Festo Handling and Positioning Profile

FESTO

Manual
FHPP

Motor controller
Type CMM...

Festo Handling and Positioning Profile

Manual
555 696
en 0708NH
[720 315]

Original ..... German text

Edition ..... en 0708NH

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

Order-no. 555 696

Internet: http://www.festo.com
E-mail: service_international@festo.com

Contents

Use for intended purpose

This description includes the Festo Handling and Position Profile (FHPP) for the CMMx product family.

This provides you with supplementary information about controlling, diagnosing and parametrizing the motor controllers via the fieldbus.

The complete set of information can be found in the documentation for the motor controller in question:

– Description P.BE-CMM...-HW-...:
  Mechanics - electrical engineering - overview of the function range.

Note
Always follow the safety-related instructions listed in the product manual for the motor controller in question.

Depending on which fieldbus is used, you can find further information in the following manuals for the CMMx product family:

– Description type P.BE-CMM...-CO-...:
  Description of the implemented CANopen protocol as per DSP402.

– Description type P.BE-CMM...-PB-...:
  Description of the implemented PROFIBUS-DP protocol.

– Description type P.BE-CMM...-DN-...:
  Description of the implemented DeviceNet protocol.

Safety instructions

When commissioning and programming positioning systems, you must always observe the safety regulations in this manual as well as those in the operating instructions for the other components used.

The user must make sure that nobody is in the operating range of the connected actuators or axis system. Access to the possible danger area must be prevented by suitable measures such as protective screens and warning signs.

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

Make sure that nobody can reach into the operating range of the axes or other connected actuators and that no objects lie in the positioning range while the system is still connected to a power supply.

Warning
Faults in parametrizing can cause injury to human beings and damage to property.

Only enable the controller if the axis system has been installed and parametrized by technically qualified staff.

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

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

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

The following pictogram marks passages in the text which describe activities using electrostatically sensitive devices.

Electrostatic sensitive devices: inappropriate handling can result in damage to components.

Identification of specific information

The following pictograms mark passages in the text which contain special information.

Pictograms

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

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

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

Text designations

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

1. Numbers denote activities which must be carried out in the sequence specified.

– Arrowheads indicate general lists.

Information on the version

This manual refers to versions set out in Table. 0/1:

Controller	Firmware	Remarks
CMMS-ST-...	Version 1.0 and later	Stepper motor controller
CMMP-AS-...	Version 1.0 and later	Servo motor controller premium (in preparation)
CMMS-AS-...	in preparation	Servo motor controller standard (in preparation)

Table. 0/1: Controller and firmware versions

Note
With newer firmware versions, check whether there is a newer version of this description available:
www.festo.com

Terms and abbreviations

The following terms and abbreviations are used in this manual:

Term/abbreviation	Meaning
0 signal	0 V present at input or output (positive logic, corresponds to LOW).
1 signal	24 V present at input or output (positive logic, corresponds to HIGH).
Axis	Complete actuator, consisting of motor, encoder and drive, optional with gear, if applicable with controller.
Axis zero point (AZ)	Dimensional reference point for the project zero point and the software end positions. The point of reference for the axis zero point is the homing point.
Controller	Control electronics, which evaluate the control signals and provide the voltage supply for the motor via the power electronics (power electronics + closed-loop controller + positioning controller).
Drive	Mechanical component of an axis, which transfers the driving power for the movement, defines the guide for the positioning motion and also enables the effective load and the reference switch to be configured.
Operation mode	Operation mode of the controller. – Positioning control – Speed adjustment – Torque control – Position control
Encoder	Electrical pulse generator (generally a rotor position transducer). The controller evaluates the electrical signals that are generated and uses them to calculate the position and speed.
Festo Configuration Tool (FCT)	Software with standardised project and data management for supported device types. The special requirements of a device type are supported with the necessary descriptions and dialogues by means of PlugIns.
Festo Handling und Positioning Profile (FHPP)	Fieldbus data profile for positioning controllers from Festo.
Festo Parameter Channel (FPC)	FHPP-specific parameter access, virtually acyclic.
FHPP operation mode	Way in which the controller is controlled or its setpoints are specified by FHPP – Record selection – Direct mode

Term/abbreviation	Meaning
Functions	Special functions in the different operation modes. – Jog mode – Homing
Jog mode	Manual positioning in positive or negative direction. Function for setting positions by approaching the target position, e.g. by teaching (teach mode) traversing records.
Homing method	Method for defining the reference position: against a fixed stop (overload current/speed evaluation) or with reference switch.
I O I/O	Input Output Input and/or output
PLC	Programmable logic controller; controller (also IPC: industrial PC).
Positioning mode (Profile position mode)	Operation mode for processing a traversing record or a direct positioning task.
Project zero point (PZ)	Measuring reference point for all positions in positioning tasks. The project zero point forms the basis for all absolute position specifications (e.g. in the traversing record table or in direct mode). The point of reference for the project zero point is the axis zero point.
Reference point (REF)	Basis point for the incremental measuring system. The reference point defines a known position/orientation within the drive's travel distance.
Reference switch	External sensor (e.g. type SMT-10) which serves for ascertaining the reference position and is connected directly to the controller.
Referencing (homing mode)	Function in which reference travel (homing) is carried out. By means of homing, the reference position and thereby the origin of the measuring reference system of the axis are defined.
Software end position	Programmable stroke limitation (basis point = axis zero point) **Software end position, pos. (upper):** max. limit position in the positive direction. **Software end position, negative:** min. limit position in the negative direction.
Traversing record	Positioning command defined in the traversing record table, consisting of target position, positioning mode, speed, acceleration, ...

Table. 0/2: Index of terms and abbreviations

Chapter 1

Contents

1.1 Festo profile for handling and positioning (FHPP)

Festo has developed an optimized data profile especially tailored to the target applications for handling and positioning tasks, the “Festo Handling and Positioning Profile (FHPP).”
The FHPP enables uniform control and programming for the various fieldbus systems and controllers from Festo.

In addition, it provides a largely standardised definition of the following for the user:

– the operation modes

– the I/O data structure

– the parameter objects

– the sequence control

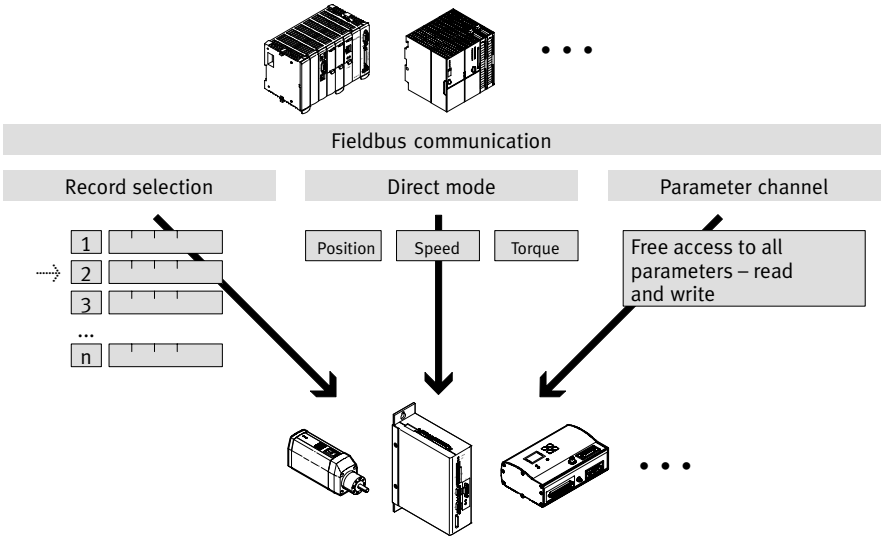


Fig. 1/1: The FHPP principle

Control and status data

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

Record selection

Saved traversing records can be processed in the Record Select mode.
For this purpose, traversing records are parametrized with the Festo Configuration Tool or taught via FHPP during commissioning.

Direct mode

In the Direct operation mode, the important positioning data is transferred directly via the control bytes.

– Target positions, speeds and torques can be ascertained and specified by the controller during run-time, depending on the operating status.

– There are no limitations relating to the number of saved traversing records.

Parameter channel

The controller can access all parameter values of the controller via the fieldbus by means of the parameter channel. A further 8 bytes of I/O data are used for this purpose.

1.2 Overview FHPP for motor controllers CMM...

Property / Function		CMMS-ST	CMMP-AS
Sequence control			
Supported operating modes			
Record selection		x	x
Direct mode		x	x
Commissioning		–	x [1]
Parameterization		–	–
Structure of I/O data			
CCON			
Enable (B0)		x	(x) [2]
Stop (B1)		x	(x) [3]
Brake (B2)		x	x
Reset (B3)		x	x
reserved (B4)		x	x
LOCK (B5)		x	x
Operation Mode	= 00 (Record selection) = 01 (Direct mode) = 10 (Commissioning) = 11 (Parameterization)	x x – –	x x x –

1) Commissioning functions 3 (current regulator identification) and 4 (phase-angle identification)
2) Corresponds to final stage enable.
3) Corresponds to controller enable.

Property / Function		CMMS-ST	CMMP-AS
CPOS			
	Halt	x	x
	Start positioning task (Start)	x	x
	Start homing (Home)	x	x
	Jog forwards (JogP)	x	x
	Jog backwards (JogN)	x	x
	Teach value	x	x
	Clear remaining path	x	x
	B7 (reserved)	x	x
CDIR			
	Move relatively (ABS)	x	x
	Control mode = 00 (Position control) = 01 (Torque control) = 10 (Speed control) = 11 (special, e.g. energy optimized)	x x x –	x x x –
	Continuous tracking mode (CONT)	–	x
	Continuous tracking mode Toggle (CTOG)	–	–
	Stroke limit deactivated (XLIM)	–	–
	Fast stop (FAST)	–	–
	Carry out function (FUNC)	–	–

Property / Function		CMMS-ST	CMMP-AS
SCON			
Controller enabled (Ready)		x	(x) [2]
Operation enabled (Open)		x	(x) [3]
Warning (Warn)		x	x
Fault (Fault)		x	x
Load voltage applied (V24L)		x	x
Control sovereignty FCT/MMI		x	x
Reply: Operation Mode	= 00 (Record selection) = 01 (Direct mode) = 10 (Commissioning) = 11 (Parameterization)	x x – –	x x x –
SPOS			
Halt		x	x
Acknowledgement Start (Ack)		x	x
Motion Complete (MC)		x	x
Acknowledgement Teach (Teach)		x	x
Axis is moving (Mov)		x	x
Drag error (Dev)		x	x
Standstill monitoring (Still)		x	x
Drive is referenced (Ref)		x	x

[2] Corresponds to final stage enable.
[3] Corresponds to controller enable.

Property / Function		CMMS-ST	CMMP-AS
SDIR			
Absolute/ Relative		x	x
Replay: Control mode	= 00 (Position control) = 01 (Torque control) = 10 (Speed control) = 11 (special)	x x x –	x x x –
Continuous tracking mode active		–	x
Speed limit reached		–	–
Stroke limit reached		–	–
Fast stop		–	–
Function		–	–
Record status byte			
First record chaining completed (RC1)		x	–
Record chaining completed (RCC)		x	–
B2		–	–
B3		–	(x) [4]
Speed limit reached		–	–
Stroke limit reached		–	–
Fast stop		–	–
Function		–	–

[4] Bit3 = Continuous tracking mode active

Property / Function	CMMS-ST	CMMP-AS
Drive modes		
Homing	x	x
Jog and teach	x	x
Record selection	x	x
Number of records (record 0 = Homing)	1…63	1...250
Record chaining (switching to the next record)	x	x
Direct mode	x	x
Move directly to desired position + speed	x	x
Continually adjust the set position (Continuous tracking mode)	–	x
Continuous set value specification with time grid (curve disc)	–	–
Specify the desired torque and ramp	(x) 5)	(x) 5)
Speed control	x	x
Standstill monitoring	x	x
Diagnosis		
Error buffer (PNU 200 ... 204)	(x) 6)	x
Warning buffer (PNU 210 ... 214)	–	x
Fault numbers	(x) 7)	(x) 7)
Parameter		
Access protection Control sovereignty FCT / PLC (SCON.Lock)	x	x
Parameter access as per FPC2	x	x

5) No ramp
6) Only PNU 201
7) Special fault numbers

Chapter 2

Contents

2.1 General information

Each controller has specific features and tasks. They therefore each have their own finite state machine and a separate database.

The Festo handling and positioning profile (FHPP) provides users with information about a controller's individual characteristics. The profile is implemented as independently as possible from each controller and fieldbus.

In addition, it provides a standardised definition of the following for the user:

- the FHPP operation modes
- the I/O data structure
- the parameter objects
- the sequence control

2.2 Setpoint specification (FHPP operation modes)

The FHPP operation modes differ in the content and meaning of the cyclic I/O data and in the functions that can be accessed in the controller.

Operation mode	Description
Record Select	A specific number of traversing records can be saved in the controller. A record contains all the parameters which are specified for a positioning task. The record number is transferred to the cyclic I/O data as the nominal or actual value.
Direct mode	The positioning task is transferred directly in the I/O telegram. The most important nominal values (position, speed, torque) are transferred in this telegram. Supplementary parameters (e.g. acceleration) are defined by the parametrizing.

Table 2/3: Overview of FHPP operation modes in CMM...

2.2.1 Switching FHPP operation modes

The FHPP operation mode is switched by the CCON control byte (see below) and indicated in the SCON status word.

Switching between record selection and direct mode is only permitted in the ”Ready” state, see section 2.7, Fig. 2/2.

2.2.2 Record selection

Each controller has a specific number of records, which contain all the information needed for one positioning job. The maximum number of records is specified separately for each controller.

The record number that the controller is to process at the next start is transferred in the PLC's output data. The input data contains the record number that was processed last. The positioning task itself does not need to be active.

The controller does not support any automatic mode, i.e. no user program. Records cannot be processed automatically with a programmed logic. The controller cannot accomplish any useful tasks with Stand Alone; close coupling to the PLC is necessary.

However, depending on the controller, it is also possible to concatenate various records and execute them one after the other with the help of a start command. It is also possible (again, dependent on the controller) to define record chaining before the target position is reached.

It is only possible to set all of the parameters for the record chaining ("route program") (e.g. the following record) using the FCT.

In this way, positioning profiles can be created without the inactive times (which arise from the transfer in the fieldbus and the PLC's cycle time) having an effect.

2.2.3 Direct mode

In the direct mode, positioning tasks are formulated directly in the PLC's output data.

The typical application calculates dynamically the nominal target values for each task or just for some tasks. This makes it possible to adjust the system to different tool sizes, for example, without having to re-parametrize the record list. The positioning data is managed completely in the PLC and sent directly to the controller.

Here also, close coupling between the PLC and the controller is necessary.

2.3 Configuration of the I/O data

2.3.1 Concept

A PLC exchanges the following data with the FHPP:

- 8-byte control and status data:
  - Control and status bytes
  - Record number or nominal position in the output data
  - Return message of actual position and record number in the input data
  - Additional mode-dependent setpoint and actual values
- If required, an additional 8 bytes of input and 8 bytes of output data for FPC parameterisation.

The FHPP protocol stipulates 8 bytes of input data and 8 bytes of output data. Of these, the first byte is fixed (the first two bytes in the FHPP operation modes Record Select and Direct mode). It remains intact in each operation mode and controls the enabling of the controller and the FHPP operation modes. The other bytes are dependent on the FHPP operation mode that was selected. Further control or status bytes and nominal and actual values can be transferred here.

In the cyclic data, a further 8 bytes of input data and 8 bytes of output data are permissible to transmit parameters according to the FPC protocol.

Overview of the control and status bytes

No.	Controller	Reply message	Description
1	CCON	SCON	Controller enable and operation mode
2	CPOS	SPOS	Positioning, homing and jogging
3	CDIR	SDIR	Positioning task in direct mode

Table 2/4: Control and status bytes

Fieldbus configuration

Fieldbus	Cyclical I/O update	Acyclic data
PROFIBUS	Two modules can be selected: – Module 1: Name = “FHPP Standard” identifier = 0xB7 (8 byte I/O consistent) – Module 2: Name = “FHPP Standard + FPC” identifier = 0xB7, 0xB7 (2 x 8 byte I/O consistent)	– (DPV0 only)
CANopen	Two PDOs can be selected: – PDO 1 with 8 byte I/O for FHPP Standard – PDO 2 with 8 byte I/O for FPC	SDO corresponding to CANopen
DeviceNet	Poll command/response Message with 16 byte I/O (FHPP Standard + FPC)	Explicit messaging

Table 2/5: Fieldbus configuration

2.3.2 I/O data in the various FHPP operation modes (control view)

Record selection

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

Direct mode

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Output data	CCON	CPOS	CDIR	Nominal value 1	Nominal value 2			
Input data	SCON	SPOS	SDIR	Actual value 1	Actual value 2			

Further 8 bytes of I/O data for parametrizing as per FPC (see section 6.1):

Festo FPC

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Output data	Reserved	Subindex	Task identifier + parameter number		Parameter value			
Input data	Reserved	Subindex	Reply identifier + parameter number		Parameter value			

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

Assignment of the control bytes (overview)

CCON (all)	**B7 OPM2**	**B6 OPM1**	**B5 LOCK**	**B4 –**	**B3 RESET**	**B2 BRAKE**	**B1 STOP**	**B0 ENABLE**
	FHPP operation mode selection		Software access blocked	–	Acknowledge fault	Release brake	Stop	Enable drive
CPOS (record selection and direct mode)	**B7 –**	**B6 CLEAR**	**B5 TEACH**	**B4 JOGN**	**B3 JOGP**	**B2 HOM**	**B1 START**	**B0 HALT**
	–	Delete remaining path	Teach value	Jogging negative	Jogging positive	Start homing	Start positioning job	HALT
CDIR (direct mode)	**B7 –**	**B6 –**	**B5 XLIM**	**B4 VLIM**	**B3 CONT**	**B2 COM2**	**B1 COM1**	**B0 ABS**
	–	**–**	Deactivate stroke limit	Velocity limit reached	Continuous tracking mode	Control mode (position, torque, speed)		absolute/relative

Assignment of the status bytes (overview)

SCON (all)	**B7 OPM2**	**B6 OPM1**	**B5 LOCK**	**B4 24VL**	**B3 FAULT**	**B2 WARN**	**B1 OPEN**	**B0 ENABLED**
	FHPP operation mode acknowledgment		Device control software	Load voltage applied	Fault	Warning	Operation enabled	Drive enabled
SPOS (record selection and direct mode)	**B7 REF**	**B6 STILL**	**B5 DEV**	**B4 MOV**	**B3 TEACH**	**B2 MC**	**B1 ACK**	**B0 HALT**
	Drive referenced	Downtime monitoring	Drag error	The axis moves	Teach acknowledgment	Motion Complete	Start acknowledgment	HALT
SDIR (direct mode)	**B7 –**	**B6 –**	**B5 XLIM**	**B4 VLIM**	**B3 CONT**	**B2 COM2**	**B1 COM1**	**B0 ABS**
	–	**–**	Stroke limit reached	Velocity limit reached	Continuous tracking mode	Control mode acknowledgment (position, torque, speed)		absolute/relative

2.5 Description of the control bytes

2.5.1 Control byte 1 (CCON)

Control byte 1 (CCON)		
Bit	**EN**	**Description**
B0 **ENABLE**	Drive **Enable**	= 1: Enable drive (controller) = 0: Drive (controller) blocked
B1 **STOP**	**Stop** 1	= 1: Operation enabled. Any fault will be deleted. = 0: Stop 1 active (cancel emergency ramp + positioning task). The drive stops with maximum braking ramp, the positioning job is reset.
B2 **BRAKE**	Open **Brake**	= 1: Release brake = 0: Activate brake Note: it is only possible to release the brake if the controller is locked. As soon as the controller is enabled, it has priority over the brake's control system.
B3 **RESET**	**Reset** Fault	With a rising edge a fault is acknowledged and the fault value is deleted.
B4 **–**	–	Reserved, must be at 0.
B5 **LOCK**	Software Access **Lock**ed	Controls access to the controller's local (integrated) diagnostic interface. = 1: The software can only observe the controller; the software cannot take over device control (HMI control) from the software. = 0: The software may take over the device control (in order to modify parameters or to control inputs).
B6 **OPM1**	Select **Op**erating **M**ode	Bit 7 6 operation mode 0 0 Record selection 0 1 Direct mode 1 x Reserved
B7 **OPM2**		

CCON controls statuses in all FHPP operation modes. For more information, see the description of the drive functions in Chapter 3.

2.5.2 Control byte 2 (CPOS)

Control byte 2 (CPOS)		
Bit	**EN**	**Description**
B0 **HALT**	**HALT**	= 1: HALT is not active = 0: HALT activated (do not cancel braking ramp + positioning task). The axis stops with a defined braking ramp, the positioning task remains active (with B6 the remaining path can be deleted).
B1 **START**	**Start** positioning task	With a **rising edge** the current nominal values will be transferred and positioning started (even if record 0 = homing travel, for example).
B2 **HOM**	Start **hom**ing	With a **rising edge** homing is started with the set parameters.
B3 **JOGP**	**Jog p**ositive	The drive moves at the specified velocity or rotational speed in the direction of larger actual values, providing the bit is set. The movement begins with the rising edge and ends with the falling edge.
B4 **JOGN**	**Jog n**egative	The drive moves at the specified velocity or rotational speed in the direction of smaller actual values, see B3.
B5 **TEACH**	**Teach** actual value	At a **falling edge** the current actual value is imported into the setpoint register of the currently addressed traversing record; see section 3.5. The Teach target is defined with PNU 520, for example. Actual values could include the position, pressure or torque, for example. The type is determined by the record status byte (RSB). See also section 3.5.
B6 **CLEAR**	**Clear** remaining position	In the “HALT” status a **rising edge** causes the positioning job to be deleted and transfer to the status “Ready.”
B7 **–**	–	Reserved, must be at 0.

CPOS controls the positioning sequences in the “Record selection” and “Direct mode” FHPP operation modes, as soon as the drive is enabled.

2.5.3 Control byte 3 (CDIR) – direct mode

Control byte 3 (CDIR) – direct mode

Bit	EN	Description
B0 **ABS**	**Abs**olute/ Relative	= 0: Nominal value is absolute = 1: Nominal value is relative to last nominal value
B1 **COM1**	**Co**ntrol **m**ode	Bit 2 1 Control mode 0 0 Position control **0 1 Torque control** 1 0 Speed adjustment 1 1 Reserved
B2 **COM2**		
B3 **CONT**	**Cont**inous Tracking mode	Activates continuous tracking mode (continuous setpoint specification): = 0: Continuous tracking mode inactive = 1: Continuous tracking mode active
B4 **VLIM**	Velocity (**V**-) **Lim**it not active	Reserved, must be at 0.
B5 **XLIM**	Torque (**X**-) **Lim**it not active	Only for torque control: = 0: Torque monitoring active = 1: Torque monitoring inactive
B6 **–**	**–**	Reserved, must be at 0.
B7 **–**	–	Reserved, must be at 0.

In direct mode, CDRI specifies the type of positioning job more precisely.

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

Control byte 4 (nominal value 1) – Direct mode		
Bit	**EN**	**Description**
B0 ... B7	Position – Velocity	Preselect depends on control mode (CDIR.B1/B2): – Position: velocity in % of the maximum velocity – Torque: reserved (= 0) – Velocity: Speed ramp in % of the maximum ramp

Control bytes 5... 8 (nominal value 2) – Direct mode		
Bit	**EN**	**Description**
B0...B31	Position, Velocity, Torque	Preselect depends on control mode (CDIR.B1/B2): – Position: Position in position unit (see appendix A.1) – Torque: Torque in % of the maximum torque – Velocity: Speed in speed unit In each case 32-bit number, low byte first.

2.5.5 Bytes 3 and 4 ... 8 – Record Select

Control byte 3 (record number) – Record select		
Bit	**EN**	**Description**
B0 ... B7	Record number	Preselect of record number for Record Select.

Control bytes 4... 8 – Record Select		
Bit	**EN**	**Description**
B0 ... B7	–	reserved (= 0)

2.6 Description of the status byte

2.6.1 Status byte 1 (SCON)

Status byte 1 (SCON)

Bit	EN	Description
B0 **ENABLED**	Drive **Enable**d	= 0: Drive blocked, controller not active = 1: Drive (controller) enabled
B1 **OPEN**	**Op**eration **En-**abled	= 0: Stop active = 1: Mode enabled, positioning possible
B2 **WARN**	**Warn**ing	= 0: Warning not applied = 1: Warning applied
B3 **FAULT**	**Fault**	= 0: No fault = 1: There is a fault or fault reaction is active. Fault code in the diagnostic memory.
B4 **24VL**	**S**upply **V**oltage is **A**pplied	= 0: No load voltage = 1: Load voltage applied
B5 **LOCK**	Drive Control by Software	= 0: Device control free (e.g. PLC/fieldbus) = 1: Device control by software (PLC control is **Lock**ed)
B6 **OPM1**	Display **Op**er-ating **M**ode	Bit 7 6 Operation mode acknowledgment 0 0 Record selection 0 1 Direct mode 1 x Reserved
B7 **OPM2**		

2.6.2 Status byte 2 (SPOS)

Status byte 2 (SPOS)		
Bit	**EN**	**Description**
B0 **HALT**	**HALT**	= 0: HALT is active = 1: HALT is not active, axis can be moved
B1 **ACK**	**Ack**nowledge Start	= 0: Ready for start (homing, jog) = 1: Start carried out (homing, jog)
B2 **MC**	**M**otion **C**omplete	= 0: Positioning job active = 1: Positioning job completed, where applicable with fault Note: MC is set after device is switched on (status “Drive blocked”)
B3 **TEACH**	Acknowledge **Teach**	= 0: Ready for teaching = 1: Teaching carried out, actual value is transferred
B4 **MOV**	Axis is **mov**ing	= 0: Speed of the axis ‹ limit value = 1: Speed of the axis ›= limit value
B5 **DEV**	Drag (**dev**i-ation) Error	= 0: No drag error = 1: Drag error active
B6 **STILL**	Stand**still** control	= 0: After MC, axis remains in tolerance window = 1: Axis has left the tolerance window after MC
B7 **REF**	Axis is **ref**erenced	= 0: Referencing must be carried out = 1: Reference information exists, homing does not need to be carried out

2.6.3 Status byte 3 (SDIR) – direct mode

The SDIR status byte acknowledges positioning mode.

Status byte 3 (SDIR) – direct mode

Bit	EN	Description
B0 **ABS**	**Abs**olute/ Relative	= 0: Nominal value is absolute = 1: Nominal value is relative to last nominal value
B1 **COM1**	**Co**ntrol **M**ode feedback	Bit 2 1 Control mode feedback 0 0 Position control **0 1 Torque control** 1 0 Speed adjustment 1 1 Reserved
B2 **COM2**		
B3 **CONT**	Continuous tracking mode	Continuous tracking mode acknowledgment (continuous setpoint specification): = 0: Continuous tracking mode inactive = 1: Continuous tracking mode active
B4 **VLIM**	Speed (**V**-) **LIM**it reached	Only for torque control: = 1: Speed limit reached = 0: Speed limit not reached
B5 **XLIM**	Torque (**X**-) **lim**it reached	Only for torque control: = 0: Torque monitoring reached = 1: Torque monitoring not reached
B6 –	–	Reserved
B7 **–**	–	Reserved

2.6.4 Bytes 4 and 5 ... 8 – direct mode

Status byte 4 (actual value 1) – direct mode		
Bit	**EN**	**Description**
B0 ... B7	 Position Torque Velocity	Reply message depends on control mode (CDIR.B1/B2): – Position: velocity in % of the maximum velocity – Torque: torque in % of the maximum torque – Velocity: Speed in % of the maximum speed

Status bytes 5 ... 8 (actual 2) – direct mode		
Bit	**EN**	**Description**
B0...B31	Position	Reply message Position for direct mode: – position in position units (see appendix A.1) (32-bit number, low byte first)

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

Status byte 3 (set number) – record selection

Bit	EN	Description
B0 ... B7	Record number	Reply message of record number for Record Select.

Status byte 4 (RSB) – record selection

Bit	EN	Description
B0 **RC1**	**1**st **R**ecord **C**haining Done	= 0: Chaining condition not configured / not reached = 1: First chaining condition reached. (The first chaining condition is reached when Motion Complete = 1 after the first chained record).
B1 **RCC**	**R**ecord **C**haining **C**omplete	Valid when MC = 1. = 0: Record chaining interrupted. At least one chaining condition was not reached. = 1: Record chain was processed to the end of the chain.
B2	–	reserved
B3	–	reserved
B4 **VLIM**	Velocity (**V**-) **LIM**it reached	Only with torque control: = 1: Speed limit reached = 0: Speed limit not reached
B5 **XLIM**	Torque (**X**-) **Lim**it reached	Only with torque control: = 0: Torque limit reached = 1: Torque limit not reached
B6	–	reserved
B7	–	reserved

Status bytes 5 ... 8 (position) – record selection

Bit	EN	Description
B0...B31	Position, ...	Reply message of position – position in position units (see appendix A.1) (32-bit number, low byte first)

2.7 Status machine FHPP

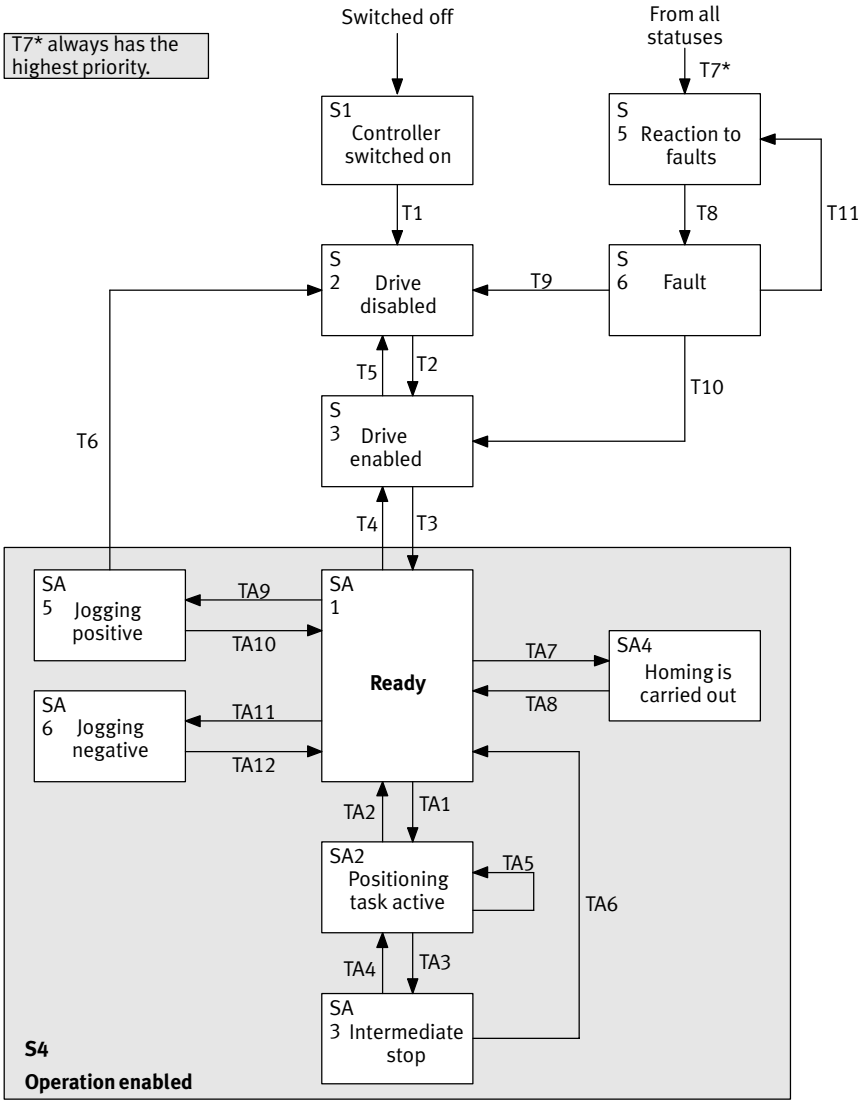


Fig. 2/2: State machine

Notes on the “Mode enabled” state

The transition T3 changes to status S4, which itself contains its own sub-status machine, the states of which are marked with “SAx” and the transitions of which are marked with “TAx” Fig. 2/2. This enables an equivalent circuit diagram (Fig. 2/3) to be used, in which the internal states SAx are omitted.

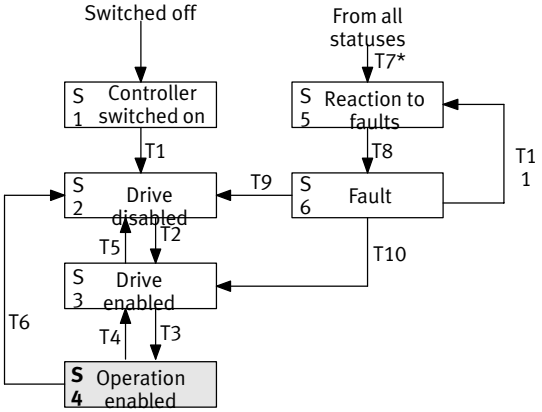


Fig. 2/3: State machine equivalent circuit diagram

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

Reaction to faults

T7 (“Fault recognized”) has the highest priority (and receives the asterisk ”*”).

T7 is then derived from S5+S6 when a fault with higher priority occurs. This means that a serious fault can suppress a simple fault.

2.7.1 Create readiness to operate

To create the ready status, additional input signals are required, depending on the controller, at DIN 4, DIN 5, DIN 13, etc., for example.
For more detailed information about this, see the description of the controller in question.

D	Internal conditions	Activities of the user
T1	Drive is switched on. A fault cannot be ascertained.	
T2	Load voltage applied. Control sovereignty for PLC.	“Enable drive” = 1 CCON = xxx0.xxx**1**
T3		“Stop” = 1 CCON = xxx0.xx**1**1
T4		“Stop” = 0 CCON = xxx0.xx**0**1
T5		“Enable drive” = 0 CCON = xxx0.xxx**0**
T6		“Enable drive” = 0 CCON = xxx0.xxx**0**
T7*	Fault recognized.	
T8	Reaction to fault completed, drive stopped.	
T9	There is no longer a fault. It was a serious fault.	“Quit fault” = 0 → 1 CCON = xxx0.**P**xxx
T10	There is no longer a fault. It was a simple fault.	“Quit fault” = 0 → 1 CCON = xxx0.**P**xx1
T11	Fault still exists.	“Quit fault” = 0 → 1 CCON = xxx0.**P**xx1
Key: P = positive edge, N = negative edge, x = any		

2.7.2 Positioning

Fundamentally, the following applies:
Transitions T4, T6 and T7* always have priority.

TA	Internal conditions	Activities of the user
TA1	Referencing is running.	Start positioning task = 0 → 1 Stop = 1 CCON = xxx0.xx11 CPOS = 0xx0.00P1
TA2	Motion Complete = 1 The current record is completed. The next record is not to be carried out automatically	"Halt" state is optional CCON = xxx0.xx11 CPOS = 0xxx.xxxx
TA3	Motion Complete = 0	Stop = 1 → 0 CCON = xxx0.xx11 CPOS = 0xxx.xxx0
TA4		Stop = 1 Start positioning task = 0 → 1 Clear remaining travel = 0 CCON = xxx0.xx11 CPOS = 00xx.xxP1
TA5	Record Selection: – An individual record is finished. – The next record is to be processed automatically.	CCON = xxx0.xx11 CPOS = 0xxx.xxx1
	Direct mode: – A new positioning task has arrived.	CCON = xxx0.xx11 CPOS = 0xxx.xx**1**1
TA6		Clear remaining travel = 0 → 1 CCON = xxx0.xx11 CPOS = 01xx.xxxx
TA7		Start reference travel = 0 → 1 Stop = 1 CCON = xxx0.xx11 CPOS = 0xx0.0Px1
Key: P = positive edge, N = negative edge, x = any		

TA	Internal conditions	Activities of the user
TA8	Referencing finished or stopped.	Only for stop: Stop = 1 → 0 CCON = xxx0.xx11 CPOS = 0xxx.xxxN
TA9		Jog positive = 0 → 1 Stop = 1 CCON = xxx0.xx11 CPOS = 0xx0.Pxx1
TA10		Either – Jog positive = 1 → 0 – CCON = xxx0.xx11 – CPOS = 0xxx.0xx1 or – Stop = 1 → 0 – CCON = xxx0.xx11 – CPOS = 0xxx.xxxN
TA11		Jog negative = 0 → 1 Stop = 1 CCON = xxx0.xx11 CPOS = 0xxP.xxx1
TA12		Either – Jog negative = 1 → 0 – CCON = xxx0.xx11 – CPOS = 0xxN.xxx1 or – Stop = 1 → 0 – CCON = xxx0.xx11 – CPOS = 0xxx.xxxN
Key: P = positive edge, N = negative edge, x = any		

2.7.3 Special features dependent on FHPP operating mode

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

2.7.4 Examples of control and status bytes

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

0. Safeguard device control
1. Create readiness to operate – record selection
2. Create readiness to operate – direct mode
3. Error treatment
4. Homing
5. Positioning record selection
6. Positioning direct mode

For information about the state machine, see section 2.7.

For all examples:
For the controller and regulator enable of the CMM... additional digital I/Os are required, see manual for the controller used.

0. Safeguard device control

Step/ description	Control bytes									Status bytes								
	Byte	B 7	B 6	B 5	B 4	B 3	B 2	B 1	B 0	Byte	B 7	B 6	B 5	B 4	B 3	B 2	B 1	B 0
0.1 Device control software = on	Byte 1	OPM2	OPM1	LOCK	–	RESET	BRAKE	STOP	EN-ABLE	Byte 1	OPM2	OPM1	LOCK	24VL	FAULT	WARN	OPEN	EN-ABLE
	CCON	0	0	0	0	0	x	0	0	SCON	0	0	**1**	**1**	0	0	0	0
	Byte 2	–	CLEAR	TEACH	JOGN	JOGP	HOM	START	HALT	Byte 2	REF	STILL	DEV	MOV	TEACH	MC	ACK	HALT
	CPOS	0	0	0	0	0	0	0	0	SPOS	0	0	0	0	0	**1**	0	0
0: 0-signal; 1: 1-signal; x: not relevant (optional); F: Edge positive																		

Table 2/6: Control and status bytes “Device control active”

Description of 0. Ensure device control:

0.1 Device control via software (e.g. Festo Configuration Tool) is activated.
To control using the fieldbus interface, device control via the software has to be deactivated first.

1.Create readiness to operate – record selection

Step/ description	Control bytes									Status bytes								
	Byte	B 7	B 6	B 5	B 4	B 3	B 2	B 1	B 0	Byte	B 7	B 6	B 5	B 4	B 3	B 2	B 1	B 0
1.1 Basic status (device control software = off)	Byte 1	OPM2	OPM1	LOCK	–	RESET	BRAKE	STOP	EN-ABLE	Byte 1	OPM2	OPM1	LOCK	24VL	FAULT	WARN	OPEN	EN-ABLE
	CCON	0	0	0	0	0	x	0	0	SCON	0	0	**0**	**1**	0	0	0	0
	Byte 2	–	CLEAR	TEACH	JOGN	JOGP	HOM	START	HALT	Byte 2	REF	STILL	DEV	MOV	TEACH	MC	ACK	HALT
	CPOS	0	0	0	0	0	0	0	0	SPOS	0	0	0	0	0	**1**	0	0
1.2 Disable device control by software	Byte 1	OPM2	OPM1	LOCK	–	RESET	BRAKE	STOP	EN-ABLE	Byte 1	OPM2	OPM1	LOCK	24VL	FAULT	WARN	OPEN	EN-ABLE
	CCON	x	x	**1**	0	x	x	x	x	SCON	x	x	**0**	x	x	x	x	x
	Byte 2	–	CLEAR	TEACH	JOGN	JOGP	HOM	START	HALT	Byte 2	REF	STILL	DEV	MOV	TEACH	MC	ACK	HALT
	CPOS	0	x	x	x	x	x	x	x	SPOS	x	x	x	x	x	x	x	x
1.3 Enable drive, enable operation (Record select)	Byte 1	OPM2	OPM1	LOCK	–	RESET	BRAKE	STOP	EN-ABLE	Byte 1	OPM2	OPM1	LOCK	24VL	FAULT	WARN	OPEN	EN-ABLE
	CCON	**0**	**0**	x	0	0	x	**1**	**1**	SCON	**0**	**0**	0	1	0	0	**1**	**1**
	Byte 2	–	CLEAR	TEACH	JOGN	JOGP	HOM	START	HALT	Byte 2	REF	STILL	DEV	MOV	TEACH	MC	ACK	HALT
	CPOS	0	0	0	0	0	0	0	**1**	SPOS	0	0	0	0	0	1	0	**1**
0: 0-signal; 1: 1-signal; x: not relevant (optional); F: Edge positive																		

Table 2/7: Control and status bytes - “Establish readiness – set selection”

Description of 1. Create readiness to operate:

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

1.2 Disable device control by software.
Optionally, assuming of device control by the software can be disabled with CCON.B5 = 1 (LOCK).
⟶ Step 1.3

1.3 Enable drive in Record Select mode.
⟶ Reference travel: Example 4, Table 2/10.

If there are faults after switching on or after setting CCON.B0 (ENABLE):
⟶ Error treatment: see example 3, Table 2/9.

2. Create readiness to operate – direct mode

Step/ description	Control bytes									Status bytes								
	Byte	B 7	B 6	B 5	B 4	B 3	B 2	B 1	B 0	Byte	B 7	B 6	B 5	B 4	B 3	B 2	B 1	B 0
2.1 Basic status (device control software = off)	Byte 1	OPM2	OPM1	LOCK	–	RESET	BRAKE	STOP	EN-ABLE	Byte 1	OPM2	OPM1	LOCK	24VL	FAULT	WARN	OPEN	EN-ABLE
	CCON	0	0	0	0	0	x	0	0	SCON	0	0	**0**	**1**	0	0	0	0
	Byte 2	–	CLEAR	TEACH	JOGN	JOGP	HOM	START	HALT	Byte 2	REF	STILL	DEV	MOV	TEACH	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 by software	Byte 1	OPM2	OPM1	LOCK	–	RESET	BRAKE	STOP	EN-ABLE	Byte 1	OPM2	OPM1	LOCK	24VL	FAULT	WARN	OPEN	EN-ABLE
	CCON	x	x	**1**	0	x	x	x	x	SCON	x	x	**0**	x	x	x	x	x
	Byte 2	–	CLEAR	TEACH	JOGN	JOGP	HOM	START	HALT	Byte 2	REF	STILL	DEV	MOV	TEACH	MC	ACK	HALT
	CPOS	0	x	x	x	x	x	x	x	SPOS	x	x	x	x	x	x	x	x
2.3 Enable drive, enable operation (direct mode)	Byte 1	OPM2	OPM1	LOCK	–	RESET	BRAKE	STOP	EN-ABLE	Byte 1	OPM2	OPM1	LOCK	24VL	FAULT	WARN	OPEN	EN-ABLE
	CCON	**0**	**1**	x	0	0	x	**1**	**1**	SCON	**0**	**1**	0	1	0	0	**1**	**1**
	Byte 2	–	CLEAR	TEACH	JOGN	JOGP	HOM	START	HALT	Byte 2	REF	STILL	DEV	MOV	TEACH	MC	ACK	HALT
	CPOS	0	0	0	0	0	0	0	**1**	SPOS	0	0	0	0	0	1	0	**1**
0: 0-signal; 1: 1-signal; x: not relevant (optional); F: Edge positive																		

Table 2/8: Control and status bytes “Create readiness to operate – Direct mode”

Description of 2. Create readiness to operate:

2.1 Basic status of the drive when the supply voltage has been switched on.
→ Step 2.2 or 2.3

2.2 Disable device control by software.
Optionally, assuming of device control by the software can be disabled with CCON.B5 = 1 (LOCK).
→ Step 2.3

2.3 Enable drive in Direct mode.
→ Reference travel: Example 4, Table 2/10.

If there are faults after switching on or after setting CCON.B0 (ENABLE):
→ Error treatment: see example 3, Table 2/9.

3. Error treatment

Step/ description	Control bytes									Status bytes								
	Byte	B 7	B 6	B 5	B 4	B 3	B 2	B 1	B 0	Byte	B 7	B 6	B 5	B 4	B 3	B 2	B 1	B 0
3.1 Fault	Byte 1	OPM2	OPM1	LOCK	–	RESET	BRAKE	STOP	EN-ABLE	Byte 1	OPM2	OPM1	LOCK	24VL	FAULT	WARN	OPEN	EN-ABLE
	CCON	x	x	x	0	x	x	x	x	SCON	x	x	x	x	**1**	x	x	x
	Byte 2	–	CLEAR	TEACH	JOGN	JOGP	HOM	START	HALT	Byte 2	REF	STILL	DEV	MOV	TEACH	MC	ACK	HALT
	CPOS	0	x	x	x	x	x	x	x	SPOS	x	x	x	x	x	**0**	x	x
3.2 Warning	Byte 1	OPM2	OPM1	LOCK	–	RESET	BRAKE	STOP	EN-ABLE	Byte 1	OPM2	OPM1	LOCK	24VL	FAULT	WARN	OPEN	EN-ABLE
	CCON	x	x	x	0	x	x	x	x	SCON	x	x	x	x	x	**1**	x	x
	Byte 2	–	CLEAR	TEACH	JOGN	JOGP	HOM	START	HALT	Byte 2	REF	STILL	DEV	MOV	TEACH	MC	ACK	HALT
	CPOS	0	x	x	x	x	x	x	x	SPOS	x	x	x	x	x	**0**	x	x
3.3 Quit fault	Byte 1	OPM2	OPM1	LOCK	–	RESET	BRAKE	STOP	EN-ABLE	Byte 1	OPM2	OPM1	LOCK	24VL	FAULT	WARN	OPEN	EN-ABLE
	CCON	0	x	x	0	**F**	x	x	1	SCON	0	x	0	1	**0**	**0**	0	0
with CCON.B3	Byte 2	–	CLEAR	TEACH	JOGN	JOGP	HOM	START	HALT	Byte 2	REF	STILL	DEV	MOV	TEACH	MC	ACK	HALT
(RESET)	CPOS	0	0	0	0	0	0	x	x	SPOS	x	0	0	0	0	**1**	**0**	1
0: 0-signal; 1: 1-signal; x: not relevant (optional); F: Edge positive; N: Negative edge																		

Table 2/9: Control and status bytes “Error treatment”

Description of 3. Error treatment

3.1 A fault is shown with SCON.B3 (FAULT).
→ Positioning can no longer be undertaken.

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

3.3 Quit fault with positive edge at CCON.B3 (RESET).
→ Fault bit SCON.B2 (FAULT) or SCON.B3 (WARN) is reset
→ SPOS.B2 (MC) will be set
→ Drive is ready to operate

Errors and warnings can be also acknowledged using DIN5 (controller enable), see description on the controller in question.

4. Homing run (requires status 1.4 or 1.5)

Step/ description	Control bytes									Status bytes								
	Byte	B 7	B 6	B 5	B 4	B 3	B 2	B 1	B 0	Byte	B 7	B 6	B 5	B 4	B 3	B 2	B 1	B 0
4.1 Start reference travel	Byte 1	OPM2	OPM1	LOCK	–	RESET	BRAKE	STOP	EN-ABLE	Byte 1	OPM2	OPM1	LOCK	24VL	FAULT	WARN	OPEN	EN-ABLE
	CCON	0	x	x	0	0	x	1	1	SCON	0	x	0	1	0	0	1	1
	Byte 2	–	CLEAR	TEACH	JOGN	JOGP	HOM	START	HALT	Byte 2	REF	STILL	DEV	MOV	TEACH	MC	ACK	HALT
	CPOS	0	0	0	0	0	**F**	0	1	SPOS	0	0	0	0	0	**0**	**1**	1
4.2 Reference travel runs	Byte 1	OPM2	OPM1	LOCK	–	RESET	BRAKE	STOP	EN-ABLE	Byte 1	OPM2	OPM1	LOCK	24VL	FAULT	WARN	OPEN	EN-ABLE
	CCON	0	x	x	0	0	x	1	1	SCON	0	x	0	1	0	0	1	1
	Byte 2	–	CLEAR	TEACH	JOGN	JOGP	HOM	START	HALT	Byte 2	REF	STILL	DEV	MOV	TEACH	MC	ACK	HALT
	CPOS	0	0	0	0	0	**1**	0	1	SPOS	0	0	0	**1**	0	0	1	1
4.3 Reference travel concluded	Byte 1	OPM2	OPM1	LOCK	–	RESET	BRAKE	STOP	EN-ABLE	Byte 1	OPM2	OPM1	LOCK	24VL	FAULT	WARN	OPEN	EN-ABLE
	CCON	0	x	x	0	0	x	1	1	SCON	0	x	0	1	0	0	1	1
	Byte 2	–	CLEAR	TEACH	JOGN	JOGP	HOM	START	HALT	Byte 2	REF	STILL	DEV	MOV	TEACH	MC	ACK	HALT
	CPOS	0	0	0	0	0	**0**	0	1	SPOS	**1**	0	0	0	0	**1**	0	1
0: 0-signal: 1: 1-signal: x: not relevant (optional); F: Edge positive																		

Table 2/10: Control and status bytes “Reference travel”

Description of 4. Reference travel:

4.1 A positive edge at CPOS.B2 (HOM, Start reference travel) starts the reference travel. The start is confirmed with SPOS.B1 (Quit Start) as long as CPOS.B2 (HOM) is set.

4.2 Movement of the axis is shown with SPOS.B4 (MOV, Axis moves).

4.3 After successful reference travel SPOS.B2 (MC, Motion Complete) and SPOS.B7 (REF) will be set.

If there are faults during reference travel:
⟶ Error treatment: see example 3, Table 2/9.

5.Positioning record selection (requires status 1.3/2.3 and 4.)

Step/ description	Control bytes									Status bytes								
	Byte	B 7	B 6	B 5	B 4	B 3	B 2	B 1	B 0	Byte	B 7	B 6	B 5	B 4	B 3	B 2	B 1	B 0
5.1 Preselect record number (control byte 3)	Byte 3	Record number								Byte 3	Record number							
	Re-cord no.	Record no. (0 ...)								Re-cord no.	Previous record no. (0 ...)							
5.2 Start order	Byte 1	OPM2	OPM1	LOCK	–	RESET	BRAKE	STOP	EN-ABLE	Byte 1	OPM2	OPM1	LOCK	24VL	FAULT	WARN	OPEN	EN-ABLE
	CCON	0	0	x	0	0	x	1	1	SCON	0	0	0	1	0	0	1	1
	Byte 2	–	CLEAR	TEACH	JOGN	JOGP	HOM	START	HALT	Byte 2	REF	STILL	DEV	MOV	TEACH	MC	ACK	HALT
	CPOS	0	0	0	0	0	0	**F**	1	SPOS	1	0	0	0	0	**0**	1	1
5.3 Order runs	Byte 1	OPM2	OPM1	LOCK	–	RESET	BRAKE	STOP	EN-ABLE	Byte 1	OPM2	OPM1	LOCK	24VL	FAULT	WARN	OPEN	EN-ABLE
	CCON	0	0	x	0	0	x	1	1	SCON	0	0	0	1	0	0	1	1
	Byte 2	–	CLEAR	TEACH	JOGN	JOGP	HOM	START	HALT	Byte 2	REF	STILL	DEV	MOV	TEACH	MC	ACK	HALT
	CPOS	0	0	0	0	0	0	**1**	1	SPOS	1	0	0	**1**	0	0	**1**	1
	Byte 3	Record number								Byte 3	Record number							
	Re-cord no.	Record no. (0 ...)								Re-cord no.	Current record no. (0 ...)							
5.4 Order concluded	Byte 1	OPM2	OPM1	LOCK	–	RESET	BRAKE	STOP	EN-ABLE	Byte 1	OPM2	OPM1	LOCK	24VL	FAULT	WARN	OPEN	EN-ABLE
	CCON	0	0	x	0	0	x	1	1	SCON	0	0	0	1	0	0	1	1
	Byte 2	–	CLEAR	TEACH	JOGN	JOGP	HOM	START	HALT	Byte 2	REF	STILL	DEV	MOV	TEACH	MC	ACK	HALT
	CPOS	0	0	0	0	0	0	**0**	1	SPOS	1	0	0	**0**	0	**1**	**0**	1
	Byte 5...8	reserved								Byte 5...8	Position							
	–	Reserved								Actual posi-tion	Actual position (increments)							
0: 0-signal; 1: 1-signal; x: not relevant (optional); F: Edge positive																		

Table 2/11: Control and status bytes “Positioning Record Select”

Description of 5. Positioning Record Select:

(steps 5.1 ... 5.4 conditional sequence)

When the readiness to operate is created and the reference travel has been carried out, a positioning task can be started.

5.1 Preselect record number: Byte 3 of the output data
0 = Homing run
1 ... = Programmable positioning records

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

5.3 Movement of the axis is shown with SPOS.B4 (MOV, Axis moves).

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

If there are faults during positioning:
→ Error treatment: see example 3, Table 2/9.

6.Positioning direct mode (requires status 1.3/2.3 and 4.)

Step/ description	Control bytes									Status bytes								
	Byte	B 7	B 6	B 5	B 4	B 3	B 2	B 1	B 0	Byte	B 7	B 6	B 5	B 4	B 3	B 2	B 1	B 0
6.1 Preselect position and speed (bytes 4 and 5...8)	Byte 4 Velocity	Velocity preselect (0...100 %)								Byte 4 Velocity	Velocity reply message (0...100 %)							
	Byte 5...8 Nominal (target) position	Nominal position (increments)								Byte 5...8 Actual position	Actual position (increments)							
6.2 Start order	Byte 1 CCON	OPM2 0	OPM1 1	LOCK x	– 0	RESET 0	BRAKE x	STOP 1	EN-ABLE 1	Byte 1 SCON	OPM2 0	OPM1 1	LOCK 0	24VL 1	FAULT 0	WARN 0	OPEN 1	EN-ABLE 1
	Byte 2 CPOS	– 0	CLEAR 0	TEACH 0	JOGN 0	JOGP 0	HOM 0	START **F**	HALT 1	Byte 2 SPOS	REF 1	STILL 0	DEV 0	MOV 0	TEACH 0	MC **0**	ACK **1**	HALT 1
	Byte 3 CDIR	FUNC 0	FAST 0	XLIM 0	VLIM 0	CONT 0	COM2 0	COM1 0	ABS **C**	Byte 3 SDIR	FUNC 0	FAST 0	XLIM 0	VLIM 0	CONT 0	COM2 0	COM1 0	ABS **C**
6.3. Order runs	Byte 1 CCON	OPM2 0	OPM1 1	LOCK x	– 0	RESET 0	BRAKE x	STOP 1	EN-ABLE 1	Byte 1 SCON	OPM2 0	OPM1 1	LOCK 0	24VL 1	FAULT 0	WARN 0	OPEN 1	EN-ABLE 1
	Byte 2 CPOS	– 0	CLEAR 0	TEACH 0	JOGN 0	JOGP 0	HOM 0	START **1**	HALT 1	Byte 2 SPOS	REF 1	STILL 0	DEV 0	MOV **1**	TEACH 0	MC 0	ACK 1	HALT 1
6.4 Order concluded	Byte 1 CCON	OPM2 0	OPM1 1	LOCK x	– 0	RESET 0	BRAKE x	STOP 1	EN-ABLE 1	Byte 1 SCON	OPM2 0	OPM1 1	LOCK 0	24VL 1	FAULT 0	WARN 0	OPEN 1	EN-ABLE 1
	Byte 2 CPOS	– 0	CLEAR 0	TEACH 0	JOGN 0	JOGP 0	HOM 0	START **0**	HALT 1	Byte 2 SPOS	REF 1	STILL 0	DEV 0	MOV **0**	TEACH 0	MC **1**	ACK 0	HALT 1
0: 0-signal; 1: 1-signal; x: not relevant (optional); F: Edge positive S: Positioning condition: 0= absolute; 1 = relative																		

Table 2/12: Control and status bytes "Positioning Direct Mode"

Description of Positioning Direct Mode:

(step 6.1 ... 6.4 conditional sequence)

When the readiness to operate is created and the reference travel has been carried out, a nominal position must be pre-selected.

6.1 The nominal position is transferred in increments in bytes 5...8 of the output word.
The nominal speed is transferred in % in byte 4 (0 = no speed; 100 = max. speed).

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

6.3 Movement of the axis is shown with SPOS.B4 (MOV, Axis moves).

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

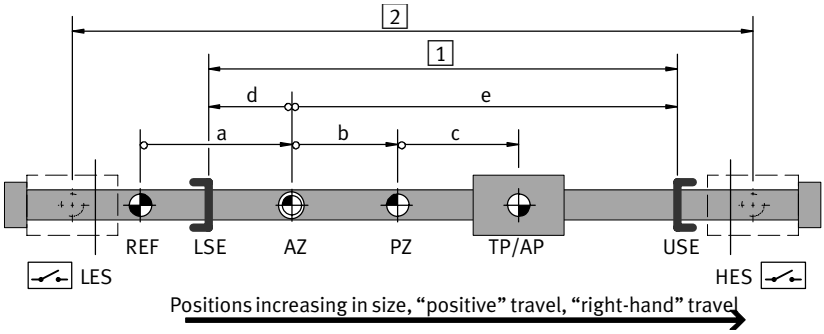
If there are faults during positioning:
→ Error treatment: see example 3, Table 2/9.

Chapter 3

Contents

3.1 Reference system for electric drives

Reference system for electric linear drives

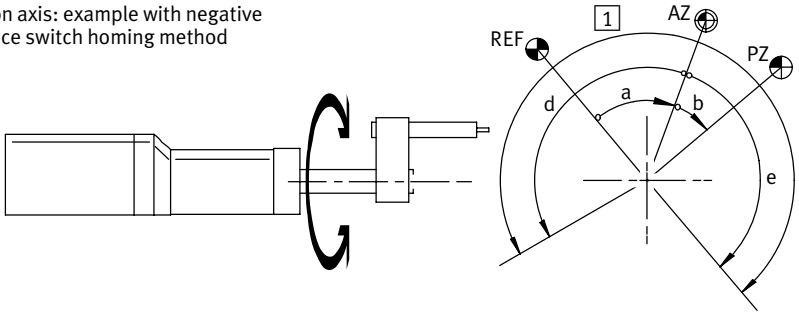


REF	Reference point	a	Offset axis zero point
AZ	Axis zero point	b	Offset project zero point
PZ	Project zero point	c	Offset target/actual position
LSE	Lower software end position	d, e	Offset software end positions
USE	Upper software end position	1	Usable stroke
LES	Lower end switch	2	Nominal stroke
HES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

Table 3/13: Reference system for electric drives

Reference system for electric rotary drives

Rotation axis: example with negative reference switch homing method



REF	Reference point: Point ascertained during the reference run: reference switch, end switch or stop, with index pulse, where applicable.
AZ	Axis zero point: reference point for the project zero point and the software end positions.
PZ	Project zero point: reference point (zero point) for actual position and absolute positions in the traversing record table (always 0 in I/O version).
a	Offset axis zero point: Distance of axis zero point AZ from reference point REF
b	Offset project zero point: distance from AZ
d, e	Offset software end positions: limit the permitted positioning range (usable stroke). Optionally: endless positioning possible
1	Effective stroke: permitted positioning range

Table 3/14: Reference system for electric rotary drives

3.2 Calculation rules for reference system

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

Table 3/15: Calculation rules for the measuring reference system with incremental measuring systems

3.3 Homing

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

Various homing modes are permitted, depending on the controller.
An overview is shown in Table 3/16 (as at May 2007).

For a description of the homing modes, see section 3.3.2.

Homing mode			Controller		
Hex	**Dec**	**Description**	**CMMS-ST**	**CMMP-AS**	**CMMS-AS**
01h	1	Negative limit switch with index pulse	**x** [1]	**x**	**x**
02h	2	Positive limit switch with index pulse	**x** [1]	**x**	**x**
07h	7	Reference switch in positive direction with index pulse	–	**x**	–
0Bh	11	Reference switch in negative direction with index pulse	–	**x**	–
11h	17	Negative limit switch	**x**	**x**	**x**
12h	18	Positive limit switch	**x**	**x**	**x**
17h	23	Reference switch in positive direction	–	**x**	–
1Bh	27	Reference switch in negative direction	–	**x**	–
21h	33	Index pulse in a negative direction	**x** [1]	**x**	**x**
22h	34	Index pulse in a negative direction	**x** [1]	**x**	**x**
23h	35	Current position	**x**	**x**	**x**
FFh	-1	Negative stop with index pulse	**x** [1]	**x**	**x**
FEh	-2	Positive stop with index pulse	**x** [1]	**x**	**x**
EFh	-17	Negative stop	**x** [1]	**x**	**x**
EEh	-18	Positive stop	**x** [1]	**x**	**x**
E9h	-23	Reference switch in positive direction with travel to stop or limit switch	–	**x**	–
E5h	-27	Reference switch in negative direction with travel to stop or limit switch	–	**x**	–
[1] Only possible for motors with an encoder					

Table 3/16: Permissible homing modes, as of May 2007

3.3.1 Homing for electric drives

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

Sequence:

1. Search for the reference point in accordance with the configured method.
2. Move relative to the reference point around the “Offset axis zero point.”
3. Set at axis zero point:
   Current position = 0 – project zero offset.

Overview of parameters involved (see also section 5.4.12)

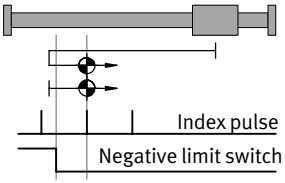
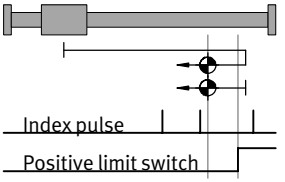
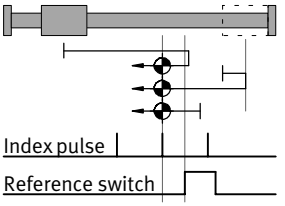
Parameters involved	Description	PNU
	Offset axis zero point	1010
	Reference travel (homing) method	1011
	Reference travel speeds	1012
	Homing accelerations	1013
	Reference travel required	1014
	CMMP only: Reference travel maximum torque	1015
Start (FHPP)	CPOS.B2 = positive edge: Start homing	
Reply message (FHPP)	SPOS.B1 = positive edge: Start acknowledgment SPOS.B7 = drive referenced	
Requirement	Device control by PLC/field bus Controller must be in status “Operation enabled” There must not be any command for jogging	

Table 3/17: Parameters involved in homing travel

3.3.2 Homing methods

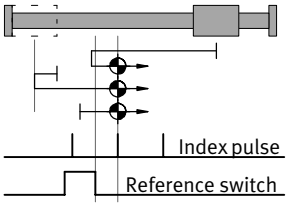
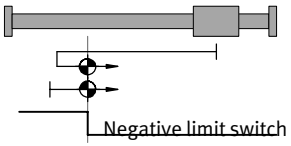
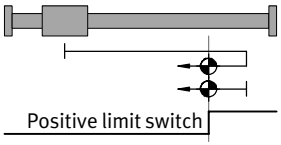
The homing methods are oriented towards CANopen DS 402.

Homing methods

hex	dec.	Description	
01h	1	**Negative limit switch with index pulse** [1)] 1. If the negative limit switch is inactive: run at search speed in negative direction to negative limit switch. 2. Run at crawl speed in positive direction until the limit switch becomes inactive, then on to first index pulse. This position is saved as a reference point. 3. If the axis zero point ≠ 0: run at travel speed to axis zero point.	
02h	2	**Positive limit switch with index pulse** [1)] 1. If the positive limit switch is inactive: run at search speed in positive direction to positive limit switch. 2. Run at crawl speed in positive direction until the limit switch becomes inactive, then on to first index pulse. This position is saved as a reference point. 3. If the axis zero point ≠ 0: run at travel speed to axis zero point.	
07h	7	**Reference switch in positive direction with index pulse** [1)] 1. If reference switch is inactive: run at search speed in positive direction to reference switch. If the stop or limit switch is reached in the process: run at search speed in positive direction to reference switch. 2. Run at crawl speed in positive direction until the reference switch becomes inactive, then on to first index pulse. This position is saved as a reference point. 3. If the axis zero point ≠ 0: run at travel speed to axis zero point.	

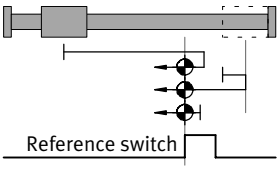
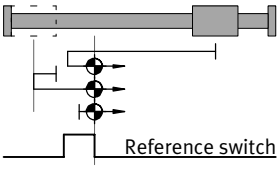
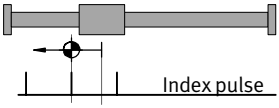
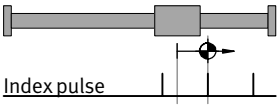
[1)] Only possible for motors with an encoder.

Homing methods

hex	dec.	Description	
0B	11	**Reference switch in negative direction with index pulse** [1)] 1. If reference switch is inactive: run at search speed in negative direction to reference switch. If the stop or limit switch is reached in the process: run at search speed in negative direction to reference switch. 2. Run at crawl speed in positive direction until the reference switch becomes inactive, then on to first index pulse. This position is saved as a reference point. 3. If the axis zero point ≠ 0: run at travel speed to axis zero point.	
11h	17	**Negative limit switch** 1. If the negative limit switch is inactive: run at search speed in negative direction to negative limit switch. 2. Run at crawl speed in positive direction until limit switch becomes inactive. This position is saved as a reference point. 3. If the axis zero point ≠ 0: run at travel speed to axis zero point.	
12h	18	**Positive limit switch** 1. If the positive limit switch is inactive: run at search speed in positive direction to positive limit switch. 2. Run at crawl speed in negative direction until proximity switch becomes inactive. This position is saved as a reference point. 3. If the axis zero point ≠ 0: run at travel speed to axis zero point.	

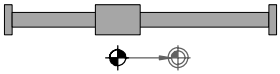
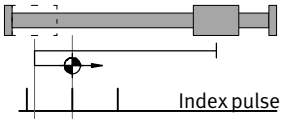
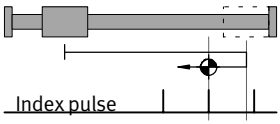
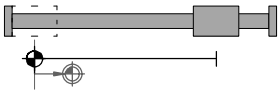
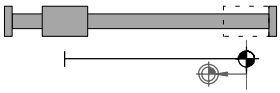
[1)] Only possible for motors with an encoder.

Homing methods

hex	dec.	Description	
17h	23	**Reference switch in positive direction** 1. If reference switch is inactive: run at search speed in positive direction to reference switch. If the stop or limit switch is reached in the process: run at search speed in positive direction to reference switch. 2. Run at crawl speed in negative direction until reference switch becomes inactive. This position is saved as a reference point. 3. If the axis zero point ≠ 0: run at travel speed to axis zero point.	
1Bh	27	**Reference switch in negative direction** 1. If reference switch is inactive: run at search speed in negative direction to reference switch. If the stop or limit switch is reached in the process: run at search speed in negative direction to reference switch. 2. Run at crawl speed in positive direction until reference switch becomes inactive. This position is saved as a reference point. 3. If the axis zero point ≠ 0: run at travel speed to axis zero point.	
21h	33	**Index pulse in a negative direction** [1)] 1. Run at crawl speed in negative direction to index pulse. This position is saved as a reference point. 2. If the axis zero point ≠ 0: run at travel speed to axis zero point.	
22h	34	**Index pulse in a positive direction** [1)] 1. Run at crawl speed in positive direction to index pulse. This position is saved as a reference point. 2. If the axis zero point ≠ 0: run at travel speed to axis zero point.	

[1)] Only possible for motors with an encoder.

Homing methods

hex	dec.	Description	
23h	35	**Current position** 1. The current position is saved as the reference point. 2. If the axis zero point ≠ 0: run at travel speed to axis zero point. Note: If the reference system is shifted, runs to the proximity switch or fixed stop are possible. This is therefore generally used for rotating axes.	
FFh	-1	**Negative stop with index pulse** [1) 2)] 1. Run at search speed in negative direction to stop. 2. Run at crawl speed in positive direction to next index pulse. This position is saved as a reference point. 3. If the axis zero point ≠ 0: run at travel speed to axis zero point.	
FEh	-2	**Positive stop with index pulse** [1) 2)] 1. Run at search speed in positive direction to stop. 2. Run at crawl speed in negative direction to next index pulse. This position is saved as a reference point. 3. If the axis zero point ≠ 0: run at travel speed to axis zero point.	
EFh	-17	**Negative stop** [1) 2) 3)] 1. Run at search speed in negative direction to stop. This position is saved as a reference point. 2. If the axis zero point ≠ 0: run at travel speed to axis zero point.	
EEh	-18	**Positive stop** [1) 2) 3)] 1. Run at search speed in positive direction to stop. This position is saved as a reference point. 2. If the axis zero point ≠ 0: run at travel speed to axis zero point.	

1) Only possible for motors with an encoder.
2) Limit switches are ignored during the run to the stop.
3) Since the axis is not intended to stay at the stop, the run has to parametrized to the axis zero point and the axis zero point offset has to be ≠ 0.

Homing methods

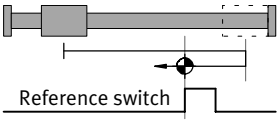
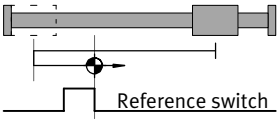
hex	dec.	Description	
E9h	-23	**Reference switch in positive direction with travel to stop or limit switch.** 1. Run at search speed in positive direction to stop **or limit switch**. 2. Run at search speed in negative direction to reference switch. 3. Run at crawl speed in negative direction until reference switch becomes inactive. This position is saved as a reference point. 4. If the axis zero point ≠ 0: run at travel speed to axis zero point.	
E5h	-27	**Reference switch in negative direction with travel to stop or limit switch.** 1. Run at search speed in negative direction to stop **or limit switch**. 2. Run at search speed in positive direction to reference switch. 3. Run at crawl speed in positive direction until reference switch becomes inactive. This position is saved as a reference point. 4. If the axis zero point ≠ 0: run at travel speed to axis zero point.	

Table 3/18: Overview of homing methods

3.4 Jog mode

In the “Operation enabled” status, the drive can be jogged in the positive/negative directions. This function is usually used for:

– Moving to teaching positions

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

– Manual positioning as normal operating mode (manually operated feed)

Sequence

1. When one of the signals “Jog positive / Jog negative” is set, the drive starts to move slowly. Due to the slow speed, a position can be defined very accurately.

2. If the signal remains set for longer than the configured “phase 1 duration”, the speed is increased until the configured maximum speed is reached. In this way large strokes can be traversed quickly.

3. If the signal changes to 0, the drive is braked with the pre-set maximum deceleration.

4. Only if the drive is referenced:
   If the drive reaches a software end position, it will stop automatically. The software end position is not exceeded, the path for stopping depends on the ramp set. Jogging operation is also exited here when Jogging = 0.

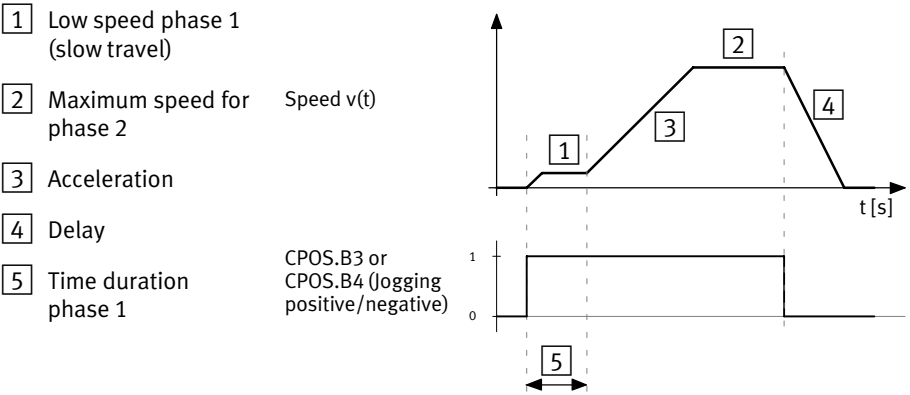


Fig. 3/4: Sequence diagram for jogging mode

Overview of parameters involved (see section 5.4.7)		
Parameters involved	**Description**	**PNU**
	Jog mode speed phase 1	530
	Jog mode speed phase 2	531
	Jog mode acceleration	532
	Jog mode delay	533
	Jog mode time phase 1 (T1)	534
Start (FHPP)	CPOS.B3 = positive edge: jog positive (towards larger actual values) CPOS.B4 = positive edge: jog negative (towards smaller actual values)	
Acknowledgment (FHPP)	SPOS.B4 = 1: drive moving SPOS.B2 = 0: (Motion Complete)	
Requirement	Device control by PLC/field bus Controller must be in status “Operation enabled”	

Table 3/19: Parameters involved in jogging mode

3.5 Teaching via field bus

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

Sequence

1. The drive will be moved to the desired position by the jogging mode or manually. This can be accomplished in jogging mode by positioning (or by moving manually in the “Drive blocked” status in the case of motors with an encoder).

2. The user must make sure that the desired parameter is selected. For this, the parameter “Teach target” and, if applicable, the correct record address must be entered.

Teach target (PNU 520)	**is taught**
= 1 (specification)	Nominal position in the position set – Record selection: Traversing record after control byte 3 – Direct mode: Traversing record after PNU=400
= 2	Axis zero point
= 3	Project zero point
= 4	Lower software end position
= 5	Upper software end position

Table 3/20: Overview of teach targets

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

1 PLC:
Prepare teaching

2 Controller:
Ready for teaching

3 PLC:
Teach now

4 Controller:
Value transferred

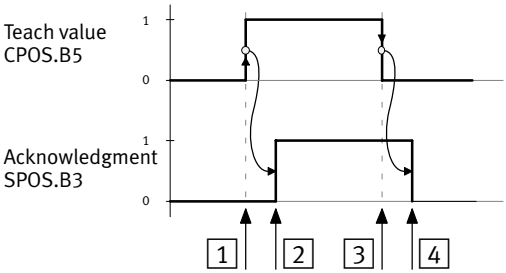


Fig. 3/5: Handshake when teaching

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

Overview of parameters involved (see sections 5.4.6 and 5.4.7)

Parameters involved	**Description**	**PNU**
	Teach target	520
	Record number	400
	Offset project zero point	500
	Software end positions	501
	Axis zero point offset (electric drives)	1010
Start (FHPP)	CPOS.B5 = falling edge: teach value	
Acknowledgment (FHPP)	SPOS.B2 = 1: value transferred	
Requirement	Device control by PLC/field bus Controller must be in status “Operation enabled”	

Table 3/21: Teach parameters involved

3.6 Carry out record (Record selection)

A record can be started in the “Drive enabled” state. This function is usually used for:

– Moving to any position in the record list by the PLC

– Processing a positioning profile by linking records

– Known target positions which seldom change (formulation change)

Sequence

1. Set the required record number in the PLC’s output data. Until the start, the controller replies with the number of the record last processed.

2. With a rising edge at CPOS.B1 (START) the controller accepts the record number and starts the positioning task.

3. The controller signals with the rising edge at Quit Start that the PLC output data has been accepted and that the positioning task is now active. The positioning command continues to be executed, even if CPOS.B1 (START) is reset to zero.

4. When the record is concluded, SPOS.B2 (MC) is set.

Causes of faults in application:

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

– The target position and/or the preselect position cannot be reached.

– Invalid record number.

– Record not initialized.

In the event of conditional record chaining (see section 3.6.3):

If a new speed and/or a new target position is specified in the movement, the remaining path to the target position must be large enough to reach a standstill with the braking ramp that was set.

Overview of parameters involved (see section 5.4.6)

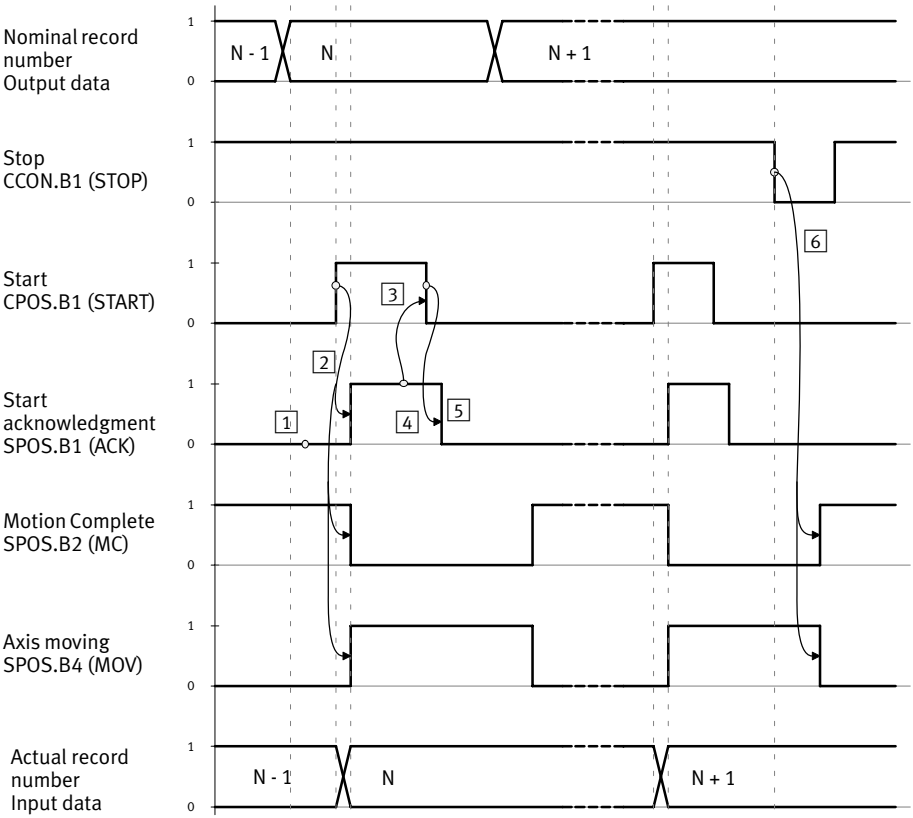
Parameters involved	Description	PNU
	Record number	400
	All parameters of the record data, see sections 3.6.2, Table 3/23	401 ... 414
Start (FHPP)	CPOS.B1 = positive edge: start Jogging and referencing have priority.	
Acknowledgment (FHPP)	SPOS.B2 = 0: Motion Complete SPOS.B1 = positive edge: start acknowledgment SPOS.B4 = 1: drive moving	
Requirement	Device control by PLC/field bus Controller must be in status “Operation enabled” Record number must be valid	

Table 3/22: Parameters involved in Record selection

3.6.1 Record selection flow diagram

Fig. 3/6, Fig. 3/7 and Fig. 3/8 show typical flow diagrams for starting and stopping a record.

Start/stop record



1 Prerequisite: "Quit Start" = 0

2 Rising edge at "Start" causes the new record number N to be accepted and "Quit Start" to be set.

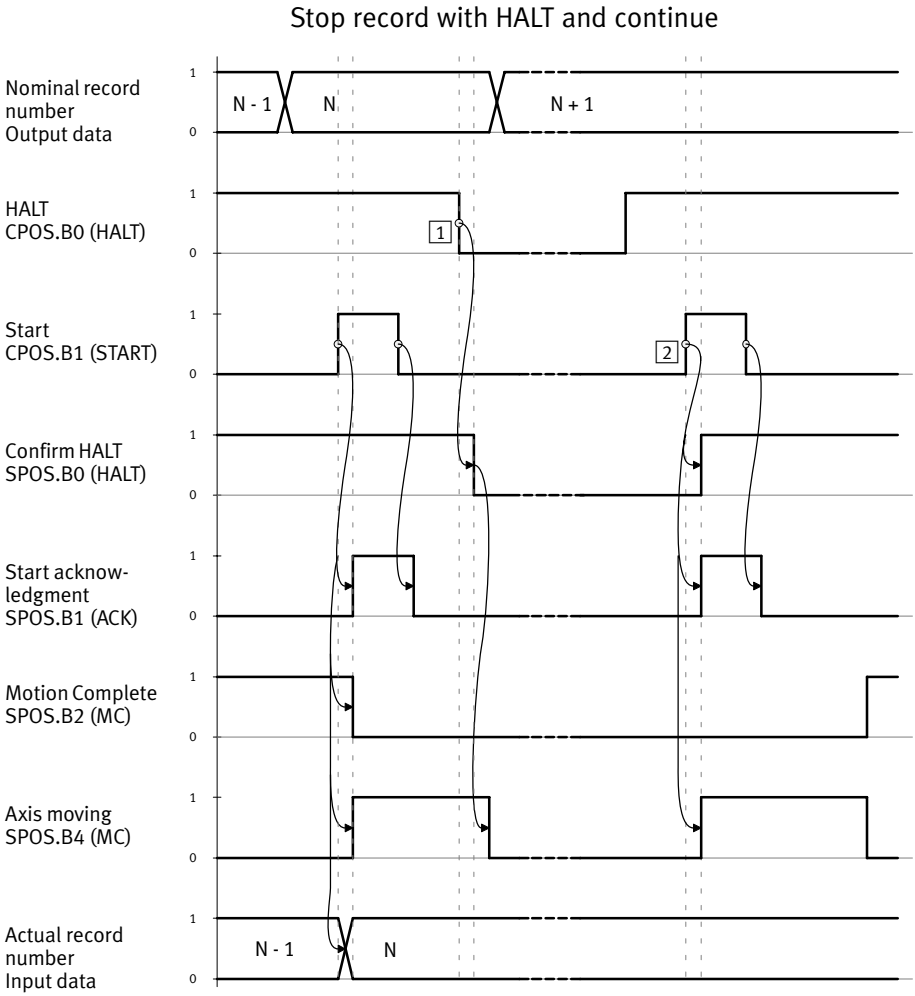
3 As soon as "Quit Start" is recognized by the PLC, "Start" may be set to 0 again.

4 The controller reacts with a falling edge at "Quit Start"

5 As soon as "Quit Start" is recognized by the PLC, the next record number may be started

6 A currently running positioning task can be stopped with "Stop"

Fig. 3/6: Sequence diagram Start/stop record



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

[2] Rising edge at "Start" starts record N again, "Confirm HALT" is set

Fig. 3/7: Sequence diagram for Stop record with HALT and Continue

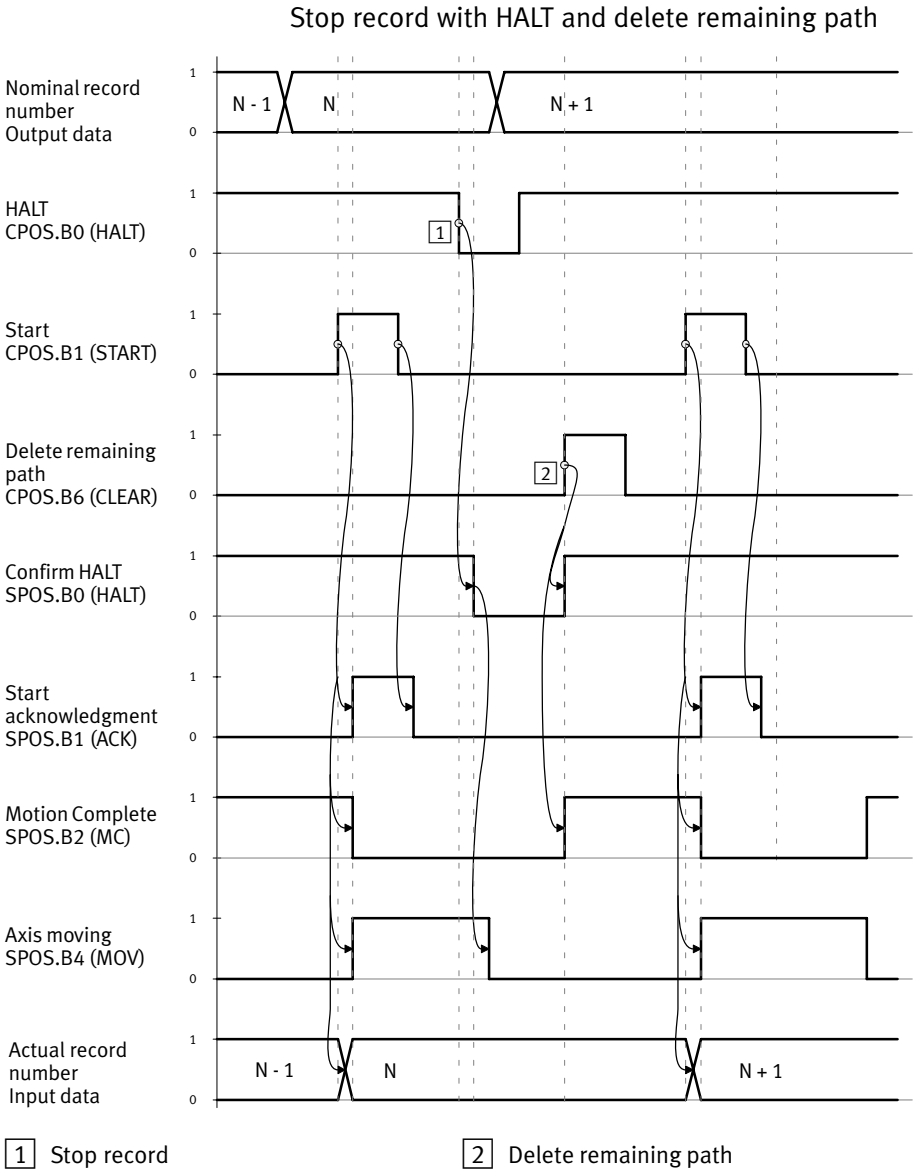


Fig. 3/8: Sequence diagram for Stop record with HALT and delete remaining path

3.6.2 Record composition

A positioning task in Record Select mode is written with one record of nominal values. Each nominal value is addressed by its own PNU. A record consists of the nominal values with the same subindex.

PNU	Name	Description
401	Record control byte 1	Setting for positioning task: absolute/relative, position/torque control, ...
402	Record control byte 2	Record control: Settings for conditional record chaining
403	Reserved	– (not supported by CMM...)
404	Setpoint value	Setpoint value as per record control byte 1
405	Preselected value	Preselected value as per record control byte 2
406	Speed	Auxiliary setpoint: nominal speed
407	Acceleration	Auxiliary setpoint: nominal acceleration during start up
408	Delay	Auxiliary setpoint: nominal acceleration during braking
409	Reserved	– (not supported by CMM...)
410	Reserved	– (not supported by CMM...)
411	Reserved	– (not supported by CMM...)
412	Reserved	– (not supported by CMM...)
413	Jerk-free filter time	Auxiliary setpoint: filter time for smoothing the profile ramps
414	Record group	CMMS-ST only: number of the traversing record profile

Table 3/23: Parameters for the traversing record

3.6.3 Conditional record chaining (PNU 402)

Record selection mode allows multiple positioning jobs to be linked. This means that, starting at CPOS.B1, various records are automatically executed one after the other. This allows a travel profile to be defined, e.g. switching to another speed after a position is reached.

To do this, the user sets a (decimal) condition in RCB2 to define that the following record is automatically executed after the current record.

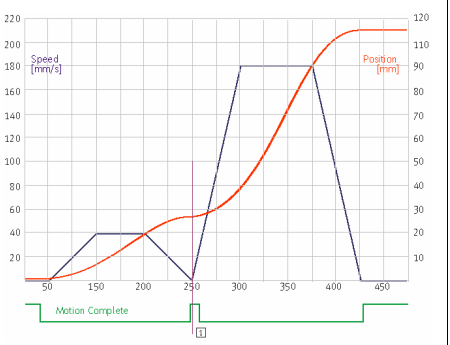
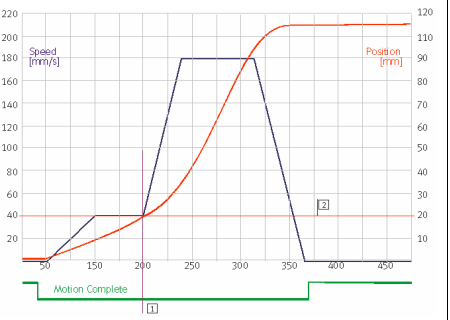
It is only possible to set all of the parameters for the record chaining (“route program”) (e.g. the following record) using the FCT.

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

Record control byte 2 (PNU 402)	
Bit 0 ... 6	Numerical value 0...128: step criterion as a list, see Table 3/25
Bit7	= 0: Record chaining (bit0...6) is not disabled (default) = 1: Record chaining disabled

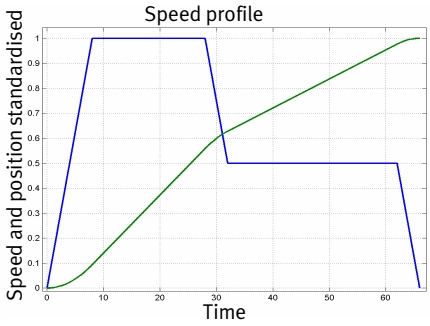
Table 3/24: Settings for conditional record chaining

Chaining criteria

Value	Condition	Description	
0	–	No automatic continuation (no chaining, no switching to the next record)	
1 [1)]	MC	The preselect value is interpreted as waiting time (delay) in milliseconds. Further switching takes place when the set target value is reached, i.e. when the MC condition is fulfilled (MC=1) and when the waiting time (delay) has also expired. Notes: Thus the axis is at a standstill for a moment during positioning. Not necessarily the case for torque control.	
2 [1)]	Position	The preselected value is interpreted as the position value 2. The next record is executed as soon as the current actual position exceeds the preselected value in the direction of travel 1: As there is no need to stop, the drive reaches its target position quicker.	
3 [1)]	Torque	The preselected value is interpreted as the torque. The next record is executed once the current actual torque exceeds the preselected value in the direction of travel. It is not absolutely necessary for a torque command to be specified here. It is also possible to position to the end point. When a specific actual torque is reached, torque control is activated.	

1) Not supported by CMM...

Chaining criteria

Value	Condition	Description
4	Standstill	The next record is executed once the drive comes to a standstill and then the time T1 specified as the preselected value has expired. (Travel to end point)
5	Time	The preselected value is interpreted as time in milliseconds. The next record is executed once this time has expired (after the start).
6	Input pos. edge	The next record is executed if a rising edge is identified at the local input. The preselected value includes the input's bit address. Preselected value = 1: NEXT1 Preselected value = 2: NEXT2
7	Input neg. edge	The next record is executed if a falling edge is identified at the local input. The preselected value includes the input's bit address. Preselected value = 1: NEXT1 Preselected value = 2: NEXT2
8 [1)]	Speed profile	The setpoint generator calculates the trajectory so that the record's nominal speed is active in the target position. The final speed is therefore not 0. The preselected value is ignored. Note: in type 1, the user only defines the switch position; the user has not influence over the speed. 
9	Input pos. edge waiting	The next record is executed after the current record ends, if a rising edge is identified at the local input. The preselected value includes the input's number. Preselected value = 1: NEXT1 Preselected value = 2: NEXT2
10	Input neg. edge waiting	The next record is executed after the current record ends, if a falling edge is identified at the local input. The preselected value includes the input's number. Preselected value = 1: NEXT1 Preselected value = 2: NEXT2

1) Not supported by CMM...

Chaining criteria

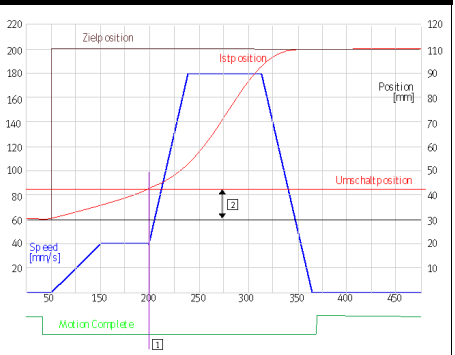
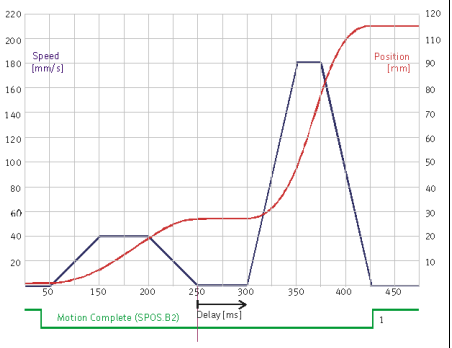
Value	Condition	Description	
11 [2)]	Position (relative)	This switching corresponds to type 2 with the difference that the position is specified not absolutely, but as relative to the last set position [2]. Switching takes place as soon as the current position has exceeded the preselect value in the direction of movement [1]. **Important:** For a reproducible switching position, the specification must be calculated relative to the last target position and not to the actual position.	
12	Internal MC condition	Like condition 1, but without external MC signal between the individual records. The external MC signal (SPOS.B2) is not set until after the last record of the further switching.	
2) Not supported by CMMS			

Table 3/25: Chaining criteria

3.7 Direct mode

In the status “Drive enabled” (Direct Mode) a task is formulated directly in the I/O data that is transmitted via the field bus. Some of the setpoint values for the position are reserved in the PLC.

The function is used in the following situations:

– Moving to any position within the work stroke

– The target positions are unknown during planning or change frequently (e.g. several different work item positions)

– A positioning profile consisting of chaining records (G24 function) is not necessary

– The drive is to continuously follow a setpoint value

If short wait times are not critical, it is possible to implement a positioning profile externally through the PLC by chaining records.

Causes of faults in application

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

– Target position cannot be reached or lies outside the software end positions.

– Load torque is too large.

Overview of parameters involved (see section 5.4.7)

Parameters involved	Description	PNU
Position specifications	Base velocity [1]	540
	Direct mode acceleration	541
	Direct mode deceleration	542
	Jerk-free filter time	546
Torque specifications (for CMMP only) [2]	Base torque ramp [1]	550
	Torque target window	552
	Damping time	553
	Permissible speed during torque control	554
Rotational speed specifications	Base acceleration ramp [1]	560
	Rotational speed target window (for CMMP only) [2]	561
	Damping time target window (for CMMP only) [2]	562
	Standstill target window (for CMMP only) [2]	563
	Standstill target window damping time (for CMMP only) [2]	564
	Torque limiting (for CMMP only) [2]	565
Start (FHPP)	CPOS.B1 = positive edge: Start CDIR.B0 = nominal position absolute/relative CDIR.B1/B2 = control mode (see section 2.5.3) CDIR.B3 = continuous following	
Acknowledgment (FHPP)	SPOS.B2 = 0: Motion Complete SPOS.B1 = positive edge: Start acknowledgment SPOS.B4 = 1: drive moving	
Requirement	Device control by PLC/field bus Controller must be in status “Operation enabled”	

[1] In the control bytes, the PLC transfers a percent value which is multiplied by the maximum permitted value, in order to achieve the final setpoint value.
[2] For supported functions, see 2.4

Table 3/26: Parameters involved in Direct Mode

3.7.1 Sequence of discrete setpoint value

1. The user sets the desired setpoint value (position, torque) and the positioning condition (absolute/relative, speed) in his or her output data.
2. With a rising edge at START (CPOS.B1) the controller accepts the setpoint values and starts the positioning task.

   After the start, a new setpoint value can be started at any time. There is no need to wait for MC.
3. Once the last setpoint position is concluded, MC (SPOS.B2) is set.

Starting the positioning task

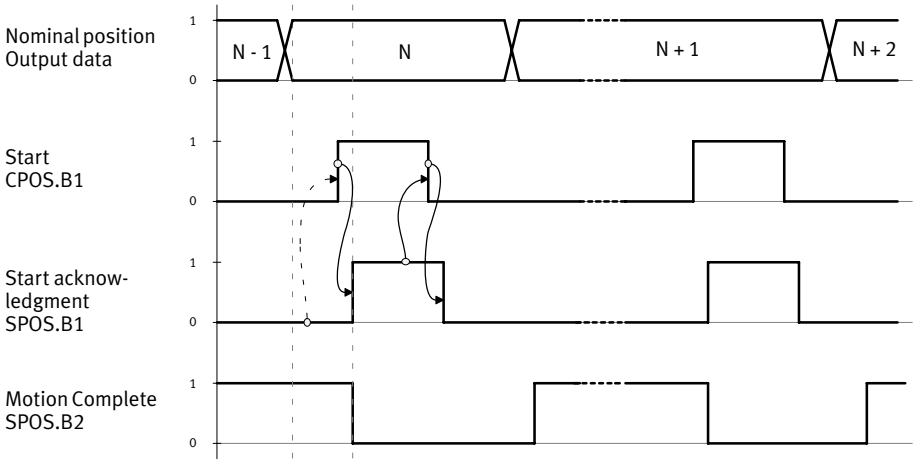


Fig. 3/9: Start the positioning task

The sequence of the remaining control and status bits as well as the functions HALT and Stop react as per the Record select function, see Fig. 3/6, Fig. 3/7 and Fig. 3/8.

3.7.2 Torque control mode sequence

Torque or moment control is prepared by switching the control mode with the CDIR - COM1/2 bits. The drive stands with the position controlled. The signal “MC” (Motion Complete) is used in this control mode to mean “Setpoint torque value reached”.

After the setpoint specification, the start signal (start bit) creates the torque/moment using the torque ramp in the direction indicated by the setpoint value’s prefix and the active torque control mode is displayed via the SDIR - COM1/2 bits.

For CMMP:
The speed is limited to the value in the parameter “Maximum speed”. Once this speed has been reached, the bit “Speed limit reached” is set in the status byte SDIR.

Once the setpoint value has been reached, taking into account the target window and the time window, the “MC” signal is set. Torque/moment continue to be controlled.

When exceeding the path set in the path/stroke monitoring (relative to the start position), the bit “Stroke limit reached” is set in the status byte SDIR. No error or warning bit is assigned. The drive is slowed with the emergency-stop ramp, held with the position controlled at the current position, and the “MC signal” is set.

Causes of faults in application

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

Setpoint specification / actual value query in direct mode torque control mode:

CCON.B6 (OPM1) = 1, CCON.B7 (OPM2) = 0
CDIR.B1 (COM1) = 1, CDIR.B2 (COM2) = 0

Direct mode

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Output data	CCON	CPOS	CDIR	Nominal value 1 (speed)	Nominal value 2 (torque)			
Input data	SCON	SPOS	SDIR	Actual value 1 (actual torque)	Actual value 2 (actual position)			

data	Meaning	Unit(s)
Nominal value 1	Limit speed (optional)	Percentage of maximum (PNU 502) or percentage of base value (PNU 540)
Nominal value 2	Torque	Percentage of nominal value (PNU 1036)
Actual value 1	Actual torque	Percentage of nominal value (PNU 1036)
Actual value 2	Actual position	Positioning unit (see PNU 1004)

3.7.3 Speed adjustment mode sequence

Speed adjustment is only supported in the “Direct mode”. Only “discrete setpoint adjustment” (compare section 3.7.1) is supported.

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

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

Motion Complete / standstill notification

The same comparator type is used to determine “speed reached” and “speed 0” and it behaves as per Fig. 3/10, see Table 3/27.

Setpoint value	Specifications for reaching MC (Motion Complete)
≠ 0	Target speed: Setpoint value as per input data Tolerance: Speed target window (PNU 561) Response time: Damping time target window (PNU 562)
= 0	Target speed: Setpoint value as per input data Tolerance: Standstill target window (PNU 563) Response time: Standstill target window damping time (PNU 564)

Table 3/27: Motion Complete / standstill notification specifications

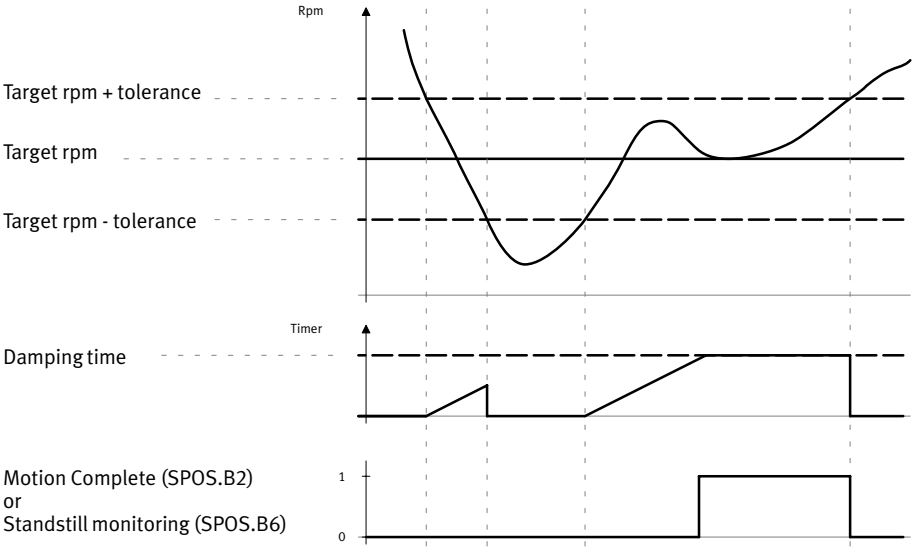


Fig. 3/10: Motion Complete / standstill notification

3.8 Standstill monitoring

With the standstill monitoring it is clear that the target position window is exited at a standstill.

Standstill monitoring is based on position control only.

When the target position has been reached and MC signalled in the status word, the drive switches to the “standstill” state, bit SPOS.B6 (standstill monitor) is reset. If, in this status, the drive is removed from the standstill position window for a defined time due to external forces or other influences, the bit SPOS.B6 will be set.

As soon as the drive is in the standstill position window again for the standstill monitoring time, the bit SPOS.B6 will be reset.

1 Target position
2 Actual position
3 Standstill monitoring (SPOS.B6)
4 Motion Complete (SPOS.B2)
5 Standstill position window
6 Target position window
7 Monitoring time (position window time)
8 Standstill monitoring time

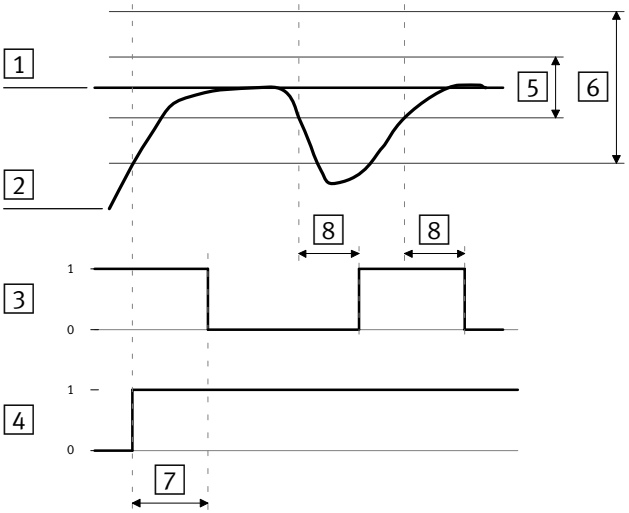


Fig. 3/11: Standstill monitoring

The standstill monitoring cannot be switched on or off explicitly. It becomes inactive when the standstill position window is set to “0”.

Overview of parameters involved (see section 5.4.12)		
Parameters involved	**Description**	**PNU**
	Target position window	1022
	Adjustment time for position	1023
	Nominal position	1040
	Current position	1041
	Standstill position window	1042
	Standstill monitoring time	1043
Start (FHPP)	SPOS.B2 = positive edge: Motion Complete	
Acknowledgment (FHPP)	SPOS.B6 = 1: drive has moved out of standstill position window	
Requirement	Device control by PLC/field bus Controller must be in status “Operation enabled”	

Table 3/28: Parameters involved in standstill monitoring

Chapter 4

Contents

4.1 Classifying the faults

We differentiate between the following types of fault:

– warnings

– malfunction type 1 (end stage not switched off)

– malfunction type 2 (end stage switched off)

The classification of the possible faults is specified separately for each controller. The basis for this classification is the way in which each controller has to behave for each fault.

The fault numbers are defined globally for all drives in a separate document, in order to enable a uniform diagnosis from the Festo profile for handling and positioning.

The controllers indicate faults by appropriate error messages or warnings. These can be evaluated via the:

– display

– status bytes (see section 2.4)

– bus-specific diagnosis (see description of the fieldbus for the controller in question)

– diagnostic memory (see section 4.2)

– FCT (see FCT help)

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

- controller and end stage remain active
- the current positioning job is not interrupted
- it is possible to start a new positioning job
- the SCON.B2 (WARN) bit is allocated
- If the cause of the warning disappears, the SCON.B2 bit is automatically deleted again.
- CMMP only:
  The warning numbers are logged in the warning register (PNU 211).

Causes of warnings

- Parameters cannot be written or read (not permitted in the operation mode, invalid PNU, ...)
- Drag error, drive has exceeded the tolerance after Motion Complete and similar minor control errors.

4.1.2 Malfunction type 1

In the event of a fault, the service that was requested cannot be provided. The drive switches from its current status to the “Fault” status.

Behaviour in the event of type 1 malfunctions

– The end stage is not switched off

– The current positioning job is interrupted

– The speed is reduced on the emergency ramp

– The sequence control switches to the Fault status No new positioning job can be carried out

– The SCON.B3 (FAULT) bit is allocated

– The ”Fault” status can be exited by switching off, with a positive edge at input CCON.B3 (RESET) or by resetting/ setting DIN5 (controller release)

– Holding brake is activated when the drive is stopped

Causes of type 1 malfunctions

– Software end positions are damaged

– Motion Complete timeout

– Drag fault monitoring

4.1.3 Malfunction type 2

In the event of a fault, the service that was requested cannot be provided. The drive switches from its current status to the “Fault” status.

Behaviour in the event of type 2 malfunctions

- The end stage is switched off
- The current positioning job is interrupted
- The drive runs down
- No new positioning job can be carried out
- The SCON.B3 (FAULT) bit is allocated
- The “Fault” status can only be exited by switching off or an edge at input CCON.B3 (RESET).

Causes of type 2 malfunctions

- Load voltage is missing (e.g. if emergency off has been implemented)
- Hardware fault
  - Measuring system fault
  - Bus fault
  - SD card fault
- Impermissible operation mode switch

4.2 Diagnostic memory

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

Structure of the diagnostic memory in CMMS	
Parameters [1)]	201
Format	uint16
Meaning	**Fault number**
Subindex 1	Latest / current diagnostic message
Subindex 2	2nd saved diagnostic message
Subindex 3	3rd saved diagnostic message
Subindex 4	4th saved diagnostic message
[1)] see section 5.4.4	

Table 4/29: Structure of diagnostic memory in CMMS

Structure of the diagnostic memory in CMMP			
Parameters [1)]	200	201	202
Format	uint8	uint16	uint32
Meaning	**Diagnostic event**	**Fault number**	**Time**
Subindex 1	Latest / current diagnostic message		
Subindex 2	2nd saved diagnostic message		
... [2)]	...		
Subindex 32	32nd saved diagnostic message		
[1)] see section 5.4.4			

Table 4/30: Structure of diagnostic memory in CMMP

CMMP only:
Configuration of the diagnostic memory with PNU 204 (see section 5.4.4)

SI	Description	Specification	Min.	Max.
1	= 1: Record incoming and outgoing faults = 2: Record only incoming faults	1	1	2
2	= 1: Resolution time stamp 10 ms = 2: Resolution time stamp 1 ms	1	1	2
3	Deleting the diagnostic memory. – Writing with value = 1 deletes the diagnostic memory – Read will always be answered with value = 0.	0	0	1
4	Number of valid entries in the diagnostic memory.	0	0	16

Table 4/31: Configuration of the diagnostic memory (CMMP only)

4.3 Warning memory (CMMP only)

Optionally, a separate warning memory can be implemented if using controllers.
The warning memory contains the codes of the last warnings that occurred. It functions in the same way as the diagnostic memory.

Structure of the warning memory			
Parameters [1)]	210	211	212
Format	uint8	uint16	uint32
Meaning	**Warning event**	**Warning number**	**Time**
Subindex 1	Most recent/current warning		
Subindex 2	Second saved warning message		
...	...		
Subindex 16	Last warning message		
[1)] (see section 5.4.4)			

Table 4/32: Structure of the warning memory

Configuration of the warning memory with PNU 214 (see section 5.4.4)				
SI	**Description**	**Specification**	**Min.**	**Max.**
1	= 1: Record incoming and outgoing faults = 2: Record only incoming faults	1	1	2
2	= 1: Resolution time stamp 10 ms = 2: Resolution time stamp 1 ms	1	1	2
3	Deleting the warning memory. – Writing with value = 1 deletes the warning memory – Read will always be answered with value = 0.	0	0	1
4	Number of valid entries in the warning memory.	0	0	16

Table 4/33: Configuration of the warning memory

4.4 Fault numbers

The controller's error messages are displayed as fault numbers, see description of type P.BE-CMM...-HW-... of the controller in question.

4.5 Diagnosis using FHPP status bytes

The controller supports the following diagnosis options using FHPP status bytes (see section 2.4):

- SCON.B2 (WARN) – Warning
- SCON.B3 (FAULT) – Fault
- SPOS.B5 (DEV) – Drag fault
- SPOS.B6 (STILL) – Standstill monitoring

Chapter 5

Contents

5.1 FHPP general parameter structure

A controller contains a parameter set with the following structure for **each axis**.

Assembly	Indices	Description
Device data	100..199	Device identification and device-specific settings, version numbers, etc.
Diagnostic memory	200...299	Memory for diagnostic events: fault numbers, fault time, incoming/outgoing event.
Processing data	300...399	Current nominal and actual values, local I/Os, status data etc.
Record list	400...499	A record contains all the nominal value parameters required for a positioning procedure.
Project data	500...599	Basic project settings. Maximum speed and acceleration, offset project zero point etc. –> Parameters are the basis for the record list.
Factor group	600...699	Parameters for scaling the nominal and actual values to the user's required dimensions or order of magnitude.
Axis data Electric drives 1	1000...1099	All axis-specific parameters for electric drives: gear factor, feed constant, reference parameter ...

Table 5/1: Parameter structure

Parameter classes	Attribute / use
Simple variable	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: record list setpoint position (PNU 404). The elements in the array are addressed using the subindex.
Struct/Record	Summary of various simple variables with different limits etc. Not used for CMM... .

Table 5/2: Parameter classes

5.2 Access protection

5.2.1 Access via PLC and FCT.

The user can prevent the drive from being operated simultaneously by PLC and FCT. The CCON.B5 bit (FCT access blocked) and the SCON.B5 bit (FCT control sovereignty) are used for this.

Preventing FCT operation: CCON.B5 (LOCK)

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

The PLC is programmed not to issue this release until the user carries out the relevant action. This generally exits automatic operation. This means that the PLC programmer can ensure that the PLC always knows when it has control over the drive.

Important: the block is active if the CCON.B5 has a 1 signal. It therefore does not need to be set compulsorily. A user who does not need this type of locking can always set the bit to 0.

Control sovereignty acknowledgment for FCT: SCON.B5 (LOCK)

This bit informs the PLC that the drive is controlled by the FCT and that the PLC no longer has any control over the drive. This bit does not need to be evaluated. The PLC can react by transferring to stop or manual operation.

5.3 Overview of FHPP parameters

The following overview (Table 5/3) shows the FHPP's parameters.

The parameters are described in sections 5.4.2 to 5.4.16.

Name	Controller	FHPP			
		PNU	Subind.	Class	Type
Device data					
Device data – standard parameter (see section 5.4.2)					
Manufacturer hardware version	All	100	–	Var	uint16
Manufacturer firmware version	All	101	–	Var	uint16
Version FHPP	All	102	–	Var	uint16
Project identification	All	113	1	Var	uint32
Controller serial number	CMMP	114	1	Var	uint32
	CMMS		1...12	Array	uint8
Device data – extended parameters (see section 5.4.3)					
Manufacturer device name	All	120	1...30	Array	uint8
User device name	All	121	1...32	Array	uint8
Manufacturer name	All	122	1...30	Array	uint8
HTTP address of manufacturer	All	123	1...30	Array	uint8
Festo order number	All	124	1...30	Array	uint8
Device control	All	125	–	Var	uint8

Name	Con-troller	FHPP			
		PNU	Subind.	Class	Type
Diagnosis (see section 5.4.4).					
Diagnostic event	CMMP	200	1...32	Array	uint8
Fault number	CMMP	201	1...32	Array	uint16
	CMMS	201	1...4	Array	uint16
Time stamp	CMMP	202	1...32	Array	uint32
Diagnostic memory parameter	CMMP	204	1, 2, 4	Array	uint8
Device Warnings	CMMP	210	1...32	Array	uint8
Warning number	CMMP	211	1...32	Array	uint16
Time stamp	CMMP	212	1...32	Array	uint32
Warning memory parameter	CMMP	214	1...4	Array	uint8
Processing data (see section 5.4.5)					
Position values	All	300	1...3	Array	int32
Torque values	All	301	1...3	Array	int32
Local digital inputs	All	303	1, 2	Array	uint8
Local digital outputs	All	304	1	Array	uint8
Maintenance parameter	All	305	3	Array	uint32
Speed values	All	310	1...3	Array	int32

Name	Con-troller	FHPP			
		PNU	Subind.	Class	Type
Record list (see section 5.4.6)					
Record status	All	400	1...3	Record	uint8
Record control byte 1	CMMP	401	0...250	Array	uint8
	CMMS	401	0...63	Array	uint8
Record control byte 2	CMMP	402	0...250	Array	uint8
	CMMS	402	0...63	Array	uint8
Traversing record setpoint value	CMMP	404	0...250	Array	int32
	CMMS	404	0...63	Array	int32
Record preselection value	CMMP	405	0...250	Array	int32
	CMMS	405	0...63	Array	int32
Traversing record speed	CMMP	406	0...250	Array	uint32
	CMMS	406	0...63	Array	uint32
Traversing record acceleration	CMMP	407	0...250	Array	uint32
	CMMS	407	0...63	Array	uint32
Traversing record deceleration	CMMP	408	0...250	Array	uint32
	CMMS	408	0...63	Array	uint32
Traversing record torque ramp	CMMP	412	0...250	Array	uint32
Traversing record jerk-free filter time	CMMP	413	0...250	Array	uint32
	CMMS	413	0...63	Array	uint32
Traversing record group	CMMS	414	0...63	Array	uint8

Name	Con-troller	FHPP			
		PNU	Subind.	Class	Type
Project data					
Project data – General project data (see section 5.4.7)					
Offset project zero point	All	500	–	Var	int32
Software end positions	All	501	1, 2	Array	int32
Max. permitted speed	All	502	–	Var	uint32
Max. permitted acceleration	All	503	–	Var	uint32
Maximum jerk-free filter time	All	505	–	Var	uint32
Project data – Teach (see section 5.4.8)					
Teach target	All	520	–	Var	uint8
Project data – Jog mode (see section 5.4.9)					
Jog mode velocity slow – phase 1	All	530	–	Var	int32
Jog mode velocity fast – phase 2	All	531	–	Var	int32
Jog mode acceleration	All	532	–	Var	uint32
Type mode deceleration	All	533	–	Var	uint32
Jog mode time phase 1	All	534	–	Var	uint32

Name	Con-troller	FHPP			
		PNU	Subind.	Class	Type
Project data – Direct mode position control (see section 5.4.10)					
Direct mode base speed	All	540	–	Var	int32
Direct mode acceleration	All	541	–	Var	uint32
Direct mode deceleration	All	542	–	Var	uint32
Direct mode jerk-free filter time	All	546	–	Var	uint32
Project data – Direct mode speed adjustment (see section 5.4.11)					
Direct mode acceleration ramp	All	560	–	Var	uint32
Direct mode speed target window	CMMP	561	–	Var	uint16
Direct mode speed target window damping time	CMMP	562	–	Var	uint16
Direct mode standstill target window	CMMP	563	–	Var	uint16
Direct mode standstill target window damping time	CMMP	564	–	Var	uint16
Direct mode torque limiting	CMMP	565	–	Var	uint32

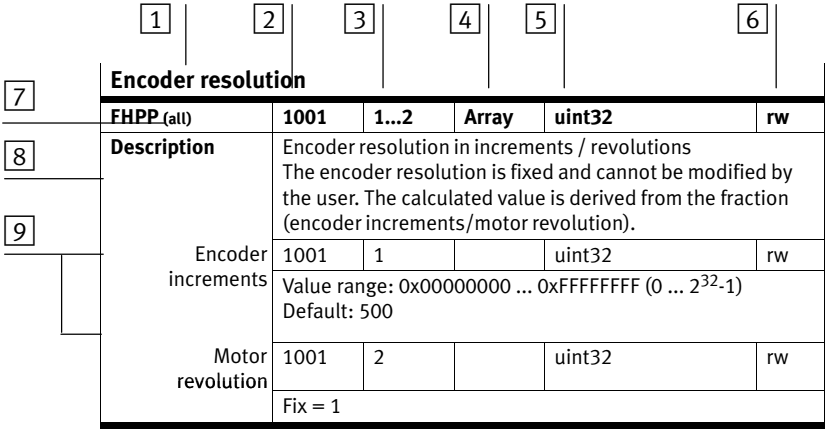
Name	Con-troller	FHPP			
		PNU	Subind.	Class	Type
Axis parameters electrical drives 1 – mechanical parameters					
Axis parameters electric drives 1 – mechanical parameters (see section 5.4.12)					
Reversal of direction	All	1000	–	Var	uint8
Encoder resolution	All	1001	1, 2	Array	uint32
Gear ratio	All	1002	1, 2	Array	uint32
Feed constant	All	1003	1, 2	Array	uint32
Position factor	All	1004	1, 2	Array	uint32
Axis parameter	All	1005	2, 3	Array	int32
Speed factor	All	1006	1, 2	Array	uint32
Acceleration factor	All	1007	1, 2	Array	uint32
Axis parameters electric drives 1 – homing parameters (see section 5.4.13)					
Offset axis zero point	All	1010	–	Var	int32
Homing method	All	1011	–	Var	int8
Homing speeds	All	1012	1, 2	Array	uint32
Homing acceleration	All	1013	1	Var	uint32
Homing (reference travel) required	All	1014	–	Var	uint8
Homing max. torque	CMMP	1015	–	Var	uint8

Name	Con-troller	FHPP			
		PNU	Subind.	Class	Type
Axis parameters electric drives 1 – controller parameters (see section 5.4.14)					
HALT option code	All	1020	–	Var	uint16
Tolerance position window	All	1022	–	Var	uint32
Adjustment time for position	All	1023	–	Var	uint16
Controller's parameters	All	1024	18...22, 32	Array	uint16
Motor data	All	1025	1, 3	Array	uint32/ uint16
Drive data	CMMP	1026	1...4, 7	Array	uint32
	CMMS	1026	1, 3, 4, 7	Array	uint32
Axis parameters electric drives 1 – electronic rating plate (see section 5.4.15)					
Maximum current	All	1034	–	Var	uint16
Rated motor current	All	1035	–	Var	uint32
Rated motor torque	All	1036	–	Var	uint32
Torque constant	All	1037	–	Var	uint32
Axis parameters electric drives 1 – Standstill monitoring (see section 5.4.16)					
Nominal position	All	1040	–	Var	int32
Current position	All	1041	–	Var	int32
Standstill position window	All	1042	–	Var	uint32
Standstill monitoring time	All	1043	–	Var	uint16

Table 5/3: Overview of FHPP parameters

5.4 Descriptions of FHPP parameters

5.4.1 Representing the parameter entries



Encoder resolution					
FHPP (all)	**1001**	**1...2**	**Array**	**uint32**	**rw**
Description	Encoder resolution in increments / revolutions The encoder resolution is fixed and cannot be modified by the user. The calculated value is derived from the fraction (encoder increments/motor revolution).				
Encoder increments	1001	1		uint32	rw
	Value range: 0x00000000 ... 0xFFFFFFFF (0 ... 2^{32}-1) Default: 500				
Motor revolution	1001	2		uint32	rw
	Fix = 1				

1 Name of the parameter in English (German in brackets)

2 PNU (parameter number)

3 Subindices of parameter, if present (–: no subindex, simple variable)
In the event of access via PROFIBUS, subindices with DPV1 are each -1 (e.g. 0...1 instead of 1...2)

4 Element class

5 Element variable type.

6 Read/write permission: ro = read only, rw = read and write

7 Identifier for general or limited validity (e.g. CMMS only)

8 Description of the parameter

9 Name and description of subindices, if present
(entry relating to FHPP, if available)

Fig. 5/12: Representing the parameter entries

5.4.2 Device data – Standard parameters

Manufacturer hardware version (manufacturer hardware version)					
FHPP (all)	**100**	**–**	**Var**	**uint16**	**ro**
Description	Coding of the hardware version, specification in BCD: xxyy (xx = main version, yy = secondary version)				

Manufacturer firmware version (manufacturer firmware version)					
FHPP (all)	**101**	**–**	**Var**	**uint16**	**ro**
Description	Coding of the firmware version, specification in BCD: xxyy (xx = main version, yy = secondary version)				

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

Project identifier					
FHPP (all)	**113**	**1**	**Var**	**uint32**	**rw**
Description	32 bit value used with the FCT PlugIn to identify the project. The identifier is specified by the user. Value range: 0x00000001 ... 0xFFFFFFFF (1 ... 2^{32}-1)				

Controller serial number					
FHPP (CMMP)	**114**	**1**	**Var**	**uint32**	**ro**
Description	Serial number for uniquely identifying the controller.				

Controller serial number					
FHPP (CMMS)	**114**	**1...12**	**Aray**	**uint8**	**ro**
Description	Serial number for uniquely identifying the controller.				

5.4.3 Device data – extended parameters

Manufacturer device name					
FHPP (all)	**120**	**1...30**	**Array**	**uint8**	**ro**
Description	Designation of the drive or controller (ASCII, 7 bit). Unused characters are filled with zero (00h='\0'). Example: "CMMS-ST"				

User device name					
FHPP (all)	**121**	**1...32**	**Array**	**uint8**	**rw**
Description	User's designation of the controller (ASCII, 7 bit). Unused characters are filled with zero (00h='\0'). Default: "motor001" (max. 32 characters)				

Drive manufacturer (manufacturer name)					
FHPP (all)	**122**	**1...30**	**Array**	**uint8**	**ro**
Description	Name of the drive's manufacturer (ASCII, 7-bit) Fixed: "Festo AG & Co. KG"				

HTTP drive catalog address (HTTP address of manufacturer)					
FHPP (all)	**123**	**1...30**	**Array**	**uint8**	**ro**
Description	Manufacturer's Internet address (ASCII, 7-bit) Fixed: "www.festo.com"				

Festo order number					
FHPP (all)	**124**	**1...30**	**Array**	**uint8**	**ro**
Description	Festo order number / order code (ASCII, 7-bit).				

Device control					
FHPP (all)	**125**	**–**	**Var**	**uint8**	**rw**
Description	Specifies which interface currently has sovereignty over the drive, in other words, which interface can be used to enable and start or stop (control) the drive. The following interfaces are taken into account: – Fieldbus: (CANopen, PROFIBUS, DeviceNet, ...) – DIN: digital I/O interface (e.g. multi-pin, I/O interface) – Parameterization interface RS 232/RS 485 (FCT) The last two interfaces are treated as equals. For all controllers of the type CMM..., the output stage enable (DIN4) and controller enable (DIN5) also have to be set in addition to the interface in question (AND operation). Value Significance SCON.B5 (LOCK) 0x00 (0) Software has control sovereignty (+ DIN) 1 0x01 (1) Fieldbus has control sovereignty (+ DIN) 0 0x02 (2) Only DIN has control sovereignty 1 Default after power on: 0x01 (1) Fieldbus has control sovereignty (+ DIN)				

5.4.4 Diagnosis

For a description of how the diagnostic memory functions, see section 4.2.

Diagnostic event					
FHPP (CMMP)	**200**	**1...32**	**Array**	**uint8**	**ro**
Description	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 0x02 (2) Reserved (Outgoing fault) 0x03 (3) Reserved 0x04 (4) Reserved (Overrun time stamp)				
Event 1	200	1		uint8	ro
	Type of latest / current diagnostic message				
Event 2	200	2		uint8	ro
	Type of second saved diagnostic message				
Event ...	200	...		uint8	ro
	...				

Fault number					
FHPP (CMMP)	**201**	**1...32**	**Array**	**uint16**	**ro**
FHPP (CMMS)	**201**	**1...4**	**Array**	**uint16**	**ro**
Description	Fault number saved in the diagnostic memory, serves for identifying the fault. See section 4.4 for fault numbers.				
Event 1	201	1		uint16	ro
	Latest / current diagnostic message				
Event 2	201	2		uint16	ro
	2nd saved diagnostic message				
Event ...	201	...		uint16	ro
	...				

Time stamp					
FHPP (CMMP)	**202**	**1...32**	**Array**	**uint32**	**ro**
Description	Time point of the diagnostic event in seconds since being switched on. In the event of an overrun the time stamp jumps from 0xFFFFFFFF to 0.				
Event 1	202	1		uint32	ro
	Time of the latest / current diagnostic message				
Event 2	202	2		uint32	ro
	Time of the second saved diagnostic message				
Event ...	202	...		uint32	ro
	...				

Diagnostic memory parameter					
FHPP (CMMP)	**204**	**1, 2, 4**	**Array**	**uint8**	**rw/ro**
Description	Configuration of the diagnostic memory.				
Fault type	204	1		uint8	rw
	Incoming and outgoing faults 0x01 (1): Reserved (Record incoming and outgoing faults) 0x02 (2): Record only incoming faults				
Resolution	204	2		uint8	rw
	Reserved (Resolution time stamp)				
Number of entries	204	4		uint8	ro
	Read out the number of valid entries in the diagnostic memory. Writing is not permitted. Value range: 0x00 ... 0x0F (0 ... 15)				

Device warnings

FHPP (CMMP)	**210**	**1...32**	**Array**	**uint8**	**ro**
Description	Type of warning or diagnostic information saved in the warning memory. Indication of whether an incoming or outgoing warning was saved. Value Type of diagnostic event 0x00 (0) No warning (or warning message deleted) 0x01 (1) Incoming warning 0x02 (2) Reserved (Outgoing warning) 0x03 (3) Power Down (with valid time stamp) 0x04 (4) Reserved (Overrun time stamp)				
Event 1	210	1		uint8	ro
	Type of latest / current warning message				
Event 2	210	2		uint8	ro
	Type of second saved warning message				
Event ...	210	...		uint8	ro
	...				

Warning number

FHPP (CMMP)	**211**	**1...32**	**Array**	**uint16**	**ro**
Description	Warning number saved in the warning memory, serves for identifying the warning. See section 4.2 for warning numbers.				
Event 1	211	1		uint16	ro
	Most recent/current warning message				
Event 2	211	2		uint16	ro
	2nd saved warning message				
Event ...	211	...		uint16	ro
	...				

Time stamp					
FHPP (CMMP)	**212**	**1...32**	**Array**	**uint32**	**ro**
Description	Time point of the warning event in seconds since being switched on. In the event of an overrun the time stamp jumps from 0xFFFFFFFF to 0.				
Event 1	212	1		uint32	ro
	Time of the latest / current warning message				
Event 2	212	2		uint32	ro
	Time of the second saved warning message				
Event ...	212	...		uint32	ro
	...				

Warning memory parameter					
FHPP (CMMP)	**214**	**1...4**	**Array**	**uint8**	**rw/ro**
Description	Configuration of the warning memory.				
Warning type	214	1		uint8	rw
	Incoming and outgoing warnings. 0x01 (1): Reserved (Record incoming and outgoing warnings) 0x02 (2): Record only incoming warnings				
Resolution	214	2		uint8	rw
	Reserved (Resolution time stamp)				
Number of entries	214	4		uint8	ro
	Read number of valid entries in the warning memory. Writing is not permitted. Value range: 0x00 ... 0x0F (0 ... 15)				

5.4.5 Process data

Position values					
FHPP (all)	**300**	**1...3**	**Array**	**int32**	**ro**
Description	Current values of the position controller in the positioning unit (see PNU 1004).				
Actual position	300	1		int32	ro
	Current actual position of the controller.				
Nominal position	300	2		int32	ro
	Current nominal position of the controller.				
Control deviation	300	3		int32	ro
	Current deviation.				

Torque values					
FHPP (all)	**301**	**1...3**	**Array**	**int32**	**ro**
Description	Current values of the torque controller in the positioning unit (see PNU 1004).				
Actual force	301	1		int32	ro
	Current actual position of the controller.				
Nominal force	301	2		int32	ro
	Current nominal position of the controller.				
Control deviation	301	3		int32	ro
	Current deviation.				

Local digital inputs

FHPP (all)	**303**	**1, 2**	**Array**	**uint8**	**ro**
Description	The controller's local digital inputs.				
Input DIN 0...7	303	1		uint8	ro
	Digital inputs: standard DIN (DIN 0 ... DIN 7)				
Input DIN 8...13	303	2		uint8	ro
	Digital inputs: standard DIN (DIN 8 ... DIN 13)				

PNU 303 allocation

Subindex 1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	DIN 7: right-hand limit switch	DIN 6: left-hand limit switch	DIN 5: controller enable	DIN 4: output stage enable	DIN 3	DIN 2	DIN 1	DIN 0
Subindex 2	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	reserved (= 0)		DIN A13	DIN A12	DIN 11	DIN 10	DIN 9	DIN 8

Local digital outputs

FHPP (all)	**304**	**1**	**Array**	**uint8**	**rw**
Description	The controller's local digital outputs.				
Outputs DOUT 0...3	303	1		uint8	rw
	Digital outputs: standard DOUT (DOUT 0 ... DOUT 3)				

PNU 304 allocation

Subindex 1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	reserved (= 0)		DOUT: READY LED	DOUT: CAN LED	DOUT 3	DOUT 2	DOUT 1	DOUT 0: controller ready for operation

Maintenance parameter					
FHPP (all)	**305**	**3**	**Array**	**uint32**	**ro**
Description	Information about the controller's or the driver's running performance.				
Operating hours	305	3		uint32	ro
	Operating hour counter in s.				

Velocity values					
FHPP (all)	**310**	**1...3**	**Array**	**int32**	**ro**
Description	Current values of the speed regulator.				
Actual speed	310	1		int32	ro
	Current actual position of the controller.				
Nominal speed	310	2		int32	ro
	Current nominal value of the controller.				
Actual deviation	310	3		int32	ro
	Speed deviation.				

5.4.6 Record list

PNU										
400	401	402	404	405	406	407	408	412	413	414
Record status (record no.)	RCB1	RCB2	Setpoint value	Preselected value	Speed	Acc. Movement to	Acc. Brake	Torque ramp	Jerk-free filter time	Record group
uint8	uint8	uint8	int32	int32	uint32	uint32	uint32	uint32	int32	uint8
0	Homing									
1	...	...	...	...	...	...	...	...	...	...
2	...	...	...	...	...	...	...	...	...	...
... [1]	...	...	...	...	...	...	...	...	...	...
[1] Number of records: For CMMP...: 1...250 (0 = homing). For CMMS...: 1...63 (0 = homing).										

Table 5/4: Structure of FHPP record list

With FHPP, record selection for reading and writing is made via the subindex of the PNUs 401 ... 414. The active record for positioning or teaching is selected with PNU 400.

Controller/ drive	PNU									
	401	402	404	405	406	407	408	412	413	414
CMMP-AS	x	x	x	x	x	x	x	x	x	–
CMMS-ST	x	x	x	x	x	x	x	–	x	x
CMMS-AS	x	x	x	x	x	x	x	–	x	x

Table 5/5: Supported elements in the record list

With CMMS the "dynamic" parameters of a record are determined together via the record group (PNU 414).
When these parameters (PNU406, 407, 408, 413) are written for a record, the profile parameters assigned to the record will be overwritten. The modified parameters are therefore effective for all records which are assigned to this profile. The assignment of the records to a record group can only be parametrized with the FPC.

Record status					
FHPP (all)	**400**	**1...3**	**Record**	**uint8**	**rw/ro**
Description	Record status.				
Nominal record number	400	1		uint8	rw
	Nominal record number. The value can be changed using FHPP. In Record Select mode, the nominal record number is always copied from the master's output data with a rising edge at START.				
Current record number	400	2		uint8	ro
	Current record number.				
Record status byte	400	3		uint8	ro
	The record status byte (RSB) contains a feedback code that is transferred in the input data. When a positioning job starts, the RSB is reset. See Table 5/6 for the allocation of the record control byte. Note: this byte is not the same as SDIR, there is only a feedback signal for dynamic states and not absolute/relative, for example. This makes is possible to provide feedback about record chaining, for example.				

RSB allocation **Bit**	**Meaning**
Bit 0 **RC1**	= 0: A step criterion was not configured/achieved. = 1: The step criterion was achieved.
Bit 1 **RCC**	Valid, if MC present. = 0: Record chaining cancelled. At least one step criterion was not achieved. = 1: Record chain was processed to the end of the chain.
Bit 2	Reserved.
Bit 3	Reserved.
Bit 4 **VLIM**	= 1: Speed reached. = 0: Speed not reached (for force control only).
Bit 5 **XLIM**	= 1: Stroke limit value reached. = 0: Stroke limit not reached (for force control only).
Bit 6	Reserved.
Bit 7	Reserved.

Table 5/6: Allocation of PNU 400/3 (RSB)

Record control byte 1					
FHPP (CMMP)	**401**	**0...250**	**Array**	**uint8**	**rw**
FHPP (CMMS)	**401**	**0...63**	**Array**	**uint8**	**rw**
Description	The record control byte 1 (RCB1) controls the most important settings for the positioning task in Record Select mode. The record control byte is bit-orientated: Bit 0: setpoint value absolute/relative Bit 1 ... 7: reserved (!= 0) Values: 0x00 (0): Nominal value is absolute (default) 0x01 (1): Nominal value is relative to the last nominal value/switch further value				
Traversing record 0	401	0		uint8	rw
	Record control byte traversing record 0 (homing).				
Traversing record 1	401	1		uint8	rw
	Record control byte traversing record 1				
Traversing record ...	401	...		uint8	rw
	Record control byte traversing record ...				

Record control byte 2					
FHPP (CMMP)	**402**	**0...250**	**Array**	**uint8**	**rw**
FHPP (CMMS)	**402**	**0...63**	**Array**	**uint8**	**rw**
Description	Record control byte 2 (RCB2) controls conditional record chaining. If a condition was defined, it is possible to prohibit automatic continuing by setting the B7 bit. This function is intended for debugging and not for normal control purposes. Bit Significance Bit 0...6 Numerical value 0...128: step criterion as a list, see section 3.6.3 Table 3/25. Bit 7 = 0: Record chaining (bit 0...6) is not disabled (default) = 1: Record chaining is locked out				
Record 0	402	0		uint8	rw
	Record control byte 2 traversing record 0 (homing).				
Record 1	402	1		uint8	rw
	Record control byte 2 traversing record 1.				
Record ...	402	...		uint8	rw
	Record control byte 2 traversing record ...				

Record setpoint value					
FHPP (CMMP)	**404**	**0...250**	**Array**	**int32**	**rw**
FHPP (CMMS)	**404**	**0...63**	**Array**	**int32**	**rw**
Description	Target position of the traversing record table. Position nominal value as per PNU 401 / RCB1 absolute or relative in positioning unit (see PNU 1004). Value range: 0x80000000 ... 0x7FFFFFFF (-2^{31}...+(2^{31}-1)) Default: 0x00000000 (0)				
Traversing record 0	404	0		int32	rw
	Nominal position value traversing record 0 (homing).				
Traversing record 1	404	1		int32	rw
	Nominal position value traversing record 1.				
Traversing record...	404	...		int32	rw
	Nominal position value traversing record ...				

regulation	**the step size**	**Default**	**Minimum**	**Maximum**
Position [1)]	1/100 mm 1/1000 inch 1/100 °	0 (= 0. mm) 0 (= 0.0 inch) 0 (= 0.0 °)	-1,000,000 (= -10.0 m) -400,000 (= -400 inch) -36,000 (= -360.0 °)	1,000,000 (= 10.0 m) 400,000 (= 400 inch) 36,000 (= 360.0 °)
[1)] Examples of positioning unit, see PNU 1004				

Table 5/7: Preselected values for positioning units in PNU 404

Record preselection value					
FHPP (CMMP)	**405**	**0...250**	**Array**	**int32**	**rw**
FHPP (CMMS)	**405**	**0...63**	**Array**	**int32**	**rw**
Description	Preselected value for conditional record chaining, see Table 5/8. Values as per step criterion from PNU 402 (RCB2), see section 3.6.3 Table 3/25. Value range: 0x80000000 ... 0x7FFFFFFF (-2^{31}...+(2^{31}-1)) Default: 0x00000000				
Traversing record 0	405	0		int32	rw
	Preselected record value 0 (homing).				
Traversing record 1	404	1		int32	rw
	Preselected record value 1.				
Traversing record ...	405	...		int32	rw
	Preselected record value ...				

regulation	**the step size**	**Default**	**Minimum**	**Maximum**
Position [1]	1/100 mm 1/1000 inch 1/100 °	0 (= 0. mm) 0 (= 0.0 inch) 0 (= 0.0 °)	-1,000,000 (= -10,0 m) -400,000 (= -400 inch) -36,000 (= -360.0 °)	1,000,000 (= 10.0 m) 400,000 (= -400 inch) 36,000 (= -360.0 °)
Force	1/10 N	0 (= 0.0 N)	-100,000 (= -10.0 kN)	100,000 (= 10.0 kN)
Time	1 msec	0 (= 0 msec)	0 (= 0 msec)	100,000 (= 100 sec)
[1] Examples of positioning unit, see PNU 1004				

Table 5/8: Preselected values in PNU 405 depend on the type of control or the condition

Record speed (traversing record speed)					
FHPP (CMMP)	**406**	**0...250**	**Array**	**uint32**	**rw**
Description	Nominal speed in unit of speed (see PNU 1006). Value range: 0x00000000 ... 0x7FFFFFFF (0...+(2^{31}-1)) Default: 0x00000000 (0)				
Traversing record 0	406	0		uint32	rw
	Nominal speed value traversing record 0 (homing).				
Traversing record 1	406	1		uint32	rw
	Nominal speed value traversing record 1				
Traversing record ...	406	...		uint32	rw
	Nominal speed value traversing record ...				

Record speed (traversing record speed)					
FHPP (CMMS)	**406**	**0...63**	**Array**	**uint32**	**rw**
Description	Nominal speed value of the record group as per PNU 414 in unit of speed (see PNU 1006). **When written, the value is effective for all records in the record group.** Value range: 0x00000000 ... 0x7FFFFFFF (0...+(2^{31}-1)) Default: 0x00000000 (0)				
Traversing record 0	406	0		uint32	rw
	Nominal speed value of the record group of traversing record 0 (homing).				
Traversing record 1	406	1		uint32	rw
	Nominal speed value of the record group of traversing record 1.				
Traversing record ...	406	...		uint32	rw
	Nominal speed value of the record group of traversing record ...				

Record acceleration					
FHPP (CMMP)	**407**	**0...250**	**Array**	**uint32**	**rw**
Description	Nominal acceleration value for start up in the unit of acceleration (see PNU 1007). Value range: 0x00000000 ... 0x7FFFFFFF (0...+(2^{31}-1)) Default: 0x00000000 (0)				
Traversing record 0	407	0		uint32	rw
	Nominal acceleration value traversing record 0 (reference travel).				
Traversing record 1	407	1		uint32	wr
	Nominal acceleration value traversing record 1				
Traversing record ...	407	...		uint32	rw
	Nominal acceleration value traversing record ...				

Record acceleration					
FHPP (CMMS)	**407**	**0...63**	**Array**	**uint32**	**rw**
Description	Nominal acceleration value for the record group as per PNU 414 for the start up in the unit of acceleration (see PNU 1007). **When written, the value is effective for all records in the record group.** Value range: 0x00000000 ... 0x7FFFFFFF (0...+(2^{31}-1)) Default: 0x00000000 (0)				
Traversing record 0	407	0		uint32	rw
	Nominal acceleration value of the record group of traversing record 0 (homing).				
Traversing record 1	407	1		uint32	wr
	Nominal acceleration value of the record group of traversing record 1.				
Traversing record ...	407	...		uint32	rw
	Nominal acceleration value of the record group of traversing record ...				

Record deceleration					
FHPP (CMMP)	**408**	**0...250**	**Array**	**uint32**	
Description	Nominal deceleration value for braking (deceleration) in unit of acceleration (see PNU 1007). Value range: 0x00000000 ... 0x7FFFFFFF (0...+(2^{31}-1)) Default: 0x00000000 (0)				
Traversing record 0	408	0		uint32	
	Nominal deceleration value traversing record 0 (reference travel).				
Traversing record 1	408	1		uint32	
	Nominal deceleration value traversing record 1				
Traversing record ...	408	...		uint32	
	Nominal deceleration value traversing record ...				

Record deceleration					
FHPP (CMMS)	**408**	**0...63**	**Array**	**uint32**	
Description	Nominal deceleration value for the record group as per PNU 414 for braking in the unit of acceleration (see PNU 1007). **When written, the value is effective for all records in the record group.** Value range: 0x00000000 ... 0x7FFFFFFF (0...+(2^{31}-1)) Default: 0x00000000 (0)				
Traversing record 0	408	0		uint32	
	Nominal deceleration value of the record group of traversing record 0 (homing).				
Traversing record 1	408	1		uint32	
	Nominal deceleration value of the record group of traversing record 1.				
Traversing record ...	408	...		uint32	
	Nominal deceleration value of the record group of traversing record ...				

Record torque ramp					
FHPP (CMMP)	**412**	**0...250**	**Array**	**uint32**	**rw**
Description	Torque ramp in unit of torque/s (mN/s). Value range: 0x00000000 ... 0x7FFFFFFF (0...+(2^{31}-1)) Default: 0x00000000 (0)				
Traversing record 0	412	0		uint32	rw
	Torque ramp traversing record 0 (homing).				
Traversing record 1	412	1		uint32	wr
	Torque ramp traversing record 1.				
Traversing record ...	412	...		uint32	rw
	Torque ramp traversing record ...				

Record jerkfree filter time					
FHPP (CMMP)	**413**	**0...250**	**Array**	**uint32**	**rw**
Description	Jerk-free filter time in ms. Specifies the filter time constant for the output filter that is used to smooth the linear movement profiles. Completely jerk-free movement is achieved if the filter time is the same as the acceleration time. Value range: 0x00000000 ... 0x7FFFFFFF (0...+(2^{31}-1)) Default: 0x00000000 (0)				
Traversing record 0	413	0		uint32	rw
	Jerk-free filter time traversing record 0 (homing).				
Traversing record 1	413	1		uint32	rw
	Nominal deceleration value traversing record 1				
Traversing record ...	413	...		uint32	rw
	Jerk-free filter time traversing record ...				

Record jerkfree filter time					
FHPP (CMMS)	**413**	**0...63**	**Array**	**uint32**	**rw**
Description	Jerk-free filter time of the record group as per PNU 414 in ms. Specifies the filter time constant for the output filter that is used to smooth the linear movement profiles. Completely jerk-free movement is achieved if the filter time is the same as the acceleration time. **When written, the value is effective for all records in the record group.** Value range: 0x00000000 ... 0x7FFFFFFF (0...+(2^{31}-1)) Default: 0x00000000 (0)				
Traversing record 0	413	0		uint32	rw
	Jerk-free filter time of the record group of traversing record 0 (homing).				
Traversing record 1	413	1		uint32	rw
	Jerk-free filter time of the record group of traversing record 1.				
Traversing record ...	413	...		uint32	rw
	Jerk-free filter time of the record group of traversing record ...				

Record group					
FHPP (CMMS)	**414**	**0...63**	**Array**	**uint8**	**rw**
Description	Specifies affiliation to a group. The traversing records are assigned to the groups (0...7). The following parameters are determined in a group: – Positioning speed (PNU 406) – Acceleration (PNU 407) – Deceleration (PNU 408) – Jerk-free filter time (PNU 413) – Maximum positioning time [1)] – Start delay [1)] – Final speed [1)] – Start during an ongoing positioning job [1)] Value range: 0x00 ... 0x07 (0 ... 7) Default: 0x00 (0)				
Traversing record 0	414	0		uint8	rw
	Record group traversing record 0 (homing).				
Traversing record 1	414	1		uint8	rw
	Record group traversing record 1.				
Traversing record ...	414	...		uint8	rw
	Record group traversing record ...				
[1)] Cannot be parameterized with FHPP, access via FCT only					

5.4.7 Project data – General project data

Project zero point (offset project zero point)					
FHPP (all)	**500**	**–**	**Var**	**int32**	**rw**
Description	Offset of axis zero point to project zero point in positioning unit. (see PNU 1004). Reference point for position values in the application (see PNU 404). Value range: 0x80000000 ... 07FFFFFF (-2^{31}...$+(2^{31}-1)$) Default: 0x00000000 (0)				

Software end positions					
FHPP (all)	**501**	**1, 2**	**Array**	**int32**	**rw**
Description	Software end positions in positioning unit (see PNU 1004) A nominal value 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 Value range: 0x80000000 ... 0x7FFFFFFF (-2^{31}...$+(2^{31}-1)$)				
Lower limit value	501	1		int32	rw
	Lower software end position Default: 0xFFFF8000 (-32768)				
Lower limit value	501	2		int32	rw
	Upper software end position Default: 0x00008000 (8000)				

Max. speed (max. permitted speed)					
FHPP (all)	**502**	**1**	**Var**	**uint32**	**rw**
Description	Max. permissible speed in unit of speed (see PNU 1006). This value limits the speed in all operation modes except torque mode. Value range: 0x00000000 ... 0xFFFFFFFF (0...$+2^{32}$) Default: 0x00000000				

Max. acceleration (max. permitted acceleration)					
FHPP (all)	**503**	**1**	**Var**	**uint32**	**rw**
Description	Max. permissible acceleration in unit of acceleration (see PNU 1007). Value range: 0x00000000 ... 0xFFFFFFFF (0...+2^{32}) Default: 0x00000000 (0)				

Max. jerkfree filter time (max. jerk-free filter time)					
FHPP (all)	**505**	**1**	**Var**	**uint32**	**rw**
Description	Max. permissible jerk-free filter time in ms. Value range: 0x00000000 ... 0x00000032 (0 ... 50) Default: 0x00000032 (50)				

5.4.8 Project data – Teaching

Teach target					
FHPP (all)	**520**	**–**	**Var**	**uint8**	**rw**
Description	The parameter defined is the one which is written with the actual position with the next Teach command (see section 3.5). Values: 0x01 (1): Nominal position in traversing record (default) – with Record select: traversing record as per FHPP control bytes – with Direct mode: traversing record as per PNU 400/1 0x02 (2): axis zero point (PNU 1010) 0x03 (3): project zero point (PNU 500) 0x04 (4): lower software end position (PNU 501/01) 0x05 (5): upper software end position (PNU 501/02)				

5.4.9 Project data – Jog mode

Jog mode velocity slow – phase 1					
FHPP (all)	**530**	**–**	**Var**	**int32**	**rw**
Description	Maximum speed for phase 1 in unit of velocity (see PNU 1006). Value range: 0x00000000 ... 0x7FFFFFFF (0...+(2^{31}-1)) Default: 0x00000000 (0)				

Jog mode velocity fast – phase 2					
FHPP (all)	**531**	**–**	**Var**	**int32**	**rw**
Description	Maximum speed for phase 2 in unit of velocity (see PNU 1006). Value range: 0x00000000 ... 0x7FFFFFFF (0...+(2^{31}-1)) Default: 0x00000000 (0)				

Jog mode acceleration					
FHPP (all)	**532**	**–**	**Var**	**uint32**	**rw**
Description	Acceleration during jogging in unit of acceleration (see PNU 1007). Value range: 0x00000000 ... 0x7FFFFFFF (0...+(2^{31}-1)) Default: 0x00000000 (0)				

Jog mode deceleration					
FHPP (all)	**533**	**–**	**Var**	**uint32**	**rw**
Description	Deceleration during jogging in unit of acceleration (see PNU 1007). Value range: 0x00000000 ... 0x7FFFFFFF (0...+(2^{31}-1)) Default: 0x00000000 (0)				

Jog mode time phase 1					
FHPP (all)	**534**	**–**	**Var**	**uint32**	**rw**
Description	Time duration of phase 1 (T1) in ms Value range: 0x00000000 ... 0xFFFFFFFF (0...2^{32}) Default: 0x000003E8 (1000)				

5.4.10 Project data – Direct mode position control

Direct mode base velocity					
FHPP (all)	**540**	**–**	**Var**	**int32**	**rw**
Description	Base velocity during direct mode position control in unit of velocity (see PNU 1006). Value range: 0x00000000 ... 0x7FFFFFFF (0...+(2^{31}-1)) Default: 0x00000000 (0)				

Direct mode acceleration					
FHPP (all)	**541**	**–**	**Var**	**uint32**	**rw**
Description	Acceleration during direct mode position control in unit of acceleration (see PNU 1007). Value range: 0x00000000 ... 0x7FFFFFFF (0...+(2^{31}-1)) Default: 0x00000000 (0)				

Direct mode deceleration					
FHPP (all)	**542**	**–**	**Var**	**uint32**	**rw**
Description	Deceleration during direct mode position control in unit of acceleration (see PNU 1007). Value range: 0x00000000 ... 0x7FFFFFFF (0...+(2^{31}-1)) Default: 0x00000000 (0)				

Direct mode jerkfree filter time					
FHPP (all)	**546**	**–**	**Var**	**uint32**	**rw**
Description	Jerk-free filter time during direct mode position control in ms. Value range: 0x00000000 ... 0x000000032 (0...50) Default: 0x00000000 (0)				

5.4.11 Project data – Direct mode speed adjustment

Direct mode base velocity ramp					
FHPP (all)	**560**	**–**	**Var**	**uint32**	**rw**
Description	Base acceleration value (speed ramp) during direct mode speed adjustment in unit of acceleration (see PNU 1007). Value range: 0x00000000 ... 0x7FFFFFFF (0...+(2^{31}-1)) Default: 0x00000000 (0)				

Direct mode velocity window (direct mode speed target window)					
FHPP (CMMP)	**561**	**–**	**Var**	**uint16**	**rw**
Description	Speed target window during direct mode speed adjustment in unit of speed (see PNU 1006). Value range: 0x0000 ... 0x7FFF (0...32767) Default: 0x00000000 (0)				

Direct mode velocity window time (direct mode speed target window damping time)					
FHPP (CMMP)	**562**	**–**	**Var**	**uint16**	**rw**
Description	Damping time for speed target window during direct mode speed adjustment in ms. Value range: 0x0000 ... 0x7FFF (0...32767) Default: 0x00000000 (0)				

Direct mode velocity threshold (direct mode standstill target window)					
FHPP (CMMP)	**563**	**–**	**Var**	**uint16**	**rw**
Description	Standstill target window during direct mode speed adjustment in unit of speed (see PNU 1006). Value range: 0x0000 ... 0x7FFF (0...32767) Default: 0x0000 (0)				

Direct mode velocity threshold time (direct mode standstill target window damping time)					
FHPP (CMMP)	**564**	**–**	**Var**	**uint16**	**rw**
Description	Damping time for standstill target window during direct mode speed adjustment in ms. Value range: 0x0000 ... 0x7FFF (0...32767) Default: 0x0000 (0)				

Direct mode torque limit					
FHPP (CMMP)	**565**	**–**	**Var**	**uint32**	**rw**
Description	Torque limiting during direct mode speed adjustment in unit of torque (mN). Value range: 0x00000000 ... 0x7FFFFFFF (0...+(2^{31}-1)) Default: 0x00000000 (0)				

5.4.12 Axis parameters electrical drives 1 – mechanical parameters

Polarity (reversal of direction)					
FHPP (all)	**1000**	**1**	**Var**	**uint8**	**rw**
Description	Direction of the position values. Values: Position value (vector) 0x00 (0) : normal 0x80 (128): inverted (multiplied by -1) Default: 0x00 (0)				

Encoder resolution					
FHPP (all)	**1001**	**1, 2**	**Array**	**uint32**	**rw**
Description	Encoder resolution in encoder increments / motor revolutions. Specified internal conversion factor. The calculated value is derived from the fraction "encoder-increments/motor revolution."				
Encoder increments	1001	1		uint32	rw
	Fixed: 0x00010000 (65536)				
Motor revolutions	1001	2		uint32	rw
	Fixed: 0x00000001 (1)				

Gear ratio					
FHPP (all)	**1002**	**1, 2**	**Array**	**uint32**	**rw**
Description	Ratio of motor revolutions to gearing spindle revolutions (drive output revolutions), see appendix A.1. Gear transmission = motor revolutions / spindle revolutions				
Motor revolutions	1002	1		uint32	rw
	Gear ratio – enumerator. Value range: 0x00000000 ... 0x7FFFFFFF (0...+(2^{31}-1)) Default: 0x00000001 (1)				
Shaft revolutions	1002	2		uint32	rw
	Gear ratio – denominator. Value range: 0x00000000 ... 0x7FFFFFFF (0...+(2^{31}-1)) Default: 0x00000001 (1)				

Feed constant					
FHPP (all)	**1003**	**1, 2**	**Array**	**uint32**	**rw**
Description	The feed constant specifies the pitch of the drive's shaft per revolution, see appendix A.1. Feed constant = feed / shaft revolution				
Feed	1003	1		uint32	rw
	Feed constant – numerator. Value range: 0x00000000 ... 0x7FFFFFFF (0...+(2^{31}-1)) Default: 0x00000001 (1)				
Shaft revolutions	1003	2		uint32	rw
	Feed constant – denominator. Value range: 0x00000000 ... 0x7FFFFFFF (0...+(2^{31}-1)) Default: 0x00000001 (1)				

Position factor					
FHPP (all)	**1004**	**1, 2**	**Array**	**uint32**	**rw**
Description	Conversion ratio for all position units (converting the user units into internal controller units). See appendix A.1 for the calculation. $\text{Positionsfaktor} = \frac{\text{Encoder-Auflösung} * \text{Getriebeübersetzung}}{\text{Vorschubkonstante}}$				
Counter	1004	1		uint32	rw
	Position factor – numerator. Value range: 0x00000000 ... 0xFFFFFFFF (0...+(2^{32}-1)) Default: 0x00010000 (65536)				
Denominator	1004	2		uint32	rw
	Position factor – denominator. Value range: 0x00000000 ... 0xFFFFFFFF (0...+(2^{32}-1)) Default: 0x00000001 (1)				

WARNING formula texts: DELETE AFTER INCLUDING IN THE FORMULA!
Position factor
Encoder resolution
Gear ratio
Feed constant

Axis parameter					
FHPP (all)	**1005**	**2, 3**	**Array**	**int32**	**rw**
Description	Specify and read out axis parameters.				
Gear numerator	1005	2		int32	rw
	Gear ratio – axis gear numerator Value range: 0x00000000 ... 0x7FFFFFFF (0 ... +(2^{31}-1)) Default: 0x00000000 (0)				
Gear denominator	1005	3		int32	rw
	Gear ratio – axis gear denominator Value range: 0x00000000 ... 0x7FFFFFFF (0 ... +(2^{31}-1)) Default: 0x00000000 (0)				

Velocity factor					
FHPP (all)	**1006**	**1, 2**	**Array**	**uint32**	**rw**
Description	Conversion ratio for all units of velocity (converting the user units into internal controller units). See appendix A.1 for the calculation. $\text{Speedfactor} = \frac{\text{Encoder resolution} * \text{Time factor\_v}}{\text{Feed constant}}$				
Counter	1006	1		uint32	rw
	Velocity factor – numerator. Value range: 0x00000000 ... 0xFFFFFFFF (0...+(2^{32}-1)) Default: 0x00001000 (4096)				
Denominator	1006	2		uint32	rw
	Velocity factor – denominator. Value range: 0x00000000 ... 0xFFFFFFFF (0...+(2^{32}-1)) Default: 0x00000001 (1)				

Acceleration factor					
FHPP (all)	**1007**	**1, 2**	**Array**	**uint32**	**rw**
Description	Conversion ratio for all units of acceleration (converting the user units into internal controller units). See appendix A.1 for the calculation. $\text{Accelerationfactor} = \frac{\text{Encoder resolution} * \text{Time factor\_a}}{\text{Feed constant}}$				
Counter	1007	1		uint32	rw
	Acceleration factor – numerator. Value range: 0x00000000 ... 0xFFFFFFFF (0...+(2^{32}-1)) Default: 0x00000100 (256)				
Denominator	1007	2		uint32	rw
	Acceleration factor – denominator. Value range: 0x00000000 ... 0xFFFFFFFF (0...+(2^{32}-1)) Default: 0x00000001 (1)				

5.4.13 Axis data electrical drives 1 - Homing parameters

Offset axis zero point					
FHPP (all)	**1010**	**1**	**Var**	**int32**	**rw**
Description	Axis zero point offset in positioning unit (see PNU 1004). The offset for the axis zero point (home offset) defines the axis zero point ‹AZ› as a dimensional reference point relative to the physical reference point ‹REF›. The axis zero point is the basis point for the project zero point ‹PZ› and for the software end positions. All positioning operations refer to the project zero point (PNU 500). The axis zero point is calculated from: AZ = REF + offset axis zero point Value range: 0x80000000 ... 0x7FFFFFFF (-2^{31}...+(2^{31}-1)) Default: 0x00000000 (0)				

Homing method (reference travel method)					
FHPP (all)	**1011**	**1**	**Var**	**int8**	**rw**
Description	Defines the method which the drive uses to carry out the homing. (See section 3.3, Table 3/16). Default: 0x11 (17) – negative limit switch				

Homing speeds					
FHPP (all)	**1012**	**1, 2**	**Array**	**uint32**	**rw**
Description	Speeds during the homing in unit of speed (see PNU 1006).				
Search speed	1012	1		uint32	rw
	Speed when searching for the homing point REF or a stop or switch. Value range: 0x00000000 ... 0x7FFFFFFF (0...+(2^{31}-1)) Default: 0x00000000 (0)				
Travel speed	1012	2		uint32	rw
	Speed of travel to the axis zero point AZ. Value range: 0x00000000 ... 0x7FFFFFFF (0...+(2^{31}-1)) Default: 0x00000000 (0)				

Homing acceleration					
FHPP (all)	**1013**	**1**	**Var**	**uint32**	**rw**
Description	Acceleration during the homing in unit of acceleration (see PNU 1007) Value range: 0x00000000 ... 0x7FFFFFFF (0...+(2^{31}-1)) Default: 0x00000000 (0)				

Homing required (reference travel necessary)					
FHPP (all)	**1014**	**–**	**Var**	**uint8**	**rw**
Description	Defines whether or not homing must be carried out after switching on in order to carry out positioning tasks. **Homing must always be carried out after Power On for all drives.** **Exception: drives with the multi-turn absolute displacement encoder only need one homing run after being installed.** Values: 0x00 (0): Reserved 0x01 (1): Homing run must be carried out Fixed: 0x01 (1)				

Homing max. torque (reference travel max. torque)					
FHPP (CMMP)	**1015**	**–**	**Var**	**uint8**	**rw**
Description	Max. torque during the reference run. Specified as a multiple of the rated torque in % (see PNU 1036). The maximum permitted torque (via current limiting) during reference travel. If this value is reached, the drive identifies the stop (REF) and travels to the axis zero point. Value range: 0x00 ... 0xFF (0 ... 255) Default: 0x64 (100)				

5.4.14 Axis parameters electrical drives 1 – Controller parameters

HALT option code					
FHPP (all)	**1020**	**–**	**Var**	**uint16**	**rw**
Description	Reaction to a HALT command (falling edge at SPOS.B0). Values: 0x00 (0): Reserved (switch off motor – coil without current, brake not actuated) 0x01 (1): Brake with stop ramp 0x02 (2): Reserved (brake with emergency stop ramp) Default: 0x01 (1)				

Position window (tolerance window position)					
FHPP (all)	**1022**	**–**	**Var**	**uint32**	**rw**
Description	Tolerance window in positioning unit (see PNU 1004). Amount by which the current position may deviate from the target position, in order that it may still be regarded as being within the target window. The width of the window is twice the value transferred, with the target position in the centre of the window. Value range: 0x00000000 ... 0xFFFFFFFF (0 ... +(2^{32}-1)) Default: 0x00000000 (0)				

Position window time (adjustment time position)					
FHPP (all)	**1023**	**–**	**Var**	**uint16**	**rw**
Description	Adjustment time in milliseconds. If the actual position has been in the target position window this amount of time, the bit “Target reached” is set in the status word. Value range: 0x0000 ... 0xFFFF (0 ... 65535) Specification: 0x0064 (100)				

Control parameter set					
FHPP (all)	**1024**	**18...22, 32**	**Array**	**uint16**	**rw**
Description	Control parameters as well as parameters for ”quasi-absolute position registering.”				
Gain position	1024	18		uint16	rw
	Gain position controller Value range: 0x0000 ... 0xFFFF (0 ... 65535) Default: 0x0034 (52)				
Gain speed	1024	19		uint16	rw
	Gain speed controller Value range: 0x0000 ... 0xFFFF (0 ... 65535) Default: 0x0080 (128)				
Speed time constant	1024	20		uint16	rw
	Time constant for the speed controller. Value range: 0x0000 ... 0xFFFF (0 ... 65535) Default: 0x1F40 (8000)				
Gain current	1024	21		uint16	rw
	Gain current controller Value range: 0x0000 ... 0xFFFF (0 ... 65535) Default: 0x0100 (256)				
Current time constant	1024	22		uint16	rw
	Time constant for the current regulator. Value range: 0x0000 ... 0xFFFF (0 ... 65535) Default: 0x07D0 (2000)				
Save position	1024	32		uint16	rw
	Save the current position when switching off (“quasi-absolute” positioning). See also PNU 1014. Values: 0x00F0 (240) = Current position will **not** be saved at power-off (default) 0x000F (15) = reserved (current position is saved at power-off)				

Motor data

FHPP (all)	1025	1, 3	Array	uint32/uint16	r/rw
Description	Motor-specific data.				
Serial number	1025	1		uint32	r
	Festo serial number and motor's serial number.				
Time max. current	1025	3		uint16	rw
	I^2t time in ms. When the I^2t time elapses, the current is limited automatically to the motor rated current in order to protect the motor (motor rated current, PNU 1035). Value range: 0x0000 ... 0xFFFF (0 ... 65535) Default: 0x03E8 (1000)				

Drive data

FHPP (CMMP)	1026	1...4, 7	Array	uint32	r/rw
FHPP (CMMS)	1026	1, 3, 4, 7	Array	uint32	r/rw
Description	General motor data				
Temp. end stage	1026	1		uint32	r
	Temperature of the end stage in °C.				
Max. temp. end stage	1026	2		uint32	r
	CMMP only: Maximum temperature of the end stage in °C.				
Rated motor current	1026	3		uint32	rw
	Rated motor current in mA, identical to PNU 1035. Value range: 0x00000000 ... 0x00007FFF (0 ... 32767) Default: 0x000005D4 (1492)				
Max. motor current	1026	4		uint32	rw
	Maximum motor current, identical to PNU 1034. Value range: 0x00000000 ... 0x00007FFF (0 ... 32767) Default: 0x0000068E (1678)				
Controller serial number	1026	7		uint32	r
	Controller's internal serial number.				

5.4.15 Axis parameters for electric drives 1 – Electronic rating plate

Maximum current					
FHPP (all)	**1034**	**–**	**Var**	**uint16**	**rw**
Description	Maximum motor current in 1/1000 of the specified rated current (PNU 1035), identical to PNU 1026. Note: Please note that the current limitation also limits the maximum possible speed and that (higher) nominal speeds may not therefore be achieved. Value range: 0x00000000 ... 0x00007FFF (0 ... 32767) Default: 0x0000068E (1678)				

Rated motor current					
FHPP (all)	**1035**	**–**	**Var**	**uint32**	**rw**
Description	The motor's rated current in mA, identical to PNU 1026/3. Value range: 0x00000000 ... 0x00007FFF (0 ... 32767) Default: 0x000005D4 (1492)				

Rated motor torque					
FHPP (all)	**1036**	**–**	**Var**	**uint32**	**rw**
Description	The motor's rated torque in 0,001 Nm. Value range: 0x00000000 ... 0x00007FFF (0 ... 32767) Default: 0x000005D4 (1492)				

Torque constant					
FHPP (all)	**1037**	**–**	**Var**	**uint32**	**rw**
Description	Ratio between the current and torque in the motor used in mNm/A. Value range: 0x00000000 ... 0xFFFFFFFF (0 ... +(2^{31}-1)) Default: 0x003E8000 (4096000)				

5.4.16 Axis parameters electric drives 1 – Standstill monitoring

Position demand value (nominal position)					
FHPP (all)	**1040**	**–**	**Var**	**int32**	**ro**
Description	Nominal target position of the last positioning task in positioning unit (see PNU 1004) Value range: 0x80000000 ... 0x7FFFFFFF (-2^{31} ... $+(2^{31}-1)$)				

Position actual value (current position)					
FHPP (all)	**1041**	**–**	**Var**	**int32**	**ro**
Description	Current position of the drive in the positioning unit (see PNU 1004). Value range: 0x80000000 ... 0x7FFFFFFF (-2^{31} ... $+(2^{31}-1)$)				

Following error window (standstill position window)					
FHPP (all)	**1042**	**–**	**Var**	**uint32**	**rw**
Description	Standstill position window in positioning unit (see PNU 1004). Amount by which the drive may move after MC, until the standstill monitoring responds. Value range: 0x00000000 ... 0xFFFFFFFF (0 ... $+(2^{31}-1)$) Default: 0x00000000 (0)				

Following error timeout (standstill monitoring time)					
FHPP (all)	**1043**	**–**	**Var**	**uint16**	**rw**
Description	Standstill monitoring time in ms. Time during which the drive must be outside the standstill position window before the standstill monitoring responds. Value range: 0x0000 ... 0xFFFF (0 ... 65535) Default: 0x0064 (100)				

Parameterization

Chapter 6

Contents

6.1 Parameterization with FHPP

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

The parameter channel serves for transmitting parameters.
The parameter channel comprises the following:

Components	Description
Parameter identifier (PKE)	Component of the parameter channel component which contains the Task and Response identifiers (AK) and the parameter number (PNU). The parameter number serves for identifying or addressing the individual parameter. The Task or Response identifier (AK) describes the task or the reply in the form of an identifier number.
Subindex (IND)	addresses an element of an array parameter (sub-parameter number)
Parameter value (PWE)	Value of the parameter. If a task of the parameter processing cannot be carried out, a fault number will be shown instead of the value in the reply telegram. The fault number describes the cause of the fault.

Tab. 6/1: Components of the parameter channel (PKW)

The parameter channel consists of 8 octets. The structure of the parameter channel as a factor of the size or type of the parameter value is shown in the following table:

FPC

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Output data	0	IND	ParID (PKE)		Value (PWE)			
Input data	0	IND	ParID (PKE)		Value (PWE)			

IND Subindex – for addressing an array element
ParID (PKE) Parameter Identifier – consists of ReqID or ResID and PNU
Value (PWE) Parameter value, parameter value:
– with double word: bytes 5...8
– with word: bytes 7, 8
– with byte: Byte 8

Tab. 6/2: Structure of parameter channel

Parameter identifier (PKE)

The parameter identifier contains the Task or Response identifier (AK) and the parameter number (PNU).

PKE

	Octet 1 (byte 3)								Octet 2 (byte 4)							
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Order	ReqID (AK)				res.	Parameter number (PNU)										
Reply	ResID (AK)				res.	Parameter number (PNU)										

ReqID (AK)	Request Identifier – task identifier (read, write, ...)
ResID (AK)	Response Identifier – response identifier (transfer value, fault, ...)
Value (PNU)	Parameter Number – identifies and addresses the relevant parameter (see section 6.1). The Task or Response identifier indicates the type of task or reply (see section 6.1.2).

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

6.1.2 Order identifiers, response identifiers and fault numbers

Theorder identifiers are shown in the following table:

ReqID	Description	Response identifier	
		positive	negative
0	No task	0	–
6	Request parameter (array)	5	7
8	Modify parameter value (array, double word)	5	7
13	Request lower limit value	5	7
14	Request upper limit value	5	7

Tab. 6/4: Order identifiers

If the task cannot be carried out, Response identifier 7 as well as the appropriate fault number will be transmitted (negative reply).

The following table shows the response identifiers:

ResID	Description
0	No reply
5	Parameter value transferred (array, double word)
7	Task cannot be carried out (with fault number) 1)
1) Fault numbers see following table	

Tab. 6/5: Response identifiers

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

No.	Check	Fault numbers		Description
1	PNU defined?	0	0x00	Non-permitted PNU. The parameter does not exist.
2	If Array: IND defined?	3	0x03	Faulty subindex
3	ReqID permitted?	101	0x65	Festo: ReqID is not supported
4	Access rights (read, write)	1	0x01	Parameter value cannot be modified (read only)
		102	0x66	Parameter is WriteOnly (with passwords)
5	If change: operating status	17	0x11	Task cannot be carried out because of operating status
6	If change: operating sovereignty	11	0x0B	No control sovereignty
7	If change: password	12	0x0C	Password incorrect
8	If change: value permitted	2	0x02	Lower or upper limit value exceeded

Tab. 6/6: Sequence of fault checks and error numbers

6.1.3 Rules for task reply processing

Rules	Description
1	If the master transmits the identifier for “No order,” the controller responds with the reply identifier for “No reply.”
2	A task or reply telegram always refers to a single parameter.
3	The master must continue to send an order until it has received the appropriate reply from the controller.
4	The master recognizes 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 controller supplies the reply until the master sends a new order.
6	a) A write task, even with cyclic repetition of the same order, will only be carried out once by the controller. b) Between two consecutive tasks with the same Task identifier (AK), parameter number (PNU) and subindex (IND), the Task identifier 0 (no task) must be sent and the Response identifier 0 (no reply) must be awaited. This is to ensure that an ”old” reply is not interpreted as a ”new” reply.

Tab. 6/7: Rules for task reply processing

Sequence of parameter processing

Caution
Observe the following when modifying parameters:
An FHPP control signal, which is to refer to a modified parameter, may only follow when the Response identifier "Parameter value transferred" is received for the relevant parameter and if applicable for the index.

If, e.g. a position value in a position register is to be modified and if a movement is then to be made to this position, the positioning command must not be given until the controller has completed and confirmed the modification of the position register.

Caution
In order to be sure that an "old" reply cannot be interpreted as a "new" reply, the Task identifier 0 (no task) must be sent and the Response identifier 0 (no reply) must be
awaited between two consecutive tasks with the same Task identifier (AK), parameter number (PNU) and subindex (IND).

Evaluating faults

In the case of tasks which cannot be carried out, the slave replies as follows:

– Output of response identifier = 7

– Output a fault number in bytes 7 and 8 of the parameter channel (FPC).

Example of parameterizing via FPC

The following tables show an example of parameterising a positioning task in the position set table via (FPC – Festo Parameter Channel).

Step 1

Output status of the 8 bytes of FPC data:

	Byte 1	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**	**Byte 7**	**Byte 8**
	Reserved	Subindex	ReqID/ResID + PNU		Parameter value			
Output data	0x00	0x00	0x**0**0	0x00	0x00	0x00	0x00	0x00
Input data	0x00	0x00	0x**0**0	0x00	0x00	0x00	0x00	0x00

Step 2

Write record number 1 with absolute positioning:
PNU 401, subindex 2 – Modify parameter value, array, byte:
ReqID 12 (0xC) with value 0x00.

	Byte 1	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**	**Byte 7**	**Byte 8**
	Reserved	Subindex	ReqID/ResID + PNU		Parameter value			
Output data	0x00	0x**02**	0x**C1**	0x**91**	Unused	Unused	Unused	0x**00**
Input data	0x00	0x**02**	0x**C1**	0x**91**	0x00	0x00	0x00	0x**00**

Step 3

After receiving the input data with ResID 0xC send output data with ReqID = 0x0 and wait for input data with ResID = 0x0:

	Byte 1	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**	**Byte 7**	**Byte 8**
	Reserved	Subindex	ReqID/ResID + PNU		Parameter value			
Output data	0x00	0x02	0x**0**1	0x91	Unused	Unused	Unused	0x00
Input data	0x00	0x02	0x**0**1	0x91	0x00	0x00	0x00	0x00

Step 4

Write record number 1 with target position 0x1234 (decimal 4660 increments):
PNU 404, subindex 2 – Modify parameter value, array, double word: ReqID 8 (0x8) with value 0x00001234.

	Byte 1	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**	**Byte 7**	**Byte 8**
	Reserved	Subindex	ReqID/ResID + PNU		Parameter value			
Output data	0x00	0x02	0x**81**	0x**94**	0x**00**	0x**00**	0x**12**	0x**34**
Input data	0x00	0x02	0x**81**	0x**94**	0x**00**	0x**00**	0x**12**	0x**34**

Step 5

After receiving the input data with ResID 0x8 send output data with ReqID = 0x0 and wait for input data with ResID = 0x0:

	Byte 1	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**	**Byte 7**	**Byte 8**
	Reserved	Subindex	ReqID/ResID + PNU		Parameter value			
Output data	0x00	0x02	0x**0**1	0x94	0x00	0x00	0x12	0x34
Input data	0x00	0x02	0x**0**1	0x94	0x00	0x00	0x12	0x34

Step 6

Write record number 1 with speed 0x7743 (decimal 30531 increments/s):
PNU 406, subindex 2 – Modify parameter value, array, double word: ReqID 8 (0x8) with value 0x00007743.

	Byte 1	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**	**Byte 7**	**Byte 8**
	Reserved	Subindex	ReqID/ResID + PNU		Parameter value			
Output data	0x00	0x02	0x**81**	0x**96**	0x**00**	0x**00**	0x**77**	0x**43**
Input data	0x00	0x02	0x**81**	0x**96**	0x**00**	0x**00**	0x**77**	0x**43**

Step 7

After receiving the input data with ResID 0x8 send output data with ReqID = 0x0 and wait for input data with ResID = 0x0:

	Byte 1	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**	**Byte 7**	**Byte 8**
	Reserved	Subindex	ReqID/ResID + PNU		Parameter value			
Output data	0x00	0x02	0x**0**1	0x94	0x00	0x00	0x77	0x43
Input data	0x00	0x02	0x**0**1	0x94	0x00	0x00	0x77	0x43

Appendix A

Contents

A.1 Conversion factors (factor group)

A.1.1 Overview

Motor controllers are used in a wide variety of applications: as direct drives, with downstream gear units, for linear drives etc.

In order to enable simple parameterization for all applications, the motor controller can be parametrized with the parameters in the “factor group” (PNU 1001 to 1007, see section 5.4.12) in such a way that factors such as the rotational speed can be directly specified at or read from the output side in the units required (e.g, in the case of a linear axis, the position values in millimetres and speeds in millimetres per second).

The motor controller then uses the factor group to calculate the entries in its internal units of measurement. One conversion factor is available for each of the physical parameters: position, speed and acceleration. These conversion factors adjust the user’s units of measurement to the application in question.
Fig. A/1 clarifies the function of the factor group:

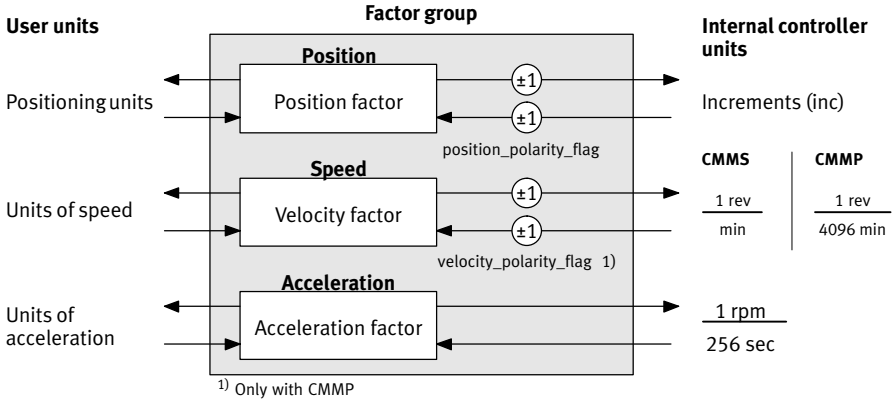


Fig. A/1: Factor group

All parameters are always saved in the motor controller in its internal units of measurement and are only converted (using the factor group) when the parameters are written or read out.

For this reason, the factor group should be set first during parameterization and should not be changed again during parameterization.

The factor group is set to the following units by default:

Parameter	Description	Unit	Explanation
Length	Positioning units	Increments	65536 increments per revolution
Speed	Units of speed	min^{-1}	Revolutions per minute
Acceleration	Units of acceleration	$(min^{-1})/s$	Rotational speed increase per second

Table A/1: Factor group default settings

A.1.2 Objects in the factor group

Table A/2 shows the parameters in the factor group.

Name	PNU	Object	Type	Access
Reversal of direction	1000	Var	uint8	rw
Position factor	1004	Array	uint32	rw
Speed factor	1006	Array	uint32	rw
Acceleration factor	1007	Array	uint32	rw

Table A/2: Overview of the factor group

Table A/3 shows the parameters involved in the conversion.

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

Table A/3: Overview of parameters involved

A.1.3 Calculating the positioning units

The **position factor** (PNU 1004, see section 5.4.12) is used to convert all the length values from the user's **positioning units** into the internal unit **increments** (65536 increments are equivalent to one motor revolution). The position factor consists of numerators and denominators.

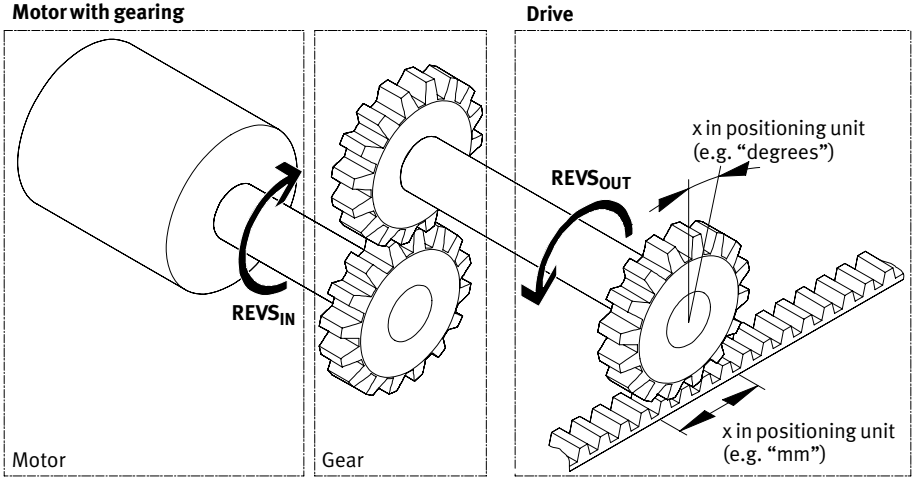


Fig. A/2: Calculating the positioning units

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

Gear ratio (gear ratio)	Gear ratio between revolutions at the input side (revs$_{IN}$) and revolutions at the output side (revs$_{OUT}$).
Feed constant (feed constant)	Ratio between movement in positioning units at the drive and revolutions at the gear unit's output (revs$_{OUT}$). (e.g. 1˚rev ≙ 63.15 mm or 1˚rev ≙ 360° degrees)

The position factor is calculated using the following formula:

$$\text{Position factor} = \frac{\text{Gear ratio} * \text{Encoder resolution}}{\text{Feed constant}}$$

The position factor must be written to the motor controller separated into numerators and denominators. This can make it necessary to bring the fraction up to whole integers by expanding it accordingly.

Example

First, the desired unit (column 1) and the desired number of decimal places (dp) have to be specified, along with the application's gear ratio and its feed constant (if applicable). The feed constant is then displayed in the desired positioning units (column 2).

In this way, all the values can be entered into the formula and the fraction can be calculated:

Sequence of calculating the position factor

Positioning units [1]	**Feed constant** [2]	**Gear ratio** [3]	**Formula** [4]	**Result shortened**
degrees, 1 dp 1/10 degree $(°/_{10})$	**1** $REVS_{OFF}$ = **3600** $\frac{°}{10}$	**1/1**	$\frac{1REVS}{1REVS} * 65536 \frac{Ink}{REVS} = \frac{65536\ Ink}{\frac{3600\ °}{10}}$	num : 4096 div : 225

Fig. A/3: Sequence of calculating the position factor

Examples of calculating the position factor

Positioning units [1]	Feed constant [2]	Gear ratio [3]	Formula [4]	Result shortened
increments, 0 dp inc.	**1** $REVS_{OFF}$ = **65536** Ink	**1/1**	$\frac{\frac{1REVS}{1REVS} * 65536\frac{Ink}{REVS}}{\frac{65536\ Ink}{1\ REVS}} = \frac{1\ Ink}{1\ Ink}$	num : 1 div : 1
degrees, 1 dp 1/10 degree ($^{\circ}/_{10}$)	**1** $REVS_{OFF}$ = **3600** $\frac{^{\circ}}{10}$	**1/1**	$\frac{\frac{1REVS}{1REVS} * 65536\frac{Ink}{REVS}}{\frac{3600\frac{^{\circ}}{10}}{1\ REVS}} = \frac{\mathbf{65536}\ Ink}{\mathbf{3600}\ \frac{^{\circ}}{10}}$	num : 4096 div : 225
revs, 2 dp 1/100 revs ($^{revs}/_{100}$)	**1** $REVS_{OFF}$ = **100** $\frac{REVS}{100}$	**1/1**	$\frac{\frac{1REVS}{1U} * 65536InkoverREVS}{\frac{100\frac{REVS}{100}}{1\ REVS}} = \frac{\mathbf{65536}\ Ink}{\mathbf{100}\ \frac{REVS}{100}}$	num : 16384 div : 25
		2/3	$\frac{\frac{2REVS}{3REVS} * 65536\frac{Ink}{REVS}}{\frac{100\frac{REVS}{100}}{1\ REVS}} = \frac{\mathbf{131072}\ Ink}{\mathbf{300}\ \frac{REVS}{100}}$	num : 32768 div : 75
mm, 1 dp 1/10 mm ($^{mm}/_{10}$)	**1** $REVS_{OFF}$ = **631,5** $\frac{mm}{10}$	**4/5**	$\frac{\frac{4REVS}{5REVS} * 65536\frac{Ink}{REVS}}{\frac{631,5\frac{mm}{10}}{1\ REVS}} = \frac{\mathbf{2621440}\ Ink}{\mathbf{31575}\ \frac{mm}{10}}$	num: 524288 div: 6315

1) Desired unit at the output
2) Positioning units per revolution ($revs_{OUT}$).
Drive's feed constant (PNU 1003) * 10^{-dp} (decimal places taken into consideration)
3) $revs_{IN}$ per $revs_{OUT}$
4) Insert values into formula.

Table A/4: Examples of calculating the position factor

A.1.4 Calculating the units of speed

The **speed factor** (PNU 1006, see section 5.4.12) is used to convert all the speed values from the user's **units of speed** into the internal units:

- with CMMS-ST: Revolutions per minute
- with: CMMP-AS: Revolutions per 4096 minutes

The speed factor consists of numerators and denominators.

The speed factor is calculated in two parts: a conversion factor from internal units of length into the user's positioning units and a conversion factor from internal units of time into user-defined units of time (e.g. from seconds to minutes). The first part is equivalent to calculating the position factor; an additional factor is required to calculate the second part:

Time factor_v	Ratio between the internal unit of time and the user-defined unit of time (e.g. with CMMS-ST: ˚1˚min˚= $^{1}/_{4096}$ 4096 min).
Gear ratio (gear ratio)	Gear ratio between revolutions at the input side ($revs_{IN}$) and revolutions at the output side ($revs_{OUT}$).
Feed constant (feed constant)	Ratio between movement in positioning units at the drive and revolutions at the gear unit's output ($revs_{OUT}$). (e. g. 1˚rev ≙ 63.15 mm or 1˚rev ≙ 360° degrees)

The speed factor is calculated using the following formula:

$$\text{Speed factor} = \frac{\text{Gear ratio} * \text{Time factor\_v}}{\text{Feed constant}}$$

Like the position factor, the speed factor also has to be written to the motor controller separated into numerators and denominators. This can make it necessary to bring the fraction up to whole integers by expanding it accordingly.

Example

First, the desired unit (column 1) and the desired number of decimal places (dp) have to be specified, along with the application's gear ratio and its feed constant (if applicable). The feed constant is then displayed in the desired positioning units (column 2).

Then, the desired unit of time is converted into the motor controller's unit of time (column 3).

In this way, all the values can be entered into the formula and the fraction can be calculated:

Sequence of calculating the speed factor

Units of speed [1]	Feed constant [2]	Time constant [3]	Gear [4]	Formula [5]	Result shortened
mm/s, 1 dp 1/10 mm/s (mm/10 s)	63,15 mm/U ⇒ 1 U_OFF = 631,5 mm/10	1 1/s = 60 1/min = 60 * 4096 1/(4096 min)	4/5	4 U/5 U * (60 * 4096 1/(4096 min)) / (1 1/s) / (631,5 mm/10 / 1 U) = 1966080 U/(4096 min) / 6315 mm/10s	num: 131072 div: 421

Fig. A/4: Sequence of calculating the speed factor (CMMP-AS)

Calculation example Speed factor with CMMS-ST

Speed units [1]	Feed constant [2]	Time constant [3]	Gear [4]	Formula [5]	Result shortened
rpm, 0 dp 1/100 revs/ min	**1** $revs_{OUT}$ = **65536** Inc	**1** $\frac{1}{min}$ = **1** $\frac{1}{min}$	**1/1**	$\frac{\frac{1\ revs}{1\ revs} * \frac{1\ revs}{1\ revs} * \frac{\frac{1}{min}}{\frac{1}{min}}}{\frac{1\ revs}{1\ revs}} = \frac{\mathbf{1}\ \frac{revs}{min}}{\mathbf{1}\ \frac{revs}{min}}$	num: 1 div: 1
°/s, 1 dp 1/10 °/s ($^{\circ}/_{10\ s}$)	**1** $revs_{OUT}$ = **3600** $\frac{^{\circ}}{10}$	**1** $\frac{1}{s}$ = **1** **60** $\frac{1}{min}$	**1/1**	$\frac{\frac{1\ revs}{1\ revs} * \frac{1\ revs}{1\ revs} * \frac{60 * \frac{1}{min}}{\frac{1}{s}}}{\frac{3600\ \frac{^{\circ}}{10}}{1\ revs}} = \frac{\mathbf{1}\ \frac{revs}{min}}{\mathbf{216000}\ \frac{^{\circ}}{10\ s}}$	num: 1 div: 216000
rpm, 2 dp 1/100 rpm ($^{revs}/_{100\ min}$)	**1** $revs_{OUT}$ = **100** $\frac{revs}{100}$	**1** $\frac{1}{min}$ = **1** $\frac{1}{min}$	**1/1**	$\frac{\frac{1\ revs}{1\ revs} * \frac{1\ revs}{1\ revs} * \frac{\frac{1}{min}}{\frac{1}{min}}}{\frac{100\ \frac{revs}{100}}{1\ revs}} = \frac{\mathbf{1}\ \frac{revs}{min}}{\mathbf{100}\ \frac{revs}{100\ min}}$	num: 1 div: 100
			2/3	$\frac{\frac{1\ revs}{1\ revs} * \frac{2\ revs}{3\ revs} * \frac{\frac{1}{min}}{\frac{1}{min}}}{\frac{100\ \frac{revs}{100}}{1\ revs}} = \frac{\mathbf{2}\ \frac{revs}{min}}{\mathbf{300}\ \frac{revs}{100\ min}}$	num: 2 div: 300
mm/s, 1 dp 1/10 mm/s ($^{mm}/_{10\ s}$)	**63.15** $\frac{mm}{revs}$ ⇒ **1** $revs_{OUT}$ = **631.5** $\frac{mm}{10}$	**1** $\frac{1}{s}$ = **1** **60** $\frac{1}{min}$	**1/1**	$\frac{\frac{1\ revs}{1\ revs} * \frac{1\ revs}{1\ revs} * \frac{60 * \frac{1}{min}}{\frac{1}{s}}}{\frac{631.5\ \frac{mm}{10}}{1\ revs}} = \frac{\mathbf{1}\ \frac{revs}{min}}{\mathbf{37890}\ \frac{mm}{10\ s}}$	num: 1 div: 37890
			4/5	$\frac{\frac{1\ revs}{1\ revs} * \frac{4\ revs}{5\ revs} * \frac{60 * \frac{1}{min}}{\frac{1}{s}}}{\frac{631.5\ \frac{mm}{10}}{1\ revs}} = \frac{\mathbf{4}\ \frac{revs}{min}}{\mathbf{189450}\ \frac{mm}{10\ s}}$	num: 2 div: 94725

1) Desired unit at the output
2) Position units per revolution ($revs_{OUT}$).
   Feed constant of drive (PNU 1003) * 10^{-dp} (positions after **d**ecimal **p**oint)
3) Time factor_v: desired time unit per internal time unit
4) Gear ratio: $revs_{IN}$ per $revs_{OUT}$
5) Enter values into formula.

Table A/5: Calculation example Speed factor with CMMS-ST

Examples of calculating the speed factor with CMMP-AS

Units of speed [1]	Feed constant [2]	Time constant [3]	Gear [4]	Formula [5]	Result shortened
rpm, 0 dp 1/100 $^{revs}/_{min}$	**1** $REVS_{OUT}$ = **1** $REVS_{OUT}$	**1** $\frac{1}{min}$ = **4096** $\frac{1}{4096\ min}$	**1/1**	$\frac{\frac{1\ REVS}{1\ REVS} * \frac{4096\ \frac{1}{4096\ min}}{1\ \frac{1}{min}}}{\frac{1\ REVS}{1\ REVS}} = \frac{\mathbf{4096}\ \frac{REVS}{4096\ min}}{\mathbf{1}\ \frac{REVS}{min}}$	num: 4096 div: 1
rpm, 2 dp 1/100 $^{revs}/_{min}$ ($^{revs}/_{100\ min}$)	**1** $revs_{OUT}$ = **100** $\frac{revs}{100}$	**1** $\frac{1}{min}$ = **4096** $\frac{1}{4096\ min}$	**2/3**	$\frac{\frac{2\ REVS}{3\ REVS} * \frac{4096\ \frac{1}{4096\ min}}{1\ \frac{1}{min}}}{\frac{100\ \frac{REVS}{100}}{1\ REVS}} = \frac{\mathbf{8192}\ \frac{REVS}{4096\ min}}{\mathbf{300}\ \frac{REVS}{100\ min}}$	num: 2048 div: 75
°/s, 1 dp 1/10 °/s ($^{\circ}/_{10\ s}$)	**1** $revs_{OUT}$ = **3600** $\frac{^{\circ}}{10}$	**1** $\frac{1}{s}$ = **60** $\frac{1}{min}$ = **60 * 4096** $\frac{1}{4096\ min}$	**1/1**	$\frac{\frac{1\ REVS}{1\ REVS} * \frac{60 * 4096\ \frac{1}{4096\ min}}{1\ \frac{1}{s}}}{\frac{3600\ \frac{^{\circ}}{10}}{1\ REVS}} = \frac{\mathbf{245760}\ \frac{REVS}{4096\ mi}}{\mathbf{3600}\ \frac{^{\circ}}{10s}}$	num: 1024 div: 15
mm/s, 1 dp 1/10 $^{mm}/_{s}$ ($^{mm}/_{10\ s}$)	**63.15** $\frac{mm}{revs}$ ⇒ **1** $revs_{OUT}$ = **631.5** $\frac{mm}{10}$	**1** $\frac{1}{s}$ = **60** $\frac{1}{min}$ = **60 * 4096** $\frac{1}{4096\ min}$	**4/5**	$\frac{\frac{4\ REVS}{5\ REVS} * \frac{60 * 4096\ \frac{1}{4096\ min}}{1\ \frac{1}{s}}}{\frac{631,5\ \frac{mm}{10}}{1\ REVS}} = \frac{\mathbf{1966080}\ \frac{REV}{4096\ r}}{\mathbf{6315}\ \frac{mm}{10s}}$	num: 131072 div: 421

1) Desired unit at the output
2) Positioning units per revolution ($revs_{OUT}$).
Drive's feed constant (PNU 1003) * 10^{-dp} (**d**ecimal **p**laces taken into consideration)
3) Time factor_v: Desired unit of time per internal unit of time
4) Gear ratio: $revs_{IN}$ per $revs_{OUT}$
5) Insert values into formula.

Table A/6: Examples of calculating the speed factor CMMP-AS

A.1.5 Calculating the units of acceleration

The **acceleration factor** (PNU 1007, see section 5.4.12) is used to convert all the acceleration values from the user's **units of acceleration** into the internal unit **revolutions per minute per 256 seconds**.
The speed factor consists of numerators and denominators.

The calculation factor is also calculated in two parts: a conversion factor from internal units of length into the user's positioning units and a conversion factor from internal units of time squared into user-defined units of time squared (e.g. from seconds^2 to minutes^2). The first part is equivalent to calculating the position factor; an additional factor is required to calculate the second part:

Time factor_a	Ratio between the internal unit of time squared and the user-defined unit of time squared (e. g.˚1˚min$^{2\circ}$= 1 min * 1 min = 60 s * 1 min = $^{60}/_{256}$ min * s).
Gear ratio (gear ratio)	Gear ratio between revolutions at the input side ($revs_{IN}$) and revolutions at the output side ($revs_{OUT}$).
Feed constant (feed constant)	Ratio between movement in positioning units at the drive and revolutions at the gear unit's output ($revs_{OUT}$). (e. g. 1˚rev ≙ 63.15 mm or 1˚rev ≙ 360° degrees)

The acceleration factor is calculated using the following formula:

$$\text{Acceleration factor} = \frac{\text{Gear ratio} * \text{Time factor\_a}}{\text{Feed constant}}$$

Like the position and speed factors, the acceleration factor also has to be written to the motor controller separated into numerators and denominators. This can make it necessary to bring the fraction up to whole integers by expanding it accordingly.

Example

First, the desired unit (column 1) and the desired number of decimal places (dp) have to be specified, along with the application's gear ratio and its feed constant (if applicable). The feed constant is then displayed in the desired positioning units (column 2).

Then, the desired unit of time^2 is converted into the motor controller's unit of time^2 (column 3).

In this way, all the values can be entered into the formula and the fraction can be calculated:

Sequence of calculating the acceleration factor

Units of acceler. 1)	Feed constant 2)	Time constant 3)	Gear 4)	Formula 5)	Result shortened
mm/s^2, 1 dp 1/10 $\frac{mm}{(10\,s^2)}$	**63,15** $\frac{mm}{U}$ $\Rightarrow$ **1** U_{AUS} = **631,5** $\frac{mm}{10}$	**1** $\frac{1}{s^2}$ = **60** $\frac{1}{min * s}$ = **60 * 256** $\frac{1}{256 * min * s}$	**4/5**	$\frac{4\,U}{5\,U} * \frac{60 * 256 \frac{1}{256\ min * s}}{1\frac{1}{s^2}} = \frac{122880 \frac{U}{256\,s}}{6315 \frac{mm}{10s^2}}$	num: 8192 div: 421

Fig. A/5: Sequence of calculating the acceleration factor

Examples of calculating the acceleration factor

Units of acceler. [1]	Feed constant [2]	Time constant [3]	Gear [4]	Formula [5]	Result shortened
rpm/s, 0 dp $^{revs}/_{min\ s}$	$\mathbf{1}\ U_{OFF} = \mathbf{1}\ U_{OFF}$	$\mathbf{1}\ \frac{1}{min * s} = \mathbf{256}\ \frac{\frac{1}{min}}{256\ *\ s}$	**1/1**	$\frac{\frac{1\ U}{1\ U} * \frac{256\ \frac{1}{256\ min\ s}}{1\ \frac{1}{min * s}}}{\frac{1\ U}{1\ U}} = \frac{\mathbf{256}\ \frac{\frac{U}{min}}{256 * s}}{\mathbf{1}\ \frac{U}{min}}$	num: 256 div: 1
$^{\circ}/s^2$, 1 dp 1/10 $^{\circ}/_{s^2}$ ($^{\circ}/_{10\ s^2}$)	$\mathbf{1}\ U_{OFF} = \mathbf{3600}\ \frac{^{\circ}}{10}$	$\mathbf{1}\ \frac{1}{s^2} = \mathbf{60}\ \frac{1}{min * s} = \mathbf{60 * 256}\ \frac{\frac{1}{min}}{256\ *\ s}$	**1/1**	$\frac{\frac{1\ U}{1\ U} * \frac{60 * 256\ \frac{1}{256\ min * s}}{1\ \frac{1}{s^2}}}{\frac{3600\ \frac{^{\circ}}{10}}{1\ U}} = \frac{\mathbf{15360}\ \frac{\frac{U}{min}}{256 * s}}{\mathbf{3600}\ \frac{^{\circ}}{10\ s^2}}$	num: 64 div: 15
rpm^2, 2 dp 1/100 $^{revs}/_{min^2}$ ($^{revs}/_{100\ min^2}$)	$\mathbf{1}\ U_{OFF} = \mathbf{100}\ \frac{U}{100}$	$\mathbf{1}\ \frac{1}{min^2} = \frac{\mathbf{1}}{\mathbf{60}}\ \frac{\frac{1}{min}}{s} = \frac{\mathbf{256}}{\mathbf{60}}\ \frac{\frac{1}{min}}{256 * s}$	**2/3**	$\frac{\frac{2\ U}{3\ U} * \frac{256\ \frac{1}{256\ min * s}}{60\ \frac{1}{min^2}}}{\frac{100\ \frac{U}{100}}{1\ U}} = \frac{\mathbf{512}\ \frac{\frac{U}{min}}{256\ s}}{\mathbf{18000}\ \frac{U}{100\ min^2}}$	num: 32 div: 1125
mm/s^2, 1 dp 1/10 $^{mm}/_{s^2}$ ($^{mm}/_{10\ s^2}$)	$\mathbf{63,15}\ \frac{mm}{U} \Rightarrow \mathbf{1}\ U_{OFF} = \mathbf{631,5}\ \frac{mm}{10}$	$\mathbf{1}\ \frac{1}{s^2} = \mathbf{60}\ \frac{1}{min * s} = \mathbf{60 * 256}\ \frac{\frac{1}{min}}{256\ *\ s}$	**4/5**	$\frac{\frac{4\ U}{5\ U} * \frac{60 * 256\ \frac{1}{256\ min * s}}{1\ \frac{1}{s^2}}}{\frac{631,5\ \frac{mm}{10}}{1\ U}} = \frac{\mathbf{122880}\ \frac{\frac{U}{min}}{256\ s}}{\mathbf{6315}\ \frac{mm}{10\ s^2}}$	num: 8192 div: 421

1) Desired unit at the output
2) Positioning units per revolution ($revs_{OUT}$).
   Drive's feed constant (PNU 1003) * 10^{-dp} (decimal places taken into consideration)
3) Time factor_a: Desired unit of time^2 per internal unit of time^2
4) Gear ratio: $revs_{IN}$ per $REVS_{OUT}$
5) Insert values into formula.

Table A/7: Examples of calculating the acceleration factor

Appendix B

A

C

D

E

F

H

J

M

O

P

R

S

T

U

V

W