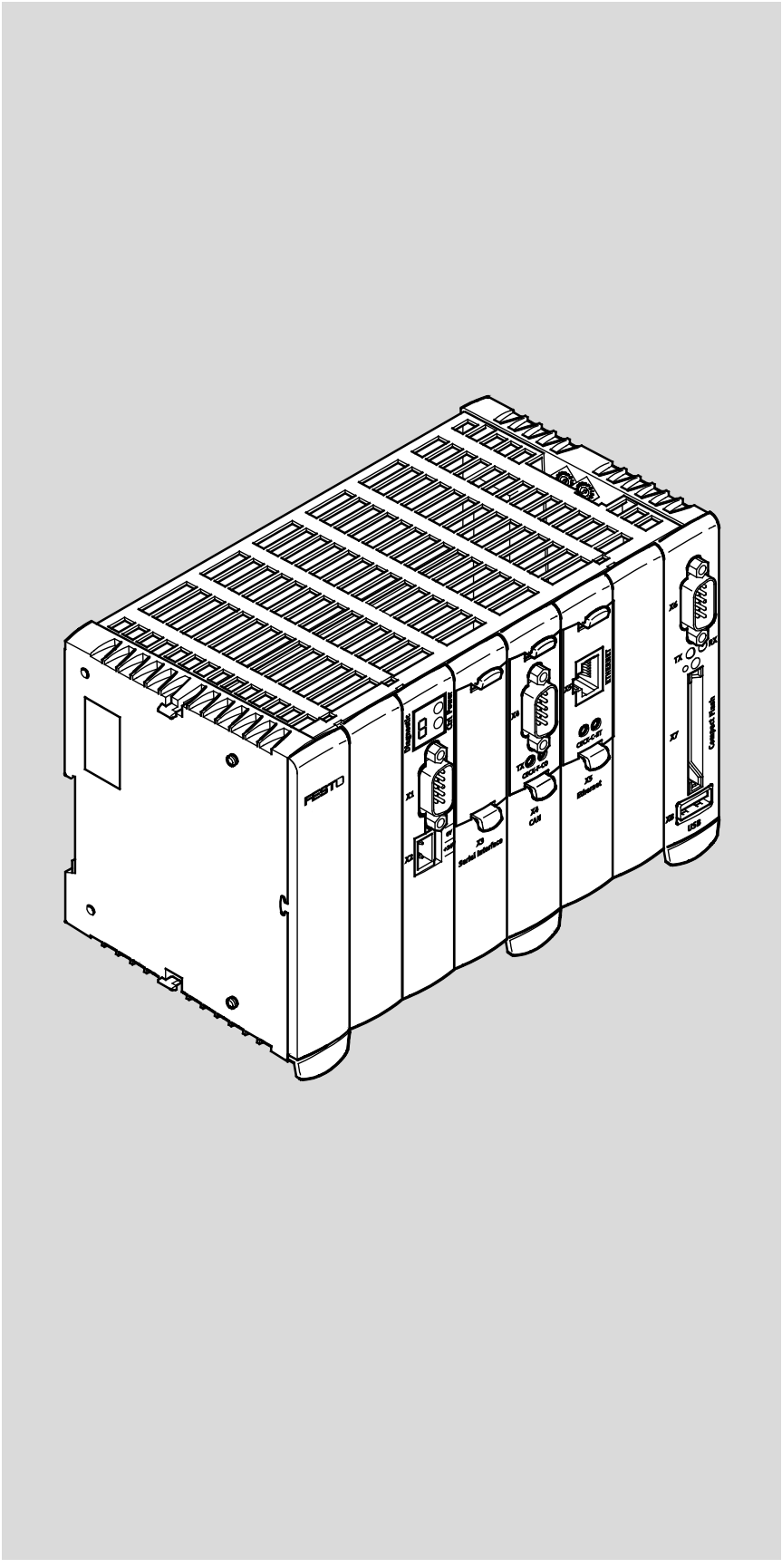
Multi-axis control system

CMXR-C1



Description

PLC interface

560328
1501b
[8039883]

Version 1501b

Designation GDCP-CMXR-F-EN

Order no. 560328

Internet: http://www.festo.com

E-Mail: service_international@festo.com

Directory of revisions			
Created by:			
Handbook title:	GDCP-CMXR-F-EN		
Filename:			
File storage location:			
Sequential no.	Description	Revisions index	Date of revision
001	Created	0805NH	09.07.2008
002	Revision	0909a	25.08.2009
003	Revision	1501b	15.01.2015

Table 1.1 Directory of revisions

Specified directives/standards

Issue status
EN ISO 13849-1:2008-12

Table 1.2 Directives/standards specified in the document

Trademarks

Microsoft® Windows® Registered trademark of Microsoft Corporation

PROFIBUS, PROFIBUS DP® registered trademarks of PROFIBUS International (PI)

Notes on this documentation

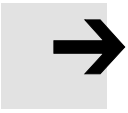
This documentation describes the MotionControl Profile (MCP) for activating the CMXR multi-axis control system via PROFIBUS as well as via the digital I/O interface.

Identification of dangers and instructions on how to avoid them:

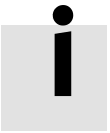
Warning
Dangers which can lead to death or serious injuries.

Caution
Dangers which can lead to slight injuries or to serious material damage.

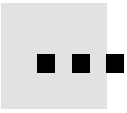
Other symbols:



Note
Material damage or failure of function.



Recommendation, tip, reference to other documentation.



Necessary or useful accessories.

Information on environmentally friendly use.

Text designation:

- Activities that can be carried out in any order.

1. Activities which should be carried out in the specified sequence.

– General lists.

Additional documentation

The total functionality of the multi-axis control system CMXR-C1 is described in the following documents:

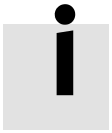
Name	Contents
GDCP-CMXR-SY-...	System description
GDCP-CMXR-HW-...	Mounting and installation
GDCP-CMXR-F-...	PLC interface (this document)
GDCP-CMXR-SW-...	Programming instructions for CMXR FTL base

Table 1.3 Further documentation – CMXR-C1 multi-axis control system

The operator unit CDSA-D1-VX also has additional documents available:

Name	Contents
GDCP-CDSA-SY-...	System manual
GDCP-CDSA-SW-...	Software manual

Table 1.4 Additional documentation – operator unit CDSA-D1-VX



The listed documents are available for download in the Festo Support Portal (➔ www.festo.com/sp).

Table of contents

1. Safety and requirements for product use

1.1 Safety

1.1.1 General safety information

Caution

Malfunction of the machine or system due to improper manipulation.

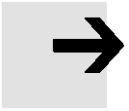
- Provide the load and operating voltage as a circuit with limited energy in accordance with IEC 61131–2 and always make sure there is fuse protection with max. 10 A.
- The power supply for the equipment must be assured by reliable isolation of the low-voltage from contact-hazardous voltages.
- Make sure that an interrupted program can be properly restarted after voltage drops and voltage failure. The occurrence of dangerous operating statuses must be avoided, even if temporary.
- Emergency stop equipment must remain effective in all operating modes. Unlocking the emergency stop equipment must not initiate an uncontrolled restart.
- Additional external measures, which guarantee a safe operating status of the entire system even in the event of an error, must be put in place everywhere that errors that occur in the control system could result in personal injuries or extensive material damage.

Warning

Dangerous voltage!

- If not otherwise specified, maintenance work must always be carried out with the system switched off! At the same time, the system must be protected against unauthorised or unintentional restart.

Measuring or test work on the system must be carried out only by electrically skilled persons.



Note

Damage to the product from incorrect handling.

- Never unplug or plug in a product when powered.
- Observe the handling specifications for electrostatically sensitive devices.

1.1.2 Intended use

The CMXR multi-axis control system is intended to operate a kinematics system in the environment of a machine or automated system.

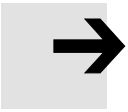
- Use the multi-axis control system only as follows:
  - in perfect technical condition
  - in original status without unauthorised modifications, except for the adaptations described in this documentation
  - within the limits of the product defined by the technical data
  - in an industrial environment.

Warning

The Festo CMXR multi-axis control system is not designed for safety-relevant control tasks (e.g. emergency stop or monitoring of reduced speeds).

The CMXR multi-axis control system conforms to category B of EN ISO 13849-1 and is thus not sufficient for the implementation of safety functions for the protection of persons.

- Additional external safeguarding that ensures the safe operating status of the entire system even in the event of a malfunction must be adopted for safety-relevant control tasks or for the safety of persons.



Note

In the event of damage caused by unauthorised manipulation or other than intended use, the guarantee is invalidated and the manufacturer is not liable for damages.

1.2 Requirements for product use

- Provide this documentation to the following persons:
  - design engineer
  - installer
  - commissioner of the machine or system
- Comply with the specifications of the documentation. Follow all accompanying documentation and the documentation of any associated accessories.
- Take the following into consideration for the destination:
  - applicable legal regulations
  - regulations and standards
  - regulations of the testing organisations and insurers
  - national specifications

For correct and safe use:

- Observe all warnings and notes.
- Comply with all load limits of the product and the connected components.

1.2.1 Qualification of specialized personnel

- The product should only be installed by specialized personnel with corresponding qualifications.

The following knowledge is required:

- installation and operation of electrical control systems
- applicable regulations for operating safety-engineering systems
- applicable regulations for accident protection and operational safety
- documentation and mode of operation of the product.

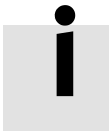
1.3 Service

- Please consult your local Festo service or write to the following e-mail address if you have any technical problems (➔ service_international@festo.com).

2. Control interfaces for external activation

The CMXR multi-axis control system can be controlled through 3 different methods:

- through a higher-level control system via digital inputs/outputs with the MotionControl Profile MPC1-EA.
- through a higher-level control system via PROFIBUS DP. The following 2 performance classes are distinguished:
  - MotionControl Profile MCP1-PB (with 12 byte data packet)
  - MotionControl Profile MCP2-PB (with 64 byte data packet)
- Operation without external control (not described here, ➔ System description GDCP-CMXR-SY-...).



The CDSA operator unit can be connected with all activating methods.

3. PLC interface MCP1

3.1 General

The PLC interface can be reached via PROFIBUS as well as via digital I/O in a simpler variant. Refer to the system manual for information on how these settings are carried out.

3.2 Activation via digital I/O

Activation via digital I/O is done through wiring with three I/O modules of type CECX-D-8E8A-NP-2. The MCP1 I/O signals are assigned to concrete inputs/outputs in the control system configuration with the Festo Configuration Tool (FCT).

The following describes the signal allocation of the three I/O modules.

I/O module - BasicIO

Signal	Name	Function	Master control required	Description
DO:0	doutError	CMXR error active	No	3.4.2
DO:1	Reserved	–		
DO:2	doutAutoSelected	Automatic operating mode set	No	3.4.10
DO:3	doutManSelected	Manual operating mode set	No	3.4.10
DO:4	Unassigned	For free use		
DO:5	Unassigned	For free use		
DO:6	Unassigned	For free use		
DO:7	Unassigned	For free use		
DI:0	dinEMStop	Emergency stop	No	System description
DI:1	dinEnabling	Enabling button	No	System description
DI:2	dinAutoSelect	Automatic operating mode	No	3.4.10
DI:3	dinManSelect	Manual operating mode	No	3.4.10
DI:4	Unassigned	For free use		
DI:5	Unassigned	For free use		
DI:6	Unassigned	For free use		
DI:7	Unassigned	For free use		

I/O module - MCPIO1

Signal	Name	Function	Master control required	Description
DO:0	CTRDY	Controller ready	No	System description
DO:1	HOLD	HOLD active	No	3.4.8
DO:2	DRRDY	Drives ready	No	System description
DO:3	DRREF	Drives referenced	No	System description
DO:4	ACCENA	Write access not assigned	No	3.4.6
DO:5	WRACC	Write access received	Yes	3.4.6
DO:6	ERROR	Error is active	No	3.4.2
DO:7	WDBIT	Watchdog bit	No	3.4.1
DI:0	–			
DI:1	–			
DI:2	–			
DI:3	DRENA	Enable drive	Yes	3.4.7
DI:4	HALTENA	Enable HOLD	Yes	3.4.8
DI:5	WRREQU	Request write access	No	3.4.6
DI:6	QUITERR	Acknowledge error	Yes	3.4.2
DI:7	WDBIT	Watchdog bit	No	3.4.1

I/O module - MCPIO2

Signal	Name	Function	Master control required	Description
DO:0	–			
DO:1	–			
DO:2	–			
DO:3	–			
DO:4	ACK	Load the program/start executed	No	3.4.11 ff.
DO:5	NACK	Load the program/start not executed	No	3.4.11 ff.
DO:6	RUNNING	Program running	No	3.4.11 ff.
DO:7	LOADED	Program loaded	No	3.4.11 ff.
DI:0	PRGNR Bit0	Selection of program no.	Yes	
DI:1	PRGNR Bit1	Selection of program no.	Yes	
DI:2	PRGNR Bit2	Selection of program no.	Yes	
DI:3	PRGNR Bit3	Selection of program no.	Yes	
DI:4	STOP	Interrupt program	Yes	3.4.14
DI:5	START	Start/continue program	Yes	3.4.13
DI:6	UNLOAD	Unload program	Yes	3.4.12
DI:7	LOAD	Load program	Yes	3.4.11

3.3 Activation via PROFIBUS

3.3.1 I/O module BasicIO (optional)

With activation via PROFIBUS, the I/O module - BasicIO can be omitted. The signals of this module are then transmitted via PROFIBUS.

I/O module - BasicIO

Signal	Name	Function	Master control required	Description
DO:0	doutError	CMXR error active	No	3.4.2
DO:1	Reserved	–		
DO:2	doutAutoSelected *	Automatic operating mode set	No	3.4.10
DO:3	doutManSelected *	Manual operating mode set	No	3.4.10
DO:4	Unassigned	For free use		
DO:5	Unassigned	For free use		
DO:6	Unassigned	For free use		
DO:7	Unassigned	For free use		
DI:0	dinEMStop	Emergency stop	No	System description
DI:1	dinEnabling	Enabling button	No	System description
DI:2	dinAutoSelect *	Automatic operating mode	No	3.4.10
DI:3	dinManSelect *	Manual operating mode	No	3.4.10
DI:4	Unassigned	For free use		
DI:5	Unassigned	For free use		
DI:6	Unassigned	For free use		
DI:7	Unassigned	For free use		
(*) The mode can also be selected via PROFIBUS as an option.				

3.3.2 Control signals in the PROFIBUS data packet

MCP1-PB includes 12-byte control signals and 12-byte status signals, which are cyclically exchanged. The signals are exchanged as I/O data packets on PROFIBUS.

Byte 1	Byte 2	Byte 3	Byte 4
CREG	CREG2	Unassigned	PRGNR
Control register 1	Control register 2		Program number

Byte 5	Byte 6	Byte 7	Byte 8
COVR	CJOG	EDATA1	EDATA2
Override	JOG register	8 input bits	8 input bits

Byte 9	Byte 10	Byte 11	Byte 12
JOGPOS			
Target position of the JOG axis			

Control register 1

Bit	7	6	5	4	3	2	1	0
CREG	WDBIT	QUITERR	WRREQU	HALTENA	DRENA	JPENA	HAND	AUTO
	Watchdog bit	Acknowledge error	Request master control	Enable HOLD	Enable drives	Enable Jog position	Manual operating mode	Automatic operating mode

Control register 2

Bit	7	6	5	4	3	2	1	0
CREG2	LOAD	UNLOAD	START	STOP	Unassigned	Unassigned	DEVENA	EMSTOP
	Load program	Unload program	Start program	Interrupt program			Enabling button	Emergency stop

Program number

Size 1 byte, possible program numbers 1 ... 255.

The program number can be used to address the desired FTL program on the control system. The program number is assigned to the FTL program via an assignment table in the FCT.

Override

Size 1 byte, possible values 0 ... 100 %.

The override can be used to limit the nominal value of the speed of the kinematics, e.g. for commissioning.

Jog register

Bit	7	6	5	4	3	2	1	0
CJOG	JOGP	JOGN	COORD		AXIS			
	Jog positive	Jog negative	JOG coordinate system		JOG axis			

EDATA1/EDATA2

Size 2 x 8 bit.

The PLC can send freely definable signals to the multi-axis control system over the 16 binary inputs.

Access in the FTL program: VarTest := plc_InBool[0-15]

Target position of the JOG axis

Size 4 bytes.

The target position is used to specify a position for the selected axis, which can be approached via the Jog command.

3.3.3 Status signals in the PROFIBUS data packet

Byte 1	Byte 2	Byte 3	Byte 4
SREG	SREG2	SREG3	Unassigned
Status register 1	Status register 2	Status register 3	

Byte 5	Byte 6	Byte 7	Byte 8
SOVR	SJOG	ADATA1	ADATA2
Override status	JOG register	8 output bits	8 output bits

Byte 9	Byte 10	Byte 11	Byte 12
ACTPOS			
Actual position of the JOG axis			

Status register 1

Bit	7	6	5	4	3	2	1	0
SREG	WDBIT	ERROR	WRACC	ACCENA	DRREF	DRRDY	HOLD	CTRDY
	Watchdog bit	Error is active	Master control received	Master control not assigned	Drives referenced	Drives ready	Halt active	Controller ready

Status register 2

Bit	7	6	5	4	3	2	1	0
SREG2	LOADED	RUNNING	NACK	ACK	JPRDY	Unassigned	HAND	AUTO
	Program loaded	Program running	Program ACK	Program NACK	Jog position enabled		Manual active	Auto active

Status register 3

Bit Bit order	7	6 2^0	5 2^1	4 2^2	3 2^0	2 2^1	1 2^2	0 2^3
SREG3	Unassigned	MSG			MSGNR			
		Message class			Message number			

Override status

Size 1 byte, possible values 0 ... 100 %.

The override status displays the active override value.

JOG register

Bit	7	6	5	4	3	2	1	0
SJOG	AXIS				COORD		ESN	ESP
	JOG axis				JOG coordinate system		Limit switch negative	Limit switch positive

ADATA1/ADATA2

Size 2 x 8 bit.

The multi-axis control system can send freely definable signals to the PLC over the 16 binary outputs.

Access in the FTL program: plc_OutBool[0-15] := VarTest

Actual position of the JOG axis

Size 4 bytes.

The actual position is used to display the current position of the selected axis.

3.4 Function of the PLC interface MCP1

3.4.1 Watchdog

The watchdog control bit CREG.WDBIT is sent from the PLC to the robot controller. This, in turn, mirrors the bit at the SREG.WDBIT output with a maximum delay in the watchdogTimeout parameter.

Condition	The running up of the control system and the interface was successful.
Action	Alternative setting and resetting of CREG.WDBIT
Reaction	The current value of CREG.WDBIT is issued in the SREG.WDBIT
Configuration	Configuration is carried out via FCT

3.4.2 Error active

The SREG.ERROR bit is used to inform the control system that at least one error is active. The cause of the error can be evaluated through the status-register 3. A faultless robot controller is a requirement for operation via the PLC interface.

3.4.3 Message system

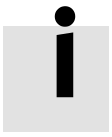
Status register 1, bit 6, is used to show whether an error is active.

In the event of an error, the message class and message number can be read out via the status-register 3.

3.4.4 Message classes and message numbers

The message class provides the approximate classification of a message. It is transmitted with 3 bits over the status-register 3 (SREG3.MSG).

The message number provides additional information on the message class. It is displayed through an additional 4 bits in the status-register 3 (SREG3.MSGNR).



The waiting message only becomes valid when SREG3.MSG is not equal to 0 or the error bit SREG.ERROR is set.

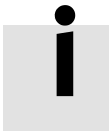
Coding of message class MSG and message number MSGNR:

The “Component” column lists the component number of the error message or error group contained in the status display of the CDSA-D1-VX operator unit and the reports of the multi-axis control system on these errors.

MSG	MSGNR	Comment	Component
PLC interface error, MSG=1			
1	1	Program could not be loaded	51.390
1	2	Program could not be unloaded	51.391
1	3	Program could not be started	51.392
1	4	Program could not be stopped	51.393
System error, MSG=2			
2	1	Error in MotionControl component	4000.xx
2	2	Error in TeachControl component	
2	3	Error in Motion Function Blocks component	
2	4	Error in RobotControl component	
2	5	Error in base system component	2050.xx
2	6	Error in IO-system component	2220.xx
2	7	Error in IEC run-time system component	
2	8	Error in communication interface component	
2	9	Error in visualisation interface component	
2	15	Error in system-component component	6338.xx

3.4.5 Acknowledge error

The control-register 1 (CREG.QUITERR) allows a higher-order controller to acknowledge errors that are present.



Acknowledging an error does not eliminate its cause.
It is advisable to use special tools for more exact investigation of the cause.

Message acknowledgement and prioritisation

Due to the fact that only one bit is available for acknowledging the messages, the following prioritisation has been introduced for acknowledging the messages.

Priority	Message class
A	PLC interface error
B	System error

An error is confirmed with a positive edge at the control-register 1 (CREG.QUITERR). Errors can only be deleted one after the other.

Therefore, the interface error should be the first to be acknowledged (if present). Then, any system errors will be acknowledged.

For reasons of error reproduction, interface errors are recorded in the system's buffer memory as an "Info message".

Condition	The running up of the control system and the interface was successful.
Action	Alternative setting for the CREG.QUITERR (trailing edge) bit.
Reaction	The last message applied is acknowledged (first the interface errors, then the control system errors).

3.4.6 Requesting the master control

To receive write access for a PLC interface and ultimately access to the multi-axis control system, it is necessary that you request this at the multi-axis control system.

The SREG.ACCENA bit shows whether write access is currently free or assigned.

To request write access, a rising edge has to be placed onto the CREG.WRREQU bit.

Requesting and receiving master control

If you receive write access for an interface, then the SREG.WRACC bit is set. The higher-order controller has complete access to the multi-axis control system as long as the multi-axis control system bit is set and the CREG.WRREQU bit remains set.

Requesting but not receiving master control

There could be a situation where, at the time of the log-in attempt, another PLC interface priority has write access. This can be recognised, on the one hand, by the bit SREG.ACCENA = FALSE and not receiving a positive reply. If despite this a log-in attempt has been started, SREG.WRACC will never be TRUE.

Surrendering master control

If the interface has master control, it can be surrendered again with a trailing edge at bit CREG.WRREQU.

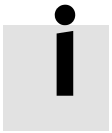
Surrendering master control is confirmed using SREG.WRACC = FALSE.

3.4.7 Switching on drives

Condition	– Master control on interface – Valid operating mode is selected – No error present – Emergency stop is not active
Action	Setting bit CREG.DRENA
Reaction	– The robot's drives are switched on. – ITF replies SREG.DRRDY = TRUE.

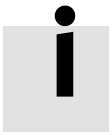
3.4.8 Programmed stop

Condition	Master control on interface
Action	Setting bit CREG.HALTENA
Reaction	– If the macro ProgHold() is located in a positioning program, the sequence will be interrupted at this position. – Interface answers with SREG.HALT "Hold active". The program is continued as soon as the bit CREG.HALTENA is reset. After that, SREG.HALT returns to FALSE.



Detailed information can be taken from the CMXR FTL-base programming instructions (➔ GDCP-CMXR-SW-...).

3.4.9 Jog



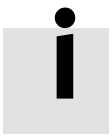
The jog function is only possible with activation via PROFIBUS (MCP1-PB) and not via digital I/O (MCP1-EA).

Jog in positive/negative direction

Condition	– Master control on interface – Drives are switched on – Configured operating mode allows jogging – CREG.JPENA bit is not activated – A jog axis CJOG.AXIS and a jog coordinate system CJOG.COORD are selected in the JOG-register CJOG.
Action	Set bit CJOG.JOGN or CJOG.JOGP
Reaction	The selected jog axis is moved in the selected reference system analogously to the jog keys on the operator unit as long as the CJOG.JOGN or CJOG.JOGP bit has the TRUE status.

Jog to position

Condition	– Master control on interface – Drives are switched on – Configured operating mode allows jogging – CREG.JPENA bit is activated – A jog axis CJOG.AXIS and a jog coordinate system CJOG.COORD are selected in the JOG-register CJOG. – An achievable position is specified in the jog-data control register JOGPOS.
Action	Set bit CJOG.JOGN or CJOG.JOGP
Reaction	When setting the CJOG.JOGN or CJOG.JOGP bit (both are possible), the jogging movement is towards the position that is in the jog-data control register. When resetting the JOGN or JOGP bit, the movement is stopped.



Wrist axes can only be run to a specified position in the axis coordinate system. This is not possible in other coordinate systems.

JOG control register CJOG

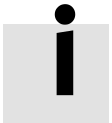
AXIS	Number of the JOG axis
	0 ... 8 corresponds to:
	Travel in axes: A1.... A6, auxiliary axes A7 ... A9
	Travel in Cartesian coordinates: X, Y, Z, A, B, C, AUX1 ... 3
	If axes which do not exist are selected, then the last existing axis will be selected, e.g. robot has 5 axes, axis 6 is selected, axis 5 is active.
COORD	JOG coordinate system
	Axis coordinate system
	World coordinate system
	Active reference system
	Tool coordinate system
JOGN	JOG bit negative direction Manual movement in negative direction.
JOGP	JOG bit positive direction
	Manual movement in positive direction. If the CREG.JPENA bit is set, then the target position is approached in the Jog position register JOGPOS.

JOG status register

AXIS	Active JOG axis
COORD	Active JOG coordinate system
ESN	Negative limit switch reached
ESP	Positive limit switch reached

3.4.10 Selecting operating mode

The interface can be used to select two operating modes, MANUAL and AUTOMATIC.



The period of time for switching from one operating mode to another valid operating mode is currently defined at 1.00 second.

The system behaviour for an invalid operating mode can be found in the system manual (➔ GDCP-CMXR-SY-...).

Note
If the system is configured in such a way that the operating mode is specified via digital I/Os and not via PROFIBUS, the inputs HAND and AUTO stay ineffective.

Control inputs

CREG.AUTO	CREG.HAND	Status
0	0	Invalid, no operating mode
0	1	Manual operating mode
1	0	Automatic operating mode
1	1	Invalid, no operating mode

Status outputs

SREG.AUTO	SREG.HAND	Status
0	0	Invalid, no operating mode
0	1	Manual operating mode
1	0	Automatic operating mode

3.4.11 Load program

The program with the transferred program number is loaded. Several programs can be loaded one after the other.

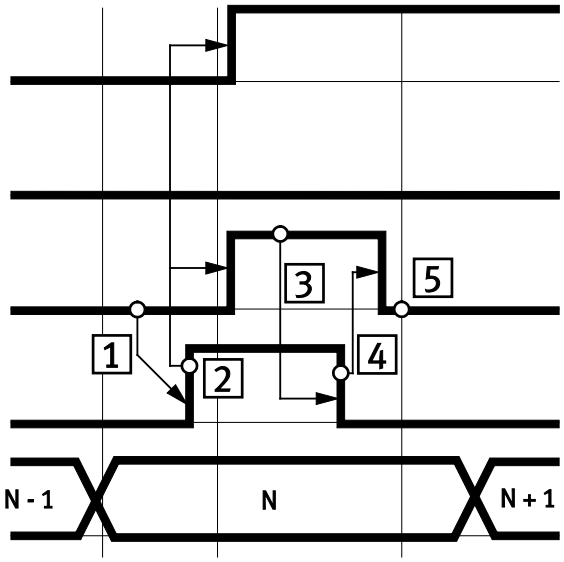
Condition	– Master control on PLC interface
	– The programs in the program table are available on the controller.
	– Program number 1 ... n is in the program register
	  – MCP1-EA: CREG.PRGNR register (4 bit)
	  – MCP1-PB: PRGNR register (1 byte)

Action	– Set bit CREG.LOAD (rising edge). – Wait for Acknowledge SREG.ACK or Not Acknowledge SREG.NACK
Reaction	Program/project is loaded.

Loading first program with positive confirmation

Condition
- SREG.ACK is set and shows that the loading process was successful.
- SREG.LOADED is set and shows that at least one program has been loaded.

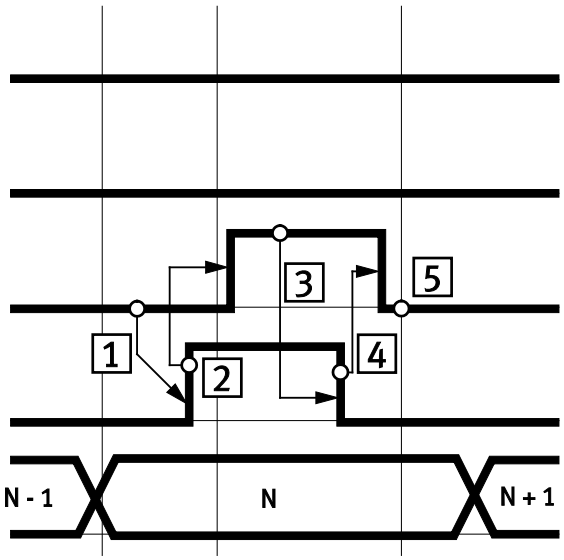
Signal curve

Min. 1 prg. loaded SREG2.B7 LOADED Acknowledge SREG2.B5 NACK Load/reset Ack SREG2.B4 ACK Load program CREG2.B7 LOAD Req. prg number Output data		1 Requirements: ACK and NACK = 0 2 Rising edge at “Load program” loads the new program number N ⚠ Condition: Program is error free and can be loaded. 3 The LOAD output must carry logic 1 for as long as it takes until the ACK input also reports logic 1. 4 As soon as ACK = 1 has been recognised by the PLC, it may reset LOAD to 0. The CMXR reacts with a trailing edge at ACK. 5 As soon as ACK = 0 has been recognised by the PLC, the controller is ready for a new job.

Loading first program with negative confirmation

Condition
- SREG.NACK is set and shows that the loading procedure was not successful.
- SREG.LOADED remains FALSE.
- SREG.ERROR shows error is present.

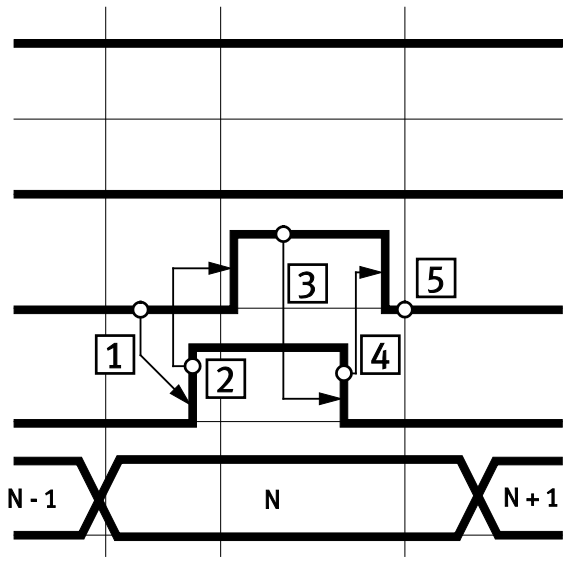
Signal curve

Min. 1 prg. loaded SREG2.B7 LOADED Acknowledge SREG2.B4 ACK Not acknowledge SREG2.B5 NACK Load program CREG2.B7 LOAD Req. prg number Output data		1 Requirements: ACK and NACK = 0 2 Rising edge at “Load program” loads the new program number N ⚠ Condition: The program cannot be loaded for one reason or another. 3 The LOAD output must carry logic 1 for as long as it takes until the ACK input also reports logic 1. 4 As soon as NACK = 1 has been recognised by the PLC, it may reset LOAD to 0. The CMXR reacts with a trailing edge at NACK. 5 As soon as NACK = 0 has been recognised by the PLC, the controller is ready for a new job. ⚠ The cause of the failed loading process is reported as an error.

Loading a further program with positive confirmation

Condition
- SREG.ACK is set and shows that the loading process was successful.
- SREG.LOADED stays set and shows that at least one program has been loaded.

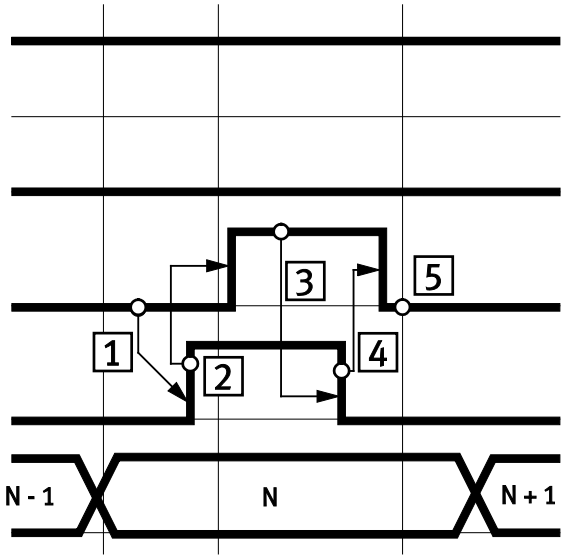
Signal curve

Min. 1 prg. loaded SREG2.B7 LOADED Acknowledge SREG2.B5 NACK Acknowledge SREG2.B4 ACK Load program CREG2.B7 LOAD Req. prg number Output data		1 Requirements: ACK and NACK = 0 2 Rising edge at “Load program” loads the new program number N ⚠ Condition: Program is error free and can be loaded. 3 The LOAD output must carry logic 1 for as long as it takes until the ACK input also reports logic 1. 4 As soon as ACK = 1 has been recognised by the PLC, it may reset LOAD to 0. The CMXR reacts with a trailing edge at ACK. 5 As soon as ACK = 0 has been recognised by the PLC, the controller is ready for a new job.

Loading a further program with negative confirmation

Condition
- SREG.NACK is set and shows that the loading procedure was not successful.
- SREG.LOADED stays set and shows that at least one program has been loaded.
- SREG.ERROR shows error is present.

Signal curve

Min. 1 prg. loaded SREG2.B7 LOADED Acknowledge SREG2.B4 ACK Not acknowledge SREG2.B5 NACK Load program CREG2.B7 LOAD Req. prg number Output data		1 Requirements: ACK and NACK = 0 2 Rising edge at “Load program” loads the new program number N ⚠ Condition: The program cannot be loaded for one reason or another. 3 The LOAD output must carry logic 1 for as long as it takes until the ACK input also reports logic 1. 4 As soon as NACK = 1 has been recognised by the PLC, it may reset LOAD to 0. The CMXR reacts with a trailing edge at NACK. 5 As soon as NACK = 0 has been recognised by the PLC, the controller is ready for a new job. ⚠ The cause of the failed loading process is reported as an error.

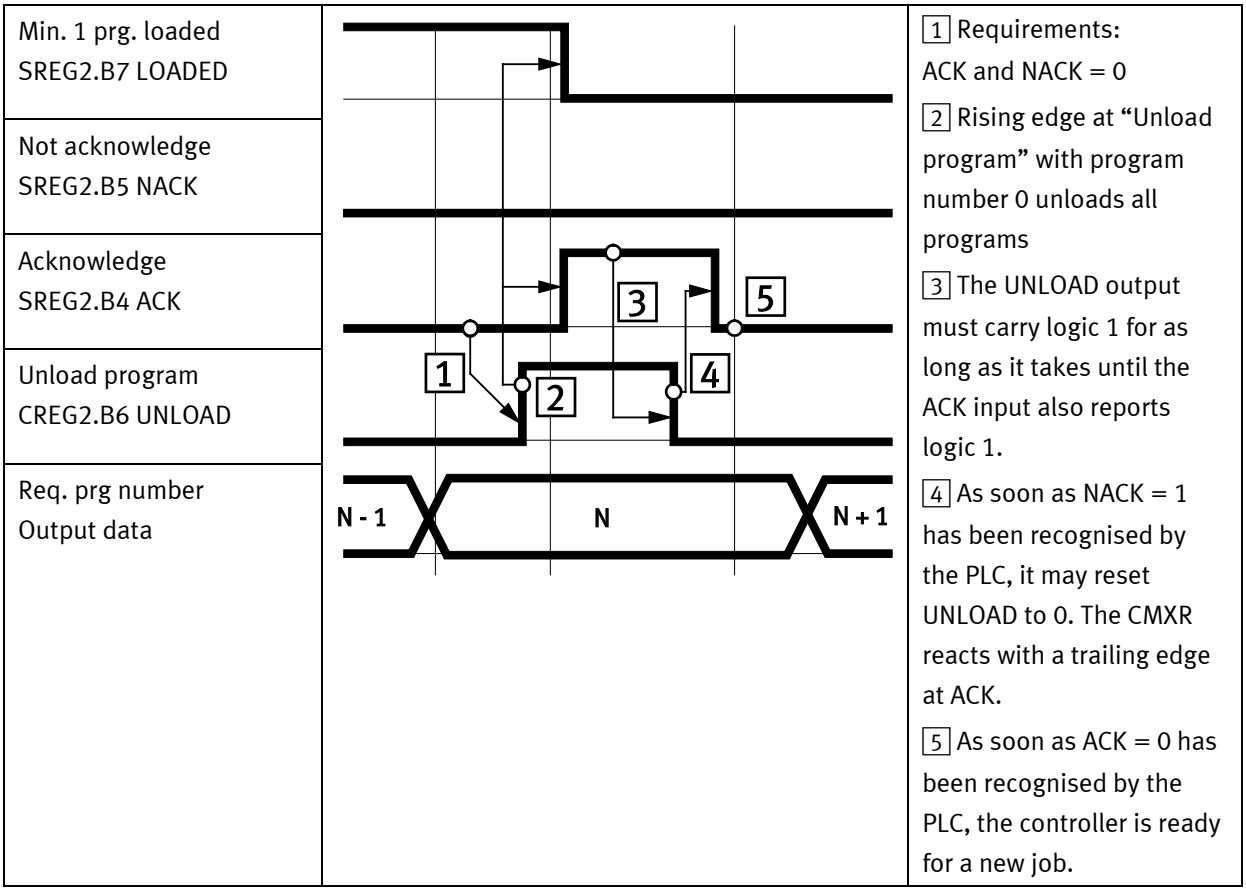
3.4.12 Unloading program

The program with the transferred program number is unloaded. It is also possible to unload all programs at once. To do this, the program number 0 has to be transferred.

Condition	– Master control on PLC interface
	– The program to be unloaded is loaded.
	– ACK and NACK are FALSE
	– Program number 1 ... n is in the program register
	  – MCP1-EA: CREG.PRGNR register (4 bit)
	  – MCP1-PB: PRGNR register (1 byte)

Action	– Set bit CREG.UNLOAD (rising edge). – Wait for Acknowledge SREG.ACK or Not Acknowledge SREG.NACK
Reaction	– A program is unloaded. – In the event of an error, SREG.ERROR is set.

Signal curve for unloading all programs



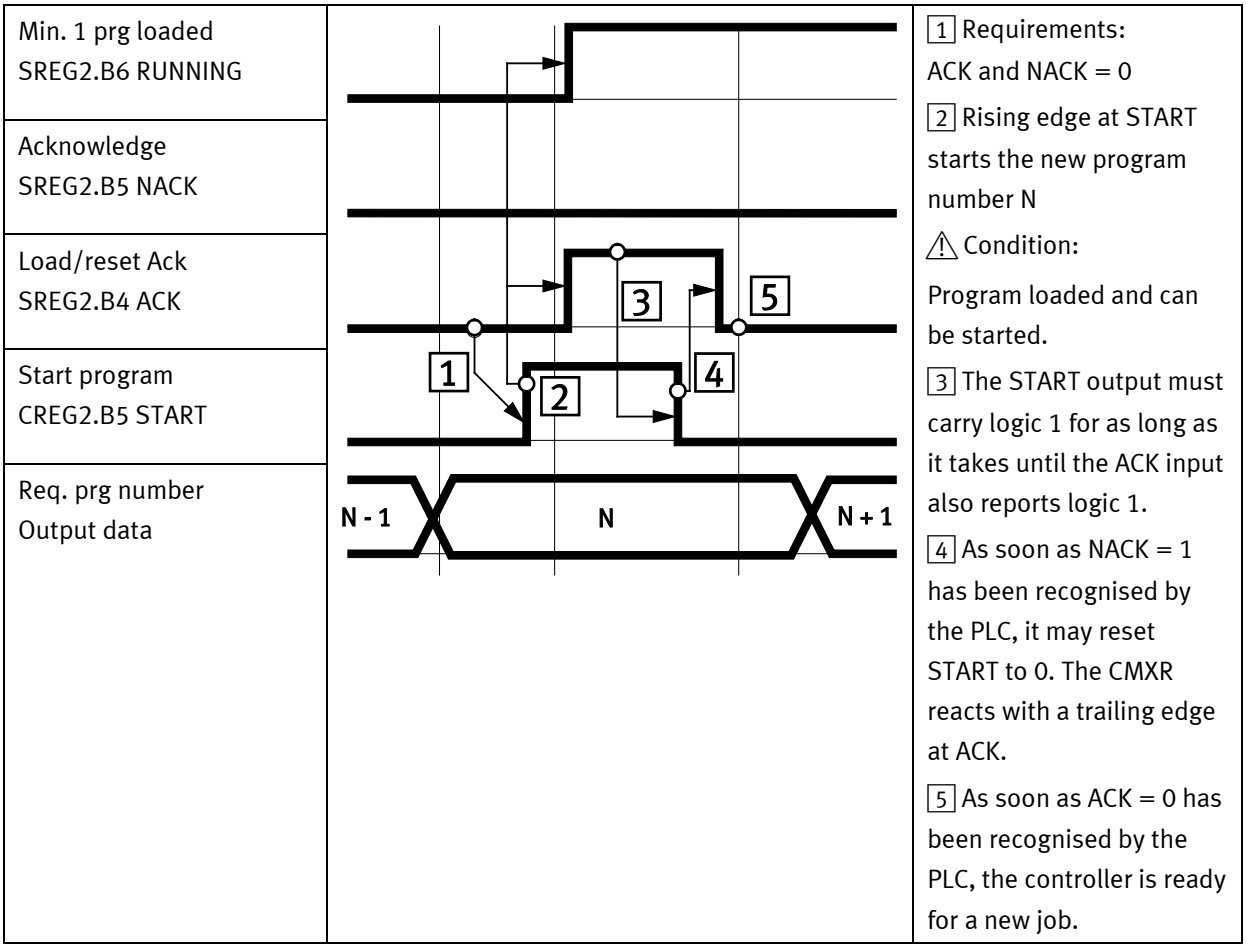
3.4.13 Starting program

The loaded program with the transferred program number is started. Only one program can be started.

Condition	– Master control on PLC interface.
	– The program to be started is loaded.
	– Drives and automatic mode are enabled.
	– No errors are present.
	– Program number 1 ... n is in the program register
	  – MCP1-EA: CREG.PRGNR register (4 bit)
	  – MCP1-PB: PRGNR register (1 byte).

Action	– Set the CREG.START bit (rising edge). – Wait for Acknowledge SREG.ACK or Not Acknowledge SREG.NACK
Reaction	– The programme is started. – In the event of an error, SREG.ERROR is set.

Signal curve



1 Requirements:
ACK and NACK = 0

2 Rising edge at START starts the new program number N

⚠ Condition:
Program loaded and can be started.

3 The START output must carry logic 1 for as long as it takes until the ACK input also reports logic 1.

4 As soon as NACK = 1 has been recognised by the PLC, it may reset START to 0. The CMXR reacts with a trailing edge at ACK.

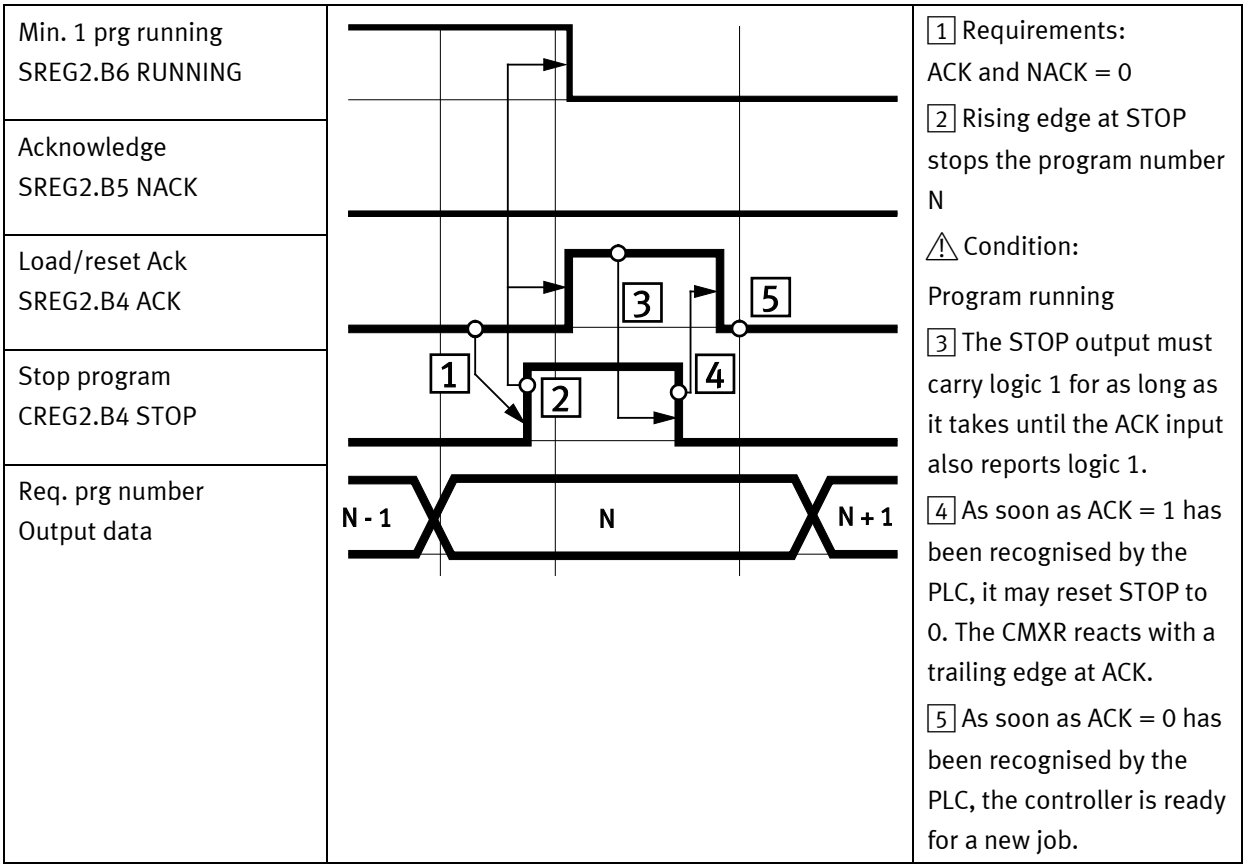
5 As soon as ACK = 0 has been recognised by the PLC, the controller is ready for a new job.

3.4.14 Stopping program

The previously started program is interrupted and ended. As only one program can be started, a program number is not required for stopping.

Condition	–	Master control on PLC interface.
	–	The program to be stopped has started.
Action	–	Set bit CREG.STOP (rising edge).
	–	Wait for Acknowledge SREG.ACK or Not Acknowledge SREG.NACK.
Reaction	–	The program will be stopped.
	–	In the event of an error, SREG.ERROR is set.

Signal curve



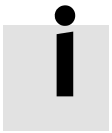
3.4.15 Parallel programs

The PLC interface can be used to start only one program. It is necessary to process parallel programs; in this way, subprograms can be activated from the started main program with the help of the RUN() instruction.

Subprograms are linked to the main program. This means that actions that are executed for the main program also affect the subprograms.

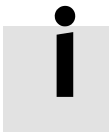
- STOP: also ends all subprograms
- UNLOAD: also unloads all subprograms

3.4.16 Override



The override function is only possible with activation via PROFIBUS (MCP1-PB) and not via digital I/O (MCP1-EA).

Override of the controller can be specified via the COVR control register. The current override is visible in the SOVR status register.



The override does not describe an absolute dynamic value, but only specifies a percentage amount of the maximum dynamic currently permitted.

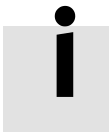
Refer to the system description to see which settings the maximum dynamic depends on (➔ GDCP-CMXR-SY-...).

Automatic mode:

- 0 Override of 0.1 % is set.
- 1 ... 100 Override between 1 and 100 % is set.

Jog operation:

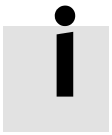
- 0 Jog override of 0.1 increment is set.
- 1 Jog override of 1 increment is set.
- 2 ... 100 Jog override between 2 and 100 % is set.



Deviating values (‹ 0 and › 100) are ignored, and the previously valid override remains unchanged.

Condition	Master control on PLC interface.
Action	Write the new override into the COVR control register.
Reaction	The specified override is accepted and reflected in the SOVR status register.

3.4.17 Exchange of cyclical I/O data



This function is only possible with activation via PROFIBUS (MCP1-PB) and not via digital I/O (MCP1-EA).

In each cycle, 2 bytes of data can be exchanged between the PLC and the multi-axis control system.

The data are transferred into the registers ADATA1 and ADATA2 or received in the registers EDATA1 and EDATA2. The content of these 16 bits can be determined by the user.

The I/O variable values can be accessed from the user program.

The names of the variables are:

- plc_InBool[0-15] for input data
- plc_OutBool[0-15] for output data

Byte order

EDATA1, ADATA1 = Low byte

EDATA2, ADATA2 = High byte

4. PLC interface MCP2

4.1 MCP2-PB

MCP2-PB includes 52-byte control signals and 52-byte status signals and has acyclically extendable commands. The signals from MCP1 and from MCP2 are transferred via PROFIBUS. The signals are exchanged as I/O data packets on PROFIBUS.

Data packet on PROFIBUS

MCP1 is always transferred along with MCP2.

	MCP1-PB		**MCP2-PB**	
	MCP1-EA		**Header**	**Parameter**
Input data	2 bytes	10 bytes	4 bytes	48 bytes
Output data	2 bytes	10 bytes	4 bytes	48 bytes

4.1.1 Control signals

Byte	**1**	**2**	**3**	**4**
Header	CREG	CODE	SUBCODE	INDEX
	Control register	Function code	Subfunction code	Data index

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CREG								EXECUTE
	Unassigned	Unassigned	Unassigned	Unassigned	Unassigned	Unassigned	Unassigned	Valid

4.1.2 Status signals

Byte	**1**	**2**	**3**	**4**
Header	SREG	CODE	SUBCODE	INDEX
	Status register	Function code	Subfunction code	Data index

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SREG	RETVAL						ERROR	DONE
	Return value						Error bit	End

4.2 Function of the PLC interface MCP2

Data can be exchanged acyclically with the multi-axis control system via the MCP2 command interface. The function of the command interface is not dependent on the master control. This means that, even if the robot is controlled via the operator unit, it is possible to exchange data via fieldbus.

The command interface has a width of 52 bytes, which is, in turn, divided into 4-byte headers and 48-byte parameters. The header is needed for the control of the transmission and the parameters contain the transmission data.

MCP2 header

Each MCP2 header consists of 4 bytes of I/O data.

- CREG, SREG Control or status register (1 byte)
- CODE Function code
- SUBCODE Subcode of the function code
- INDEX Index of the function code

Command sequence

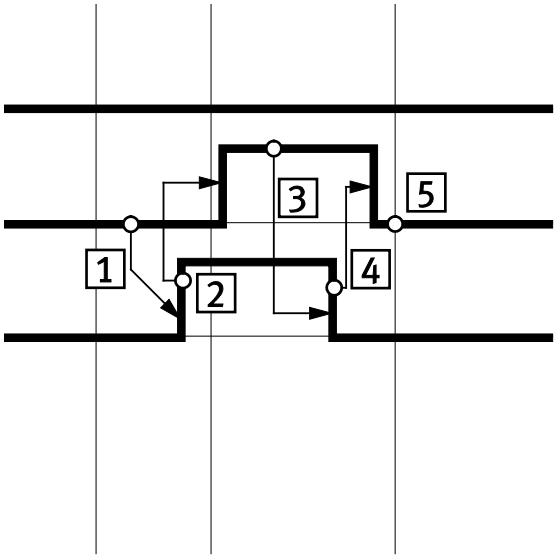
Before a command or job can be executed, the function must be specified in the header. A job always consists of CODE, SUBCODE and INDEX. If one of them is not needed, a 0 must be transferred there.

The command is executed with the EXECUTE bit in the CREG byte. The interface answers in each case with DONE = TRUE. If a transmission error occurs, the controller also replies with ERROR and returns a 6-bit error code RETVAL in the SREG byte.

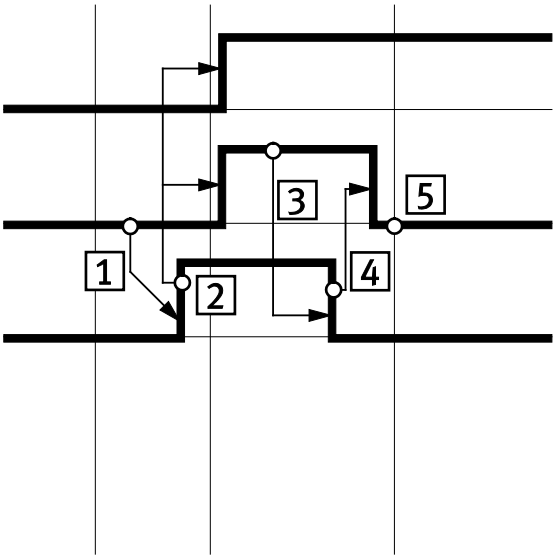
The transferred parameters are only valid if the interface reports back DONE = TRUE. In the event of an error, the error code RETVAL is also only valid if the ERROR and DONE bits both report TRUE.

Issuing signal curve command

Transmission successful

Error SREG.B1 ERROR Command executed SREG.B5 DONE Execute command CREG.B0 EXECUTE		1 Requirements: DONE = FALSE 2 Rising edge at EXECUTE transmits the command present 3 When the CMXR has processed the job, it replies with DONE = TRUE 4 A trailing edge at EXECUTE ends the transmission 5 A job can be executed again as soon as DONE goes back to FALSE.

Transmission not successful

Error SREG.B1 ERROR Command executed SREG.B5 DONE Execute command CREG.B0 EXECUTE		1 Requirements: DONE = FALSE 2 Rising edge at EXECUTE transmits the present (faulty) command 3 When the CMXR has processed the job, it replies with DONE = TRUE and ERROR = TRUE 4 A trailing edge at EXECUTE ends the faulty transmission 5 A job can be executed again as soon as DONE goes back to FALSE.

4.3 Function codes

Access to communication variables

Function	CODE	SUBCODE		INDEX	Parameter
Read communication variable	1	1 2 3 4	CARTPOSREG AXISPOSREG DINTREG CARTRSREG	0 ... 255 0 ... 255 0 ... 255 0 ... 15	Value is placed into PROFIBUS output data range (MCP2 – ADATA), depending on the data type
Write communication variable	2	1 2 3 4	CARTPOSREG AXISPOSREG DINTREG CARTRSREG	0 ... 255 0 ... 255 0 ... 255 0 ... 15	Value is read by PROFIBUS-input data range (MCP2 – EDATA), depending on the data type
Teach communication variable	3	1 2	CARTPOSREG AXISPOSREG	0 ... 255 0 ... 255	X

Access to actual values

Function	CODE	SUBCODE		INDEX	Parameter
Read setpoint position	8	1 2 3	Axes World Object	x	Setpoint position Format: position variable (➔ 5 Communication variables)
Read actual position	9	1 2 3	Axes World Object	x	Actual position Format: position variable (➔ 5 Communication variables)

4.3.1 Return values

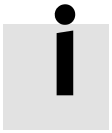
The following codes are located in the RETVAL register after a command has been executed.

RETVAL	Significance
0	Command executed correctly
1	Incorrect CODE was transmitted
2	Incorrect SUBCODE was transmitted
3	Incorrect INDEX was transmitted

5. Communication variables

The robot operating system provides globally visible variables for all robot programs for the free exchange of data with an external control system:
- Position variables (Cartesian positions and axis positions)
- Reference system variables
- Word variables

These variables can be described and read externally via every installed PLC interface (not MCP1-EA).



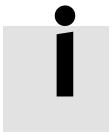
In the robot program, the communication variables are directly visible with predefined names.

5.1 Position variables

5.1.1 Access from the robot program

The interface includes 256 Cartesian and 256 axis positions respectively.

These are predefined as follows:
- plc_CartPos [0 ... 255]: CARTPOS
- plc_AxisPos [0 ... 255]: AXISPOS



The data types CARTPOS and AXISPOS are standard data types for representing positions.

Example: Use in the robot program
- Ptp (plc_CartPos [0])
- Lin (plc_AxisPos [0])

5.1.2 Interface format

In the PLC interface, the position and angle values are transferred in integer form. The conversion to flow point form is done via a configurable factor (posResolution).

Format of Cartesian position in the interface (CARTPOSREG type)

Variable	Significance
x, y, z : DINT	Position of the tool
a, b, c : DINT	Orientation of the tool
aux1 ... 3 : DINT	Position of the auxiliary axes

Format of axis position in the interface (AXISPOSREG type)

Variable	Significance
a1 ... 6 : DINT	Position of the basic axes
aux1 ... 3 : DINT	Position of the auxiliary axes

5.2 Reference system variables

5.2.1 Access from the robot program

The PLC interface includes 16 reference systems. These are predefined as follows:
- plc_RefSys [0 ... 15] : REFSYSDATA

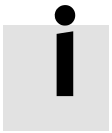
Example: Use in the robot program
- SetRefSys (plc_RefSys [0])

5.2.2 Interface format

In the PLC interface, the position and angle values are transferred in integer form. The conversion is just the same as with the position values.

Format of the Cartesian position in the REFSYSDATA interface

Variable	Significance
baseRS : DINT	Index reference system, –1 = WORLD
x, y, z : DINT	Position of the tool
a, b, c : DINT	Orientation of the tool



baseRS ≠ index current variable

A reference “to itself” is not allowed.

5.3 Word variables

5.3.1 Access from the robot program

The PLC interface includes 255 integer variables of type DINT (4 bytes).

These are predefined as follows:

– plc_Dint [0 ... 255] : DINT

Example: Use in the robot program

– plc_Dint [0] := 100

5.3.2 Interface format

The DINT variables are not subjected to a conversion and are transmitted 1:1 into the multi-axis control system or PLC.

Phone:
+49 711 347 0

Fax:
+49 711 347 2144

E-mail:
service_international@festo.com

Internet:
www.festo.com

Original: de
Version: 1501b