mode is switched by the control byte CCON and acknowledged in the status word SCON.

The switch into commissioning or parametrisation is not possible in the status S4 “Operation enabled”.

The switch between record selection mode and direct mode is possible in every status.

The statuses are described in chapter 2.6, status machine, p. 2-25.

2.1.2 Record selection mode

Per axis group, the CPX-CMXX has over 1024 records that contain all information necessary for a positioning task.

The record number that the CPX-CMXX is to process at the next start is transferred in the controller's output data.

The CPX-CMXX does not support any automatic mode, i.e. no user program. Records cannot be processed automatically with a programmable logic. The CPX-CMMX thus cannot accomplish any practical tasks as a stand-alone – close coupling with the controller is always necessary.

However, it is possible to link various records and execute them one after the other with the help of a start command.

In this way, positioning profiles can be created without any effect by the inactive times which arise from the transfer in the fieldbus and the controller's cycle time.

2.1.3 Direct mode

In the direct mode, it is possible to dynamically calculate the parameters for each positioning task. This makes it possible to adjust the system to different tool sizes, for example, without having to re-parametrise the positioning record table. The positioning data are managed completely in the controller and sent directly to the CPX-CMXX.

The parameters for a dynamic positioning task are loaded via the record register in the positioning record table of the CPX-CMXX. The positioning task is worked off like a record in the record selection operating mode.

2.1.4 Commissioning

In this operating mode, the following functions are available:

– Jogging

– Homing

These functions apply only to the chosen axis.

Switching to the commissioning operating mode is not possible in the status S4 “Operation enabled”.

The statuses are described in chapter 2.6, status machine, p. 2-25.

The control and status signals have different effects in the commissioning and record selection operating modes.

2.1.5 Parametrisation

In this operating mode, parameters are transferred via the cyclical data of the FHPP-MAX.

The first CCON control byte is transferred to control the enabling and operating mode of the axis/axis group. The seven additional bytes are occupied by the Festo Parameter Channel (FPC).

Via additional parameters (PNU's), CAN objects of the individual drives can be read using an SDO query. Data transparency is then guaranteed, particularly in the case of error (drive status).

Switching to the parametrisation operating mode is not possible in the status S4 "Operation enabled".

The statuses are described in chapter 2.6, status machine, p. 2-25.

During parametrisation, a drive cannot be moved, but a vertical drive can be held stationary.

2.2 Configuration of the I/O data

2.2.1 Concept

The controller exchanges the following data with the CPX-CMXX via FHPP-MAX:

- control and status bytes,
- record number or record register content in the output data,
- acknowledgement of actual position in the input data,
- parameters as per FPC.

Fundamentally, 8 byte input and 8 byte output data are planned per axis group. Of these, the first byte is permanently occupied. It remains intact in each operating mode and controls the enabling of the motor controller and the operating modes. The other bytes are dependent on the chosen operating mode. Additional control or status bytes and target and actual values can be transferred here.

Overview of the control and status bytes

No.	Control	Acknowledge-ment	Description
1	CCON	SCON	Enabling and operating mode
2	CPOS	SPOS	Positioning, homing and jogging
3	CASEL	SASEL	In the commissioning operating mode: Selection of the axis In the direct operating mode: Load register
4	Reserved	ErrorNo	Error number

Tab. 2/2: Control and status bytes

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

Record selection mode

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Output data	CCON	CPOS	CASEL	Reserved	Record number			
Input data	SCON	SPOS	SASEL	Error No.	Actual position			

Tab. 2/3: Control and status bytes in the record selection mode

Direct mode

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Output data	CCON	CPOS	CASEL	Reserved	Record number			
				Register No.	Register value			
Input data	SCON	SPOS	SASEL	Error No.	Actual position of the selected axis			
					Register value			

Tab. 2/4: Control and status bytes in the direct mode

Commissioning								
	Byte 1	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**	**Byte 7**	**Byte 8**
Output data	CCON	CPOS	CASEL	Speed in %	Jog interval			
Input data	SCON	SPOS	SASEL	Error No.	Actual position of the selected axis			

Tab. 2/5: Control and status bytes in the commissioning operating mode

Parametrisation (FPC)								
	Byte 1	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**	**Byte 7**	**Byte 8**
Output data	CCON	Sub-index	Request identifier + parameter number		Value			
Input data	SCON	Sub-index	Response identifier + parameter number		Value			

Tab. 2/6: Control and status bytes in the parametrisation (FPC) operating mode

All values that occupy several bytes are stored in the Intel format (Little Indian).

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

Assignment of the control bytes (overview)								
CCON	**B7** **OPM2**	**B6** **OPM1**	**B5** **LOCK**	**B4** **–**	**B3** **RESET**	**B2** **–**	**B1** **STOP**	**B0** **ENABLE**
	Operating mode selection		Lock FCT access	–	Acknowledge malfunction	–	Stop	Enable axis/axis group
CPOS	**B7** **–**	**B6** **CLEAR**	**B5** **–**	**B4** **JOGN**	**B3** **JOGP**	**B2** **HOM**	**B1** **START**	**B0** **HALT**
	–	Delete remaining path	–	Jog negative	Jog positive	Start Homing	Start positioning task	Halt
CASEL	**B7** **LOAD_R**	**B6** **CMD2**	**B5** **CMD1**	**B4** **CMD0**	**B3** **–**	**B2** **–**	**B1** **SELAX1**	**B0** **SELAX0**
	Load register	Load commands			–	–	Axis selection	

Tab. 2/7: Assignment of the control bytes

Assignment of the status bytes (overview)								
SCON	**B7** **OPM2**	**B6** **OPM1**	**B5** **LOCK**	**B4** **DREADY**	**B3** **FAULT**	**B2** **WARN**	**B1** **OPEN**	**B0** **ENABLED**
	Acknowledgement Operating mode		FCT control	Ready	Malfunc-tion	Warning	Opera-tion enabled	Axis/Axis group enabled
SPOS	**B7** **REF**	**B6** –	**B5** –	**B4** **MOV**	**B3** –	**B2** **MC**	**B1** **ACK**	**B0** **HALT**
	Refer-enced	–	–	Position-ing task active	–	Motion complete	Acknow-ledge-ment	Halt
SASEL	**B7** **Load_** **CPL**	**B6** **CMD2**	**B5** **CMD1**	**B4** **CMD0**	**B3** –	**B2** –	**B1** **SELAX1**	**B0** **SELAX0**
	Loading executed	Acknowledgement of the load commands			–	–	Chosen axis	

Tab. 2/8: Assignment of the status bytes

2.4 Description of the control bytes

The following tables contain the German and English designations of the functions/status information. The designations of the bits have been derived from the English designations of the functions/status information.

The interaction of the control bits is explained in chapter 3, Drive functions.

2.4.1 Control byte 1 (CCON)

CCON controls the statuses that must be available in all operating modes.

Control byte 1 (CCON)		
Bit	**English**	**Description**
B0 **ENABLE**	**Enable** axis/ axis group	= 1: Enable axis/axis group = 0: Block axis/axis group
B1 **STOP**	**Stop**	= 1: Enable operation of the axis/axis group. = 0: Cancel positioning task. The axis/axis group stops independently of the device type with a braking ramp. The positioning task is reset.
B2 **–**	–	Reserved
B3 **RESET**	**Reset** Fault	= 0 → 1: With the rising edge, a fault is acknowledged and the fault value is deleted.
B4 **–**	–	Reserved, must be at 0.
B5 **LOCK**	**Lock** FCT Access	Controls FCT access. = 1: FCT must not take over drive control. = 0: FCT may take over drive control.
B6 **OPM1**	Select **Op**erating **M**ode	Bit 7 6 Operating mode 0 0 Record selection 0 1 Direct mode 1 0 Commissioning 1 1 Parametrisation
B7 **OPM2**		

Tab. 2/9: Control byte 1 (CCON)

Operating-mode-dependent effect of CCON

Bit	Operating mode		
	Record selection	**Direct mode**	**Commissioning**
CCON.ENABLE CCON.STOP	Effect on all axes in the group.		Effect only on the chosen axis; see CASEL.

Tab. 2/10: Operating-mode-dependent effect of CCON

2.4.2 Control byte 2 (CPOS)

CPOS controls the positioning sequences as soon as the chosen axis has been enabled.

Control byte 2 (CPOS)		
Bit	**EN**	**Description**
B0 **HALT**	**Halt**	= 1: Halt is not active = 0: Halt activated (Do not cancel positioning task). The axis/axis group stops independently of the device with a braking ramp; the positioning task remains active. With CPOS.B6, the remaining position can be cleared. The Halt command can be activated and deactivated with parameter 522, bit 0.
B1 **START**	**Start** Positioning Task	= 0 → 1: With the positive edge, the current positioning record is taken over and positioning of all participating axes started.
B2 **HOM**	Start **Hom**ing	= 0 → 1: With the positive edge, the homing run of the chosen axis is started with the set parameters.
B3 **JOGP**	**Jog p**ositive	The drive of the chosen axis moves at the specified speed in the direction of larger actual values, as long as the bit is set. The movement begins with the positive edge and ends with the negative edge.
B4 **JOGN**	**Jog n**egative	The drive of the chosen axis moves at the specified speed in the direction of smaller actual values, as long as the bit is set. The movement begins with the positive edge and ends with the negative edge.
B5 **–**	–	Reserved
B6 **CLEAR**	**Clear** Remaining Position	= 0 → 1: In the “Halt” status, the positive edge causes deletion of the positioning task and transition to the “Ready” status. MC is set again.
B7 **–**	–	Reserved, must be at 0.

Tab. 2/11: Control byte 2 (CPOS)

Operating-mode-dependent effect of CPOS

Bit	Operating mode		
	Record selection	**Direct mode**	**Commissioning**
CPOS.CLEAR	Effect on all axes in the group.		No effect.
CPOS.START	Has effect only in these operating modes.		No effect.
CPOS.HOM CPOS.HALT	Effect on all axes in the group.		Effect only on the chosen axis; see CASEL.
CPOS.JOGP CPOS.JOGN	No effect.		Effect only on the chosen axis; see CASEL.

Tab. 2/12: Operating-mode-dependent effect of CPOS

2.4.3 Control byte 3 (CASEL)

CASEL chooses the axis in the commissioning operating mode.

Control byte 3 (CASEL)		
Bit	**EN**	**Description**
B0 **SELAX0** **B1** **SELAX1**	**Sel**ect **Ax**is	Bit 1 0 Axis 0 0 1st axis (X) 0 1 2nd axis (Y) 1 0 3rd axis (Z) 1 1 4th axis (U)
B2 **–**	–	Reserved
B3 **–**	–	Reserved
B4 **CMD0** **B5** **CMD1** **B6** **CMD2**	Load **Com**man**d**	Bit 6 5 4 Command 0 0 0 No action 0 0 1 Write register 0 1 0 Add register 0 1 1 Sub register 1 0 0 Read register Reg. no. in byte 4, register value in byte 5,6,7,8
B7 **LOAD_R**	**Load R**egister	= 0 → 1: With the positive edge, the action in CMD0, CMD1 and CMD2 is performed.

Tab. 2/13: Control byte 3 (CASEL)

Operating-mode-dependent effect of CASEL

Bit	Operating mode		
	Record selection	**Direct mode**	**Commissioning**
CASEL.SELAX0 CASEL.SELAX1	Axis choice for display of actual position	Axis choice for display of actual position	Axis choice for commissioning
CASEL.CMD0 CASEL.CMD1 CASEL.CMD2	Not relevant	Load commands	Not relevant
CASEL.LOAD_R	Reserved, must be at 0!	= 0 → 1: With the positive edge, the action in Load Command is carried out.	Reserved, must be at 0!

Tab. 2/14: Operating-mode-dependent effect of CASEL

2.5 Description of the status byte

The following tables contain the German and English designations of the functions/status information. The designations of the bits have been derived from the English designations of the functions/status information.

2.5.1 Status byte 1 (SCON)

Status byte 1 (SCON)		
Bit	**EN**	**Description**
B0 **ENABLED**	Axis/Axis Group **Enabled**	= 0: Axis/axis group not enabled = 1: Axis/axis group enabled
B1 **OPEN**	**Op**eration **En**abled	= 0: Stop active = 1: Operation enabled, positioning possible
B2 **WARN**	**Warn**ing	= 0: Warning not registered = 1: Warning registered
B3 **FAULT**	**Fault**	= 0: No fault = 1: There is a fault or fault reaction is active. Fault code in the diagnostic memory and also in the error byte of FHPP-MAX.
B4 **DREADY**	All **D**rives are **Ready**	= 0: One or more axes are initialised = 1: All axes in the group are ready Target-actual configuration is OK
B5 **LOCK**	Drive Control by FCT	= 0: The controller controls the CPX-CMXX = 1: The FCT controls the CPX -CMXX (Drive control by FCT) (PLC control is **Lock**ed)
B6 **OPM1** **B7** **OPM2**	Display **Op**erating **M**ode	Bit 7 6 Operating mode displayed 0 0 Record selection 0 1 Direct mode 1 0 Commissioning 1 1 Parametrisation

Tab. 2/15: Status byte 1 (SCON)

Operating-mode-dependent effect of SCON

Bit	Operating mode		
	Record selection	**Direct mode**	**Commissioning**
SCON.ENABLED SCON.OPEN	Status of all axes in the group (AND link of the statuses of all individual axes in the group)		Status of the chosen axis, see CASEL.

Tab. 2/16: Operating-mode-dependent effect of SCON

2.5.2 Status byte 2 (SPOS)

Status byte 2 (SPOS)		
Bit	**EN**	**Description**
B0 **HALT**	**Halt**	= 0: Halt is active = 1: Halt is not active, drive can be moved
B1 **ACK**	**Ack**nowledge	= 0: Ready for start (positioning task, homing, jogging) = 1: Start executed (positioning task, homing, jogging)
B2 **MC**	**M**otion **C**omplete	= 0: Positioning task active = 1: Positioning task completed Notes: – If a positioning task was completed with an error, MC is set only after acknowledgement of the error. – MC is set after device is switched on (status “Drive blocked”).
B3 **–**	–	Reserved
B4 **MOV**	Axis is **mov**ing	= 0: Selected axis at standstill. = 1: Selected axis is moving.
B5 **–**	–	Reserved
B6 **–**	–	Reserved
B7 **REF**	Axis is **ref**erenced	= 0: Referencing must be carried out = 1: Reference information present, homing not necessary

Tab. 2/17: Status byte 2 (SPOS)

Operating-mode-dependent effect of SPOS

Bit	Operating mode		
	Record selection	**Direct mode**	**Commissioning**
SPOS.HALT SPOS.MC SPOS.MOV SPOS.REF	Status of all axes in the group (AND link of the statuses of all individual axes in the group)		Status of the chosen axis, see CASEL.

Tab. 2/18: Operating-mode-dependent effect of SPOS

2.5.3 Status byte 3 (SASEL)

The status byte 3 (SASEL) reports back on the chosen axis in the commissioning operating mode.

Status byte 3 (SASEL)

Bit	EN	Description
B0 SELAX0 **B1 SELAX1**	**Sel**ected **Ax**is	Bit 1 0 Selected axis 0 0 1st axis (X) 0 1 2nd axis (Y) 1 0 3rd axis (Z) 1 1 4th axis (U)
B2 –	–	Reserved
B3 –	–	Reserved
B4 CMD0 **B5 CMD1** **B6 CMD2**	Load **Com**man**d**	Bit 6 5 4 Command 0 0 0 No action 0 0 1: Write Register *) 0 1 0: Add Register *) 0 1 1: Sub Register *) 1 0 0: Read Register *) *) Reg. no. in byte 4, value in bytes 5,6,7,8. Value remains stationary as long as Load_CPL is active.
B7 Load_ CPL	**Load** com**pl**eted	= 0 → 1: With the positive edge, the action in CMD0, CMD1 and CMD2 was performed. Load_CPL remains stationary as long as Load_R (CASEL.B7) is active.

Tab. 2/19: Status byte 3 (SASEL)

Operating-mode-dependent effect of SASEL

Bit	Operating mode		
	Record selection	**Direct mode**	**Commissioning**
SASEL.SELAX0 SASEL.SELAX1	Chosen axis for display of actual position	Chosen axis for display of actual position	Chosen axis for commissioning
SASEL.CMD0 SASEL.CMD1 SASEL.CMD2	No effect	Acknowledgement of load commands	No effect
SASEL.LOAD_CPL	Reserved, must be at 0!	= 0 → 1: With the positive edge, the action in Load Command was carried out.	Reserved, must be at 0!

Tab. 2/20: Operating-mode-dependent effect of SASEL

2.6 FHPP-MAX status machine

2.6.1 General representation

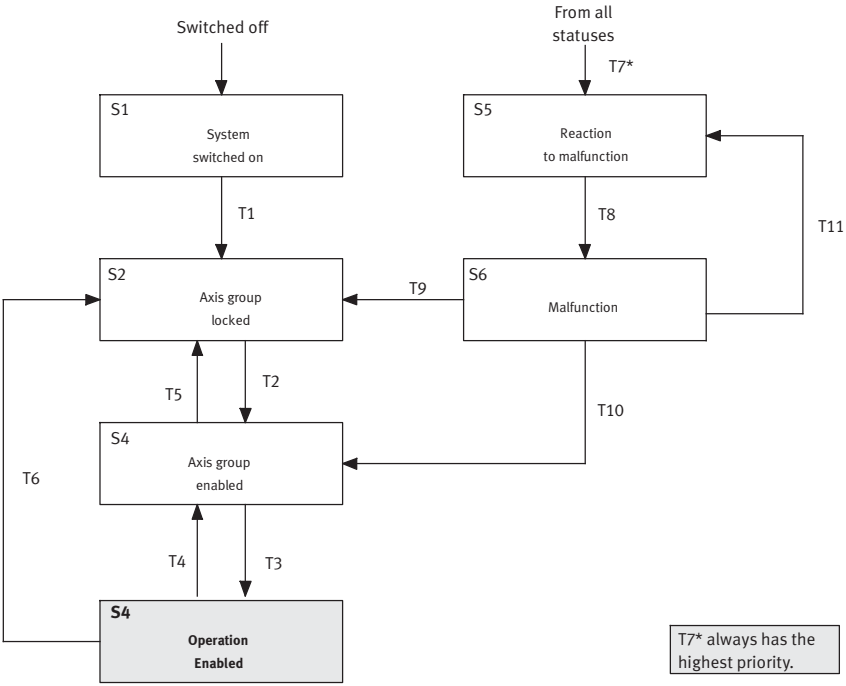


Fig. 2/1: FHPP-MAX status machine

S1...S6: Statuses

T1...T11: Transitions (Status transitions)

Description of the transitions Tx

T	Internal conditions	Actions of the user	Remarks
T1	System is switched on. There is no fault.		
T2	Load voltage applied. Controller controls CPX-CMXX.	“Axis” group enabled = 1 CCON = xxx0.xxx**1**	
T3		“Stop” = 1 CCON = xxx0.xx**1**1	Not permitted in the parametrisation mode, since the status S4 “Operation enabled” does not exist in this operating mode.
T4		“Stop” = 0 CCON = xxx0.xx**0**1	Not present in the parametrisation mode.
T5		“Axis group enabled” = 0 CCON = xxx0.xxx**0**	
T6		“Axis group enabled” = 0 CCON = xxx0.xxx**0**	Not present in the parametrisation mode.
T7*	Malfunction recognised.		T7* has the highest priority. T7 is executed from the statuses S5 and S6 when a fault with higher priority occurs. This means that a serious error can displace a less serious error.
T8	Reaction to fault complete; axis group stopped.		
T9	Fault type 2 is no longer present.	“Acknowledge malfunction” = 0 → 1 CCON = xxx0.**P**xxx	
T10	Fault type 1 is no longer present.	“Acknowledge malfunction” = 0 → 1 CCON = xxx0.**P**xx1	
T11	Malfunction still exists.	“Acknowledge malfunction” = 0 → 1 CCON = xxx0.**P**xx1	
Key: 0 → 1 = P = positive edge, 1 → 0 = N = negative edge, x = any, * highest priority			

Tab. 2/21: Description of the transitions Tx

2.6.2 Status S4 “Operation enabled”

The transition T3 changes to status S4, which itself contains its own sub-status machine, the statuses of which are marked with “SAx” and the transitions of which are marked with “TAx”, see Fig. 2/2.

Transitions T4, T6 and T7* are executed from every sub-state SAx and automatically have a higher priority than any transition TAx.

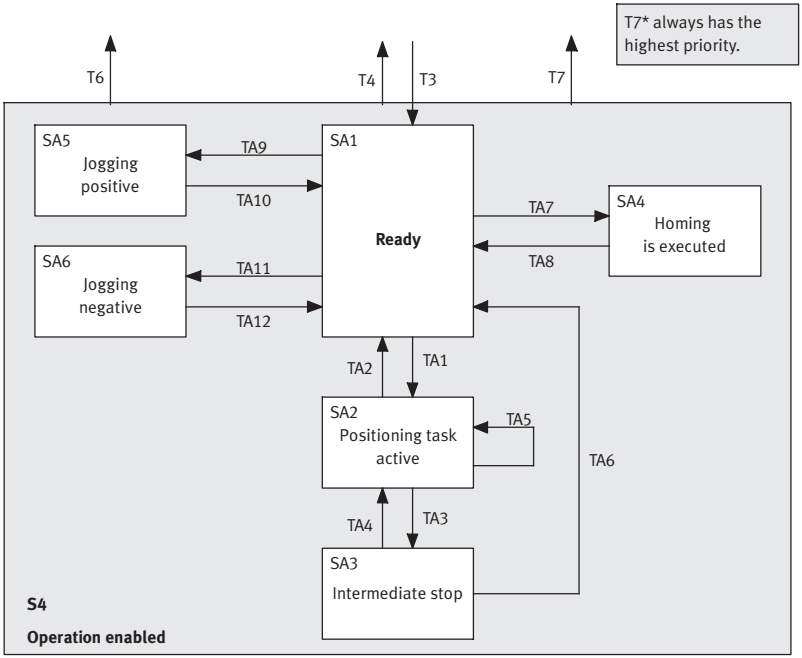


Fig. 2/2: Status S4 “Operation enabled”

SA1...SA6: Sub-statuses of the status S4

TA1...TA12: Transitions (status transitions) within status S4

Description of the transitions TAx

Fundamentally, the following applies:
Transitions T4, T6 and T7* always have priority!
Halt is only effective if parameter 522, bit 1 = 0.

TA	Internal conditions	Actions of the user	Remarks
TA1	Referencing is running.	Start positioning task = 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 processed automatically.	“Halt” status is any CCON = xxx0.xx11 CPOS = 0xxx.xxxx	
TA3	Motion Complete = 0	Halt = 1 → 0 CCON = xxx0.xx11 CPOS = 0xxx.xxx**0**	In the commissioning operating mode, with Halt = 1 → 0, the switch is made into the status SA1. In the commissioning mode, there is no status SA3 “Intermediate Stop”.
TA4		Halt = 1 Start positioning task = 0→1 Clear remaining position = 0 CCON = xxx0.xx11 CPOS = 0**0**xx.xx**P1**	Not present in the commissioning mode.
TA5			Not present in the commissioning mode.
	Record selection and direct mode: – A single record is finished. – The next record is processed automatically.	CCON = xxx0.xx11 CPOS = 0xxx.xxx1	
TA6		Clear remaining position = 0 → 1 CCON = xxx0.xx11 CPOS = 0**1**xx.xxxx	
Key: 0 → 1 = P = positive edge, 1 → 0 = N = negative edge, x = any			

TA	Internal conditions	Actions of the user	Remarks
TA7		Start homing = 0→1 Halt = 1 CCON = xxx0.xx11 CPOS = 0xx0.0P**x1**	
TA8	Referencing finished or stopped.	Only for halt: Halt = 1 → 0 CCON = xxx0.xx11 CPOS = 0xxx.xxx**N**	
TA9		Jog positive = 0 → 1 Halt = 1 CCON = xxx0.xx11 CPOS = 0xx0.**P**xx**1**	
TA10		Either – Jog positive = 1 → 0 – CCON = xxx0.xx11 – CPOS = 0xxx.**0**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**.xxx**1**	
TA12		Either – Jog negative = 1 → 0 – CCON = xxx0.xx11 – CPOS = 0xx**N**.xxx1 or – Halt = 1 → 0 – CCON = xxx0.xx11 – CPOS = 0xxx.xxx**N**	
Key: 0 → 1 = P = positive edge, 1 → 0 = N = negative edge, x = any			

Tab. 2/22: Description of the transitions TAx

2.6.3 Special features in the commissioning mode

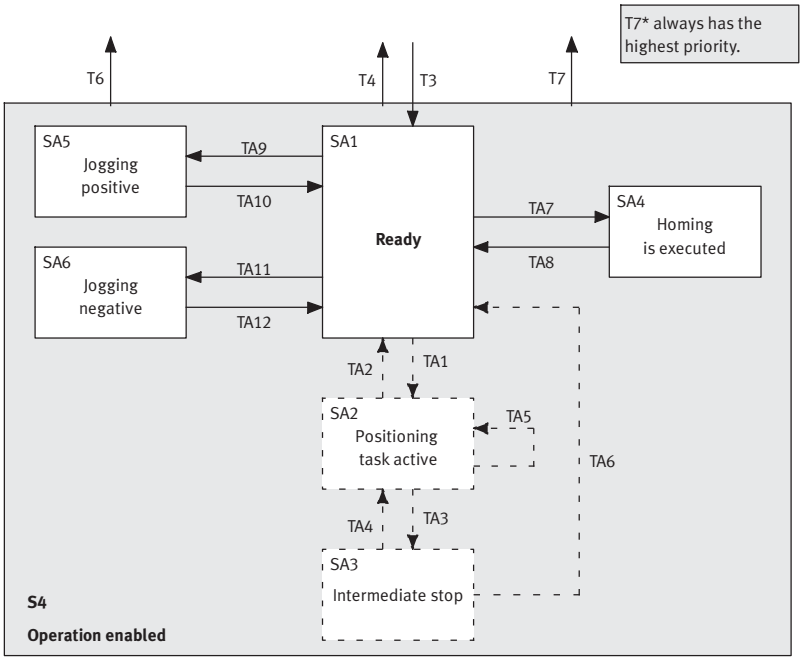


Fig. 2/3: Status S4 “Operation enabled” in the commissioning operating mode

SA2 and SA3: Does not exist here. If Halt=0, the axis group changes after SA1.

TA1, TA2, TA3, TA4, TA5 and TA6: The transitions are not present.

2.6.4 Special features in the parametrisation mode

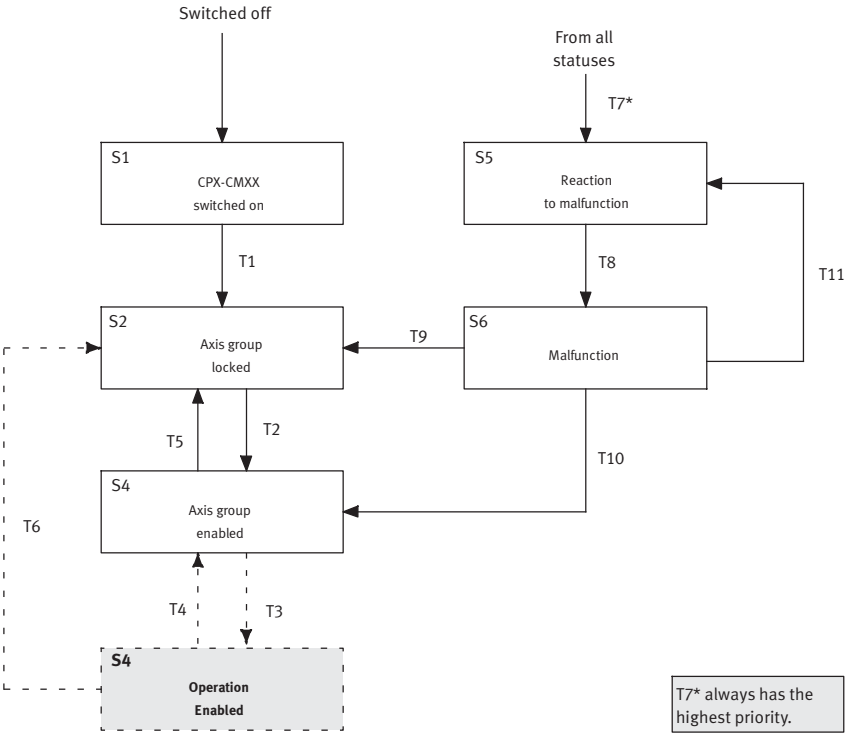


Fig. 2/4: FHPP-MAX status machine in the parametrisation operating mode

S4: Does not exist here.

T3: The transition is not permitted.

2.7 Examples of control and status bytes

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

- Safeguard device control
- Establish readiness to operate – Record Selection
- Establish readiness to operate – Direct mode
- Fault handling
- Homing
- Positioning record selection
- Positioning direct mode

The following examples are presented in tables. The column control bytes contains the setpoints and the column status bytes the acknowledgements.

1. Safeguard device control

Step/ description	Control bytes									Status bytes								
	Byte	B7	B6	B5	B4	B3	B2	B1	B0	Byte	B7	B6	B5	B4	B3	B2	B1	B0
1.1 Device control FCT on, SCON.B5 = 1	Byte 1	OPM2	OPM1	LOCK	–	RESET	–	STOP	ENABL	Byte 1	OPM2	OPM1	LOCK	DRDY	FAULT	WARN	OPEN	ENABL
	CCON	0	0	0	0	0	0	0	0	SCON	0	0	**1**	1	0	0	0	0
	Byte 2	–	CLEAR	–	JOGN	JOGP	HOM	START	HALT	Byte 2	REF	–	–	MOV	–	MC	ACK	HALT
	CPOS	0	0	0	0	0	0	0	0	SPOS	0	0	0	0	0	**1**	0	0
0: logic 0: 1: logic 1: x: not relevant (any); P: positive edge; N: negative edge																		

Tab. 2/23: Control and status bytes “Device control active”

Description of 1. Safeguard device control:

1.1 The device control through the FCT (Festo Configuration Tool) is activated.
For control over FHPP-MAX, the device control must first be deactivated through the FCT; see online help for FCT plug-in CMXX.

2. Establish readiness to operate – Record Selection

Step/ description	Control bytes									Status bytes								
	Byte	B7	B6	B5	B4	B3	B2	B1	B0	Byte	B7	B6	B5	B4	B3	B2	B1	B0
2.1 Basic status (Device control FCT off, SCON.B5 = 0)	Byte 1	OPM2	OPM1	LOCK	–	RESET	–	STOP	ENABL	Byte 1	OPM2	OPM1	LOCK	DRDY	FAULT	WARN	OPEN	ENABL
	CCON	0	0	0	0	0	0	0	0	SCON	0	0	**0**	1	0	0	0	0
	Byte 2	–	CLEAR	–	JOGN	JOGP	HOM	START	HALT	Byte 2	REF	–	–	MOV	–	MC	ACK	HALT
	CPOS	0	0	0	0	0	0	0	0	SPOS	0	0	0	0	0	**1**	0	0
2.2 Disable device control FCT	Byte 1	OPM2	OPM1	LOCK	–	RESET	–	STOP	ENABL	Byte 1	OPM2	OPM1	LOCK	DRDY	FAULT	WARN	OPEN	ENABL
	CCON	x	x	**1**	0	x	0	x	x	SCON	x	x	**0**	1	x	x	x	x
	Byte 2	–	CLEAR	–	JOGN	JOGP	HOM	START	HALT	Byte 2	REF	–	–	MOV	–	MC	ACK	HALT
	CPOS	0	x	0	x	x	x	x	x	SPOS	x	x	x	x	x	x	x	x
2.3 Enable drive, enable operation (record selection)	Byte 1	OPM2	OPM1	LOCK	–	RESET	–	STOP	ENABL	Byte 1	OPM2	OPM1	LOCK	DRDY	FAULT	WARN	OPEN	ENABL
	CCON	**0**	**0**	x	0	0	0	**1**	**1**	SCON	**0**	**0**	0	1	0	0	**1**	**1**
	Byte 2	–	CLEAR	–	JOGN	JOGP	HOM	START	HALT	Byte 2	REF	–	–	MOV	–	MC	ACK	HALT
	CPOS	0	0	0	0	0	0	0	**1**	SPOS	0	0	0	0	0	1	0	**1**
0: logic 0: 1: logic 1: x: not relevant (any); P: positive edge; N: negative edge																		

Tab. 2/24: Control and status bytes – “Establish readiness to operate – Record selection”

Description of 2. Establish readiness to operate:

2.1 Basic status of the axis group when the supply voltage has been switched on.
→ Continue with step 2.2 or 2.3

2.2 Disable device control by FCT.
Optionally, the takeover of device control by the FCT can be disabled with CCON.B5 = 1 (LOCK).
→ Continue with step 2.3

2.3 Enable axis group in Record Select mode.
→ See example 5. Homing run, Tab. 2/27.

If there are malfunctions after switching on or after setting CCON.B0 (ENABLE):
→ See example 4. Fault handling, Tab. 2/26.

3. Establish readiness to operate – Direct mode

Step/ description	Control bytes									Status bytes								
	Byte	B7	B6	B5	B4	B3	B2	B1	B0	Byte	B7	B6	B5	B4	B3	B2	B1	B0
3.1 Basic status (Device control FCT off, SCON.B5 = 0)	Byte 1	OPM2	OPM1	LOCK	–	RESET	-	STOP	ENABL	Byte 1	OPM2	OPM1	LOCK	DRDY	FAULT	WARN	OPEN	ENABL
	CCON	0	0	0	0	0	0	0	0	SCON	0	0	**0**	1	0	0	0	0
	Byte 2	–	CLEAR	–	JOGN	JOGP	HOM	START	HALT	Byte 2	REF	–	–	MOV	–	MC	ACK	HALT
	CPOS	0	0	0	0	0	0	0	0	SPOS	0	0	0	0	0	**1**	0	0
3.2 Disable device control FCT	Byte 1	OPM2	OPM1	LOCK	–	RESET	-	STOP	ENABL	Byte 1	OPM2	OPM1	LOCK	DRDY	FAULT	WARN	OPEN	ENABL
	CCON	x	x	**1**	0	x	0	x	x	SCON	x	x	0	1	x	x	x	x
	Byte 2	–	CLEAR	-	JOGN	JOGP	HOM	START	HALT	Byte 2	REF	-	-	MOV	-	MC	ACK	HALT
	CPOS	0	x	0	x	x	x	x	x	SPOS	x	0	0	x	0	x	x	x
3.3 Enable drive, enable operation (direct mode)	Byte 1	OPM2	OPM1	LOCK	–	RESET	-	STOP	ENABL	Byte 1	OPM2	OPM1	LOCK	DRDY	FAULT	WARN	OPEN	ENABL
	CCON	**0**	**1**	x	0	0	0	**1**	**1**	SCON	**0**	**1**	0	1	0	0	**1**	**1**
	Byte 2	–	CLEAR	-	JOGN	JOGP	HOM	START	HALT	Byte 2	REF	-	-	MOV	-	MC	ACK	HALT
	CPOS	0	0	0	0	0	0	0	**1**	SPOS	0	0	0	0	0	1	0	**1**
0: logic 0: 1: logic 1: x: not relevant (any); P: positive edge; N: negative edge																		

Tab. 2/25: Control and status bytes “Establish readiness to operate – Direct mode”

Description of 3. Establish readiness to operate:

3.1 Basic status of the axis group when the supply voltage has been switched on.
→ Step 3.2 or 3.3

3.2 Disable device control FCT.
Optionally, the takeover of device control by the FCT can be disabled with CCON.B5 = 1 (LOCK).
→ Step 3.3

3.3 Enable axis group in the direct mode.
→ See example 5. Homing run, Tab. 2/27.

If there are malfunctions after switching on or after setting CCON.B0 (ENABLE):
→ See example 4. Fault handling, Tab. 2/26.

4. Fault handling

Step/ description	Control bytes									Status bytes								
	Byte	B7	B6	B5	B4	B3	B2	B1	B0	Byte	B7	B6	B5	B4	B3	B2	B1	B0
4.1 Error	Byte 1	OPM2	OPM1	LOCK	–	RESET	–	STOP	ENABL	Byte 1	OPM2	OPM1	LOCK	DRDY	FAULT	WARN	OPEN	ENABL
	CCON	x	x	x	0	x	0	x	x	SCON	x	x	x	1	**1**	x	x	x
	Byte 2	–	CLEAR	–	JOGN	JOGP	HOM	START	HALT	Byte 2	REF	–	–	MOV	–	MC	ACK	HALT
	CPOS	0	x	0	x	x	x	x	x	SPOS	x	0	0	x	x	x	x	x
4.2 Warning	Byte 1	OPM2	OPM1	LOCK	–	RESET	–	STOP	ENABL	Byte 1	OPM2	OPM1	LOCK	DRDY	FAULT	WARN	OPEN	ENABL
	CCON	x	x	x	0	x	0	x	x	SCON	x	x	x	1	x	**1**	x	x
	Byte 2	–	CLEAR	–	JOGN	JOGP	HOM	START	HALT	Byte 2	REF	–	–	MOV	–	MC	ACK	HALT
	CPOS	0	x	0	x	x	x	x	x	SPOS	x	0	0	x	0	x	x	x
4.3 Acknowledge fault	Byte 1	OPM2	OPM1	LOCK	–	RESET	–	STOP	ENABL	Byte 1	OPM2	OPM1	LOCK	DRDY	FAULT	WARN	OPEN	ENABL
	CCON	0	x	x	0	**P**	0	x	1	SCON	0	x	0	1	**0**	**0**	0	0
with CCON.B3 (RESET)	Byte 2	–	CLEAR	–	JOGN	JOGP	HOM	START	HALT	Byte 2	REF	–	–	MOV	–	MC	ACK	HALT
	CPOS	0	0	0	0	0	0	x	x	SPOS	x	0	0	0	0	**1**	**0**	1
0: logic 0: 1: logic 1: x: not relevant (any); P: positive edge; N: negative edge																		

Tab. 2/26: Control and status bytes “Fault handling”

Description of 4. Fault handling

4.1 An error is shown with SCON.B3 (FAULT).
⟶ Positioning can no longer be undertaken.

4.2 A warning is shown with SCON.B2 (WARN).
⟶ Positioning can still be undertaken.

4.3 Acknowledge fault with positive edge at CCON.B3 (RESET).
⟶ Fault bit SCON.B2 (FAULT) or SCON.B3 (WARN) is reset
⟶ SPOS.B2 (MC) is set
⟶ Axis group is ready for operation

5. Homing

Step/ description	Control bytes									Status bytes								
	Byte	B7	B6	B5	B4	B3	B2	B1	B0	Byte	B7	B6	B5	B4	B3	B2	B1	B0
5.1 Start homing	Byte 1	OPM2	OPM1	LOCK	–	RESET	–	STOP	ENABL	Byte 1	OPM2	OPM1	LOCK	DRDY	FAULT	WARN	OPEN	ENABL
	CCON	0	x	x	0	0	0	1	1	SCON	0	x	0	1	0	0	1	1
	Byte 2	–	CLEAR	–	JOGN	JOGP	HOM	START	HALT	Byte 2	REF	–	–	MOV	–	MC	ACK	HALT
	CPOS	0	0	0	0	0	**P**	0	1	SPOS	0	0	0	0	0	**0**	**1**	1
5.2 Homing is running	Byte 1	OPM2	OPM1	LOCK	–	RESET	–	STOP	ENABL	Byte 1	OPM2	OPM1	LOCK	DRDY	FAULT	WARN	OPEN	ENABL
	CCON	0	x	x	0	0	0	1	1	SCON	0	x	0	1	0	0	1	1
	Byte 2	–	CLEAR	–	JOGN	JOGP	HOM	START	HALT	Byte 2	REF	–	–	MOV	–	MC	ACK	HALT
	CPOS	0	0	0	0	0	**1**	0	1	SPOS	0	0	0	**1**	0	0	1	1
5.3 Homing is finished	Byte 1	OPM2	OPM1	LOCK	–	RESET	–	STOP	ENABL	Byte 1	OPM2	OPM1	LOCK	DRDY	FAULT	WARN	OPEN	ENABL
	CCON	0	x	x	0	0	0	1	1	SCON	0	x	0	1	0	0	1	1
	Byte 2	–	CLEAR	TEACH	JOGN	JOGP	HOM	START	HALT	Byte 2	REF	–	–	MOV	–	MC	ACK	HALT
	CPOS	0	0	0	0	0	**0**	0	1	SPOS	**1**	0	0	0	0	**1**	0	1
0: logic 0: 1: logic 1: x: not relevant (any); P: positive edge; N: negative edge																		

Tab. 2/27: Control and status bytes “Homing run”

Description of 5. Homing run:

5.1 A positive edge at CPOS.B2 (HOM, Start homing) starts the homing. The start is confirmed with SPOS.B1 (Acknowledge Start) as long as CPOS.B2 (HOM) is set.

5.2 After successful reference travel SPOS.B2 (MC, Motion Complete) and SPOS.B7 (REF) is set.

If there are faults during homing:
→ See example 4. Fault handling, Tab. 2/26.

6. Positioning record select / direct mode

Step/ description	Control bytes									Status bytes								
	Byte	B7	B6	B5	B4	B3	B2	B1	B0	Byte	B7	B6	B5	B4	B3	B2	B1	B0
6.1 Preselect record number (control bytes 5 and 6)	Byte5	Record number byte 5																
	Byte6 Re-cord no.	Record number byte 6 Record no. (1 ...)																
6.2 Start task	Byte 1	OPM2	OPM1	LOCK	–	RESET	–	STOP	ENABL	Byte 1	OPM2	OPM1	LOCK	DRDY	FAULT	WARN	OPEN	ENABL
	CCON	0	0	x	0	0	0	1	1	SCON	0	0	0	1	0	0	1	1
	Byte 2	–	CLEAR	–	JOGN	JOGP	HOM	START	HALT	Byte 2	REF	–	–	MOV	–	MC	ACK	HALT
	CPOS	0	0	0	0	0	0	**P**	1	SPOS	1	0	0	0	0	**0**	1	1
6.3 Task running	Byte 1	OPM2	OPM1	LOCK	–	RESET	–	STOP	ENABL	Byte 1	OPM2	OPM1	LOCK	DRDY	FAULT	WARN	OPEN	ENABL
	CCON	0	0	x	0	0	0	1	1	SCON	0	0	0	1	0	0	1	1
	Byte 2	–	CLEAR	–	JOGN	JOGP	HOM	START	HALT	Byte 2	REF	–	–	MOV	–	MC	ACK	HALT
	CPOS	0	0	0	0	0	0	**1**	1	SPOS	1	0	0	**1**	0	0	**1**	1
6.4 Task finished	Byte 1	OPM2	OPM1	LOCK	–	RESET	–	STOP	ENABL	Byte 1	OPM2	OPM1	LOCK	DRDY	FAULT	WARN	OPEN	ENABL
	CCON	0	0	x	0	0	0	1	1	SCON	0	0	0	1	0	0	1	1
	Byte 2	–	CLEAR	–	JOGN	JOGP	HOM	START	HALT	Byte 2	REF	–	–	MOV	–	MC	ACK	HALT
	CPOS	0	0	0	0	0	0	**0**	1	SPOS	1	0	0	**0**	0	**1**	**0**	1
										Byte 5...8 Act. pos.	position Actual position (1/1000 mm or 1/1000 °)							
0: logic 0: 1: logic 1: x: not relevant (any); P: positive edge; N: negative edge																		

Tab. 2/28: Control and status bytes “Positioning record selection / direct mode”

Description of 6. Positioning record selection:

(steps 6.1 ... 6.3 conditional sequence)

When the readiness to operate is established and the homing run has been carried out, a positioning task can be started.

6.1 Preselect record number: Byte 5 and 6 of the output data

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

6.3 At the end of the positioning job, SPOS.B2 (MC, Motion Complete) will be set.

If there are faults during positioning:
→ See example 4. Fault handling, Tab. 2/26.

7. Write record register – direct mode

Step/ description	Control bytes									Status bytes								
	Byte	B7	B6	B5	B4	B3	B2	B1	B0	Byte	B7	B6	B5	B4	B3	B2	B1	B0
7.1 Set data	Byte 3 CA-SEL	LD_R 0	CMD2 0	CMD1 0	CMD0 **1**	– 0	– 0	SAX1 x	SAX0 x									
	Byte 4 Re-gister	Register number byte 4 Register number																
	Byte 5...8 Value	Register number byte 5...8 New register value																
7.2 Load	Byte 3 CA-SEL	LD_R **1**	CMD2 0	CMD1 0	CMD0 1	– 0	– 0	SAX1 x	SAX0 x	Byte 3 SA-SEL	L_CPL **1**	CMD2 0	CMD1 0	CMD0 1	– 0	– 0	SAX1 x	SAX0 x
7.3 Loading ended	Byte 3 CA-SEL	LD_R 0	CMD2 0	CMD1 0	CMD0 1	– 0	– 0	SAX1 x	SAX0 x	Byte 3 SA-SEL	L_CPL 0	CMD2 0	CMD1 0	CMD0 1	– 0	– 0	SAX1 x	SAX0 x
0: logic 0: 1: logic 1: x: not relevant (any); P: positive edge; N: negative edge																		

Tab. 2/29: Write record register – direct mode

Description to 7. Write record register – direct mode:

7.1 Register preselection:
Register number: Byte 4 of the output data
Register value: Byte 5 -8 of the output data

7.2 The register is loaded with CASEL.B7 (LOAD_R). The task is confirmed with SASEL.B7 (Load_CPL).

7.3 At the end of the load process, CASEL.B7 (LOAD_R) is reset. The end of the load process is confirmed by resetting SASEL.B7 (LOAD_CPL).

If there are faults while writing record register:
→ See example 4. Fault handling, Tab. 2/26.

8. Add record register – direct mode

Step/ description	Control bytes									Status bytes								
	Byte	B7	B6	B5	B4	B3	B2	B1	B0	Byte	B7	B6	B5	B4	B3	B2	B1	B0
8.1 Set data	Byte 3 CA-SEL	LD_R 0	CMD2 0	CMD1 **1**	CMD0 0	– 0	– 0	SAX1 x	SAX0 x									
	Byte 4 Re-gister	Register number byte 4 Register number																
	Byte 5...8 Value	Register number byte 5...8 Addition value																
8.2 Load	Byte 3 CA-SEL	LD_R **1**	CMD2 0	CMD1 1	CMD0 0	– 0	– 0	SAX1 x	SAX0 x	Byte 3 SA-SEL	L_CPL **1**	CMD2 0	CMD1 1	CMD0 0	– 0	– 0	SAX1 x	SAX0 x
8.3 Loading ended	Byte 3 CA-SEL	LD_R 0	CMD2 0	CMD1 1	CMD0 0	– 0	– 0	SAX1 x	SAX0 x	Byte 3 SA-SEL	L_CPL 0	CMD2 0	CMD1 1	CMD0 0	– 0	– 0	SAX1 x	SAX0 x
0: logic 0: 1: logic 1: x: not relevant (any); P: positive edge; N: negative edge																		

Tab. 2/30: Add record register – direct mode

Description for 8. Add record register – direct mode:

8.1 Register preselection:
Register number: Byte 4 of the output data
Addition value: Byte 5-8 of the output data

8.2 With CASEL.B7 (LOAD_R), the addition value is added to the register content. The task is confirmed with SASEL.B7 (Load_CPL).

8.3 At the end of the load process, CASEL.B7 (LOAD_R) is reset. The end of the load process is confirmed by resetting SASEL.B7 (LOAD_CPL).

If there are faults while adding record register:
→ See example 4. Fault handling, Tab. 2/26.

9. Subtract record register – direct mode

Step/ description	Control bytes									Status bytes								
	Byte	B7	B6	B5	B4	B3	B2	B1	B0	Byte	B7	B6	B5	B4	B3	B2	B1	B0
9.1 Set data	Byte 3 CA-SEL	LD_R 0	CMD2 0	CMD1 **1**	CMD0 **1**	– 0	– 0	SAX1 x	SAX0 x									
	Byte 4 Re-gister	Register number byte 4 Register number																
	Byte 5...8 Value	Register number byte 5...8 Subtraction value																
9.2 Load	Byte 3 CA-SEL	LD_R **1**	CMD2 0	CMD1 1	CMD0 1	– 0	– 0	SAX1 x	SAX0 x	Byte 3 SA-SEL	L_CPL **1**	CMD2 0	CMD1 1	CMD0 1	– 0	– 0	SAX1 x	SAX0 x
9.3 Loading ended	Byte 3 CA-SEL	LD_R 0	CMD2 0	CMD1 1	CMD0 1	– 0	– 0	SAX1 x	SAX0 x	Byte 3 SA-SEL	L_CPL 0	CMD2 0	CMD1 1	CMD0 1	– 0	– 0	SAX1 x	SAX0 x
0: logic 0: 1: logic 1: x: not relevant (any); P: positive edge; N: negative edge																		

Tab. 2/31: Subtract record register – direct mode

Description for 9. Subtract record register – direct mode:

9.1 Register preselection:
Register number: Byte 4 of the output data
Subtraction value: Byte 5-8 of the output data

9.2 With CASEL.B7 (LOAD_R), the subtraction value is subtracted from the register content. The task is confirmed with SASEL.B7 (Load_CPL).

9.3 At the end of the load process, CASEL.B7 (LOAD_R) is reset. The end of the load process is confirmed by resetting SASEL.B7 (LOAD_CPL).

If there are faults while subtracting record register:
→ See example 4. Fault handling, Tab. 2/26.

10. Read record register – direct mode

Step/ description	Control bytes									Status bytes								
	Byte	B7	B6	B5	B4	B3	B2	B1	B0	Byte	B7	B6	B5	B4	B3	B2	B1	B0
10.1 Set data	Byte 3 CA-SEL	LD_R 0	CMD2 **1**	CMD1 0	CMD0 0	– 0	– 0	SAX1 x	SAX0 x									
	Byte 4 Re-gister	Register number byte 4 Register number																
10.2 Load	Byte 3 CA-SEL	LD_R **1**	CMD2 1	CMD1 0	CMD0 0	– 0	– 0	SAX1 x	SAX0 x	Byte 3 SA-SEL	L_CPL **1**	CMD2 1	CMD1 0	CMD0 0	– 0	– 0	SAX1 x	SAX0 x
10.3 Loading ended	Byte 3 CA-SEL	LD_R 0	CMD2 1	CMD1 0	CMD0 0	– 0	– 0	SAX1 x	SAX0 x	Byte 3 SA-SEL	L_CPL 0	CMD2 1	CMD1 0	CMD0 0	– 0	– 0	SAX1 x	SAX0 x
0: logic 0: 1: logic 1: x: not relevant (any); P: positive edge; N: negative edge																		

Tab. 2/32: Read record register – direct mode

Description of 10. Read record register – direct mode:

10.1 Register preselection:
Register number: Byte 4 of the output data

10.2 The register content is read with CASEL.B7 (LOAD_R). The task is confirmed with SASEL.B7 (Load_CPL).

10.3 At the end of the load process, CASEL.B7 (LOAD_R) is reset. The end of the load process is confirmed by resetting SASEL.B7 (LOAD_CPL).

If there are faults while loading record register:
→ See example 4. Fault handling, Tab. 2/26.

Drive functions

Chapter 3

Contents

3.1 Jog mode

When the commissioning operating mode is selected, the axes of the axis group in the status S4 “Operation enabled” can be driven positively/negatively by jogging.
This function is usually used for:

– Moving axes out of the way (e.g. after a system fault)

– Manual traversing as a normal operating mode (manually operated feed).

Procedure

1. Choose axis with CASEL.SELAX0 (B0) and CASEL.SELAX1 (B1).

2. Set positioning speed in output byte 4:
   1 ... 100% of Vmax.

3. Reference axis.

4. When one of the signals “Jog positive (JOGP) / Jog negative (JOGN)” is set, the axis starts to move.

   – If PNU522 bit 6 = 0
     The axis runs, dependent on the jog interval, in the output bytes 5..8 as follows:
     Jog interval = 0: Axis runs up to the SW limit switch.
     Jog interval › 0: Axis runs the path in jog interval.

   – If PNU522 bit 6 = 1
     The axis is moved continuously in the speed control mode if supported by the motor controller. As a result, simultaneous moving of several axes (through switching of the axis choice with SASEL.SELAX0 and SASEl.SELAX1) is possible.

5. If the signal changes to 0, the axis is braked with the preset maximum deceleration.

Overview of the parameters involved	
Parameters involved	**Description**
PNU 508 **Acceleration**	Acceleration in Jog Mode; value range 1 ... 2^{31}-1 Default: Max. Acceleration (PNU 503)
PNU 509 **Deceleration**	Deceleration in Jog Mode; value range 1 ... 2^{31}-1 Default: Max. Deceleration (PNU 507)
PNU 522 bit 6 **Jog mode via speed mode**	= 0: Jogging with evaluation of the jog interval = 1: Axis is continously moved in the speed mode
Start	CPOS.B3 = 0 → 1: Jogging positive (direction of larger actual values) CPOS.B4 = 0 → 1: Jogging negative (direction of smaller actual values)
Speed	Output byte 4 (in % of Vmax)
Jog interval	Output byte 5, 6, 7 and 8; value range 0 ... 2^{32}-1

Tab. 3/1: Parameters involved in jogging mode

3.2 Record selection mode

A position record can be started in the status S4 “Operation enabled”.
This function is usually used for:

– Freely selected start-up through the controller of position records of the CPX-CMXX position record table.

– Processing of a positioning profile by linking position records,

– known target positions, which seldom change (e.g. with formulation change).

A total of 1024 position records is available.

The position records are configured in the CPX-CMXX FCT plug-in.

3.2.1 Record selection mode procedure

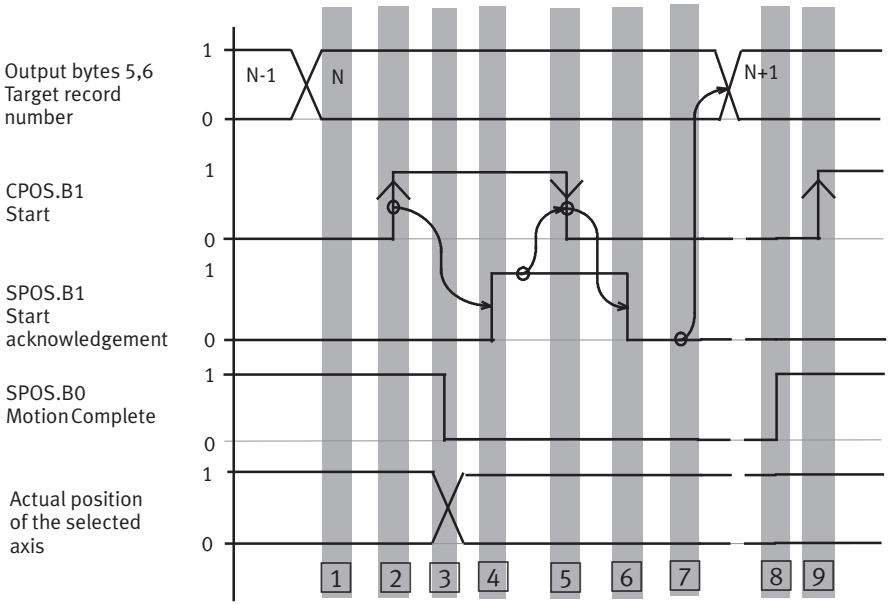


Fig. 3/1: Record start process

1 Set the desired record number in the controller's output data.

2 If ACK (SPOS.B1, acknowledgement) = 0, the controller can intitiate execution of the record with a rising edge at START (CPOS.B1).

3 The CPX-CMXX takes over the record number and starts positioning.
In the input data of the controller, the actual position of the selected axis is set and MC (SPOS.B2) reset.

4 The CPX-CMXX now signals with the rising edge at ACK (SPOS.B1) that the output data of the controller has been taken over and the positioning task is active.

5 The controller recognises the acknowledgement ACK (SPOS.B1) = 1 in its input data and resets START (CPOS.B1) in its output data.
Resetting of CPOS.START (B1) to 0 does not influence the positioning task.

6 With the resetting of ACK (SPOS.B1), the CPX-CMXX acknowledges the resetting of START (CPOS.B1).

7 After the controller detects ACK (SPOS.B1) = 0, it can write a new record number into its output data. The CPX-CMXX ignores this until the next start.

8 The CPX-CMXX sets SPOS.MC as soon as all axes involved have reached their target position.

9 The next positioning task can be started.

Notes:

– As soon as the controller detects the rising edge at ACK (SPOS.B1), it can assume that SPOS.MC is valid.

– If the actual position is within the tolerance of the target position, the handshake is carried out but the positioning task is not started. Motion Complete then stays set. The positioning task is ended immediately.

Typical causes of errors in applications:

– Referencing has not been carried out.

– Selection of an invalid record number or a record that has not been initialised.

– The target value lies outside the software end positions.

– Error in the record parameters, e.g. invalid record linking.

– Subsequent record with further record switching not initialised.

Problems in applications:

– Why does the CPX-CMXX not react to the rising edge at START (CPOS.B1)?
If the CPX-CMXX does not react to the rising edge at START, please check whether ACK (SPOS.B1) has really been reset. After the controller sets start = 0 (Fig. 3/1 6), it must wait for ACK (SPOS.B1) = 0 (Fig. 3/1 7). Otherwise, it may be that the time for START = 0 is too short to be recognised by the CPX-CMXX.

Overview of the parameters involved

Parameters involved	Description
Start	CPOS.B1 = 0 → 1: Start Jogging and homing have priority
Target record number	Output byte 5 and 6
Acknowledgement	SPOS.ACK = 0 → 1: Acknowledgement of start SPOS.MC = 0: Motion Complete

Tab. 3/2: Record selection parameters involved

3.2.2 Record linking

The record selection mode and direct mode make it possible to link several positioning tasks. This means that, starting at CPOS.START, various positioning records are automatically executed one after the other. With this, a positioning profile can be defined.

With the “record linking” positioning record parameter, the record number of the positioning record to be executed as next record is specified. This permits any jumping desired in the positioning record table.

If the value of the positioning record parameter “record linking” = 0, no further switching takes place.

If the value of the “record linking” positioning record parameter › 0, further switching occurs after the target value is reached, i.e. when the motion-complete condition is fulfilled (MC=1).

3.2.3 Changeover from direct mode to record selection mode

Warning
Danger of injury through uncontrolled movements of drive axes.

Through the direct operating mode, the random access memory of the CPX-CMXX can contain a positioning record table different from the positioning record table in the permanent memory of the CPX-CMXX. Switching from direct mode to record selection mode does not overwrite the positioning record table in the random access memory of the CPX-CMXX.

- Note that execution of positioning records can lead to undesired movements of the axis group.

3.3 Direct mode

The direct mode is an extension of the record selection operating mode.

All parameters of the positioning record table can be overwritten by the controller.

This function is normally used:

– for freely selected movement to positions

– if the target positions are unknown during planning or if they change frequently (several different work item positions).

Note
Changed positioning record parameters can be lost:
– after the power supply is switched off and back on; overwritten positioning record parameters are stored only in the volatile memory of the CPX-CMXX.
– through the FCT plug-in functions download and comparison of the FCT-plug in to the CPX-CMXX; the positioning records in the volatile and permanent memory are overwritten.

Note
The FCT plug-in functions upload and comparison from the CPX-CMXX to the FCT plug-in only access the positioning records in the permanent memory of the CPX-CMXX. The positioning records in the random access memory of the CPX-CMXX are not transferred to the FCT plug-in.

The positioning record parameters are written to the positioning record selected via the record register. Each record register corresponds to a parameter of the selected positioning record. The structure of the record register can be found in the following section.

The positioning records are started as in the record selection operating mode.

3.3.1 Structure of the record register

RegNo	Designation	Description	Value range	Unit
0	Record number	Selects the record for all coming register commands.	1 ... 1024	
1	Record control word	Bit 0 = 0: absolute positioning Bit 0 = 1: relative positioning Bit 1..4: Axes 1 ... 4 are controlled (PTP-synchronous) ous) in synchronisation Bit 5 = 0: Record not configured Bit 5 = 1: Record configured		
2	Target position axis 1		-2^{31} ... $2^{31}-1$	1/1000 mm or 1/1000 °
3	Target position axis 2		-2^{31} ... $2^{31}-1$	1/1000 mm or 1/1000 °
4	Target position axis 3		-2^{31} ... $2^{31}-1$	1/1000 mm or 1/1000 °
5	Target position axis 4		-2^{31} ... $2^{31}-1$	1/1000 mm or 1/1000 °
6	Speed axis 1	[1]	0 ... $2^{31}-1$	1/1000 mm/s or 1/1000 °/s

[1] Value can only be specified for positioning axes. With portal axes, the value is not active.

RegNo	Designation	Description	Value range	Unit
7	Speed axis 2	1)	0 ... 2^{31}-1	1/1000 mm/s or 1/1000 °/s
8	Speed axis 3	1)	0 ... 2^{31}-1	1/1000 mm/s or 1/1000 °/s
9	Speed axis 4	1)	0 ... 2^{31}-1	1/1000 mm/s or 1/1000 °/s
10	Acceleration axis 1	1)	0 ... 2^{31}-1	1/1000 mm/s^2 or 1/1000 °/s^2
11	Acceleration axis 2	1)	0 ... 2^{31}-1	1/1000 mm/s^2 or 1/1000 °/s^2
12	Acceleration axis 3	1)	0 ... 2^{31}-1	1/1000 mm/s^2 or 1/1000 °/s^2
13	Acceleration axis 4	1)	0 ... 2^{31}-1	1/1000 mm/s^2 or 1/1000 °/s^2
14	Speed path		0 ... 2^{31}-1	1/1000 mm/s or 1/1000 °/s
15	Acceleration path		0 ... 2^{31}-1	1/1000 mm/s^2 or 1/1000 °/s^2
16	Rounding radius	0: No rounding Note: Values greater than zero are rounding radii	0 ... 2^{31}-1	1/1000 mm or 1/1000 °
17	Record linking	0: No linking 1 ... 1024 The selected record is executed as next record	0 ... 1024	
18	Response time	Time from reaching the target position to the start of the linked record.	0 ... 2^{32}	ms
20	Deceleration axis 1		0 ... 2^{31}-1	1/1000 mm/s^2 or 1/1000 °/s^2

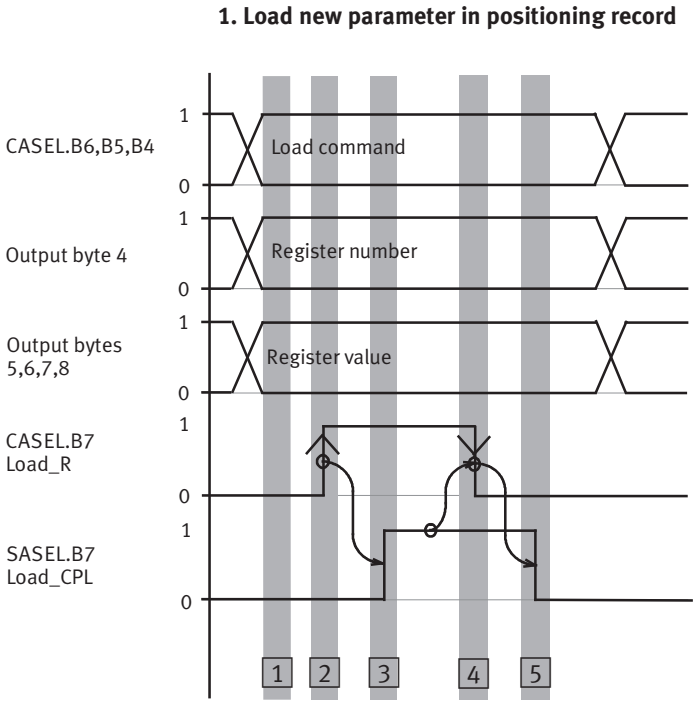
1) Value can only be specified for positioning axes. With portal axes, the value is not active.

RegNo	Designation	Description	Value range	Unit
21	Deceleration axis 2	1)	0 ... 2^{31}-1	1/1000 mm/s^2 or 1/1000 °/s^2
22	Deceleration axis 3	1)	0 ... 2^{31}-1	1/1000 mm/s^2 or 1/1000 °/s^2
23	Deceleration axis 4	1)	0 ... 2^{31}-1	1/1000 mm/s^2 or 1/1000 °/s^2
24	Deceleration path	1)	0 ... 2^{31}-1	1/1000 mm/s^2 or 1/1000 °/s^2

1) Value can only be specified for positioning axes. With portal axes, the value is not active.

Tab. 3/3: Record register for direct mode

3.3.2 Direct mode process



1 Place new register value in the output bytes 5, 6, 7, 8, the register number in byte 4 and the load action (e.g. here: =1 for write) in CASEL.B4, B5, B6.

2 With the rising edge at LOAD_R (CASEL.B7), the CPX-CMXX takes over the register value and writes it into the selected position record parameter.

3 The CPX-CMXX with rising edge at LOAD_CPL (SASEL.B7) reports that the write procedure is completed.

4 The controller takes back the LOAD_R signal as soon as it has detected the LOAD_CPL (SASEL.B7).

5 The CPX-CMXX takes back LOAD_CPL as soon as the controller takes away the LOAD_R signal.
The steps 1 to 5 are repeated until all desired positioning record parameters have been changed. The first command must always access register 0 to select the desired positioning record number.

Fig. 3/1: Load new parameter in positioning record

2. Record start

The process of record start is identical to the process of record start in the record selection operating mode, see chapter 3.2.1, Fig. 3/1.

All positioning records can be started, even the unchanged positioning records

Fault behaviour and diagnosis

Chapter 4

Contents

4.1 Classification of faults

We differentiate between the following types of fault:

– Warnings,

– Type 1 fault (axis group remains enabled),

– Type 2 fault (axis group is blocked).

The CPX-CMMX indicates errors or faults through appropriate fault messages or warnings. These can be evaluated via the following:

– status bytes, see section 2.5,

– diagnostic memory, see section 4.3,

– FCT, see online help for FCT.

4.1.1 Warnings

A warning provides the user with information that does not have any effect on the drive's behaviour.

Behaviour in the event of warnings

- Motor controller and end stage remain active,
- Positioning run is not interrupted,
- Start of a new positioning run is possible,
- The SCON.B2 (WARN) bit is set,
- If the cause of the warning disappears, the SCON.B2 bit is automatically deleted again.

Examples for causes of warnings

- Parameters cannot be written or read (examples: not permitted in the current operating status; invalid PNU, ...),
- Following error
- Drive has left the tolerance range after Motion Complete
- Slight control errors.

4.1.2 Fault type 1

If there is a fault, the required performance is not provided by a drive axis of the axis group. The status of the CPX-CMXX changes to the “Fault” status.

The “Fault” status can only be exited by switching off or through a positive edge at input CCON.B3 (RESET).

Behaviour in the event of type 1 faults

– The end stages of the motor controllers are **not** switched off

– The current positioning run is interrupted

– A new positioning run is not possible

– The SCON.B3 (FAULT) bit is set.

Examples of causes of type 1 faults

– Software end positions are damaged

– Time overrun with Motion Complete (Timeout).

4.1.3 Fault type 2

If there is a fault, the required performance is not provided by a drive axis of the multi-axis system. The status of the CPX-CMXX changes to the “Fault” status.

The “Fault” status can only be exited by switching off or through a positive edge at input CCON.B3 (RESET).

Behaviour in the event of type 2 faults

– The end stages of the motor controllers are switched off

– The current positioning run is interrupted

– A new positioning run is not possible

– The SCON.B3 (FAULT) bit is set.

Examples of causes of type 2 faults

– Load voltage is missing (e.g. if emergency off has been implemented)

– Hardware fault (e.g. measurement system fault, bus fault)

– Impermissible operating mode change

4.2 Diagnostic status

The diagnostic status, parameter 220, contains the codes of the last diagnostic messages. In addition, extended device faults and the parameters 205, 207 ... 209 can be read. These parameters are described in section 5.4.3.

4.3 Diagnostic memory

The diagnostic memory contains the codes of the last diagnostic messages that occurred. It is backed up if possible in the event of power failure. If the diagnostic memory is full, the oldest element will be overwritten (ring-memory principle).

The diagnostic memory includes the parameters 200 ... 203. The diagnostic memory is configured with parameter 204. These parameters are described in section 5.4.3.

4.4 Fault numbers

Fault no.	Mal-function type	Designation (cause)	Error handling	CPX error category
1	1	Axis not referenced	• Reference the axis. Non-referenced axes can be – moved in the jog mode if PNU522 bit 6=1 and the motor controller supports the speed control operating mode; see also section 3.1. – moved by hand, after the motor controller enable has been switched off.	101
2	1	Target position outside the permitted positioning range (Selected target position lies outside the SW end positions of the axis or selected record was not configured)	• Check the target position and SW end positions. SW end positions of the FCT plug-in CPX-CMXX differentiate themselves from the SW end positions or HW end positions of the axes. • Configure the positioning record.	101
3	1	Positioning record invalid (Number of the selected positioning record outside the range 1 ... 1024)	• Start a valid positioning record.	102
4	2	Positioning record memory not initialised (Positioning records could not be written from file {permanent memory} into internal cache {volatile memory})	• Consult your local Festo Service.	104
5	2	Invalid FHPP-MAX command combination (Two FHPP-MAX commands were set simultaneously, e.g. START and LOAD_R; this is not permitted)	• Rework the control program.	105

Fault no.	Mal-function type	Designation (cause)	Error handling	CPX error category
8	2	Axis not initialised (Either the device is factory-new or an axis could not be initialised)	1. Determine the specific error in the malfunction buffer of the FCT plug-in. 2. Check and correct the configuration in the FCT plug-in accordingly.	100
9	2	Axis is not enabled	• Enable the axis.	107
10	2	Error active (Positioning task is sent although another error is active)	• Acknowledge the error.	101
11	Warning	Homing interrupted (Positioning task is sent to axis while homing is active)	• Send a positioning task only when the homing run has been ended.	0
12	Warning	Positioning interrupted (positioning task is sent to axis while positioning record is active)	• Send a positioning task only when the current positioning record has been carried out.	0
13	Warning	Reset interrupted (Positioning task is sent to axis while reset is active)	• Send a positioning task only when reset has been ended.	0
14	Warning	Axis stopped (Positioning task is sent to axis while axis stops)	• Send a positioning task only when the stop process has been ended.	0
26	Warning	Speed cannot be achieved (Limit values of the axes are incorrect or acceleration path is too short)	• Check the positioning record data and the axis parameterisation.	0
27	Warning	Speed ‹ minimum speed	• Check the positioning record data and the axis parameterisation.	103
28	Warning	Axis acceleration ‹ minimum acceleration	• Check the positioning record data and the axis parameterisation.	0

Fault no.	Mal-function type	Designation (cause)	Error handling	CPX error category
29	Warning	Axis deceleration < minimum deceleration	• Check the positioning record data and the axis parameterisation.	0
30	1	Time overrun during command execution (Axis command could not be ended in the specified time)	• Check the motor controller and the axis.	101
31	Warning	Following error (Controller reports following error: following error window or time out)	• Check the positioning record data, motor controller settings and the axis.	0
32	2	Error in homing run (Interruption of the homing run (Halt bit), both limit switches actuated simultaneously, search section traveled larger than positioning space)	• Check the motor controller and the axis.	107
36	2	Timeout with Stop command	• Check the motor controller and the axis.	107
37	2	Timeout when changing the operating mode	• Check the motor controller.	107
42	2	Time overrun during activation	• Check the motor controller.	107
43	2	Axis status undefined (DS402)	• Check the motor controller.	107
44	1	Axis in the FAULT condition (_REACTION_ACTIVE)	1. Check the motor controller status – Additional messages can be found directly at the motor controller: – Diagnostic memory – LED – Display 2. Read out the status with FCT plug-in of the motor controller or via display.	107

Fault no.	Mal-function type	Designation (cause)	Error handling	CPX error category
45	Warning	Warning (Warning bit of the motor controller is set. A direction of rotation is blocked, since the limit switch has been actuated)	• Check the positioning record data and the axis.	0
46	2	Time overrun during deactivation	• Check the motor controller.	107
47	2	Command for inactive axis	• Check the triggering or configuration.	101
48	2	Command for inactive group	• Check the triggering or configuration.	101
49	2	Error during activation of the “Interpolated position mode”	• Check the motor controller.	107
56	2	Time overrun when starting the CAN node	• Check the CAN bus line and terminating resistor.	100
57	2	CAN ID not present (No CANOpen device with CAN ID of the axis in controller configuration)	• Check the CAN ID.	100
58	2	No live signal from FCT (Connection between FCT plug-in and CPX-CMXX interrupted (network connection, FCT ended))	• Check the connection.	105
59	1	Error when sending/receiving from SDO	• Check the CAN bus and motor controller.	105
60	1	Status error when sending/receiving from SDO	• Check the gantry parameterisation, CAN bus and motor controller.	105
61	1	Time overflow when sending/receiving from SDO	• Check the CAN bus and motor controller.	105

Fault no.	Mal-function type	Designation (cause)	Error handling	CPX error category
62	2	Invalid device type recognised	• Check or change the configuration. • Close the device configured in the FCT plug-in.	100
63	2	Invalid product code recognised	• Check or change the configuration. • Close the device configured in the FCT plug-in.	100
64	2	Invalid firmware recognised	• Check or change the configuration. • Close the device configured in the FCT plug-in.	100
65	Warning	Warning: Recognised device is not completely supported	The warning serves as information that possibly not all functions of the device are supported; operation is still possible.	0
66	2	Error when resetting a node	• Check the CAN bus and the configuration of the node involved.	105
67	2	Internal error at node start	• Consult your local Festo Service.	105
68	2	Serious internal error	• Consult your local Festo Service.	105
74	2	Opening of positioning records failed (Error when opening file in CPX-CMXX)	• Consult your local Festo Service.	104
75	2	Reading of positioning records failed (Error when reading file in CPX-CMXX)	• Consult your local Festo Service.	104

Fault no.	Mal-function type	Designation (cause)	Error handling	CPX error category
76	2	Closing of positioning records failed (Error when closing file in CPX-CMXX)	• Consult your local Festo Service.	104
77	2	Writing of data failed (Error when writing to file in CPX-CMXX)	• Consult your local Festo Service.	104
121	2	CAN bus offline (No participant was recognised at the CAN bus)	• Check the CAN bus line and terminating resistor.	71
125	2	CAN bus error switch-off (Communication errors were detected at the CAN bus)	• Check the CAN bus line and terminating resistor.	71
126	2	CAN bus node monitoring (A CAN bus participant has failed)	• Check the participant.	71

Tab. 4/4: Error messages of the CPX-CMXX

4.5 Diagnostics using FHPP-MAX status bytes

The CPX-CMXX supports the following diagnostics options using FHPP-MAX status bytes; see section 2.5:

– SCON.B2 (WARN) – Warning

– SCON.B3 (FAULT) – Fault

– Byte 4 – Fault number

Parameters

Chapter 5

Contents

5.1 FHPP_MAX General Parameter Structure

The CPX-CMXX contains a parameter set with the following structure.

Complete parameterisation of an axis group as well as changing of the motor types, number of axes or portal configuration is only possible over FCT.

Group	Indices	Description
Device data	100 ... 199	Device identification and device-specific settings, version numbers, identifier words, etc.
Diagnostics Memory	200 ... 299	Memory for diagnostic events: fault numbers, fault time, incoming/outgoing event.
Process data	300 ... 399	Current nominal and actual values, local I/Os, status data etc.
Project data	500 ... 599	Fundamental project settings. Maximum speeds and accelerations etc. -› Parameters are the basis for the position record table
Factor group	600 ... 699	Parameter for scaling of the target and actual values
Axis data axis 1 Electrical drives	1000 ... 1099	Axis-specific parameters for electric drives:
Axis data axis 2 Electrical drives	1100 ... 1199	Axis-specific parameters for electric drives
Axis data axis 3 Electrical drives	1200 ... 1299	Axis-specific parameters for electric drives
Axis data axis 4 Electrical drives	1300 ... 1399	Axis-specific parameters for electric drives

Tab. 5/1: Parameter structure

Parameter classes	Attribute / use
Simple variable (Var)	Contains only one value. Its significance, limits, unit etc. differentiate it from other simple variables. The subindex does not have a function.
Array	Contains multiple values that all have the same significance, the same limits, the same unit, etc. Example: Max. speed (PNU 502). The elements in the array are addressed using the subindex.
Struct/Record	Compilation of several single variables with different limit values etc.

Tab. 5/2: Parameter classes

5.2 Access protection

5.2.1 Access via controller and FCT

Simultaneous operation through controller and FCT is not possible.

You can block the takeover of the drive control over the CPX-CMXX through the FCT with bit CCON.B5. The status bit SCON.B5 (drive control by FCT) shows whether the drive control over the CPX-CMXX is exercised through the FCT.

Preventing FCT operation: CCON.B5 (LOCK)

By setting the CCON.B5 control bit, the controller prevents the FCT from taking over drive control. So if the LOCK is set, FCT cannot write parameters or control the CPX-CMXX, execute homing etc.

Program the controller so that it enables this only through an appopriate user action. This generally causes exit from automatic operation. As a result, you as controller programmer can ensure that the controller always knows when it has control over the CPX -CMXX.

Important: the block is active if the CCON.B5 has a logic 1. It therefore does not need to be set compulsorily. If you do not need this type of locking, you can always leave it at 0.

Acknowledgment, drive control by FCT: SCON.B5 (LOCK)

This bit informs the controller that the CPX-CMXX is controlled through the FCT and the controller has no more control over the CPX-CMXX. This bit does not need to be evaluated. The controller can react by transferring to stop or manual operation.

5.3 Overview of parameters

The following overview (Tab. 5/3) shows the FHPP-MAX parameters

The parameters are described in sections 5.4.2 to 5.4.10.

Name	FHPP-MAX			
	PNU	Subind.	Class	Type
Device data (see section 5.4.2)				
Manufacturer Hardware Version	100	-	Var	uint16
Manufacturer Firmware Version	101	-	Var	uint16
Version FHPP-MAX	102	-	Var	uint16
Supported drive modes	112	-	Var	uint32
Controller serial number	114	1 ... 12	Array	char
Manufacturer device name	120	1 ... 255	Array	char
User device name	121	1 ... 255	Array	char
Drive manufacturer	122	1 ... 18	Array	char
Data Memory Control After “Load data” and “Store data”, the values of the parameters are reset to zero	127	1 ... 3	Array	uint8
Password PC/ Diag	130	1 ... 8	Array	char
Axis 1 Name	180	1 ... 128	Array	char
Axis 2 Name	181	1 ... 128	Array	char
Axis 3 Name	182	1 ... 128	Array	char
Axis 4 Name	183	1 ... 128	Array	char
Diagnostics (see section 5.4.3)				
Diagnostic event	200	1 ... 16	Array	uint8
Fault number	201	1 ... 16	Array	uint8
Time stamp	202	1 ... 16	Array	uint32
Additional Data	203	1 ... 16	Array	uint32
Diagnostics Memory Parameter	204	1 ... 4	Array	uint8
Extended Device Error Axis 1	205	-	Var	uint16
Extended Device Error Axis 2	207	-	Var	uint16
Extended Device Error Axis 3	208	-	Var	uint16
Extended Device Error Axis 4	209	-	Var	uint16

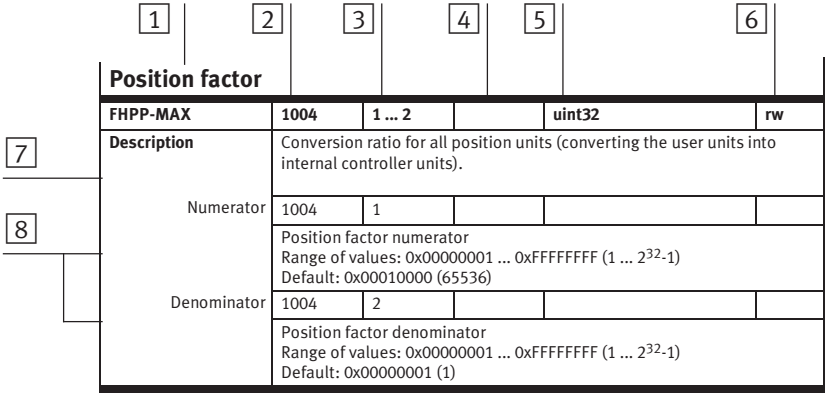
Name	FHPP-MAX			
	PNU	Subind.	Class	Type
Diagnostics Status	220	1 ... 5	Array	int32
CANopen tunnel axis 1	230	1 ... 3	Var	uint32
CANopen tunnel axis 2	231	1 ... 3	Var	uint32
CANopen tunnel axis 3	232	1 ... 3	Var	uint32
CANopen tunnel axis 4	233	1 ... 3	Var	uint32
Processing data (see section 5.4.4)				
Position Values	300	1 ... 12	Array	int32
Maintenance Parameter	305	1 ... 3	Array	uint32
Project data (see section 5.4.5)				
Software End Position s	501	1 ... 8	Array	int32
Max. Speed	502	1 ... 4	Array	uint32
Max. Acceleration	503	1 ... 4	Array	uint32
Max. Acceleration jerk	504	1 ... 4	Var	uint32
Max. Deceleration jerk	505	1 ... 4	Var	uint32
Max. Deceleration (Max. permissible deceleration)	507	1 ... 4	Var	uint32
Jog Acceleration	508	1 ... 4	Var	uint32
Jog Deceleration	509	1 ... 4	Var	uint32
FHPP-MAX Features Supported	522	-	Var	uint32
Factor group (see section 5.4.6)				
Position Notation Index	600	1 ... 4	Array	int8
Position Dimension Index	601	1 ... 4	Array	uint8
Velocity Notation Index	602	1 ... 4	Array	int8
Velocity Dimension Index	603	1 ... 4	Array	uint8
Acceleration Notation Index	604	1 ... 4	Array	int8
Acceleration Dimension Index	605	1 ... 4	Array	uint8
Jerk Notation Index	608	1 ... 4	Array	int8
Jerk Dimension Index	609	1 ... 4	Array	uint8

Axis data electric drives				
Axis data electric drives – axis 1 (see section 5.4.7)				
Polarity	1000	-	Var	uint8
Position factor	1004	1 ... 2	Array	uint32
Controller type	1030	1 ... 128	Array	char
Kinematic Type	1031	-	Var	uint8
Usage	1072	-	Var	uint8
Axis data electric drives – axis 2 (see section 5.4.8)				
Polarity	1100	-	Var	uint8
Position factor	1104	1 ... 2	Array	uint32
Controller type	1130	1 ... 128	Array	char
Kinematic Type	1131	-	Var	uint8
Usage	1172	-	Var	uint8
Axis data electric drives – axis 3 (see section 5.4.9)				
Polarity	1200	-	Var	uint8
Position Factor	1204	1 ... 2	Array	uint32
Controller type	1230	1 ... 128	Array	char
Kinematic Type	1231	-	Var	uint8
Usage	1272	-	Var	uint8
Axis data electric drives – axis 4 (see section 5.4.10)				
Polarity	1300	-	Var	uint8
Position factor	1304	1 ... 2	Array	uint32
Controller type	1330	1 ... 128	Array	char
Kinematic Type	1331	-	Var	uint8
Usage	1372	-	Var	uint8

Tab. 5/3: Overview of FHPP-MAX parameters

5.4 Descriptions of parameters according to FHPP-MAX

5.4.1 Representation of the parameter entries



Position factor					
FHPP-MAX	**1004**	**1 ... 2**		**uint32**	**rw**
Description	Conversion ratio for all position units (converting the user units into internal controller units).				
Numerator	1004	1			
	Position factor numerator Range of values: 0x00000001 ... 0xFFFFFFFF (1 ... 2^{32}-1) Default: 0x00010000 (65536)				
Denominator	1004	2			
	Position factor denominator Range of values: 0x00000001 ... 0xFFFFFFFF (1 ... 2^{32}-1) Default: 0x00000001 (1)				

1 Name of the parameter in English

2 PNU (parameter number)

3 Subindices of the parameter with arrays.
If the parameter is a simple variable, this field remains empty.

4 Parameter class

5 Variable type of the parameter / subindices.

6 Read/write permission:
ro = read only,
rw = read and write,
wo = write only; read access returns the status, not the content

7 Description of the parameter
Global parameters apply for both axis groups!

8 Name and description of the subindices for arrays.

Fig. 5/1: Representation of the parameter entries

5.4.2 Device data

Manufacturer Hardware Version					
FHPP-MAX	**100**	**–**	**Var**	**uint16**	**ro**
Description	Global parameter Coding of the hardware version, specification in BCD: xxyy (xx = main version, yy = secondary version)				

Manufacturer Firmware Version					
FHPP-MAX	**101**	**–**	**Var**	**uint16**	**ro**
Description	Global parameter Coding of the firmware version, specification in BCD: xxyy (xx = main version, yy = secondary version)				

Version FHPP-MAX					
FHPP-MAX	**102**	**–**	**Var**	**uint16**	**ro**
Description	Global parameter Version number of the FHPP-MAX, specification in BCD: xxyy (xx = main version, yy = secondary version)				

Supported drive modes					
FHPP-MAX	**112**	**–**	**Var**	**uint32**	**ro**
Description	Global parameter The parameter describes the supported modes of the motor controllers. Bit control mode 0 Position control (Profile position mode – pp) 1 Velocity control (Velocity mode – vl) 2 Velocity control (profile velocity mode – pv) 3 Pressure / power control (profile torque mode – tq) 4 Reserved (reserved) 5 Homing (homing mode – hm) 6 Interpolated position control (Interpolated positioning mode – ip) 7-15 Reserved (reserved) 16-31 Manufacturer-specific (customer-specific) CPX-CMXX supports the following control modes Position control (pp) bit 0 = 1 Velocity control (vl) bit 1 = 1 Homing run (hm) bit 5 = 1 The value of the parameter is 35.				

Serial Number					
FHPP-MAX	**114**	**Length is dependent on content**	**Array**	**char**	**ro**
Description	Global parameter Serial number for unique identification of the CPX-CMXX. Characters not used are filled with zero (0x00=‘\0’).				

Manufacturer Device Name					
FHPP-MAX	**120**	**Length is dependent on content**	**Array**	**char**	**ro**
Description	Global parameter Designation of the device “CPX-CMXX” Characters not used are filled with zero (0x00=‘\0’).				

User Device Name					
FHPP-MAX	**121**	**1 ... 255**	**Array**	**char**	**rw**
Description	Global parameter User's designation of the CPX-CMXX (ASCII, 7 bit). Characters not used are filled with zero (0x00='\0').				

Drive Manufacturer					
FHPP-MAX	**122**	**Length is dependent on content**	**Array**	**char**	**ro**
Description	Global parameter Name of the manufacturer (ASCII, 7 bit). "Festo AG & Co. KG" Characters not used are filled with zero (0x00='\0').				

Data Memory Control [1)]					
FHPP-MAX	**127**	**1 ... 3**	**Array**	**uint8**	**rw**
Description	With this parameter, the volatile RAM data can be written to the non-volatile flash memory.				
Load data	127	1			
	Load the volatile memory (RAM) with the data of the non-volatile storage medium (flash) Value 0x10: Load				
Store data	127	2			
	Store the data of the volatile memory (RAM) in the non-volatile storage area (flash) Value 0x01: Save				
No function	127	3			
	Reserved				

[1)] After a load or storage process, the parameter is automatically reset to zero.

Password PC/Diag					
FHPP-MAX	**130**	**1 ... 8**	**Array**	**char**	**wo**
		–	Var	uint8	ro
Description	Global parameter The user can use a password to protect parameter changing through the PC via the diagnostics interface. Blocked: – changing of parameters Permitted: – display of parameters – project upload – display of actual values, target values, diagnostic data If a password is defined, the connection with the FCT is only built up after the password has been correctly entered. If read access takes place on parameter 130, the status of the current password protection is displayed. Value Meaning 0 No access protection active 1 Access protection active, access blocked 2 Access protection active, access enabled				

Axis 1 Name					
FHPP-MAX	**180**	**1 ... 128**	**Array**	**char**	**rw**
Description	Individual name for axis 1. Note: In a multi-axis system, the axis name permits a textual description of each axis. The device name is present only once.				

Axis 2 Name					
FHPP-MAX	**181**	**1 ... 128**	**Array**	**char**	**rw**
Description	Individual name for axis 2. Note: In a multi-axis system, the axis name permits a textual description of each axis. The device name is present only once.				

Axis 3 Name					
FHPP-MAX	**182**	**1 ... 128**	**Array**	**char**	**rw**
Description	Individual name for axis 3. Note: In a multi-axis system, the axis name permits a textual description of each axis. The device name is present only once.				

Axis 4 Name					
FHPP-MAX	**183**	**1 ... 128**	**Array**	**char**	**rw**
Description	Individual name for axis 4. Note: In a multi-axis system, the axis name permits a textual description of each axis. The device name is present only once.				

5.4.3 Diagnostics

For a description of how the diagnostic memory functions, see section 4.3.

Diagnostic Event					
FHPP-MAX	**200**	**1 ... 16**	**Array**	**uint8**	**ro**
Description	Global parameter Type of fault or diagnostic information saved in the diagnostic memory. Displays whether an incoming or outgoing fault is saved. Value Type of diagnostic event 0x00 (0) No fault (or fault message deleted) 0x01 (1) Incoming fault = occurrence of the fault 0x02 (2) Outgoing fault = acknowledgement of the fault 0x03 (3) Switch-off 0x04 (4) Overflow of the internal time stamp				
Event 1	200	1			
	Type of latest / current diagnostic message				
Event 2	200	2			
	Type of second saved diagnostic message				
Event ...	200	...			
	...				

Fault Number					
FHPP-MAX	**201**	**1 ... 16**	**Array**	**uint8**	**ro**
Description	Global parameter Fault number stored in the diagnostic memory, serves to identify the fault. See section 4.4 for error numbers.				
Event 1	201	1			
	Latest / current diagnostic message				
Event 2	201	2			
	2. Saved diagnostic message				
Event ...	201	...			
	...				

Time Stamp

FHPP-MAX	202	1 ... 16	Array	uint32	ro
Description	Global parameter Time of the diagnostics event in milliseconds after switch-on. In case of overflow, the time stamp jumps from 0xFFFFFFFF to 0.				
Event 1	202	1			
	Time of the latest / current diagnostic message				
Event 2	202	2			
	Time of the second saved diagnostic message				
Event ...	202	...			
	...				

Additional Data

FHPP-MAX	203	1 ... 16	Array	uint32	ro

Description

Global parameter
The groups and axes are encoded in the additional information. This identifies the source of the error.

Bit 9	8	2	1	0	Source of the diagnostic event
0	0	0	0	0	CPX-CMXX
0	1	0	0	0	Group 1
0	1	0	0	1	Group 1, axis 1
0	1	0	1	0	Group 1, axis 2
0	1	0	1	1	Group 1, axis 3
0	1	1	0	0	Group 1, axis 4
1	0	0	0	0	Group 2
1	0	0	0	1	Group 2, axis 1
1	0	0	1	0	Group 2, axis 2
1	0	0	1	1	Group 2, axis 3
1	0	1	0	0	Group 2, axis 4

Bits 3 ... 7 reserved

Diagnostics Memory Parameter					
FHPP-MAX	**204**	**1 ... 4**	**Array**	**uint8**	**rw/ro**
Description	Global parameter Configuration of the diagnostic memory.				
Fault type	204	1			rw
	Recording of faults. 0x01 (1) Record all faults 0x02 (2): Record only incoming faults (default)				
Resolution	204	2			rw
	Resolution time stamp: 0x01 (1): Resolution time stamp 10 ms (default) 0x02 (2): Resolution time stamp 1 ms				
Clear memory	204	3			rw
	Clear diagnostic memory: Write with value = 1 deletes the memory Reading is always answered with value = 0				
Number of entries	204	4			ro
	Read out the number of valid entries in the diagnostic memory. Writing is not permitted! Range of values: 0x00 ... 0x0F (0 ... 15)				

Extended Device Error Axis 1					
FHPP-MAX	**205**	**–**		**uint16**	**ro**
Description	Read-out of the fault numbers axis 1.				

Extended Device Error Axis 2					
FHPP-MAX	**207**	**–**		**uint16**	**ro**
Description	Read-out of the fault numbers axis 2.				

Extended Device Error Axis 3					
FHPP-MAX	**208**	**–**		**uint16**	**ro**
Description	Read-out of the fault numbers axis 3.				

Extended Device Error Axis 4					
FHPP-MAX	**209**	**–**		**uint16**	**ro**
Description	Read-out of the fault numbers axis 4.				

Diagnosis Status					
FHPP-MAX	**220**	**1 ... 5**	**Array**	**int32**	**ro**
Description	All faults and warnings present can be read out with this object. While the diagnostic buffers show the history, here it can be determined which faults and warnings are present.				
	220	1			
	Fault / warning axis group				
	220	2			
	Fault / warning axis 1				
	220	3			
	Fault / warning axis 2				
	220	4			
	Fault / warning axis 3				
	220	5			
	Fault / warning axis 4				

5.4.4 Process data

CANopen tunnel axis 1					
FHPP-MAX	**230**	**1 ... 3**	**Var**	**uint32**	**rw/ro**
Description	This object allows read access to the CANopen objects of axis 1 in the group concerned.				
	230	1			rw
	Index + subindex of the CANopen object				
	230	2			ro
	Value: Value of the object.				
	230	3			ro
	Value: Cancellation code in the case of error				

CANopen tunnel axis 2					
FHPP-MAX	**231**	**1 ... 3**	**Var**	**uint32**	**rw/ro**
Description	This object allows read access to the CANopen objects of axis 2 in the group concerned.				
	231	1			rw
	Index + subindex of the CANopen object				
	231	2			ro
	Value: Value of the object.				
	231	3			ro
	Value: Cancellation code in the case of error				

CANopen tunnel axis 3					
FHPP-MAX	**232**	**1 ... 3**	**Var**	**uint32**	**rw/ro**
Description	This object allows read access to the CANopen objects of axis 3 in the group concerned.				
	232	1			rw
	Index + subindex of the CANopen object				
	232	2			ro
	Value: Value of the object.				
	232	3			ro
	Value: Cancellation code in the case of error				

CANopen tunnel axis 4					
FHPP-MAX	**233**	**1 ... 3**	**Var**	**uint32**	**rw/ro**
Description	This object allows read access to the CANopen objects of axis 4 in the group concerned.				
	233	1			rw
	Index + subindex of the CANopen object				
	233	2			ro
	Value: Value of the object.				
	233	3			ro
	Value: Cancellation code in the case of error				

Position Values					
FHPP-MAX	**300**	**1 ... 12**	**Array**	**int32**	**ro**
Description	Current position values.				
Axis 1					
Actual Position	300	1			
	Current actual position axis 1				
Nominal Position	300	2			
	Current target position axis 1				
Actual Deviation	300	3			
	Current deviation axis 1				
Axis 2					
Actual Position	300	4			
	Current actual position axis 2				
Nominal Position	300	5			
	Current target position axis 2				
Actual Deviation	300	6			
	Current deviation axis 2				
Axis 3					
Actual Position	300	7			
	Current actual position axis 3				
Nominal Position	300	8			
	Current target position axis 3				

Position Values					
Actual Deviation	300	9			
	Current deviation axis 3				
Axis 4					
Actual Position	300	10			
	Current actual position axis 4				
Nominal Position	300	11			
	Current target position axis 4				
Actual Deviation	300	12			
	Current deviation axis 4				

Maintenance Parameter					
FHPP-MAX	**305**	**1 ... 3**		**uint32**	**ro**
Description	Global parameter Support of the user with information about performance.				
Electric drives					
Number of cycles	305	1			
	Reserved, fixed at 0				
Total path of the axis group	305	2			
	Reserved, fixed at 0				
Operating time counter	305	3			
	Reserved, fixed at 0				

5.4.5 Project data

Software End Positions					
FHPP-MAX	**501**	**1 ... 8**	**Array**	**int32**	**rw**
Description	Lower and upper software end positions A setpoint specification (position) outside the end positions is not permitted and will lead to a fault. The offset to the axis zero point is entered. Plausibility rule: min. limit ≤ max. limit Wertebereich: 0x80000000 ... 0x7FFFFFFF (-2^{31} ... $+2^{31}-1$)				
Axis 1					
Lower Limit	501	1			
	Lower software end position axis 1 Default: 0xFFFF8000 (-32768)				
Upper Limit	501	2			
	Upper software end position axis 1 Default: 0x00008000 (32768)				
Axis 2					
Lower Limit	501	3			
	Lower software end position axis 2 Default: 0xFFFF8000 (-32768)				
Upper Limit	501	4			
	Upper software end position axis 2 Default: 0x00008000 (32768)				
Axis 3					
Lower Limit	501	5			
	Lower software end position axis 3 Default: 0xFFFF8000 (-32768)				
Upper Limit	501	6			
	Upper software end position axis 3 Default: 0x00008000 (32768)				
Axis 4					
Lower Limit	501	7			
	Lower software end position axis 4 Default: 0xFFFF8000 (-32768)				
Upper Limit	501	8			
	Upper software end position axis 4 Default: 0x00008000 (32768)				

Max. Speed

FHPP-MAX	502	1 ... 4	Var	uint32	rw
Description	Max. permitted speed. This value limits the speed in all operating modes. Range of values: 0x00000001 ... 0xFFFFFFFF (1 ... $+2^{32-1}$) Default: 0x00000001				
Axis 1	502	1			
	Max. permitted speed axis 1.				
Axis 2	502	2			
	Max. permitted speed axis 2.				
Axis 3	502	3			
	Max. permitted speed axis 3.				
Axis 4	502	4			
	Max. permitted speed axis 4.				

Max. Acceleration

FHPP_Max	503	1 ... 4	Var	uint32	rw
Description	Max. permitted acceleration. Range of values: 0x00000001 ... 0xFFFFFFFF (1 ... $+2^{32}$-1) Default: 0x00000001 (1)				
Axis 1	503	1			
	Max. permitted acceleration axis 1.				
Axis 2	503	2			
	Max. permitted acceleration axis 2.				
Axis 3	503	3			
	Max. permitted acceleration axis 3.				
Axis 4	503	4			
	Max. permitted acceleration axis 4.				

Max. Acceleration jerk

FHPP_Max	504	1 ... 4	Var	uint32	rw
Description	Max. permitted acceleration jerk. Range of values: 0x00000001 ... 0xFFFFFFFF (1 ... $+2^{32}$-1) Default: 0x00000001 (1)				
Axis 1	504	1			
	Max. permitted acceleration jerk axis 1.				
Axis 2	504	2			
	Max. permitted acceleration jerk axis 2.				
Axis 3	504	3			
	Max. permitted acceleration jerk axis 3.				
Axis 4	504	4			
	Max. permitted acceleration jerk axis 4.				

Max. Deceleration jerk

FHPP_Max	505	1 ... 4	Var	uint32	rw
Description	Max. permissible deceleration. Range of values: 0x00000001 ... 0xFFFFFFFF (1 ... $+2^{32}$-1) Default: 0x00000001 (1)				
Axis 1	505	1			
	Max. permitted deceleration jerk axis 1.				
Axis 2	505	2			
	Max. permitted deceleration jerk axis 2.				
Axis 3	505	3			
	Max. permitted deceleration jerk axis 3.				
Axis 4	505	4			
	Max. permitted deceleration jerk axis 4.				

Max. Deceleration

FHPP_Max	507	1 ... 4	Var	uint32	rw
Description	Max. permissible deceleration. Range of values: 0x00000001 ... 0xFFFFFFFF (1 ... $+2^{32}$-1) Default: 0x00000001 (1)				
Axis 1	507	1			
	Max. permitted deceleration axis 1.				
Axis 2	507	2			
	Max. permitted deceleration axis 2.				
Axis 3	507	3			
	Max. permitted deceleration axis 3.				
Axis 4	507	4			
	Max. permitted deceleration axis 4.				

Jog Acceleration

FHPP_Max	508	1 ... 4	Var	uint32	rw
Description	Acceleration in jog mode. Range of values: 0x00000001 ... 0xFFFFFFFF (1 ... $+2^{32}$-1) Default: Max. Acceleration (PNU 503)				
Axis 1	508	1			
	Acceleration in Jog Mode axis 1.				
Axis 2	508	2			
	Acceleration in Jog Mode axis 2.				
Axis 3	508	3			
	Acceleration in Jog Mode axis 3.				
Axis 4	508	4			
	Acceleration in Jog Mode axis 4.				

Jog Deceleration

FHPP_Max	509	1 ... 4	Var	uint32	rw
Description	Deceleration in jog mode. Range of values: 0x00000001 ... 0xFFFFFFFF (1 ... $+2^{32}$-1) Default: Max. Deceleration (PNU 507)				
Axis 1	509	1			
	Deceleration in Jog Mode axis 1.				
Axis 2	509	2			
	Deceleration in Jog Mode axis 2.				
Axis 3	509	3			
	Deceleration in Jog Mode axis 3.				
Axis 4	509	4			
	Deceleration in Jog Mode axis 4.				

FHPP-MAX Features Supported

FHPP-MAX	522	–	Var	uint32	rw
Description	Global parameter With this parameter, specific features of the FHPP-MAX can be activated or deactivated. Bit 0 = 0 “Halt” status is permitted. Bit 0 = 1 “Halt” status is not supported. Bit 1 ... 3 reserved. Bit 4 = 0 Inactive axes are deactivated (DISABLED). Bit 4 = 1 Inactive axes remain activated (ENABLED). Bit 5 = 0 Parallel homing not possible. Bit 5 = 1 Parallel homing possible. Bit 6 = 0 Jog mode via velocity mode off. Parallel jog mode not possible Bit 6 = 1 Jog mode via speed mode on. Parallel jog mode possible Bit 7 ... 31 reserved. Default: PNU 522 = 1, i.e. bit 0 = 1.				

5.4.6 Factor group

Position Notation Index					
FHPP-MAX	**600**	**1 ... 4**	**Array**	**int8**	**ro**
Description	Tens' exponent for position values = –6 for linear axes = –3 for rotative axes Default: = –6				
Axis 1	600	1			
	Tens' exponent for position values axis 1				
Axis 2	600	2			
	Tens' exponent for position values axis 2				
Axis 3	600	3			
	Tens' exponent for position values axis 3				
Axis 4	600	4			
	Tens' exponent for position values axis 4				

Position dimension index					
FHPP-MAX	**601**	**1 ... 4**	**Array**	**uint8**	**ro**
Description	Unit of measurement for position values = 0x01: [m] for linear axes = 0x41: [°] for rotative axes Default: = 0x01				
Axis 1	601	1			
	Unit of measurement for position values axis 1				
Axis 2	601	2			
	Unit of measurement for position values axis 2				
Axis 3	601	3			
	Unit of measurement for position values axis 3				
Axis 4	601	4			
	Unit of measurement for position values axis 4				

Velocity Notation Index

FHPP-MAX	602	1 ... 4	Array	int8	ro
Description	Tens' exponent for speed values = –6 for linear axes = –3 for rotative axes Default: = –6				
Axis 1	602	1			
	Tens' exponent for speed values axis 1				
Axis 2	602	2			
	Tens' exponent for speed values axis 2				
Axis 3	602	3			
	Tens' exponent for speed values axis 3				
Axis 4	602	4			
	Tens' exponent for speed values axis 4				

Velocity Dimension Index

FHPP-MAX	603	1 ... 4	Array	uint8	ro
Description	Unit of measurement for speed values = 0xA6 [m/sec] for linear axes = 0xF2 [°/sec] for rotative axes Default: = 0xA6				
Axis 1	603	1			
	Unit of measurement for speed values axis 1				
Axis 2	603	2			
	Unit of measurement for speed values axis 2				
Axis 3	603	3			
	Unit of measurement for speed values axis 3				
Axis 4	603	4			
	Unit of measurement for speed values axis 4				

Acceleration Notation Index					
FHPP-MAX	**604**	**1 ... 4**	**Array**	**int8**	**ro**
Description	Tens' exponent for acceleration values = –6 for linear axes = –3 for rotative axes Default: = –6				
Axis 1	604	1			
	Tens' exponent for acceleration values axis 1				
Axis 2	604	2			
	Tens' exponent for acceleration values axis 2				
Axis 3	604	3			
	Tens' exponent for acceleration values axis 3				
Axis 4	604	4			
	Tens' exponent for acceleration values axis 4				

Acceleration Dimension Index					
FHPP-MAX	**605**	**1 ... 4**	**Array**	**uint8**	**ro**
Description	Unit of measurement for acceleration values = 0xF3 [m/sec^2] for linear axes = 0xF5 [°/sec^2] for rotative axes Default: = 0xF3				
Axis 1	605	1			
	Unit of measurement for acceleration values axis 1				
Axis 2	605	2			
	Unit of measurement for acceleration values axis 2				
Axis 3	605	3			
	Unit of measurement for acceleration values axis 3				
Axis 4	605	4			
	Unit of measurement for acceleration values axis 4				

Jerk Notation Index

FHPP-MAX	608	1 ... 4	Var	int8	ro
Description	Tens' exponent for jerk values = –6 for linear axes = –3 for rotative axes Default: = –6				
Axis 1	608	1			
	Tens' exponent for jerk values axis 1				
Axis 2	608	2			
	Tens' exponent for jerk values axis 2				
Axis 3	608	3			
	Tens' exponent for jerk values axis 3				
Axis 4	608	4			
	Tens' exponent for jerk values axis 4				

Jerk Dimension Index

FHPP-MAX	609	1 ... 4	Array	uint8	ro
Description	Unit of measurement for acceleration values = 0xF6 [m/sec^2] for linear axes = 0xF7 [°/sec^2] for rotative axes Default: = 0xF6				
Axis 1	609	1			
	Unit of measurement for jerk values axis 1				
Axis 2	609	2			
	Unit of measurement for jerk values axis 2				
Axis 3	609	3			
	Unit of measurement for jerk values axis 3				
Axis 4	609	4			
	Unit of measurement for jerk values axis 4				

5.4.7 Axis data electric drives axis 1

Polarity Axis 1					
FHPP-MAX	**1000**	**–**	**Var**	**uint8**	**rw**
Description	Direction of the position values for axis 1. Values: Position value (vector) 0x00 (0): normal 0x80 (128): inverted (multiplied by -1) Default: 0x00 (0) When writing, all values not equal to zero are picked up as 0x80				

Position Factor Axis 1					
FHPP-MAX	**1004**	**1 ... 2**	**Array**	**uint32**	**rw**
Description	Conversion factor for all position units (converting the user units into internal controller units).				
Numerator	1004	1			
	Position factor – numerator. Range of values: 0x00000001 ... 0xFFFFFFFF (1 ... $+2^{32}$-1) Default: 0x00010000 (65536 = 2^{16})				
Denominator	1004	2			
	Position factor – denominator. Range of values: 0x00000001 ... 0xFFFFFFFF (1 ... $+2^{32}$-1) Default: 0x00000001 (1)				

Controller Type Axis 1					
FHPP-MAX	**1030**	**1 ... 128**	**Array**	**char**	**ro**
Description	Name of the motor controller or motor unit, e.g. ‘MTR-DCI’				

Kinematic Type Axis 1					
FHPP-MAX	**1031**	**–**	**Var**	**uint8**	**ro**
Description	= 0 Linear axis = 2 Rotative axis Default = 0				

Usage Axis 1					
FHPP-MAX	**1072**	**–**	**Var**	**uint8**	**ro**
Description	= 0 Not used = 1 Positioning axis = 2 Gantry axis Default = 0				

5.4.8 Axis data electric drives axis 2

Polarity Axis 2					
FHPP-MAX	**1100**	**–**	**Var**	**uint8**	**rw**
Description	Direction of the position values for axis 2. Values: Position value (vector) 0x00 (0): normal 0x80 (128): inverted (multiplied by -1) Default: 0x00 (0) When writing, all values not equal to zero are picked up as 0x80				

Position Factor Axis 2					
FHPP-MAX	**1104**	**1 ... 2**	**Array**	**uint32**	**rw**
Description	Conversion factor for all position units (converting the user units into internal controller units).				
Numerator	1104	1			
	Position factor – numerator. Range of values: 0x00000001 ... 0xFFFFFFFF (1 ... $+2^{32}$-1) Default: 0x00010000 (65536 = 2^{16})				
Denominator	1104	2			
	Position factor – denominator. Range of values: 0x00000001 ... 0xFFFFFFFF (1 ... $+2^{32}$-1) Default: 0x00000001 (1)				

Controller Type Axis 2					
FHPP-MAX	**1130**	**1 ... 128**	**Array**	**char**	**ro**
Description	Name of the motor controller or motor unit, e.g. 'MTR-DCI'				

Kinematic Type Axis 2					
FHPP-MAX	**1131**	**–**	**Var**	**uint8**	**ro**
Description	= 0 Linear axis = 2 Rotative axis Default = 0				

Usage Axis 2					
FHPP-MAX	**1172**	**–**	**Var**	**uint8**	**ro**
Description	= 0 Not used = 1 Positioning axis = 2 Gantry axis Default = 0				

5.4.9 Axis data electric drives axis 3

Polarity Axis 3					
FHPP-MAX	**1200**	**–**	**Var**	**uint8**	**rw**
Description	Direction of the position values for axis 3. Values: Position value (vector) 0x00 (0): normal 0x80 (128): inverted (multiplied by -1) Default: 0x00 (0) When writing, all values not equal to zero are picked up as 0x80				

Position Factor Axis 3					
FHPP-MAX	**1204**	**1 ... 2**	**Array**	**uint32**	**rw**
Description	Conversion factor for all position units (converting the user units into internal controller units).				
Numerator	1204	1			
	Position factor – numerator. Range of values: 0x00000001 ... 0xFFFFFFFF (1 ... $+2^{32}$-1) Default: 0x00010000 (65536 = 2^{16})				
Denominator	1204	2			
	Position factor – denominator. Range of values: 0x00000001 ... 0xFFFFFFFF (1 ... $+2^{32}$-1) Default: 0x00000001 (1)				

Controller Type Axis 3					
FHPP-MAX	**1230**	**1 ... 128**	**Array**	**char**	**ro**
Description	Name of the motor controller or motor unit, e.g. 'MTR-DCI'				

Kinematic Type Axis 3					
FHPP-MAX	**1231**	**–**	**Var**	**uint8**	**ro**
Description	= 0 Linear axis = 2 Rotative axis Default = 0				

Usage Axis 3					
FHPP-MAX	**1272**	**–**	**Var**	**uint8**	**ro**
Description	= 0 Not used = 1 Positioning axis = 2 Gantry axis Default = 0				

5.4.10 Axis data electric drives axis 4

Polarity Axis 4					
FHPP-MAX	**1300**	**–**	**Var**	**uint8**	**rw**
Description	Direction of the position values for axis 4. Values: Position value (vector) 0x00 (0): normal 0x80 (128): inverted (multiplied by -1) Default: 0x00 (0) When writing, all values not equal to zero are picked up as 0x80				

Position Factor Axis 4					
FHPP-MAX	**1304**	**1 ... 2**	**Array**	**uint32**	**rw**
Description	Conversion factor for all position units (converting the user units into internal controller units).				
Numerator	1304	1			
	Position factor – numerator. Range of values: 0x00000001 ... 0xFFFFFFFF (1 ... $+2^{32}$-1) Default: 0x00010000 (65536 = 2^{16})				
Denominator	1304	2			
	Position factor – denominator. Range of values: 0x00000001 ... 0xFFFFFFFF (1 ... $+2^{32}$-1) Default: 0x00000001 (1)				

Controller Type Axis 4					
FHPP-MAX	**1330**	**1 ... 128**	**Array**	**char**	**ro**
Description	Name of the motor controller or motor unit, e.g. ‘MTR-DCI’				

Kinematic Type Axis 4					
FHPP-MAX	**1331**	–	**Var**	**uint8**	**ro**
Description	= 0 Linear axis = 2 Rotative axis Default = 0				

Usage Axis 4 (use of axis 4)					
FHPP-MAX	**1372**	–	**Var**	**uint8**	**ro**
Description	= 0 Not used = 1 Positioning axis = 2 Gantry axis Default = 0				

Parametrisation

Chapter 6

Contents

6.1 Parameterisation with FHPP-MAX

6.1.1 Festo parameter channel (FPC) for cyclic data (I/O data)

The parameter channel is used only for transfer of parameters in the cyclic I/O data. The parameter channel consists of 8 bytes. The following tables show the structure of the parameter channel.

FPC

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Output data	CCON	IND	ParID		Value			
Input data	SCON	IND	ParID		Value			

CCON	Control byte 1: CCON.OPM1 = 1 and CCON.OPM2 = 1: Parametrisation operating mode
SCON	Status byte 1: SCON.OPM1 = 1 and SCON.OPM2 = 1: Parametrisation operating mode
IND	Subindex – for addressing an array element, see Tab. 6/2
ParID	Parameter Identifier – consists of ReqID or ResID and PNU, see Tab. 6/2 and Tab. 6/3
Value	Parameter value or fault number, see Tab. 6/2

Tab. 6/1: Structure of parameter channel

Element	Description
Subindex (IND)	Addresses an element of an array parameter (sub-parameter number)
Parameter identifier (ParID)	Component of the parameter channel that contains the request or response identifier (ReqID / ResID) and the parameter number (PNU). The parameter number serves to identify or address the respective parameter. The request or response identifiers (ReqID / ResID) describe the task or the reply in the form of an identifier number, see Tab. 6/3.
Parameter Value	Value of the parameter. If a parameter processing task cannot be carried out, a fault number will be shown instead of the value in the reply telegram. The error number describes the cause of the error.

Tab. 6/2: Components of the parameter channel (FPC)

Parameter identifier (ParID)

The parameter identifier contains the request or response identifier (ReqID / ResID) and the parameter number (PNU).

Parameter Identifier

	Byte 4								Byte 3							
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Request	ReqID				res.	PNU										
Res-ponse	ResID				res.	PNU										

ReqID	Request Identifier – task identifier (read, write, ...)
ResID	Response Identifier – response identifier (transfer value, fault, ...)
PNU	Parameter Number – serves to identify or address the relevant parameter (see section 6.1). The request or response identifier identifies the type of job or response (see section 6.1.2).

Tab. 6/3: Structure of parameter identifier (ParID)

6.1.2 Request and response identifier and fault numbers

The request and response identifiers are shown in the following table:

ReqID	Request	Response Identifier	
		Positive	Negative
0x0 (0)	No request	0x0 (0	–
0x6 (6)	Request parameters	0x5 (5)	0x7 (7)
0x8 (8)	Change parameter value	0x5 (5)	0x7 (7)
0xD (13)	Request lower limit value	0x5 (5)	0x7 (7)
0xE (14)	Request upper limit value	0x5 (5)	0x7 (7)

Tab. 6/4: Request and Response Identifier

If the task cannot be carried out, response identifier 0x7 as well as the appropriate fault number will be transmitted (negative reply).

The following table shows the response identifiers:

ResID	Description
0x0 (0)	No reply
0x5 (5)	Parameter value transferred (array, double word)
0x7 (7)	Task cannot be carried out (with error number in parameter value) [1]
[1] Error numbers see following table	

Tab. 6/5: Response Identifier

If the task of the parameter processing cannot be carried out, an appropriate fault number will be transmitted in the reply telegram (byte 5, 6, 7 and 8 of the FPC).
The order of fault checking and the possible error numbers are shown in the following table:

Order of fault checking	Checking of	Error numbers	Description
1	PNU defined	0x00 (0)	Impermissible PNU. The parameter does not exist.
2	If array: IND defined	0x03 (3)	Faulty subindex
3	ReqID permissible	0x65 (101)	ReqID is not supported
4	Access rights (read, write)	0x01 (1)	Parameter value cannot be changed (read only)
5	If change: value permissible	0x02 (2)	Lower or upper value limit exceeded
6	Cancellation message of the SDO transfer	0x6E (110)	Cancellation message received. The cancellation code can be read via subindex 3.
7	Timeout message of the SDO transfer	0x6F (111)	Timeout; no answer was received.

Tab. 6/6: Order of fault checking and error numbers

The cancellation codes of the CMMx-xx motor controller can be found in the respective system description. The cancellation codes of the other supported motor controllers can be found in the document “CANopen application layer and communication profile” (www.can-cia.org › Downloads › CiA specifications).

6.1.3 Sequence of parameter processing

Rules for request-response processing

Rule	Description
1	If the controller sends the request identifier for “No task”, the CPX-CMXX reacts with the response identifier for “No response”.
2	A task or response message always refers to a single parameter.
3	The controller must continue to send a task until it receives the appropriate reply from the CPX-CMXX.
4	The controller recognises the reply to the task 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 CPX-CMXX provides the reply until the controller sends a new task.
6	a) A write task, even with cyclic repetition of the same task, will only be carried out once by the CPX-CMMX. b) Between two consecutive tasks with the same request identifier (ReqID), parameter number (PNU) and subindex (IND), the request identifier 0 (no task) 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. 6/7: Rules for request-response processing

Evaluating errors

In the case of tasks which cannot be carried out, the CPX-CMMX replies as follows:

– Output of response identifier = 7

– Output of a fault number in bytes 5, 6, 7 and 8 of the parameter channel (FPC), see Tab. 6/6.

Examples

Read parameters

The following tables show the process for reading parameters via FPC (Festo Parameter Channel).

Step 1 Initialise

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
	CCON/ SCON	Subindex (IND)	ParID (ReqID/ResID+PNU)		Parameter value			
Output data	0xC0	0x00	0x0000		0x00000000			
Input data	0xC0	0x00	0x0000		0x00000000			

Step 2 Read parameters

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
	CCON/ SCON	Subindex (IND)	ParID (ReqID/ResID+PNU)		Parameter value			
Output data	0xC0	IND	0x6000 + PNU		0x00000000			
Input data	0xC0	0x00	0x0000		0x00000000			

Step 3 Reading ended

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
	CCON/ SCON	Subindex (IND)	ParID (ReqID/ResID+PNU)		Parameter value			
Output data	0xC0	IND	0x6000 + PNU		0x00000000			
Input data	0xC0	IND	0x5000 + PNU		Value			

Read parameter – interrupted with fault

The following tables show the process for reading parameters via FPC (Festo Parameter Channel) – interrupted with fault.

Step 1 Initialise

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
	CCON/ SCON	Subindex (IND)	ParID (ReqID/ResID+PNU)		Parameter value			
Output data	0xC0	0x00	0x0000		0x00000000			
Input data	0xC0	0x00	0x0000		0x00000000			

Step 2 Read parameters

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
	CCON/ SCON	Subindex (IND)	ParID (ReqID/ResID+PNU)		Parameter value			
Output data	0xC0	IND	0x6000 + PNU		0x00000000			
Input data	0xC0	0x00	0x0000		0x00000000			

Step 3 Reading interrupted with fault

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
	CCON/ SCON	Subindex (IND)	ParID (ReqID/ResID+PNU)		Parameter value			
Output data	0xC0	IND	0x6000 + PNU		0x00000000			
Input data	0xC0	IND	0x7000 + PNU		Error code, see Tab. 6/6			

Write parameters

The following tables show an example for writing parameters via FPC (Festo Parameter Channel).

Step 1 Initialise

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
	CCON/ SCON	Subindex (IND)	ParID (ReqID/ResID+PNU)		Parameter value			
Output data	0xC0	0x00	0x0000		0x00000000			
Input data	0xC0	0x00	0x0000		0x00000000			

Step 2 Write parameters

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
	CCON/ SCON	Subindex (IND)	ParID (ReqID/ResID+PNU)		Parameter value			
Output data	0xC0	IND	0x8000 + PNU		Value			
Input data	0xC0	0x00	0x0000		0x00000000			

Step 3 Writing ended

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
	CCON/ SCON	Subindex (IND)	ParID (ReqID/ResID+PNU)		Parameter value			
Output data	0xC0	IND	0x8000 + PNU		Value			
Input data	0xC0	IND	0x5000 + PNU		Value			

Write parameter – interrupted with fault

The following tables show an example for writing parameters via FPC (Festo Parameter Channel) – interrupted with fault.

Step 1 Initialise

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
	CCON/ SCON	Subindex (IND)	ParID (ReqID/ResID+PNU)		Parameter value			
Output data	0xC0	0x00	0x0000		0x00000000			
Input data	0xC0	0x00	0x0000		0x00000000			

Step 2 Write parameters

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
	CCON/ SCON	Subindex (IND)	ParID (ReqID/ResID+PNU)		Parameter value			
Output data	0xC0	IND	0x8000 + PNU		Value			
Input data	0xC0	0x00	0x0000		0x00000000			

Step 3 Writing interrupted with fault

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
	CCON/ SCON	Subindex (IND)	ParID (ReqID/ResID+PNU)		Parameter value			
Output data	0xC0	IND	0x8000 + PNU		Value			
Input data	0xC0	IND	0x7000 + PNU		Error code, see Tab. 6/6			

Read lower limit value of a parameter

The following tables show an example for reading the lower limit value of a parameter via FPC – (Festo Parameter Channel).

Step 1 Initialise

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
	CCON/ SCON	Subindex (IND)	ParID (ReqID/ResID+PNU)		Parameter value			
Output data	0xC0	0x00	0x0000		0x00000000			
Input data	0xC0	0x00	0x0000		0x00000000			

Step 2 Read lower limit value of the parameter with the parameter number PNU

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
	CCON/ SCON	Subindex (IND)	ParID (ReqID/ResID+PNU)		Parameter value			
Output data	0xC0	IND	0xD000 + PNU		0x00000000			
Input data	0xC0	0x00	0x0000		0x00000000			

Step 3 Reading lower limit value of the parameter with the parameter number PNU ended

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
	CCON/ SCON	Subindex (IND)	ParID (ReqID/ResID+PNU)		Parameter value			
Output data	0xC0	IND	0xD000 + PNU		0x00000000			
Input data	0xC0	IND	0x5000 + PNU		Value			

Read lower limit value of a parameter – interrupted with fault

The following tables show an example for reading the lower limit value of a parameter via FPC – (Festo Parameter Channel) – interrupted with fault.

Step 1 Initialise

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
	CCON/ SCON	Subindex (IND)	ParID (ReqID/ResID+PNU)		Parameter value			
Output data	0xC0	0x00	0x0000		0x00000000			
Input data	0xC0	0x00	0x0000		0x00000000			

Step 2 Read lower limit value of the parameter with the parameter number PNU

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
	CCON/ SCON	Subindex (IND)	ParID (ReqID/ResID+PNU)		Parameter value			
Output data	0xC0	IND	0xD000 + PNU		0x00000000			
Input data	0xC0	0x00	0x0000		0x00000000			

Step 3 Reading lower limit value of the parameter with the parameter number PNU interrupted with fault.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
	CCON/ SCON	Subindex (IND)	ParID (ReqID/ResID+PNU)		Parameter value			
Output data	0xC0	IND	0xD000 + PNU		0x00000000			
Input data	0xC0	IND	0x7000 + PNU		Error code, see Tab. 6/6			

Read upper limit value of a parameter

The following tables show an example for reading the upper limit value of a parameter via FPC (Festo Parameter Channel).

Step 1 Initialise

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
	CCON/ SCON	Subindex (IND)	ParID (ReqID/ResID+PNU)		Parameter value			
Output data	0xC0	0x00	0x0000		0x00000000			
Input data	0xC0	0x00	0x0000		0x00000000			

Step 2 Read upper limit value of the parameter with the parameter number PNU

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
	CCON/ SCON	Subindex (IND)	ParID (ReqID/ResID+PNU)		Parameter value			
Output data	0xC0	IND	0xE000 + PNU		0x00000000			
Input data	0xC0	0x00	0x0000		0x00000000			

Step 3 Reading upper limit value of the parameter with the parameter number PNU ended.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
	CCON/ SCON	Subindex (IND)	ParID (ReqID/ResID+PNU)		Parameter value			
Output data	0xC0	IND	0xE000 + PNU		0x00000000			
Input data	0xC0	IND	0x5000 + PNU		Value			

Read upper limit value of a parameter – interrupted with fault

The following tables show an example for reading the upper limit value of a parameter via FPC (Festo Parameter Channel) – interrupted with fault.

Step 1 Initialise

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
	CCON/ SCON	Subindex (IND)	ParID (ReqID/ResID+PNU)		Parameter value			
Output data	0xC0	0x00	0x0000		0x00000000			
Input data	0xC0	0x00	0x0000		0x00000000			

Step 2 Read upper limit value of the parameter with the parameter number PNU

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
	CCON/ SCON	Subindex (IND)	ParID (ReqID/ResID+PNU)		Parameter value			
Output data	0xC0	IND	0xE000 + PNU		0x00000000			
Input data	0xC0	0x00	0x0000		0x00000000			

Step 3 Reading upper limit value of the parameter with the parameter number PNU interrupted with fault.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
	CCON/ SCON	Subindex (IND)	ParID (ReqID/ResID+PNU)		Parameter value			
Output data	0xC0	IND	0xE000 + PNU		0x00000000			
Input data	0xC0	IND	0x7000 + PNU		Error code, see Tab. 6/6			

Technical appendix

Appendix A

Contents

A.1 Program Examples with FST

The FST is used for programming control tasks with the CPX-FEC. The functions and characteristics of the software are described in the document P.BE-FST4-B1-...

A.1.1 Program example, start record

Description

Starting the program P11 (name chosen arbitrarily) starts record selection. The program must be started by another program, for example triggered by a command from a connected FED. After the movement ends, the program stops automatically.

The following variables must be defined in the “Allocation List” of the FST:

- CHALT:
  This is the HALT bit in the FHPP-MAX control word (CPOS.B0).
- CSET: Bytes 5 and 6 of the FHPP-MAX output data.
- START: START bit in the FHPP-MAX control word (CPOS.B1)
- SHALT: HALT bit in the FHPP-MAX status word (SPOS.B0)
- ACK: ACK bit in the FHPP-MAX status word (SPOS.B1)
- MC: MC bit in the FHPP-MAX status word (SPOS.B2)
- sxNoHalt:
  Internal flag that displays whether or not the HALT feature is deactivated.
- siSetNr: Number of the record that should be started.

The FST program consists of the following steps:

– Init: Here the flags are initialised and the record number loaded. If the HALT feature is activated (sxNoHalt == false), the CHALT flag is set.

– Execute0: After acknowledgement of the SHALT flag, the START flag is set. If the HALT feature is deactivated, the START flag is set immediately.

– Execute1: In this step, the START/ACK handshake is executed.

– Execute2: This step waits until the MC flag is set. After that, the program is ended.

Start program listing record

```
"" ================================================
""  Init timer and flags
"" ================================================
STEP Init
 IF                  NOP
 THEN
       LOAD    N     sxNoHalt       'Disable HALT feature.
         TO          CHALT          'Control HALT
       LOAD          siSetNr   'Set number for readd/write/select set
         TO          CSET           'Set number

"" ================================================
""  Wait for Halt
"" ================================================
STEP Execute0
"" halt set? -> start set
 IF                  SHALT          'Status Halt
       OR      N     CHALT          'Control HALT
 THEN  SET           START          'Start Positioning Task

"" ================================================
""  START/ACK
"" ================================================
STEP Execute1
"" Acknowledge?
 IF                  START          'Start Positioning Task
       AND           ACK            'Acknowledge Start
 THEN  RESET         START          'Start Positioning Task

"" Acknowledge cleared?
 IF            N     START          'Start Positioning Task
       AND     N     ACK            'Acknowledge Start
 THEN  JMP TO Execute2


"" ================================================
""  Wait for MC
"" ================================================
STEP Execute2
"" Motion complete?
 IF                  MC             'Motion Complete
 THEN  RESET         P11            'Set selection
```

Fig. A/1: Program example, start record

A.1.2 Program example, homing

Description:

Starting the program P10 (name chosen arbitrarily) starts the homing run. The program must be started by another program, for example triggered by a command from a connected FED. Once the homing run is ended, the program stops automatically and signals successful homing via a flag (sxHome). The program exists in two versions – with and without time monitoring. In the more complex version with time monitoring, it is also stopped if a time monitoring fault occurs.

The following variables must be defined in the “Allocation List” of the FST:

- CHALT: This is the HALT bit in the FHPP-MAX control word (CPOS.B0).
- HOM: HOM bit in the FHPP-MAX control word (CPOS.B2)
- SHALT: HALT bit in the FHPP-MAX status word (SPOS.B0)
- ACK: ACK bit in the FHPP-MAX status word (SPOS.B1)
- REF: REF bit in the FHPP-MAX status word (SPOS.B7)
- sxNoHalt: Internal flag that displays whether or not the HALT feature is deactivated.
- sxHome: Internal flag that displays whether the axis/axes is/are referenced or not.

The FST program consists of the following steps:

– Init: The flags are initialised here.
  If the HALT feature is activated (sxNoHalt == false), the CHALT flag is set.

– Execute0: After acknowledgement of the SHALT flag, the HOM flag is set.
  If the HALT feature is deactivated, the HOM flag is set immediately.

– Execute1: After acknowledgement of the ACK flag, the HOM flag is reset again.

– Execute2: After acknowledgement of the REF flag, the sxHome flag is set and the program ended.

Program listing version 1 Homing run without time monitoring

```
"" ================================================
""  Init flags
"" ================================================
STEP Init
 IF                  NOP
 THEN
       LOAD    N     sxNoHalt       'Disable HALT feature.
         TO          CHALT          'Control HALT
       RESET         sxHome         'Homing complete

"" ================================================
""  Wait for Halt
"" ================================================
STEP Execute0
"" halt set? --> start homing
 IF                  SHALT          'Status Halt
       OR      N     CHALT          'Control HALT
 THEN  SET           HOM            'Start homing

"" ================================================
""  Wait for ACK
"" ================================================
STEP Execute1
"" Acknowledge received?
 IF                  ACK            'Acknowledge Start
 THEN  RESET         HOM            'Start homing
       JMP TO Execute2

"" ================================================
""  Wait for homing complete
"" ================================================
STEP Execute2
"" Referenced?
 IF                  REF            'Axis is referenced.
 THEN  SET           sxHome         'Homing complete
       RESET         P10            'Homing
```

Fig. A/2: Program example, homing without time monitoring

Additional description version 2:

For the version with time monitoring, the individual steps are still equipped with time monitoring. When a timer runs out, the corresponding fault bit is set and the homing run interrupted.

For the variant with time monitoring, the following additional variables must be defined in the “Allocation List” of the FST:

– THome: Timer for time monitoring of the homing run.

– TPHome: Timer preselect for THome – is loaded with 60s.

– THomeAck: Timer for time monitoring of the HOM/ACK or the CHALT/SHALT handshake.

– TPHomeAck: Timer preselect for THomeAck – is loaded with 1s.

– exHome: Internal flag that displays a fault in the homing run.

Program listing version 2 homing run with time monitoring

```
"" ================================================
""  Init timer and flags
"" ================================================
STEP Init
 IF                  NOP
 THEN
       LOAD          V6000          "60sec timeout
         TO          TPHome         'Preselect homing timer
       LOAD          V100           "1sec timeout
         TO          TPHomeAck      'Preselect home ack timer
       RESET         THome          'Homing surveillance
       LOAD    N     sxNoHalt       'Disable HALT feature.
         TO          CHALT          'Control HALT
       SET           THomeAck       'Home acknowledge timer
       RESET         sxHome         'Homing complete
"" ================================================
""  Wait for Halt
"" ================================================
STEP Execute0
"" timeout error
 IF            N     THomeAck       'Home acknowledge timer
 THEN  SET           exHome         'Homing error
       RESET         CHALT          'Control HALT
       RESET         P10            'Homing
"" halt set? --> start homing
 IF                  SHALT          'Status Halt
       OR      N     CHALT          'Control HALT
 THEN  SET           HOM            'Start homing
"" ================================================
""  Wait for ACK
"" ================================================
STEP Execute1
"" timeout error
 IF            N     THomeAck       'Home acknowledge timer
 THEN  SET           exHome         'Homing error
       RESET         HOM            'Start homing
       RESET         CHALT          'Control HALT
       RESET         P10            'Homing
"" Acknowledge received?
 IF                  ACK            'Acknowledge Start
 THEN  RESET         HOM            'Start homing
       SET           THome          'Homing surveillance
       JMP TO Execute2
```

```
"" ================================================
""  Wait for homing complete
"" ================================================
STEP Execute2
"" timeout error
 IF            N      THome          'Homing surveillance
 THEN  SET            exHome         'Homing error
       RESET          CHALT          'Control HALT
       RESET          P10            'Homing
"" Referenced?
 IF                   REF            'Axis is referenced.
 THEN  SET            sxHome         'Homing complete
       RESET          P10            'Homing
```

Fig. A/3: Program example, homing with time monitoring

Index

Appendix B

Contents

A

C

D

E

F

G

I

J

L

M

O

P

R

S

T

V