Rexroth
Bosch Group

Rexroth MTC 200/ISP 200/ MTA 200/TRANS 200 Function Interface 08VRS

R911299217
Edition 01

Reference

Title	Rexroth MTC 200/ISP 200/ MTA 200/TRANS 200 Function Interface 08VRS
Type of Documentation	Reference
Document Typecode	DOK-CONTRL-FUN*COM*V08-AW01-EN-P
Internal File Reference	Document Number 120-0401-B313-01/EN
Purpose of Documentation	This documentation describes the structure and availability of function interface commands subdevided in device groups.

Record of Revisions

Description	Release Date	Notes
120-0401-B313-01/EN	03.2004	Valid from version 23

Copyright

Validity

Published by

Bosch Rexroth AG
Bgm.-Dr.-Nebel-Str. 2 • D-97816 Lohr a. Main

Telephone +49 (0)93 52/40-0 • Tx 68 94 21 • Fax +49 (0)93 52/40-48 85

http://www.boschrexroth.com/

Dept. BRC/ESS2 (FW)

Dept. BRC/ESM6 (DiHa)

Note

This document has been printed on chlorine-free bleached paper.

Contents

9 FI Commands - MWSX Device Group (ISP 200) 9-1

10 FI Commands MWAX Device Group (MTA 200) 10-1

11 FI Commands - MSYX Device Group (SYNAX200) 11-1

12 FI Commands - MWYX Device Group (SYNAXISP200) 12-1

1 Introduction

1.1 Division of Documentation

The “Function Interface V08” documentation was divided into two documents.

- Function Interface V08 – Application manual,
  DOK-CONTRL-FUN*INT*V08-AW01-EN-P, R911295419
- Function Interface V08 – Reference,
  DOK-CONTRL-FUN*COM*V08-AW01-EN-P, R911299217

1.2 Application Manual

The document describes the structure of function interface, explains the installation and gives notes for programming. Further it contains a list of the new added commands. Listing the error codes completes this documentation part.

1.3 Reference (of this Document)

In this document, the structure and availability of function interface command are described. Divided in device groups, the individual function interface commands are described in detail with examples.

2 Construction and Availabilty of the FI Commands

2.1 Elements of the FI Command

The function interface commands are subdivided into the following elements:

- Identifier,
- Selector and
- Data code.

Identifier

The identifier is composed of 11 ASCII characters for the device address, separators, interface designator, command, function code and function descriptor.

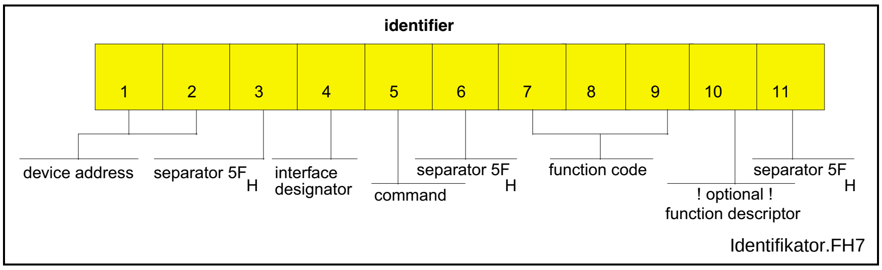


Fig. 2-1: Identifier

device address

The device address corresponds to the system address within the Bosch Rexroth GUI. This means, for example, that device 00 corresponds to system 0. Please observe, however, that the Bosch Rexroth GUI always requires a device 00. The addresses are listed specific to the device group in the following table as well as in the chapter "Function Interface Commands".

Address	**Group**	**Affiliated device types**
[xx]	MPCX	PCs
[00...63]	MWCX	MTC200-P-G2, MTC200-R-G2, MTVNC
[00]	MSCX	SERCANS-A, SERCANS-P
[00...63]	MVMX	VM-P, VM-R
[00...63]	MWMX	VMISP200-P-G2, VMISP200-R-G2
[00...63]	MWSX	ISP200-P-G2, ISP200-R-G2
[00...63]	MWAX	MTA200-P (MTA 200-controller)
[00...63]	MSYX	SYNAX200-P, SYNAX200-R
[00...63]	MWYX	SYNAXISP200-P-G2, SYNAXISP200-R-G2

Separator

The separator “_” separates the individual elements and is therefore a fixed component of the identifier.

Interface designator

Management process
Controller (logic process and communication process)

All data access via the interface identifier “M” are managed by the management process so as to ensure, for instance, that a user program can access the data from a controller via function calls. When the function call “BR_ASM5” (active system fault messages) is used, among other things the message number is fetched from the controller and the affiliated text is taken from the corresponding message file on the hard disk. When a file is opened, e.g., for editing with an editor, the management process ensures that a different user program cannot open the same file again.

Data access by way of the interface identifier “C” is managed by the logic process and by the communication process, thus enabling access to data of the relevant device groups (MWCX and MWSX, etc.).

Read and Write Commands

W = Single **W**rite	(Writing)
R = Single **R**ead	(Reading)
C = **C**yclic Read	(Cyclic reading)
B = **B**reak Cyclic Read	(Interrupt cyclic reading)

Read command “**R**”

A read request is passed on to the function interface with DataTransfer (“00_CR_PPS_1_0_15_10”). On returning from this function, the user program is notified of a result buffer (*acBuffer) of a specific length (*lLen). In the result buffer, the requested data is made available in the requested data code. In the event of an error, the “DataTransfer” routine is ended with an error (return value <>0) and it may be necessary to branch to an error routine. If the reply consists of several partial results (e.g., X1 125.4567 [mm]), the result must be interpreted with the “ReadGroupItem” routine (see Chapter 4, “Programming”)

Write command “**W**”

A new value for the specified PLC variable is passed on to the function interface in the specified buffer (*pcValue) with DataTransfer (“00_CW_PVS_TEST”). On returning from the function, and with a return value “0”, this function has been executed successfully. In the event of an error (return value <>0), it may be necessary to branch to an error routine. The write command may have to be repeated in the event of an error.

Function code

The three letters of the function code provide information about the data to be accessed. The identifier is encoded in the form of the data type designation. After the three letters, occasionally a function descriptor for the respective function code may be necessary. This is described explicitly in the list of the access functions.

Example	Access without function descriptor
CR_PPS_1_0_1_2	Read from the NC memory A, in the NC process 0, from the parts program 1 the NC block N0002.

Example:	Access with function descriptor
CR_NPA2_S00.00.022_S00.00.025	Read system parameters lines 22 to 25.

The function descriptor is necessary whenever several combinations can be read via the data code (e.g., 1 line, 1 element or several lines). In this case, the selectors would be assigned different meanings for the respective requests. This is why this access is specified in greater detail with a function descriptor.

Selector

Example

Cyclic reading of the current axis speed of the 1st axis of device address 00.

The selector consists of a minimum of 1 character and a maximum of 17 characters. The selector is encoded in the form of numeric numbers that are separated by a separator ($5F_H$, 95_{Dec}). The selector directly depends on the addressed data type. Addressing is described in the sections dealing with the individual function calls.

Example 1 (cyclic reading)

Cyclic reading of the current axis speed of the 1st axis, of device address 00.

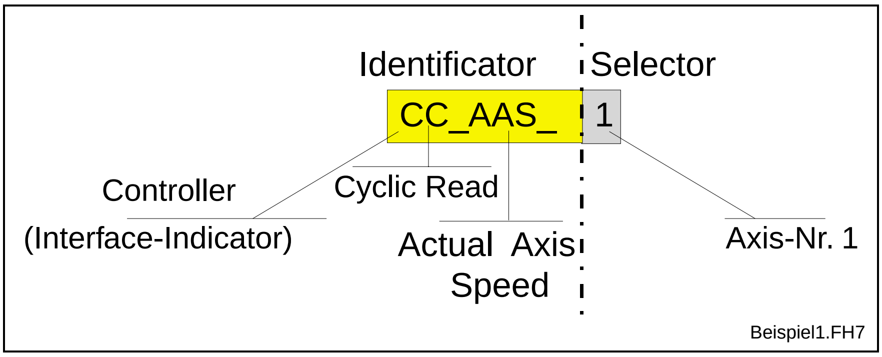


Fig. 2-2: Example 1: Cyclic reading in ASCII code

Example 2 (cyclic reading)

Cyclic reading of the current feedrate in the NC process 3 of device address 02.

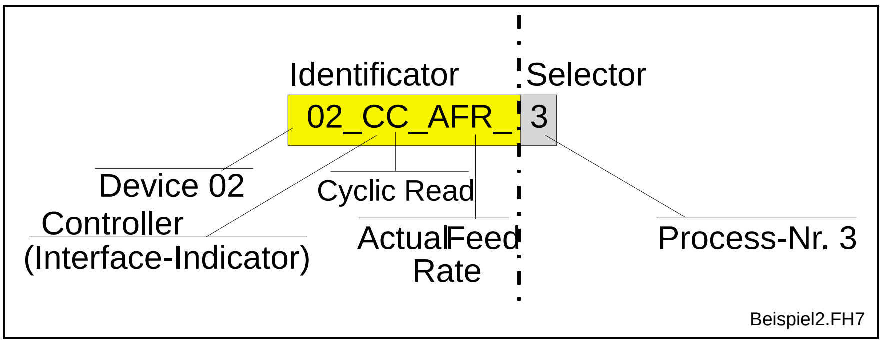


Fig. 2-3: Example 2: Cyclic reading in ASCII code

Data Code

The data code is an optional identifier in the function call. It can be specified after the selector, separated by a slash “/”. The specified data code defines the code of the data to be read; in the case of write functions, it defines the encoding of the request strings and the response in the result buffer.

The following coding types are supported:

1 = ASCII Preset !

2 = Binary

3 = ANSI

4 = Unicode (not yet implemented).

Note: The data in the control is generally filed in ASCII.

Example 3 (cyclic reading)

Cyclic reading of the current spindle speed of the 2nd spindle in the NC process 1, of the device address 01 in the “Binary” data code.

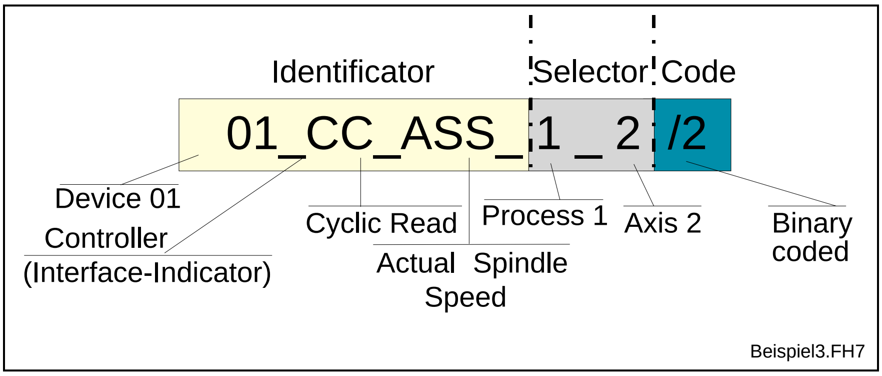


Fig. 2-4: Example 3: Cyclic reading in binary code

2.2 Data Tables

The most frequently used parameters and their value ranges for the various device groups are listed in the following data tables.

General Parameters for the MWCX Device Groups

Parameters	**Value Range**
Axis number	1...32
NC memory	1=A, 2=B
NC block No.	0...9999
NC program number	0...99
NC packet	1...99
Zero point database	0...9
Spindle number	S1, S2, S3
NC process number	0...6
Mechanism number	0...31
Drive address	0...254
Tool number	0...9999999
Duplo No.	1...9999
Data block	0 = basic tool data 1...9 = tool edge data
Data element	1...28 for basic tool data 1...40 for tool edge data
Memory	M = magazine/turret S = spindle G = gripper X = index data
Location	1...999 for M 1...4 for S,G 0...16770215 for X

Meanings of the Axes for the MWCX Device Group

Code	Axis meaning	Axis type
0	X axis	Main axis
1	Y axis	Main axis
2	Z axis	Main axis
3	U axis	Secondary axis
4	V axis	Secondary axis
5	W axis	Secondary axis
6	A axis	Rotary axis
7	B axis	Rotary axis
8	C axis	Rotary axis
9	S1 axis	spindle
10	S2 axis	spindle
11	S3 axis	spindle
20	Turret axis	Special type

Axis Types for the MWCX Device Group

No.	Axis types	Comment
0_H	AXIS_NOT_DEFINED	Axis not defined
1_H	ANALOG_LINEAR_AXIS	Analog linear axis
2_H	ANALOG_ROTARY_AXIS	Analog rotary axis
3_H	ANALOG_MAIN_SPINDLE	Analog spindle
4_H	ANALOG_COMB_TURRET_AXIS	Analog turret axis
5_H	C_AXIS	C axis
80_H	DYNAMIC_AXIS	Dynamically assignable axis
81_H	DIGITAL_LINEAR_AXIS	Linear axis
82_H	DIGITAL_ROTARY_AXIS	Rotary axis
83_H	DIGITAL_MAIN_SPINDLE	Spindle
84_H	DIGITAL_COM_TURRET_AXIS	Digital turret axis
85_H	DIGITAL_C_AXIS	Digital C axis
87_H	DIGITAL_SERCOS_E_A	Digital Sercos I/O

Base Units

Base unit	Measurement System			
	Linear in mm	Linear in inch	Rotatory in units	Specific to main spindle
velocity	mm/min	inch/min	units/min	1/min
Feed constant	mm	inch	units	--
acceleration	mm/s^2	$inch/s^2$	$units/s^2$	rad/s^2
Distance	mm	inch	units	deg
Speed	rpm	rpm	rpm	rpm
Cutting speed	m/min	inch/min	units/min	--

2.3 Overview of FI Commands

The following table presents an overview of the available FI commands, arranged according to device groups.

Note: A detailed description is contained in the following Chapters “Function Interface Commands”.

Overview of the MPCX Device Group

Com.	**Description**
ADA1	**A**ctive **D**evice **A**ddress
BCI1	**B**us **C**onfiguration **I**nfo
BCI2	**B**us **C**onfiguration **I**nfo
BCI3	**B**us **C**onfiguration **I**nfo
BFJ1	**B**reak-**F**unction-Interface **J**obs
BFJ2	**B**reak-**F**unction-Interface **J**obs
CAM1	**C**lose **A**pplication **M**anager
CCP1	**C**ell **C**onfiguration **P**arameter
CCP2	**C**ell **C**onfiguration **P**arameter
CCP3	**C**ell **C**onfiguration **P**arameter
CCP4	**C**ell **C**onfiguration **P**arameter
CCP5	**C**ell **C**onfiguration **P**arameter
CDA1	**C**hange **D**evice **A**ddress
CEI1	**C**ommunication **E**rror **I**nfo
CFL1	**C**ontrol of **F**unctioninterface **L**ogin
CPR1	**C**reate **PR**ocess
CPR2	**C**reate **PR**ocess
CPR3	**C**reate **PR**ocess
CWL1	**C**urrent **W**indow **L**ist
DCN1	**D**ialog **C**o**N**trol Listbox
DFJ1	**D**elete **F**unction Interface **J**ob
DFJ2	**D**elete **F**unction Interface **J**ob
DFS1	**D**elete I**F** Command **S**tack
DIF1	**D**evice **I**n**F**ormation
DPR1	**D**elete **PR**ocess
DPR2	**D**elete **PR**ocess
ERI1	**ER**ror **I**nformation
FCD1	**F**ile **C**ommand**D**
FCP1	**F**ar Device **C**onfiguration **P**arameter
FCP2	**F**ar Device **C**onfiguration **P**arameter
FCP3	**F**ar Device **C**onfiguration **P**arameter
FDC1	**F**ar **D**evice **C**onfiguration
FIT1	**F**urther **I**nfo **T**ext

FPC1	**F**ar **P**C **C**onfiguration
GDB1	**G**lobal **D**ata **B**uffer
ICA1	**I**nitialisation **C**ommunication **A**ddress
IFJ1	**I**nformation about **F**unction Interface **J**obs
IFJ2	**I**nformation about **F**unction Interface **J**obs
IFS1	**IF** Command **S**tack Info
LDT1	PC **L**ocal **D**ate **T**ime
LNG	Active **L**a**NG**uage
MSG	**M**e**S**sa**G**e
NST1	**N**T-**S**hu**T**-Down
NST2	**N**T-**S**hu**T**-Down
PAF1	**PA**rameter **F**ile Converted
PHD1	**Ph**ysical **D**irectory
POB1	**PO**rt **B**yte Access
POW1	**PO**rt **W**ord Access
RPR1	**R**eady **PR**ocess
SDM1	**S**hut**D**own **M**anager
SDM2	**S**hut**D**own **M**anager
SDP1	**S**tart **D**evice **P**olling
SFW1	**S**et **F**ocus to **W**indow
SFW2	**S**et **F**ocus to **W**indow
SID1	**S**oftware **I**nstallation **D**ata
SSM1	**S**et **S**ys-**M**essage
SSM2	**S**et **S**ys-**M**essage

Fig. 2-5: Overview of the MPCX device group

Overview of the MTCX Device Group

Com.	**Description**
AAC1	**A**ctual **AC**celeration
AAD	**A**ctive **A**ngle **D**imension
AAS1	**A**ctual **A**xis **S**peed
AAS2	**A**ctual **A**xis **S**peed
ABI	**A**ctual NC-**B**lock **I**nformation
ACS	**A**ctual **C**utting **S**peed
ADN1	**A**ctive **D**-Correction **N**umber
AEM	**A**ctive **E**vent **M**onitoring
AEN	**A**ctive **E**dge-**N**umber
AFO1	**A**ctive **F**eedrate **O**verride
AFR	**A**ctive **F**eed**R**ate
AGF	**A**ctive **G**-**F**unction
AMF	**A**ctive **M**-**F**unction
AMM1	**A**ctive **M**echanism **M**essage
AMM2	**A**ctive **M**echanism **M**essage
AMM3	**A**ctive **M**echanism **M**essage
AMM4	**A**ctive **M**echanism **M**essage
AMM5	**A**ctive **M**echanism **M**essage
ANM	**A**ctive **N**C **M**emory Size
API1	**A**ctual **P**arameter **I**ndex
API2	**A**ctual **P**arameter **I**ndex
APM	**A**ctive **P**art-Program **M**essage
APN	**A**ctive **P**art-Program Message Number
APO1	**A**ctual Machine **PO**sition
APO2	**A**ctual Machine **PO**sition
APP	**A**ctive **P**art-**P**rogram number
ARF	**A**xis **R**eference **F**lags
ARO1	**A**ctual **R**apid **O**verride
ART	**A**xis**R**eference**T**able
ART	**A**xis**R**eference**T**able
ASD	**A**ctual **S**pindel **D**ata
ASF	**A**ctual **S**pindle **F**or Process
ASG	**A**ctual **S**pindle **G**ear
ASM1	**A**ctive **S**ystem-Fault **M**essage
ASM2	**A**ctive **S**ystem-Fault **M**essage
ASM3	**A**ctive **S**ystem-Fault **M**essage
ASM4	**A**ctive **S**ystem-Fault **M**essage
ASM5	**A**ctive **S**ystem-Fault **M**essage
ASN	**A**ctual **S**equence **N**umber

ASO1	**A**ctual **S**pindle **O**verride
ASS	**A**ctual **S**pindle **S**peed
ATN	**A**ctive **T**ool-**N**umber
ATP1	**A**ctual **T**ool **P**lace Information
ATP2	**A**ctual **T**ool **P**lace Information
ATP3	**A**ctual **T**ool **P**lace Information
ATR	**A**ctual **T**ooldata **R**ecord
ATU	**A**ctual **T**ooldata **U**pdate
AZB1	**A**ctive **Z**ero Offset **B**ank
CCA1	N**C**-**C**ycle **A**ccess
CNI1	**C**urrent **N**C **I**nformation
CNI2	**C**urrent **N**C **I**nformation
CPI1	**C**urrent **P**rocess **I**nformation
CPO1	**C**ommand **PO**sition (SOLL)
CPO2	**C**ommand **PO**sition by log.AxisNr
CRT	**C**ontrol **R**ese**T**
DAC1	**D**evice **A**xis **C**onfiguration Parameter
DAC2	**D**evice **A**xis **C**onfiguration Parameter
DCA1	NC-**D**-**C**orrection **A**ccess
DCD1	**D**-**C**orrection **D**ata
DCP1	**D**evice **C**onfiguration **P**arameter
DCP2	**D**evice **C**onfiguration **P**arameter
DCR1	**D**-**C**orrection **R**ecord
DCT1	**D**evice **C**ommunication **T**imeout
DCT2	**D**evice **C**ommunication **T**imeout
DEM1	**D**elete FI **E**xclusive **M**ode
DIF1	**D**evice **I**n**F**ormation
DIF2	**D**evice **I**n**F**ormation
DIF3	**D**evice **I**n**F**ormation
DIS1	**D**ata **I**dentifikation **S**tring Parameter
DIS3	**D**ata **I**dentifikation **S**tring NC Packet
DIS4	**D**ata **I**dentifikation **S**tring Tool List
DIS5	**D**ata **I**dentifikation **S**tring Machine
DIS6	**D**ata **I**dentifikation **S**tring NC Program
DPN	**D**elete **P**rogramm **NC**
DPP	**D**elete **P**rogram **P**ackage
DSI1	**D**evice **S**tatus **I**nformation
DSI2	**D**evice **S**tatus **I**nformation
DTC1	**D**evice **T**ool Management **C**onfiguration
DTC2	**D**evice **T**ool Management **C**onfiguration
DTG1	**D**istance **T**o **G**o

DTG2	**Di**stance **T**o **G**o by log. AxisNr
DTY1	**D**evice **TY**pe
DWD1	**D**iagnosis **W**indow **D**ata
DWD2	**D**iagnosis **W**indow **D**ata
ECI1	**E**rror **C**omponent **I**nformation
EDE1	**E**xisting **D**iagnosis **E**rror
EDE2	**E**xisting **D**iagnosis **E**rror
EDW1	**E**xisting **D**iagnosis **W**indow
EDW2	**E**xisting **D**iagnosis **W**indow
EDW3	**E**xisting **D**iagnosis **W**indow
END1	**E**xisting **N**C **D**iagnosis
END2	**E**xisting **N**C **D**iagnosis
EPD1	**E**xisting **P**LC **D**iagnosis
EPD2	**E**xisting **P**LC **D**iagnosis
EPD3	**E**xisting **P**LC **D**iagnosis
EPO1	ProgrammEd **PO**sition (END)
EPO2	ProgrammEd **PO**sition (END)
EPT1	**E**xisting **P**roVi **T**ypes
EST1	**E**rror **ST**ate
EXD1	**EX**ecution **D**isplay
EXD2	**EX**ecution **D**isplay
GPC1	**G**lobal **P**rocess **C**onfiguration
GPC2	**G**lobal **P**rocess **C**onfiguration
GPP1	**G**lobal **P**rocess **P**arameter
GPP2	**G**lobal **P**rocess **P**arameter
IPP	**I**nsert NC-**P**rogram **P**ackage
MAP1	**M**odule **A**ssign of **P**rocess
MCD1	**M**odule **C**onfigration: **D**evice Information
MCM1	**M**odule **C**onfigration: **M**odule Information
MCP1	**M**odule **C**onfigration: **P**rocess Information
MCS1	**M**odule **C**onfigration: **S**FC- Information
MDA1	**M**achine **D**ata **A**ccess
MDA2	**M**achine **D**ata **A**ccess
MDA3	**M**achine **D**ata **A**ccess
MDA4	**M**achine **D**ata **A**ccess
MDI	**M**anual **D**ata **I**nput
MDS1	**M**achine **D**ata **S**ingle
MFD1	**M**essage **F**iles **D**ownload
MFO1	**M**aximal **F**eedrate **O**verride
MFR	**M**aximal **F**eed**R**ate
MKS	**M**achine **K**ey **S**tatus

MKT1	**M**achine **K**ey **T**able
MRO1	**M**aximal **R**apid **O**verride
MSG	**M**e**S**sa**G**e
MSO1	**M**aximal **S**pindle **O**verride
MSS	**M**aximal **S**pindle **S**peed
MTC	**MT**-**C**NC Slot Software Version
MTC1	**MT**-**C**NC Slot Software Version
MTD	**M**achine **T**able **D**ata
NCA1	**NC**-Program **A**ccess
NCA3	**NC**-Program **A**ccess
NCM1	**NC M**essages
NCM2	**NC M**essages
NEA1	**N**C-**E**vent **A**ccess
NEV	**N**C **Ev**ent
NMM	**N**C **M**e**M**ory selection
NPA1	**N**C **PA**rameter
NPA2	**N**C **PA**rameter
NPA3	**N**C **PA**rameter
NPA4	**N**C **PA**rameter
NPA5	**N**C **PA**rameter
NPC1	**N**C-**P**ackage **C**ompiling
NPD1	**N**C-**P**ackage **D**ownload
NPI	**N**C-**P**ackage D**I**rectory
NPS	**N**C **P**rogram **S**election
NTN	**N**ext **T**ool-**N**umber
NUA1	**N**C-Offset Data **A**ccess
NVA1	**N**C-**V**ariable **A**ccess
NVS	**N**C **V**ariable **S**ingle
OPD1	**O**ptimal **P**osition **D**istance by Axis sign.
OPD2	**O**ptimal **P**osition **D**istance by phys. AxisNr
PAA1	**PA**rameter **A**ccess
PAA2	**PA**rameter **A**ccess
PAC1	**P**rocess **A**xis **C**onfiguration Parameter
PAC2	**P**rocess **A**xis **C**onfiguration Parameter
PAD1	**PA**rameter **D**eactivate
PAS1	**PA**rameter **S**et Active
PDT	**P**arameter **D**efinition **T**able
PFR	**P**rogrammed **F**eed**R**ate
PPA	**P**art **P**rogram **A**ctive
PPD	**P**art-**P**rogram **D**irectory
PPN	**P**art-**P**rogram **NC**

PPP	**P**art **P**rogram **P**ackage
PPS	**P**art-**P**rogram-**S**equence
PSS	**P**rogrammed **S**pindle **S**peed
PTC1	**P**rocess **T**ool Management **C**onfiguration
PTC2	**P**rocess **T**ool Management **C**onfiguration
PTC3	**P**rocess **T**ool Management **C**onfiguration
PVM1	**P**ro**V**i **M**essages
PVM2	**P**ro**V**i **M**essages
PVM3	**P**ro**V**i **M**essages
PVM4	**P**ro**V**i **M**essages
REP1	**REP**ositioning Data
REP2	**REP**ositioning Data
RPO	**R**elative Axis **PO**sition
RPO	**R**elative Axis **PO**sition
SDD1	**S**fc **D**iagnosis **D**ata
SDD2	**S**fc **D**iagnosis **D**ata
SDD3	**S**fc **D**iagnosis **D**ata
SDD4	**S**fc **D**iagnosis **D**ata
SDD5	**S**fc **D**iagnosis **D**ata
SDD6	**S**fc **D**iagnosis **D**ata
SDS1	**S**et **D**evice **S**tatus
SDS2	**S**et **D**evice **S**tatus
SEM1	**S**et FI **E**xclusive **M**ode
SFD1	**SF**c **D**ata
SFD2	**SF**c **D**ata
SFD3	**SF**c **D**ata
SFE1	**SF**c **E**rror
SFE2	**SF**c **E**rror
SFM1	**SF**c **M**ode
SID1	**S**oftware **I**nstallation **D**ata
SLA1	Actual **S**ervo **LA**g
SLA2	Actual **S**ervo **LA**g
SLI	PLC (**S**PS) **L**ong **I**dentification
SPA1	**S**ERCOS **PA**rameter
SPH1	**S**ERCOS **PH**ase
SPP	**S**elected **P**art **P**rogram Number
TDA1	**T**ool **DA**ta
TDA2	**T**ool **DA**ta
TDD	**T**ool **D**ata **D**ownload
TDE	**T**ool List **D**ownload **E**scape
TDF	**T**ool List **D**ownload **F**inish

TDI	**T**ool List **D**ownload **I**nitialize
TDR1	**T**ool **D**ata **R**ecord of Place
TDR2	**T**ool **D**ata **R**ecord
TIF	**T**ool **I**nsert **F**inish
TII	**T**ool **I**nsert **I**nitiated
TLB1	**T**oo**L** **B**asicdata List
TLB2	**T**oo**L** **B**asicdata List
TLD1	**T**oo**L** **D**ata of Place
TLD2	**T**oo**L** **D**ata of Tool
TLD3	**T**oo**L** **D**ata of Place
TLD4	**T**oo**L** **D**ata of Tool
TLE1	**T**oo**L** **E**dgedata List
TLE2	**T**oo**L** **E**dgedata List
TMV	**T**ool **M**o**V**e
TPI1	**T**ool **P**osition **I**nformation
TPI2	**T**ool **P**osition **I**nformation
TQE1	Actual **T**or**Q**u**E**
TQE2	Actual **T**or**Q**u**E**
TRM	**T**ool **R**e**M**ove
TRS	**T**ool **R**e**S**et
ZOD	**Z**ero **O**ffset **D**ata
ZOD1	**Z**ero **O**ffset **D**ata
ZOD2	**Z**ero **O**ffset Data

Fig. 2-6: Overview of the MTCX device group

Overview of the MWCX Device Group

Com.	Description
AAC1	**A**ctual **AC**celeration
AAD	**A**ctive **A**ngle **D**imension
AAS1	**A**ctual **A**xis **S**peed
AAS2	**A**ctual **A**xis **S**peed
ABI	**A**ctual NC-**B**lock **I**nformation
ACS	**A**ctual **C**utting **S**peed
ADN1	**A**ctive **D**-Correction **N**umber
ADW	**A**ctive **D**iagnosis **W**indow
AEM	**A**ctive **E**vent **M**onitoring
AEN	**A**ctive **E**dge-**N**umber
AFO1	**A**ctive **F**eedrate **O**verride
AFR	**A**ctive **F**eed**R**ate
AGF	**A**ctive **G**-**F**unction
AMF	**A**ctive **M**-**F**unction
AMM1	**A**ctive **M**echanism **M**essage
AMM2	**A**ctive **M**echanism **M**essage
AMM3	**A**ctive **M**echanism **M**essage
AMM4	**A**ctive **M**echanism **M**essage
AMM5	**A**ctive **M**echanism **M**essage
ANM	**A**ctive **N**C **M**emory Size
API1	**A**ctual **P**arameter **I**ndex
API2	**A**ctual **P**arameter **I**ndex
APM	**A**ctive **P**art-Program **M**essage
APN	**A**ctive **P**art-Program Message Number
APO1	**A**ctual Machine **PO**sition
APO2	**A**ctual Machine **PO**sition
APP	**A**ctive **P**art-**P**rogram number
ARF	**A**xis **R**eference **F**lags
ARO1	**A**ctual **R**apid **O**verride
ART	**A**xis**R**eference**T**able
ART	**A**xis**R**eference**T**able
ASD	**A**ctual **S**pindel **D**ata
ASE	**A**ctual **S**ystem **E**rror
ASF	**A**ctual **S**pindle **F**or Process
ASG	**A**ctual **S**pindle **G**ear
ASM1	**A**ctive **S**ystem-Fault **M**essage
ASM2	**A**ctive **S**ystem-Fault **M**essage
ASM3	**A**ctive **S**ystem-Fault **M**essage
ASM4	**A**ctive **S**ystem-Fault **M**essage

ASM5	**A**ctive **S**ystem-Fault **M**essage
ASN	**A**ctual **S**equence **N**umber
ASO1	**A**ctual **S**pindle **O**verride
ASS	**A**ctual **S**pindle **S**peed
ATN	**A**ctive **T**ool-**N**umber
ATP1	**A**ctual **T**ool **P**lace Information
ATP2	**A**ctual **T**ool **P**lace Information
ATP3	**A**ctual **T**ool **P**lace Information
ATR	**A**ctual **T**ooldata **R**ecord
ATU	**A**ctual **T**ooldata **U**pdate
AZB1	**A**ctive **Z**ero Offset **B**ank
BCD1	**B**us **C**onfiguration **D**ata
BCD2	**B**us **C**onfiguration **D**ata
BCD3	**B**us **C**onfiguration **D**ata
BCD4	**B**us **C**onfiguration **D**ata
BCD5	**B**us **C**onfiguration **D**ata
BCD6	**B**us **C**onfiguration **D**ata
BCD7	**B**us **C**onfiguration **D**ata
CCA1	N**C**-**C**ycle **A**ccess
CMD1	**C**reate **M**I Import **D**ata
CMD2	**C**reate **M**I Import **D**ata
CMD3	**C**reate **M**I Import **D**ata
CMD4	**C**reate **M**I Import **D**ata
CNI1	**C**urrent **N**C **I**nformation
CNI2	**C**urrent **N**C **I**nformation
CPI1	**C**urrent **P**rocess **I**nformation
CPO1	**C**ommand **PO**sition (SOLL)
CPO2	**C**ommand **PO**sition by log.AxisNr
CRT	**C**ontrol **R**ese**T**
CVA1	**C**heck **V**irtual **A**xis
DAC1	**D**evice **A**xis **C**onfiguration Parameter
DAC2	**D**evice **A**xis **C**onfiguration Parameter
DCA1	NC-**D**-**C**orrection **A**ccess
DCD1	**D**-**C**orrection **D**ata
DCI1	**D**evice **C**omponent **I**nformation
DCP1	**D**evice **C**onfiguration **P**arameter
DCP2	**D**evice **C**onfiguration **P**arameter
DCR1	**D**-**C**orrection **R**ecord
DCT1	**D**evice **C**ommunication **T**imeout
DCT2	**D**evice **C**ommunication **T**imeout
DEM1	**D**elete FI **E**xclusive **M**ode

DIF1	**D**evice **I**n**F**ormation
DIF2	**D**evice **I**n**F**ormation
DIF3	**D**evice **I**n**F**ormation
DIS1	**D**ata **I**dentifikation **S**tring Parameter
DIS2	**D**ata **I**dentifikation **S**tring PLC Program
DIS3	**D**ata **I**dentifikation **S**tring NC Packet
DIS4	**D**ata **I**dentifikation **S**tring Tool List
DIS5	**D**ata **I**dentifikation **S**tring Machine
DIS6	**D**ata **I**dentifikation **S**tring NC Program
DPA1	**D**rive **P**arameter **A**ccess
DPA2	**D**rive **P**arameter **A**ccess
DPA3	**D**rive **P**arameter **A**ccess
DPA4	**D**rive **P**arameter **A**ccess
DPN	**D**elete **P**rogramm **N**C
DPP	**D**elete **P**rogram **P**ackage
DSI1	**D**evice **S**tatus **I**nformation
DSI2	**D**evice **S**tatus **I**nformation
DTC1	**D**evice **T**ool Management **C**onfiguration
DTC2	**D**evice **T**ool Management **C**onfiguration
DTG1	**D**istance **T**o **G**o
DTG2	**D**istance **T**o **G**o by log. AxisNr
DTY1	**D**evice **TY**pe
DWD1	**D**iagnosis **W**indow **D**ata
DWD2	**D**iagnosis **W**indow **D**ata
ECI1	**E**rror **C**omponent **I**nformation
EDE1	**E**xisting **D**iagnosis **E**rror
EDE2	**E**xisting **D**iagnosis **E**rror
EDW1	**E**xisting **D**iagnosis **W**indow
EDW2	**E**xisting **D**iagnosis **W**indow
EDW3	**E**xisting **D**iagnosis **W**indow
END1	**E**xisting **N**C **D**iagnosis
END2	**E**xisting **N**C **D**iagnosis
EPD1	**E**xisting **P**LC **D**iagnosis
EPD2	**E**xisting **P**LC **D**iagnosis
EPD3	**E**xisting **P**LC **D**iagnosis
EPO1	ProgrammEd **PO**sition (END)
EPO2	ProgrammEd **PO**sition (END)
EPT1	**E**xisting **P**roVi **T**ypes
EST1	**E**rror **ST**ate
EXD1	**EX**ecution **D**isplay
EXD2	**EX**ecution **D**isplay

GPC1	**G**lobal **P**rocess **C**onfiguration
GPC2	**G**lobal **P**rocess **C**onfiguration
GPP1	**G**lobal **P**rocess **P**arameter
GPP2	**G**lobal **P**rocess **P**arameter
IPP	**I**nsert NC-**P**rogram **P**ackage
MAP1	**M**odule **A**ssign of **P**rocess
MAR	**M**ap **A**bsolut PCL-**R**eferenz
MCD1	**M**odule **C**onfigration: **D**evice Information
MCM1	**M**odule **C**onfigration: **M**odule Information
MCP1	**M**odule **C**onfigration: **P**rocess Information
MCS1	**M**odule **C**onfigration: **S**FC- Information
MDA1	**M**achine **D**ata **A**ccess
MDA2	**M**achine **D**ata **A**ccess
MDA3	**M**achine **D**ata **A**ccess
MDA4	**M**achine **D**ata **A**ccess
MDI	**M**anual **D**ata **I**nput
MDS1	**M**achine **D**ata **S**ingle
MFD1	**M**essage **F**iles **D**ownload
MFO1	**M**aximal **F**eedrate **O**verride
MFR	**M**aximal **F**eed**R**ate
MKS	**M**achine **K**ey **S**tatus
MKT1	**M**achine **K**ey **T**able
MRO1	**M**aximal **R**apid **O**verride
MSG	**M**e**S**sa**G**e
MSO1	**M**aximal **S**pindle **O**verride
MSS	**M**aximal **S**pindle **S**peed
MTC	**MT**-**C**NC Slot Software Version
MTC1	**MT**-**C**NC Slot Software Version
MTD	**M**achine **T**able **D**ata
NCA1	**NC**-Program **A**ccess
NCA3	**NC**-Program **A**ccess
NCM1	**NC** **M**essages
NCM2	**NC** **M**essages
NEA1	**N**C-**E**vent **A**ccess
NEV	**N**C **Ev**ent
NMM	**N**C **M**e**M**ory selection
NPA1	**N**C **PA**rameter
NPA2	**N**C **PA**rameter
NPA3	**N**C **PA**rameter
NPA4	**N**C **PA**rameter
NPA5	**N**C **PA**rameter

NPC1	**N**C-**P**ackage **C**ompiling
NPD1	**N**C-**P**ackage **D**ownload
NPI	**N**C-**P**ackage D**I**rectory
NPS	**N**C **P**rogram **S**election
NTN	**N**ext Tool-**N**umber
NUA1	**N**C-Offset Data **A**ccess
NVA1	**N**C-**V**ariable **A**ccess
NVS	**N**C **V**ariable **S**ingle
OPD1	**O**ptimal **P**osition **D**istance by Axis sign.
OPD2	**O**ptimal **P**osition **D**istance by phys. AxisNr
PAA1	**PA**rameter **A**ccess
PAA2	**PA**rameter **A**ccess
PAC1	**P**rocess **A**xis **C**onfiguration Parameter
PAC2	**P**rocess **A**xis **C**onfiguration Parameter
PAD1	**PA**rameter **D**eactivate
PAS1	**PA**rameter **S**et Active
PDD1	**P**rovi **D**iagnosis **D**ata
PDD2	**P**rovi **D**iagnosis **D**ata
PDD3	**P**rovi **D**iagnosis **D**ata
PDD4	**P**rovi **D**iagnosis **D**ata
PDD5	**P**rovi **D**iagnosis **D**ata
PDT	**P**arameter **D**efinition **T**able
PFR	**P**rogrammed **F**eed**R**ate
PMI	**P**lc **M**emory **I**nformation
PPA	**P**art **P**rogram **A**ctive
PPD	**P**art-**P**rogram **D**irectory
PPN	**P**art-**P**rogram **N**C
PPP	**P**art **P**rogram **P**ackage
PPS	**P**art-**P**rogram-**S**equence
PSM	**P**CL **S**ys **M**essage
PSS	**P**rogrammed **S**pindle **S**peed
PTC1	**P**rocess **T**ool Management **C**onfiguration
PTC2	**P**rocess **T**ool Management **C**onfiguration
PTC3	**P**rocess **T**ool Management **C**onfiguration
PVA1	**P**RO**V**I-Messages **A**ccess
PVA2	**P**RO**V**I-Messages **A**ccess
PVF	**P**LC **V**ariable **F**ormated
PVM1	**P**ro**V**i **M**essages
PVM2	**P**ro**V**i **M**essages
PVM3	**P**ro**V**i **M**essages
PVM4	**P**ro**V**i **M**essages

PVR1	**P**LC **V**araible **R**etain Backup
PVT	**P**LC **V**ariable **T**ype
REP1	**REP**ositioning Data
REP2	**REP**ositioning Data
RPO	**R**elative Axis **PO**sition
SDD1	**S**fc **D**iagnosis **D**ata
SDD2	**S**fc **D**iagnosis **D**ata
SDD3	**S**fc **D**iagnosis **D**ata
SDD4	**S**fc **D**iagnosis **D**ata
SDD5	**S**fc **D**iagnosis **D**ata
SDD6	**S**fc **D**iagnosis **D**ata
SDD7	**S**fc **D**iagnosis **D**ata
SDS1	**S**et **D**evice **S**tatus
SDS2	**S**et **D**evice **S**tatus
SEM1	**S**et FI **E**xclusive **M**ode
SFD1	**SF**c **D**ata
SFD2	**SF**c **D**ata
SFD3	**SF**c **D**ata
SFE1	**SF**c **E**rror
SFE2	**SF**c **E**rror
SFM1	**SF**c **M**ode
SID1	**S**oftware **I**nstallation **D**ata
SLA1	Actual **S**ervo **LA**g
SLA2	Actual **S**ervo **LA**g
SLI	PLC (**S**PS) **L**ong **I**dentification
SPA1	**S**ercos **PA**rameter
SPA3	**S**ercos **PA**rameter
SPA4	**S**ercos **PA**rameter
SPH1	**S**ercos **PH**ase
SPP	**S**elected **P**art **P**rogram Number
TDA1	**T**ool **DA**ta
TDA2	**T**ool **DA**ta
TDD	**T**ool **D**ata **D**ownload
TDE	**T**ool List **D**ownload **E**scape
TDF	**T**ool List **D**ownload **F**inish
TDI	**T**ool List **D**ownload **I**nitialize
TDR1	**T**ool **D**ata **R**ecord of Place
TDR2	**T**ool **D**ata **R**ecord
TIF	**T**ool **I**nsert **F**inish
TII	**T**ool **I**nsert **I**nitiated
TLA1	**T**ool **L**ist **A**ccess

TLA2	**T**ool **L**ist **A**ccess
TLA3	**T**ool **L**ist **A**ccess
TLB1	**T**oo**L** **B**asicdata List
TLB2	**T**oo**L** **B**asicdata List
TLD1	**T**oo**L** **D**ata of Place
TLD2	**T**oo**L** **D**ata of Tool
TLD3	**T**oo**L** **D**ata of Place
TLD4	**T**oo**L** **D**ata of Tool
TLE1	**T**oo**L** **E**dgedata List
TLE2	**T**oo**L** **E**dgedata List
TMV	**T**ool **M**o**V**e
TPI1	**T**ool **P**osition **I**nformation
TPI2	**T**ool **P**osition **I**nformation
TQE1	Actual **T**or**Q**u**E**
TQE2	Actual **T**or**Q**u**E**
TRM	**T**ool **R**e**M**ove
TRS	**T**ool **R**e**S**et
WLA1	**W**atch **L**ist **A**llocation
WLF1	**W**atch **L**ist **F**ree
WLF2	**W**atch **L**ist **F**ree
ZOD	**Z**ero **O**ffset **D**ata
ZOD1	**Z**ero **O**ffset **D**ata
ZOD2	**Z**ero **O**ffset Data

Fig. 2-7: Overview of the MWCX device group

Overview of the MSCX Device Group

Com.	Description
ASE	**A**ctual **S**ystem **E**rror
CSE	**C**lear **S**ystem **E**rror
DCT1	**D**evice **C**ommunication **T**imeout
DCT2	**D**evice **C**ommunication **T**imeout
DEM1	**D**elete FI **E**xclusive **M**ode
DIF1	**D**evice **I**n**F**ormation
DSI1	**D**evice **S**tatus **I**nformation
DSI2	**D**evice **S**tatus **I**nformation
DTY1	**D**evice **TY**pe
MSG	**M**e**S**sa**G**e
SDS1	**S**et **D**evice **S**tatus
SDS2	**S**et **D**evice **S**tatus
SEM1	**S**et FI **E**xclusive **M**ode
SID1	**S**oftware **I**nstallation **D**ata
SPA1	**S**ercos **PA**rameter
SPH	**S**ercos **PH**ase

Fig. 2-8: Overview of the MSCX device group

Overview of the MVMX Device Group

Kom.	Beschreibung
ASM2	**A**ctive **S**ystem Fault **M**essage
ASM5	**A**ctive **S**ystem Fault **M**essage
CRT	**C**ontrol **R**ese**T**
DCT1	**D**evice **C**ommunication **T**imeout
DCT2	**D**evice **C**ommunication **T**imeout
DEM1	**D**elete FI **E**xclusive **M**ode
DFS1	**D**elete I**F** Command **S**tack
DIF1	**D**evice **I**n**F**ormation
DSI1	**D**evice **S**tatus **I**nformation
DSI2	**D**evice **S**tatus **I**nformation
DTY1	**D**evice **Ty**pe
ECI1	**E**rror **C**omponent **I**nformation
MCD1	**M**odule **C**onfigration: **D**evice Information
MCM1	**M**odule **C**onfigration: **M**odule Information
SDS1	**S**et **D**evice **S**tatus
SDS2	**S**et **D**evice **S**tatus
SEM1	**S**et FI **E**xclusive **M**ode
SID1	**S**oftware **I**nstallation **D**ata
VMD1	**V**isual **M**otion **D**ata

Fig. 2-9: Overview of the MVMX device group

Overview of the MWMX Device Group

Com.	Description
ADW	**A**ctive **D**iagnosis **W**indow
ASM2	**A**ctive **S**ystem Fault **M**essage
ASM5	**A**ctive **S**ystem Fault **M**essage
CMD1	**C**reate **M**I Import **D**ata
CMD2	**C**reate **M**I Import **D**ata
CMD3	**C**reate **M**I Import **D**ata
CMD4	**C**reate **M**I Import **D**ata
CRT	**C**ontrol **R**ese**T**
DCI1	**D**evice **C**omponent **I**nformation
DCT1	**D**evice **C**ommunication **T**imeout
DCT2	**D**evice **C**ommunication **T**imeout
DEM1	**D**elete FI **E**xclusive **M**ode
DIF1	**D**evice **I**n**F**ormation
DIS2	**D**ata **I**dentifikation **S**tring PLC Program
DSI1	**D**evice **S**tatus **I**nformation
DSI2	**D**evice **S**tatus **I**nformation
DTY1	**D**evice **Ty**pe
DWD1	**D**iagnosis **W**indow **D**ata
DWD2	**D**iagnosis **W**indow **D**ata
ECI1	**E**rror **C**omponent **I**nformation
EDE1	**E**xisting **D**iagnosis **E**rror
EDE2	**E**xisting **D**iagnosis **E**rror
EDW1	**E**xisting **D**iagnosis **W**indow
EDW2	**E**xisting **D**iagnosis **W**indow
EDW3	**E**xisting **D**iagnosis **W**indow
EPD1	**E**xisting **P**LC **D**iagnosis
EPD2	**E**xisting **P**LC **D**iagnosis
EPD3	**E**xisting **P**LC **D**iagnosis
EPT1	**E**xisting **P**roVi **T**ypes
EST1	**E**rror **ST**ate
EXD1	**EX**ecution **D**isplay
EXD2	**EX**ecution **D**isplay
MAR	**M**ap **A**bsolut PCL-**R**eferenz
MCD1	**M**odule **C**onfigration: **D**evice Information
MCM1	**M**odule **C**onfigration: **M**odule Information
MCS1	**M**odule **C**onfigration: **S**FC- Information
MFD1	**M**essage **F**iles **D**ownload
MKS	**M**achine **K**ey **S**tatus
MKT1	**M**achine **K**ey **T**able

MSG	**M**e**S**sa**G**e
MTC	**MT**-**C**NC Slot Software Version
PDD1	**P**rovi **D**iagnosis **D**ata
PDD2	**P**rovi **D**iagnosis **D**ata
PDD3	**P**rovi **D**iagnosis **D**ata
PDD4	**P**rovi **D**iagnosis **D**ata
PDD5	**P**rovi **D**iagnosis **D**ata
PMI	**P**lc **M**emory **I**nformation
PSM	**P**CL **S**ys **M**essage
PVA1	**P**RO**V**I-Messages **A**ccess
PVA2	**P**RO**V**I-Messages **A**ccess
PVF	**P**LC **V**ariable **F**ormated
PVM1	**P**ro**V**i **M**essages
PVM2	**P**ro**V**i **M**essages
PVM3	**P**ro**V**i **M**essages
PVM4	**P**ro**V**i **M**essages
PVR1	**P**LC **V**araible **R**etain Backup
PVT	**P**LC **V**ariable **T**ype
SDD1	**S**fc **D**iagnosis **D**ata
SDD2	**S**fc **D**iagnosis **D**ata
SDD3	**S**fc **D**iagnosis **D**ata
SDD4	**S**fc **D**iagnosis **D**ata
SDD5	**S**fc **D**iagnosis **D**ata
SDD6	**S**fc **D**iagnosis **D**ata
SDD7	**S**fc **D**iagnosis **D**ata
SDS1	**S**et **D**evice **S**tatus
SDS2	**S**et **D**evice **S**tatus
SEM1	**S**et FI **E**xclusive **M**ode
SFD1	**SF**c **D**ata
SFD2	**SF**c **D**ata
SFD3	**SF**c **D**ata
SFE1	**SF**c **E**rror
SFE2	**SF**c **E**rror
SFM1	**SF**c **M**ode
SID1	**S**oftware **I**nstallation **D**ata
SLI	PLC (**S**PS) **L**ong **I**dentification
VMD1	**V**isual **M**otion **D**ata
WLA1	**W**atch **L**ist **A**llocation
WLF1	**W**atch **L**ist **F**ree
WLF2	**W**atch **L**ist **F**ree

Fig. 2-10: Overview of MWMX device group

Overview of the MWSX Device Group

Com.	**Description**
ADW	**A**ctive **D**iagnosis **W**indow
ASM1	**A**ctive **S**ystem Fault **M**essage
ASM2	**A**ctive **S**ystem Fault **M**essage
ASM3	**A**ctive **S**ystem Fault **M**essage
ASM4	**A**ctive **S**ystem Fault **M**essage
ASM5	**A**ctive **S**ystem Fault **M**essage
BCD1	**B**us **C**onfiguration **D**ata
BCD2	**B**us **C**onfiguration **D**ata
BCD3	**B**us **C**onfiguration **D**ata
BCD4	**B**us **C**onfiguration **D**ata
BCD5	**B**us **C**onfiguration **D**ata
BCD6	**B**us **C**onfiguration **D**ata
BCD7	**B**us **C**onfiguration **D**ata
CMD1	**C**reate **M**I Import **D**ata
CMD2	**C**reate **M**I Import **D**ata
CMD3	**C**reate **M**I Import **D**ata
CMD4	**C**reate **M**I Import **D**ata
CRT	**C**ontrol **R**ese**T**
DCI1	**D**evice **C**omponent **I**nformation
DCT1	**D**evice **C**ommunication **T**imeout
DCT2	**D**evice **C**ommunication **T**imeout
DEM1	**D**elete FI **E**xclusive **M**ode
DIF1	**D**evice **I**n**F**ormation
DIF2	**D**evice **I**n**F**ormation
DIF3	**D**evice **I**n**F**ormation
DIS2	**D**ata **I**dentifikation **S**tring PLC Program
DSI1	**D**evice **S**tatus **I**nformation
DSI2	**D**evice **S**tatus **I**nformation
DTY1	**D**evice **Ty**pe
DWD1	**D**iagnosis **W**indow **D**ata
DWD2	**D**iagnosis **W**indow **D**ata
ECI1	**E**rror **C**omponent **I**nformation
EDE1	**E**xisting **D**iagnosis **E**rror
EDE2	**E**xisting **D**iagnosis **E**rror
EDW1	**E**xisting **D**iagnosis **W**indow
EDW2	**E**xisting **D**iagnosis **W**indow
EDW3	**E**xisting **D**iagnosis **W**indow
EPD1	**E**xisting **P**LC **D**iagnosis
EPD2	**E**xisting **P**LC **D**iagnosis

EPD3	**E**xisting **P**LC **D**iagnosis
EPT1	**E**xisting **P**roVi **T**ypes
EST1	**E**rror **ST**ate
EXD1	**EX**ecution **D**isplay
EXD2	**EX**ecution **D**isplay
MAR	**M**ap **A**bsolut PCL-**R**eferenz
MCD1	**M**odule **C**onfigration: **D**evice Information
MCM1	**M**odule **C**onfigration: **M**odule Information
MCS1	**M**odule **C**onfigration: **S**FC- Information
MFD1	**M**essage **F**iles **D**ownload
MKS	**M**achine **K**ey **S**tatus
MKT1	**M**achine **K**ey **T**able
MSG	**M**e**S**sa**G**e
MTC	**MT**-**C**NC Slot Software Version
MTC1	**MT**-**C**NC Slot Software Version
PDD1	**P**rovi **D**iagnosis **D**ata
PDD2	**P**rovi **D**iagnosis **D**ata
PDD3	**P**rovi **D**iagnosis **D**ata
PDD4	**P**rovi **D**iagnosis **D**ata
PDD5	**P**rovi **D**iagnosis **D**ata
PMI	**P**lc **M**emory **I**nformation
PSM	**P**CL **S**ys **M**essage
PVA1	**P**RO**V**I-Messages **A**ccess
PVA2	**P**RO**V**I-Messages **A**ccess
PVF	**P**LC **V**ariable **F**ormated
PVM1	**P**ro**V**i **M**essages
PVM2	**P**ro**V**i **M**essages
PVM3	**P**ro**V**i **M**essages
PVM4	**P**ro**V**i **M**essages
PVR1	**P**LC **V**araible **R**etain Backup
PVT	**P**LC **V**ariable **T**ype
SDD1	**S**fc **D**iagnosis **D**ata
SDD2	**S**fc **D**iagnosis **D**ata
SDD3	**S**fc **D**iagnosis **D**ata
SDD4	**S**fc **D**iagnosis **D**ata
SDD5	**S**fc **D**iagnosis **D**ata
SDD6	**S**fc **D**iagnosis **D**ata
SDD7	**S**fc **D**iagnosis **D**ata
SDS1	**S**et **D**evice **S**tatus
SDS2	**S**et **D**evice **S**tatus
SEM1	**S**et FI **E**xclusive **M**ode

SFD1	**SF**c **D**ata
SFD2	**SF**c **D**ata
SFD3	**SF**c **D**ata
SFE1	**SF**c **E**rror
SFE2	**SF**c **E**rror
SFM1	**SF**c **M**ode
SID1	**S**oftware **I**nstallation **D**ata
SLI	PLC (**S**PS) **L**ong **I**dentification
WLA1	**W**atch **L**ist **A**llocation
WLF1	**W**atch **L**ist **F**ree
WLF2	**W**atch **L**ist **F**ree

Fig. 2-11: Overview of the MWSX device group

Overview of the MWAX Device Group

Kom.	**Beschreibung**
AAS2	**A**ctual **A**xis **S**peed
ADB1	Get **A**ctual **D**ata **B**ase
ADM1	**MTA 200 M**essages
ADM2	**MTA 200 M**essages
ADM3	**MTA 200 M**essages
ADW	**A**ctive **D**iagnosis **W**indow
AFR	**A**ctual **F**eed **R**ate
AMM7	**A**ctive **M**echanism **M**essage
APO2	**A**ctual Machine **PO**sition
ASM1	**A**ctive **S**ystem Fault **M**essage
ASM2	**A**ctive **S**ystem Fault **M**essage
ASM3	**A**ctive **S**ystem Fault **M**essage
ASM4	**A**ctive **S**ystem Fault **M**essage
ASM5	**A**ctive **S**ystem Fault **M**essage
BCD1	**B**us **C**onfiguration **D**ata
BCD2	**B**us **C**onfiguration **D**ata
BCD3	**B**us **C**onfiguration **D**ata
BCD4	**B**us **C**onfiguration **D**ata
BCD5	**B**us **C**onfiguration **D**ata
BCD6	**B**us **C**onfiguration **D**ata
BCD7	**B**us **C**onfiguration **D**ata
CMA	**CM**OS RAM **A**SCII Parameter
CMD1	**C**reate **M**I Import **D**ata
CMD2	**C**reate **M**I Import **D**ata
CMD3	**C**reate **M**I Import **D**ata
CMD4	**C**reate **M**I Import **D**ata
CMF	**CM**OS RAM **F**loatingpoint Parameter

CMI	**CM**OS RAM **I**nteger Parameter
CNP1	**C**onvert **N**C **P**rogram
CNP2	**C**onvert **N**C **P**rogram
CPI1	**C**urrent **P**rocess **I**nformation
CRT	**C**ontrol **R**ese**T**
CVA1	**C**heck **V**irtual **A**xis
DAC1	**D**evice **A**xis **C**onfiguration Parameter
DAP1	**D**ownload **A**nlogC-**P**rogramm
DCI1	**D**evice **C**omponent **I**nformation
DCP1	**D**evice **C**onfiguration **P**arameter
DCP2	**D**evice **C**onfiguration **P**arameter
DCT1	**D**evice **C**ommunication **T**imeout
DCT2	**D**evice **C**ommunication **T**imeout**D**elete FI **E**xclusive **M**ode
DEM1	**D**elete FI **E**xclusive **M**ode
DFH1	**D**atabase **F**ile **H**andling
DIF1	**D**evice **I**n**F**ormation
DIF2	**D**evice **I**n**F**ormation
DIF3	**D**evice **I**n**F**ormation
DIS1	**D**ata **I**dentifikation **S**tring Parameter
DIS2	**D**ata **I**dentifikation **S**tring PLC Program
DPA1	**D**rive **P**arameter **A**ccess
DPA2	**D**rive **P**arameter **A**ccess
DPA3	**D**rive **P**arameter **A**ccess
DPA4	**D**rive **P**arameter **A**ccess
DSI1	**D**evice **S**tatus **I**nformation
DSI2	**D**evice **S**tatus **I**nformation
DTG2	**D**istance **T**o **G**o by log. AxisNr
DTY1	**D**evice **Ty**pe
DWD1	**D**iagnosis **W**indow **D**ata
DWD2	**D**iagnosis **W**indow **D**ata
EAD1	**E**xisting **MTA 200 D**iagnosis
EAD2	**E**xisting **MTA 200 D**iagnosis
ECI1	**E**rror **C**omponent **I**nformation
EDE1	**E**xisting **D**iagnosis **E**rror
EDE2	**E**xisting **D**iagnosis **E**rror
EDW1	**E**xisting **D**iagnosis **W**indow
EDW2	**E**xisting **D**iagnosis **W**indow
EDW3	**E**xisting **D**iagnosis **W**indow
EPT1	**E**xisting **P**roVi **T**ypes
EST1	**E**rror **ST**ate
EXD1	**EX**ecution **D**isplay

EXD2	**EX**ecution **D**isplay
GDF1	**G**et **D**atabase-**F**ilelist
GMF1	**G**et active **M**ain-**F**ileinformation
GPP1	**G**lobal **P**rocess **P**arameter
GPP2	**G**lobal **P**rocess **P**arameter
HPF1	**H**and-**P**arameter **F**lotingpoint
HPI1	**H**and-**P**arameter **I**nteger
IFR1	**I**dent **F**ile-**R**ange
IPA1	**I**dentdatei-**P**arameter **A**SCII
IPF1	**I**dentdatei-**P**arameter **F**LP
IPI1	**I**dentdatei-**P**arameter **I**nteger
MAP1	**M**odule **A**ssign of **P**rocess
MAR	**M**ap **A**bsolut PCL-**R**eferenz
MCD1	**M**odule **C**onfigration: **D**evice Information
MCM1	**M**odule **C**onfigration: **M**odule Information
MCP1	**M**odule **C**onfigration: **P**rocess Information
MCS1	**M**odule **C**onfigration: **S**FC- Information
MDI1	**M**anual **D**ata **I**nput
MFD1	**M**essage **F**iles **D**ownload
MIS1	**M**M_**I**NCH-**S**tatus
MKS	**M**achine **K**ey **S**tatus
MKT1	**M**achine **K**ey **T**able
MSG	**M**e**S**sa**G**e
MTC	**MT**-**C**NC Slot Software Version
MTC1	**MT**-**C**NC Slot Software Version
NPA5	**N**C **PA**rameter
PAA2	**PA**rameter **A**ccess
PDD1	**P**rovi **D**iagnosis **D**ata
PDD2	**P**rovi **D**iagnosis **D**ata
PDD3	**P**rovi **D**iagnosis **D**ata
PDD4	**P**rovi **D**iagnosis **D**ata
PDD5	**P**rovi **D**iagnosis **D**ata
PFR	**P**rogrammed **F**eed **R**ate
PMI	**P**lc **M**emory **I**nformation
POI	**PO**sition **I**nformation
PPS7	**P**art **P**rogram **S**equence
PSD1	Set **P**re**S**et **D**ata
PSM	**P**CL **S**ys **M**essage
PVA1	**P**ROV**I**-Messages **A**ccess
PVA2	**P**ROV**I**-Messages **A**ccess
PVF	**P**LC **V**ariable **F**ormated

PVM1	**P**ro**V**i **M**essages
PVM2	**P**ro**V**i **M**essages
PVM3	**P**ro**V**i **M**essages
PVM4	**P**ro**V**i **M**essages
PVR1	**P**LC **V**araible **R**etain Backup
PVT	**P**LC **V**ariable **T**ype
SCO1	**S**ercos **CO**nnection
SDD1	**S**fc **D**iagnosis **D**ata
SDD2	**S**fc **D**iagnosis **D**ata
SDD3	**S**fc **D**iagnosis **D**ata
SDD4	**S**fc **D**iagnosis **D**ata
SDD5	**S**fc **D**iagnosis **D**ata
SDD6	**S**fc **D**iagnosis **D**ata
SDD7	**S**fc **D**iagnosis **D**ata
SDS1	**S**et **D**evice **S**tatus
SDS2	**S**et **D**evice **S**tatus
SEM1	**S**et FI **E**xclusive **M**ode
SFD1	**SF**c **D**ata
SFD2	**SF**c **D**ata
SFD3	**SF**c **D**ata
SFE1	**SF**c **E**rror
SFE2	**SF**c **E**rror
SFM1	**SF**c **M**ode
SID1	**S**oftware **I**nstallation **D**ata
SLI	PLC (**S**PS) **L**ong **I**dentification
SPA1	**S**ercos **PA**rameter
SPA3	**S**ercos **PA**rameter
SPA4	**S**ercos **PA**rameter
SPH	**S**ercos **PH**ase
TDL1	**T**ool **D**ata **L**ist
TLD7	**T**oo**L** **D**ata
TQE2	**T**or**Q**u**E**
WLA1	**W**atch **L**ist **A**llocation
WLF1	**W**atch **L**ist **F**ree
WLF2	**W**atch **L**ist **F**ree
ZOD7	**Z**ero **O**ffset **D**ata
ZOD8	**Z**ero **O**ffset **D**ata

Fig. 2-12: Overview of the MWAX device group

Overview of the MSYX Device Group

Com.	**Description**
ADW	**A**ctive **D**iagnosis **W**indow
ASE	**A**ctual **S**ystem **E**rror
CRT	**C**ontrol **R**ese**T**
CSE	**C**lear **S**ystem **E**rror
DCT1	**D**evice **C**ommunication **T**imeout
DCT2	**D**evice **C**ommunication **T**imeout
DEM1	**D**elete FI **E**xclusive **M**ode
DIF1	**D**evice **I**n**F**ormation
DSI1	**D**evice **S**tatus **I**nformation
DSI2	**D**evice **S**tatus **I**nformation
DTY	**D**evice **Ty**pe
DWD1	**D**iagnosis **W**indow **D**ata
DWD2	**D**iagnosis **W**indow **D**ata
ECI1	**E**rror **C**omponent **I**nformation
EDE1	**E**xisting **D**iagnosis **E**rror
EDW1	**E**xisting **D**iagnosis **W**indow
LNG	Active **L**a**NG**uage
MSG	**M**e**S**sa**G**e
SDS1	**S**et **D**evice **S**tatus
SDS2	**S**et **D**evice **S**tatus
SEM1	**S**et FI **E**xclusive **M**ode
SID1	**S**oftware **I**nstallation **D**ata
SPA1	**S**ercos **PA**rameter
SPH	**S**ercos **PH**ase

Fig. 2-13: Overview of the MSYX device group

Overview of the MWYX Device Group

Com.	**Description**
ADW	**A**ctive **D**iagnosis **W**indow
ASE	**A**ctual **S**ystem **E**rror
ASM2	**A**ctive **S**ystem Fault **M**essage
CMD1	**C**reate **M**I Import **D**ata
CMD2	**C**reate **M**I Import **D**ata
CMD3	**C**reate **M**I Import **D**ata
CMD4	**C**reate **M**I Import **D**ata
CRT	**C**ontrol **R**ese**T**
CSE	**C**lear **S**ystem **E**rror
DCI1	**D**evice **C**omponent **I**nformation
DCT1	**D**evice **C**ommunication **T**imeout

DCT2	**D**evice **C**ommunication **T**imeout
DEM1	**D**elete FI **E**xclusive **M**ode
DIF1	**D**evice **I**n**F**ormation
DIS2	**D**ata **I**dentifikation **S**tring PLC Program
DSI1	**D**evice **S**tatus **I**nformation
DSI2	**D**evice **S**tatus **I**nformation
DTY1	**D**evice **Ty**pe
DWD1	**Di**agnosis **W**indow **D**ata
DWD2	**Di**agnosis **W**indow **D**ata
ECI1	**E**rror **C**omponent **I**nformation
EDE1	**E**xisting **D**iagnosis **E**rror
EDE2	**E**xisting **D**iagnosis **E**rror
EDW1	**E**xisting **D**iagnosis **W**indow
EDW2	**E**xisting **D**iagnosis **W**indow
EDW3	**E**xisting **D**iagnosis **W**indow
EPD1	**E**xisting **P**LC **D**iagnosis
EPD2	**E**xisting **P**LC **D**iagnosis
EPD3	**E**xisting **P**LC **D**iagnosis
EPT1	**E**xisting **P**roVi **T**ypes
EST1	**E**rror **ST**ate
EXD1	**EX**ecution **D**isplay
EXD2	**EX**ecution **D**isplay
LNG	Active **L**a**NG**uage
MAR	**M**ap **A**bsolut PCL-**R**eferenz
MCD1	**M**odule **C**onfigration: **D**evice Information
MCM1	**M**odule **C**onfigration: **M**odule Information
MCS1	**M**odule **C**onfigration: **S**FC- Information
MFD1	**M**essage **F**iles **D**ownload
MKS	**M**achine **K**ey **S**tatus
MKT1	**M**achine **K**ey **T**able
MSG	**M**e**S**sa**G**e
MTC	**MT**-**C**NC Slot Software Version
PDD1	**P**rovi **D**iagnosis **D**ata
PDD2	**P**rovi **D**iagnosis **D**ata
PDD3	**P**rovi **D**iagnosis **D**ata
PDD4	**P**rovi **D**iagnosis **D**ata
PDD5	**P**rovi **D**iagnosis **D**ata
PMI	**P**lc **M**emory **I**nformation
PSM	**P**CL **S**ys Message**P**lc **M**emory **I**nformation
PVA1	**P**ROVI-Messages **A**ccess
PVA2	**P**ROVI-Messages **A**ccess

PVF	**P**LC **V**ariable **F**ormated
PVM1	**P**ro**V**i **M**essages
PVM2	**P**ro**V**i **M**essages
PVM3	**P**ro**V**i **M**essages
PVM4	**P**ro**V**i **M**essages
PVR1	**P**LC **V**araible **R**etain Backup
PVT	**P**LC **V**ariable **T**ype
SDD1	**S**fc **D**iagnosis **D**ata
SDD2	**S**fc **D**iagnosis **D**ata
SDD3	**S**fc **D**iagnosis **D**ata
SDD4	**S**fc **D**iagnosis **D**ata
SDD5	**S**fc **D**iagnosis **D**ata
SDD6	**S**fc **D**iagnosis **D**ata
SDD7	**S**fc **D**iagnosis **D**ata
SDS1	**S**et **D**evice **S**tatus
SDS2	**S**et **D**evice **S**tatus
SEM1	**S**et FI **E**xclusive **M**ode
SFD1	**SF**c **D**ata
SFD2	**SF**c **D**ata
SFD3	**SF**c **D**ata
SFE1	**SF**c **E**rror
SFE2	**SF**c **E**rror
SFM1	**SF**c **M**ode
SID1	**S**oftware **I**nstallation **D**ata
SLI	PLC (**S**PS) **L**ong **I**dentification
SPA1	**S**ercos **PA**rameter
SPH	**S**ercos **PH**ase
WLA1	**W**atch **L**ist **A**llocation
WLF1	**W**atch **L**ist **F**ree
WLF2	**W**atch **L**ist **F**ree

Fig. 2-14: Overview of the MWYX device group

Logical Connection Between FI Commands

In the following table, all FI commands are grouped in a logical order.

Group	Device group	FI Commands
Axes	MTCX	AAD, AAS1, AAS2, ARF, ART, CPO1, CPO2, DAC1, DAC2, DTG1, DTG2, EPO1, EPO2, OPD1, OPD2, PAC1, PAC2, REP1, REP2, RPO, SLA1, SLA2, TQE1, TQE2
	MWCX	AAD, AAS1, AAS2, ARF, ART, CPO1, CPO2, CVA1, DAC1, DAC2, DPA1, DPA2, DPA3, DTG1, DTG2, EPO1, EPO2, OPD1, OPD2, PAC1, PAC2, REP1, REP2, RPO, SLA1, SLA2, TQE1, TQE2
	MWAX	AAS2, CVA1, DAC1, DPA1, DPA2, DPA3, DTG2, TQE2
Axis Parameters	MTCX	PAA1, PAA2, PAD1, PAS1
	MWCX	PAA1, PAA2, PAD1, PAS1
	MWAX	PAA2
D-correction	MTCX	ADN1, DCA1, DCD1, DCR1
	MWCX	ADN1, DCA1, DCD1, DCR1
Diagnosis	MTCX	DIF1, DIF2, DIF3, DWD1, DWD2, ECI1, EDE1, EDE2, EDW1, EDW2, EDW3, END1, END2
	MWCX	ADW1, DIF1, DIF2, DIF3, DWD1, DWD2, ECI1, EDE1, EDE2, EDW1, EDW2, EDW3, END1, END2, EPD1, EPD2, EPD3, SDD1, SDD2, SDD3, SDD4, SDD5, SDD6, SDD7
	MWMX	ADW1, DWD1, DWD2, ECI1, EDE1, EDE2, EDW1, EDW2, EDW3, EPD1, EPD2, EPD3, PDD1, PDD2, PDD3, PDD4, PDD5, PDD6, PVM1, PVM2, PVM3, PVM4, SDD1, SDD2, SDD3, SDD4, SDD5, SDD6, SDD7, SFE1, SFE2
	MWSX	ADW1, DIF1, DIF2, DIF3, DWD1, DWD2, ECI1, EDE1, EDE2, EDW1, EDW2, EDW3, EPD1, EPD2, EPD3, PDD1, PDD2, PDD3, PDD4, PDD5, PDD6, PVM1, PVM2, PVM3, PVM4, SDD1, SDD2, SDD3, SDD4, SDD5, SDD6, SDD7, SFE1, SFE2
	MWAX	ADW1, DIF1, DIF2, DIF3, DWD1, DWD2, EAD1, EAD2, ECI1, EDE1, EDE2, EDW1, EDW2, EDW3, PDD1, PDD2, PDD3, PDD4, PDD5, PDD6, PVM1, PVM2, PVM3, PVM4, SDD1, SDD2, SDD3, SDD4, SDD5, SDD6, SDD7, SFE1, SFE2
	MSYX	ADW1, DWD1, ECI1
	MWYX	ADW1, DWD1, DWD2, ECI1, EDE1, EDE2, EDW1, EDW2, EDW3, EPD1, EPD2, EPD3, PVM1, PVM2, PVM3, PVM4, SDD1, SDD2, SDD3, SDD4, SDD5, SDD6, SDD7, SFE1, SFE2
Download/ Upload	MTCX	CCA1, DCA1, DPA1, DPA2, DPA3, MDA1, MDA2, MDA4, MFD1, NCA1, NEA1, NUA1, NVA1, PAA1, PAA2
	MWCX	CCA1, DCA1, DPA1, DPA2, DPA3, MDA1, MDA2, MFD1, NCA1, NEA1, NUA1, NVA1, PAA1, PAA2, PVR1
	MWMX	MFD1, PVR1
	MWSX	MFD1, PVR1
	MWAX	DPA1, DPA2, DPA3, MFD1, PAA2, PVR1
	MWYX	MFD1, PVR1
Event	MTCX	AEM, NEA1, NEV
	MWCX	AEM, NEA1, NEV
Device	MPCX	CCP1, CCP2, CCP3, CCP4, CCP5, FCP1, FCP2, FCP3, FDC1
	MTCX	DCP1, DCP2, DSI1, DSI2, DTC1, DTY1, MCD1, SDS1, SDS2
	MWCX	DCI1, DCP1, DCP2, DSI1, DSI2, DTC1, DTY1, MCD1, SDS1, SDS2
	MSCX	DSI1, DSI2, DTY1, SDS1, SDS2

Group	Device group	FI Commands
	MWMX	DCI1, DSI1, DSI2, DTY1, MCD1, SDS1, SDS2
	MWSX	DCI1, DSI1, DSI2, DTY1, MCD1, SDS1, SDS2
	MWAX	DCI1, DCP1, DCP2, DSI1, DSI2, DTY1, MCD1, SDS1, SDS2
	MSYX	DSI1, DSI2, DTY1, SDS1, SDS2
	MWYX	DCI1, DSI1, DSI2, DTY1, MCD1, SDS1, SDS2
Configuration	MPCX	BCI1, BCI2, BCI3, CCP1, CCP2, CCP3, CCP4, CCP5, FCP1, FCP2, FCP3, FDC1, LNG
	MTCX	DAC1, DAC2, DCP1, DCP2, DTY1, GPC1, GPC2, PAC1, PAC2, PTC1, PTC2
	MWCX	BCD1, BCD2, BCD3, BCD4, BCD5, BCD6, BCD7, DAC1, DAC2, DCP1, DCP2, DTY1, GPC1, GPC2, PAC1, PAC2, PTC1, PTC2
	MSCX	DTY1
	MWMX	DTY1
	MWSX	BCD1, BCD2, BCD3, BCD4, BCD5, BCD6, BCD7, DTY1
	MWAX	BCD1, BCD2, BCD3, BCD4, BCD5, BCD6, BCD7, DCP1, DCP2, DTY1
	MWYX	DTY1
Machine Data	MTCX	DIS5, MDA1, MDA2, MDA4, MDS1, MKS, MTD
	MWCX	DIS5, MDA1, MDA2, MDA4, MDS2, MKS, MTD
Messages	MPCX	ERI1, FIT1, MSG, SSM1, SSM2
	MTCX	AMM1, AMM2, AMM3, AMM4, AMM5, ASM1, ASM2, ASM3, ASM4, ASM5, MSG, NCM1, NCM2, PVM1, PVM2, PVM3, PVM4, SLI
	MWCX	ASE, AMM1, AMM2, AMM3, AMM4, AMM5, ASM1, ASM2, ASM3, ASM4, ASM5, MSG, NCM1, NCM2, PDD1, PDD2, PDD3, PDD4, PDD5, PSM1, PVM1, PVM2, PVM3, PVM4, SLI
	MSCX	ASE, CSE, MSG
	MWMX	ASM2, ASM5, MSG, PDD1, PDD2, PDD3, PDD4, PDD5, PSM1, PVM1, PVM2, PVM3, PVM4, SLI
	MWSX	ASM1, ASM2, ASM3, ASM4, ASM5, MSG, PDD1, PDD2, PDD3, PDD4, PDD5, PSM1, PVM1, PVM2, PVM3, PVM4, SLI
	MWAX	ADM1, ADM2, ADM3, AMM7, ASM1, ASM2, ASM3, ASM4, ASM5, MSG, PDD1, PDD2, PDD3, PDD4, PDD5, PSM1, PVM1, PVM2, PVM3, PVM4, SLI
	MSYX	ASE
	MWYX	ASE, ASM2, MSG, PDD1, PDD2, PDD3, PDD4, PDD5, PSM1, PVM1, PVM2, PVM3, PVM4, SLI
Modules	MTCX	MAP1, MCD1, MCM1, MCP1, MCS1
	MWCX	MAP1, MCD1, MCM1, MCP1, MCS1
	MWMX	MCD1, MCM1, MCS1
	MWSX	MCD1, MCM1, MCS1
	MWAX	MAP1, MCD1, MCM1, MCP1, MCS1
	MWYX	MCD1, MCM1, MCS1
NC processing	MTCX	ABI, AGF, AMF, ANM, APM, APN, APP, ASN, CCA1, DCA1, DIS1, DIS3, DIS6, DPN, DPP, IPP, MDI, NCA1, NCA3, NCM1, NCM2, NEA1, NEV, NMM, NPA1, NPA2, NPA3, NPA4, NPC1, NPD1, NPI, NPS, NUA1, NVA1, NVS, PPA, PPD, PPN, PPP, PPS, SPP
	MWCX	ABI, AGF, AMF, ANM, APM, APN, APP, ASN, CCA1, DCA1, DIS1, DIS3, DIS6, DPN, DPP, IPP, MDI, NCA1, NCA3, NCM1, NCM2, NEA1, NEV, NMM, NPA1, NPA2, NPA3, NPA4, NPC1, NPD1, NPI, NPS, NUA1, NVA1, NVS, PPA, PPD, PPN, PPP, PPS, SPP
	MWAX	ADB1, CNP1, CNP2, MDI1, MIS1, NPA1, NPA2, NPA5

Group	Device group	FI Commands
Override	MTCX	AFO1, ARO1, ASO1, MFO1, MRO1, MSO1
	MWCX	AFO1, ARO1, ASO1, MFO1, MRO1, MSO1
Position	MTCX	APO1, APO2, CPO1, CPO2, DTG1, DTG2, EPO1, EPO2, OPD1, OPD2, REP1, REP2, SLA1, SLA2
	MWCX	APO1, APO2, CPO1, CPO2, DTG1, DTG2, EPO1, EPO2, OPD1, OPD2, REP1, REP2, SLA1, SLA2
	MWAX	APO2, DTG2, POI1, PSD1
Process	MPCX	CPR1, CPR2, DPR1, NST1, NST2, RPR1
	MTCX	CPI1, GPC1, GPC2, GPP1, GPP2, MAP1, MCP1, PAC1, PAC2, PTC1, PTC2
	MWCX	CPI1, GPC1, GPC2, GPP1, GPP2, MAP1, MCP1, PAC1, PAC2, PTC1, PTC2
	MWAX	MAP1, MCP1
Cutters	MTCX	AEN, TLE1, TLE2
	MWCX	AEN, TLE1, TLE2
Sercos	MTCX	SPA1, SPH1, SPH2
	MWCX	ASE, DPA1, DPA2, DPA3, SPA1, SPA3, SPH1, SPH2
	MSCX	ASE, SPA1, SPH
	MWAX	DPA1, DPA2, DPA3, SCO, SPA1, SPA3, SPH
	MSYX	ASE, SPA1, SPH
	MWYX	ASE, SPA1, SPH
Spindle	MTCX	AAD, AAS1, AAS2, ACS, ASD, ASF, ASG, ASO1, ASS, MSO1, MSS, PSS
	MWCX	AAD, AAS1, AAS2, ACS, ASD, ASF, ASG, ASO1, ASS, MSO1, MSS, PSS
	MWAX	AAS2
PLC	MWCX	CMD1, CMD2, CMD3, CMD4, DIS2, EDE1, EDE2, EPD1, EPD2, EPD3, EPT1, EST1, EXD1, EXD2, MAR, MKT1, PVF, PVT, SDD1, SDD2, SDD3, SDD4, SDD5, SDD6, SFD1, SFD2, SFE1, SFE2, SFM1, SLI
	MWMX	CMD1, CMD2, CMD3, CMD4, DIS2, EDE1, EDE2, EPD1, EPD2, EPD3, EPT1, EST1, EXD1, EXD2, MAR, MKT1, PMI, PVA1, PVA2, PVF, PVT, SDD1, SDD2, SDD3, SDD4, SDD5, SDD6, SFD1, SFD2, SFE1, SFE2, SFM1, SLI
	MWSX	CMD1, CMD2, CMD3, CMD4, DIS2, EDE1, EDE2, EPD1, EPD2, EPD3, EPT1, EST1, EXD1, EXD2, MAR, MKT1, PMI, PVA1, PVA2, PVF, PVT, SDD1, SDD2, SDD3, SDD4, SDD5, SDD6, SFD1, SFD2, SFE1, SFE2, SFM1, SLI
	MWAX	CMD1, CMD2, CMD3, CMD4, DIS2, EDE1, EDE2, EST1, EPT1, EXD1, EXD2, MAR, MKT1, PMI, PVA1, PVA2, PVF, PVT, SDD1, SDD2, SDD3, SDD4, SDD5, SDD6, SFD1, SFD2, SFE1, SFE2, SFM1, SLI
	MWYX	CMD1, CMD2, CMD3, CMD4, DIS2, EDE1, EDE2, EPD1, EPD2, EPD3, EPT1, EST1, EXD1, EXD2, MAR, MKT1, PMI, PVA1, PVA2, PVF, PVT, SDD1, SDD2, SDD3, SDD4, SDD5, SDD6, SFD1, SFD2, SFE1, SFE2, SFM1, SLI
Feed	MTCX	AAC1, AAD, AAS1, AAS2, ACS, ADN1, AFO1, AFR, ARO1, AZB1, CPO1, CPO2, DCD1, DCR1, DTG1, DTG2, MFO1, MFR, MRO1, OPD1, OPD2, PFR, PSS, REP1, REP2, SLA1, SLA2, TQE1, TQE2, ZOD, ZOD1, ZOD2
	MWCX	AAC1, AAD, AAS1, AAS2, ACS, ADN1, AFO1, AFR, ARO1, AZB1, CPO1, CPO2, DCD1, DCR1, DTG1, DTG2, MFO1, MFR, MRO1, OPD1, OPD2, PFR, PSS, REP1, REP2, SLA1, SLA2, TQE1, TQE2, ZOD, ZOD1, ZOD2
	MWAX	AAS2, AFR, DTG2, PFR
Tool	MTCX	AEN, ATN, ATP1, ATP2, ATP3, ATR, ATU, DIS4, DTC1, DTC2, NTN, PTC1, PTC2, TDA1, TDA2, TDE, TDF, TDI, TDR1, TDR2, TIF, TII, TLB1, TLB2, TLD1, TLD2, TLD3, TLD4, TLE1, TLE2, TMV, TPI1, TPI2, TRM, TRS

Group	Device group	FI Commands
	MWCX	AEN, ATN, ATP1, ATP2, ATP3, ATR, ATU, DIS4, DTC1, DTC2, NTN, PTC1, PTC2, TDA1, TDA2, TDE, TDF, TDI, TDR1, TDR2, TIF, TII, TLB1, TLB2, TLD1, TLD2, TLD3, TLD4, TLE1, TLE2, TMV, TPI1, TPI2, TRM, TRS
	MWAX	TDA7, TDL1, TLD7

Fig. 2-15: Logical conjunctions of FI commands

2.4 Command Execution Times

Legends for the Command Execution Times

The command execution times determined are typical measured values. Their capacity for reproduction depends on many factors. Among these factors are the type of computer used (processor, memory, etc.), the existing device configuration, (device; communication port DPR, V24 etc.) as well as the more or less heavy load caused by processes running in parallel.

The measured values determined are subject to a rasterization of 10ms. In principle, therefore, a tolerance of +/- 10ms should be assumed. In addition, sporadic measured values will be determined that lie outside this tolerance range. It cannot therefore be based on deterministic behavior.

The execution times determined do, however, help you to get a feeling for the processing times of the commands. You can therefore try numerous ways of accessing the device "at your desk" and find the best means of access.

For better comparison, the specifications of the PC and device configuration with which the command execution times have been determined are listed below.

Computer Type

The type of computer with which the following measured values have been determined has the following specifications:

Processor	RAM	Operating System
Pentium 166 MHz	32 MByte	Windows NT 4.0

Fig. 2-16: Computer identification data

Device Configuration

To determine the command execution times, a representative device was selected from each device class and the complete range of commands for the device was tested. The communication port used between the PC and the device is of critical importance. As should be expected, access via the DPR involves shorter execution times when compared to access via the serial interface. This should be taken into account when comparing the command execution times.

Refer to the following table for the respective representative devices of the device classes; the execution times have not been determined for each device family.

	Device	**PLC Compo-nents**	**NC Compo-nents**	**Communication configuration**
MPCX	PC	None	None	None
MWCX	MTC200-P-G2, MTC200-R-G2	MTS-P	MTC-P	DPR, TCON
MSCX	SERCANS-A, SERCANS-P	None	None	V24 19200 Baud TCON
MVMX	VM-P, VM-R	None	None	DPR, TCON
MWMX	VMISP200-P-G2, VMISP200-R-G2	MTS-P02.2	None	DPR, TCON
MWSX	ISP200-P-G2, ISP200-R-G2 ISP200-I-G2	MTS-P02.2	None	DPR, TCON SHM
MWAX	MTA200-P	MTS-P	None	SHM
MSYX	SYNAX200-P, SYNAX200-R	None	None	DPR, TCON
MWYX	SYNAXISP200-P-G2, SYNAXISP200-R-G2	MTS-P02.2	None	DPR, TCON

Fig. 2-17: Representative devices

Parallel Processes

The following processes are running while the command execution times are running:

- The application used in determining the times.
- The processes belonging to the function interface, i.e. COMINTFC.EXE, LOGINTFC.EXE, BOFINTFC.EXE.
- The MTA200.EXE process as communication driver to the MTA200-P.
- The MTVNC40V.EXE process as communication driver to the virtual MTC 200.
- The NETINTFC.EXE process as communication driver to the PC network.

Note: *1) The command marked is a job command.
The time given refers to the start of the job. The time it takes for the job to work in the background must be added to the complete time of command execution.
*2) For weighting the command execution time, the note is of decisive importance.

Command Execution Times for the MPCX Device Group

Com.	Description	Example	[ms]
CCP1	Cell Configuration Parameter	XX_BR_CCP1	30
CCP2	Cell Configuration Parameter	XX_BR_CCP2_MTC200-P	20
CCP3	Cell Configuration Parameter	XX_BR_CCP3_1	10
CCP4	Cell Configuration Parameter	XX_BR_CCP4_MWCX	20
CCP5	Cell Configuration Parameter	XX_BR_CCP5_02	20
DFJ1	Delete Function Interface Job	XX_BR_DFJ1	10
DFJ2	Delete Function Interface Job	XX_BR_DFJ2_1	20
FCP1	Far Device Configuration Parameter	XX_BR_FCP1	10
FCP2	Far Device Configuration Parameter	XX_BR_FCP2_MWCX	10
FCP3	Far Device Configuration Parameter	XX_BR_FCP3_MTC200	10
FDC1	Far Device Configuration	XX_BR_FDC1	20
FIT1	Further Info Text	XX_BR_FIT1_1_5	20
FPC1	Far PC Configuration	XX_BR_FPC1	10
IFJ1	Information about Function Interface Jobs	XX_BR_IFJ1	10
IFJ2	Information about Function Interface Jobs	XX_BR_IFJ2_1	10
LNG	Active LaNGuage	XX_BR_LNG	10

Fig. 2-18: Command execution times of the MPCX device group

Command Execution Times for the MWCX Device Group

Com.	Description	Example	[ms]
AAC1	Actual Acceleration	00_CR_AAC1_0	20
AAD	Active Angle Dimension	00_CR_AAD_0	20
AAS1	Actual Axis Speed	00_CR_AAS1_0_1	20
AAS2	Actual Axis Speed	00_CR_AAS2_2	20
ABI	Actual NC Block Information	00_BR_ABI_0	40
ABN	Active Conditional Banner No.	00_BR_ABN_0	30
ACS	Actual Cutting Speed	00_CR_ACS_0	20
ADN1	Active D-Correction Number	00_CR_ADN1_0	20
AEM	Active Event Monitoring	00_CR_AEM_0	20
AEN	Active Edge Number	00_CR_AEN_0	20
AFO1	Active Feedrate Override	00_CR_AFO1_0	20
AFR	Active FeedRate	00_CR_AFR_0	20
AGF	Active G-Function	00_CR_AGF_0	20
AMF	Active M-Function	00_CR_AMF_0	20
AMM1	Active Mechanism Message	00_BR_AMM1	100
AMM2	Active Mechanism Message	00_BR_AMM2	30
AMM3	Active Mechanism Message	00_BR_AMM3_0	70
AMM4	Active Mechanism Message	00_BR_AMM4_02.0	70
AMM5	Active Mechanism Message	00_BR_AMM5_0_69_0	40
API1	Actual Parameter Index	00_BR_API1	100

Com.	Description	Example	[ms]
API2	Actual Parameter Index	00_BR_API2	60
APM	Active Part Program Message	00_CR_APM_0	20
APN	Active Part Program Message Number	00_CR_APN_0	20
APO	Actual Machine POsition	00_CR_APO_0_2_1	20
APO1	Actual Machine POsition	00_CR_APO1_0_2_1	20
APO2	Actual Machine POsition	00_CR_APO2_3_1	20
APP	Active Part Program Number	00_CR_APP_0	20
ARO1	Actual Rapid Override	00_CR_AFO1_0	20
ASF	Actual Spindle For Process	00_CR_ASF_0	20
ASG	Actual Spindle Gear	00_CR_ASG_0_1	20
ASM1	Active System Fault Message	00_BR_ASM1	60
ASM2	Active System Fault Message	00_BR_ASM2	30
ASM3	Active System Fault Message	00_BR_ASM3_02	30
ASM4	Active System Fault Message	00_BR_ASM4_MWCX	50
ASM5	Active System Fault Message	00_BR_ASM5_74_0	30
ASN	Actual Sequence Number	00_CR_ASN_0	20
ASO1	Actual Spindle Override	00_CR_ASO1_0_1	20
ASS	Actual Spindle Speed	00_CR_ASS_0_1	20
ATN	Active Tool Number	00_CR_ATN_0	20
ATP1	Actual Tool Place Information	00_CR_ATP1_0	20
ATP2	Actual Tool Place Information	00_CR_ATP2_0	20
ATP3	Actual Tool Place Information	00_CR_ATP3_0	20
AZB1	Active Zero Offset Bank	00_CR_AZB1_0	20
CPO1	Command POsition (COMMAND)	00_CR_CPO1_0_2_1	20
CPO2	Command POsition by log.Axis No	00_CR_CPO2_3_1	20
CRT	Control ReseT		20
DAC1	Device Axis Configuration Parameter	00_BR_DAC1	20
DAC2	Device Axis Configuration Parameter	00_BR_DAC2_1	20
DCD1	D-Correction Data	00_CR_DCD1_0_1_1	20
DCP1	Device Configuration Parameter	00_BR_DCP1	30
DCP2	Device Configuration Parameter	00_BR_DCP2	20
DCR1	D-Correction Record	00_CR_DCR_0_1	20
DIS1	Data Identification String Parameter	00_CR_DIS1	20
DIS2	Data Identification String PLC Program	00_CR_DIS2	20
DIS3	Data Identification String NC Program	00_CR_DIS3_1	20
DIS4	Data Identification String Tool List	00_CR_DIS4_0	20
DIS5	Data Identification String Machine	00_CR_DIS5	20
DIS6	Data Identification String Machine	00_CR_DIS6_1_0_1	20
DPN	Delete Part Program NC		140
DPP	Delete Part Program Package	00_BW_DPP_2	40
DTC1	Device Tool Management Configuration	00_BR_DTC1	20
DTG1	Distance To Go	00_CR_DTG1_0_2_1	20

Com.	Description	Example	[ms]
DTG2	Distance To Go by log. Axis No	00_CR_DTG2_3_1	20
DTY1	Device TYpe	00_CR_DTY1	20
EPO1	ProgrammEd POsition (END)	00_CR_EPO1_0_2_1	20
EPO2	ProgrammEd POsition (END)	00_CR_EPO2_3_1	20
GPC1	Global Process Configuration	00_BR_GPC1	100
GPC2	Global Process Configuration	00_BR_GPC2_0	120
GPP1	Global Process Parameter	00_BR_GPP1	20
GPP2	Global Process Parameter	00_BR_GPP2_0	20
MAP1	Module Assign of Process	00_BR_MAP1_4	20
MCD1	Module Configuration: Device Information	00_BR_MCD1	20
MCM1	Module Configuration: Module Information	00_BR_MCM1	20
MCP1	Module Configuration: Process Information	00_BR_MCP1_1	20
MCS1	Module Configuration: SFC- Information	00_BR_MCS1_1	30
MFO1	Maximum Feedrate Override	00_CR_MFO1_0	20
MFR	Maximum FeedRate	00_CR_MFR_0	20
MRO1	Maximum Rapid Override	00_CR_MRO1_0	20
MSO1	Maximum Spindle Override	00_CR_MSO1_0_1	20
MSS	Maximum Spindle Speed	00_CR_MSS_0_1	20
MTD	Machine Table Data	00_CR_MTD_90_0_0_1_7	20
NEV	NC EVent	00_CR_NEV_0_1	20
NMM	NC MeMory selection	00_CW_NMM Value: 2	10
NPA1	NC PArameter	00_BR_NPA1_01_A00.000	90
NPA2	NC PArameter	00_BR_NPA2_01_A00.000_A00.004	90
NPA3	NC PArameter	00_BR_NPA3_01_A00.000_3	100
NPA4	NC PArameter	00_BR_NPA4_01_A00.000	120
NPS	NC Program Selection	00_CW_NPS_0 Value: 2	10
NTN	Next Tool Number	00_CR_NTN_0	20
NVS	NC Variable Single	00_CR_NVS_0_0	20
OPD1	Optimum Position Distance	00_CR_OPD1_0_2	20
OPD2	Optimum Position Distance by log. Axis No	00_CR_OPD2_3	20
PAC1	Process Axis Configuration Parameter	00_BR_PAC1	10
PAC2	Process Axis Configuration Parameter	00_BR_PAC2_0	20
PFR	Programmed FeedRate	00_CR_PFR_0	20
PPD	Part Program Directory		10
PPN	Part Program NC	00_BR_PPN_1_0_1_1	60
PPP	Part Program Package	00_BA_PPP_1/1 Value: PROGNAM	20
PPS	Part Program Sequence	00_CR_PPS_1_0_1_1	20
PSS	Programmed Spindle Speed	00_CR_PSS_0_1	20
PTC1	Process Tool Management Configuration	00_BR_PTC1	20
PTC2	Process Tool Management Configuration	00_BR_PTC2_0	20
PVF	PLC Variable Formatted	00_CR_PVF_VAR1	20
PVS	PLC Variable Single	00_CR_PVS_ErrorFlg	20

Com.	Description	Example	[ms]
PVT	PLC Variable Type	00_BR_PVT_VAR1	10
SID1	Software Installation Data	00_BR_SID1	30
SLA1	Actual Servo LAg	00_CR_SLA1_0_2	20
SLA2	Actual Servo LAg	00_CR_SLA2_3	20
SLI	PLC (SPS) Long Identification	00_BR_SLI	30
SPA1	Sercos PArameter	00_BR_SPA1_1_S-0-0001_40	120
SPH	Sercos PHase	00_CW_SPH_1 Value: 2	20
SPP	Selected Part Program Number	00_CR_SPP_0	20
TDA1	Tool DAta	00_BR_TDA1_0_M_21	60
TDA2	Tool DAta	00_BR_TDA2_0_1_1	70
TDR1	Tool Data Record of Place	00_CR_TDR1_0_M_21_0	30
TDR2	Tool Data Record	00_CR_TDR2_0_1_1_0	20
TIF	Tool Insert Finish	00_CR_TIF_0_M_25	20
TII	Tool Insert Initiated	00_CR_TII_0_M_25	20
TLB1	TooL Basic Data List	00_BR_TLB1_0_M_1_10_2_5_6_7	380 *2)
TLB2	TooL Basic Data List	00_BR_TLB2_0_2_5_6_7	700 *2)
TLD1	TooL Data of Place	00_CR_TLD1_0_M_1_1_1	20
TLD2	TooL Data of Tool	00_CR_TLD2_0_1_1_0_5	20
TLD3	TooL Data of Place	00_CR_TLD3_0_M_2_1	30
TLD4	TooL Data of Tool	00_CR_TLD4_0_1_1_1	30
TLE1	TooL Edge Data List	00_BR_TLE1_0_1_M_1_3_2_3	260 *2)
TLE2	TooL Edge Data List	00_BR_TLE2_0_1_3_4_5_9	770 *2)
TMV	Tool MoVe	00_CR_TMV_0_M_24_M_25	20
TQE1	Actual TorQuE	00_CR_TQE_0_2	20
TQE2	Actual TorQuE	00_CR_TQE1_0_2	20
TRM	Tool ReMove	00_CR_TRM_0_M_25	20
TRS	Tool ReSet	00_CR_TRS_0_M_25	20
ZOD	Zero Offset Data	00_CR_ZOD_1_0_0_4_1	20
ZOD1	Zero Offset Data	00_CR_ZOD1_1_0_0_4	20
ZOD2	Zero Offset Data	00_CR_ZOD2_1_0_0_4_1	20

Fig. 2-19: Command execution times of the MWCX device group

Command Execution Times for the MSCX Device Group

Com.	Description	Example	[ms]
ASE	Actual System Error	00_CR_ASE	20
CSE	Clear System Error	00_CW_CSE No Value	20
DTY1	Device Type	00_CR_DTY1	60
SPA1	Sercos Parameter	00_BR_SPA1_1_S-0-0001_40	150
SPH	Sercos Phase	00_CW_SPH_1 Value: 2	30

Fig. 2-20: Command execution times of MSCX device groups

Command Execution Times for the MWSX Device Group

Com.	Description	Example	[ms]
ASM1	Active System Fault Message	00_BR_ASM1	60
ASM2	Active System Fault Message	00_BR_ASM2	20
ASM3	Active System Fault Message	00_BR_ASM3_02	10
ASM4	Active System Fault Message	00_BR_ASM4_MWCX	10
ASM5	Active System Fault Message	00_BR_ASM5_74_0	10
CRT	Control ReseT		20
DIS2	Data Identification String PLC Program	00_CR_DIS2	20
DTY1	Device TYpe	00_CR_DTY1	20
MCD1	Module Configuration: Device Information	00_BR_MCD1	10
MCM1	Module Configuration: Module Information	00_BR_MCM1	10
MCP1	Module Configuration: Process Information	00_BR_MCP1_1	
MCS1	Module Configuration: SFC- Information	00_BR_MCS1_1	10
PVF	PLC Variable Formatted	00_CR_PVF_VAR1	20
PVS	PLC Variable Single	00_CR_PVS_ErrorFlg	20
PVT	PLC Variable Type	00_BR_PVT_VAR1	10
SID1	Software Installation Data	00_BR_SID1	20
SLI	PLC (SPS) Long Identification	00_BR_SLI	10

Fig. 2-21: Command execution times of the MWSX device group

Command Execution Times for the MWAX Device Group

Com.	Description	Example	[ms]
AMM7	Active Mechanism Message	01_BR_AMM7	10
APO2	Actual Machine Position	00_CR_APO2_3_1	20
ASM1	Active System Fault Message	00_BR_ASM1	70
ASM2	Active System Fault Message	00_BR_ASM2	60
ASM3	Active System Fault Message	00_BR_ASM3_02	80
ASM4	Active System Fault Message	00_BR_ASM4_MWCX	60
ASM5	Active System Fault Message	00_BR_ASM5_74_0	20
CMA	CMOS RAM ASCII Parameter	00_CR_CMA_10	20
CMF	CMOS RAM Floating Point Parameter	00_CR_CMF_10	20
CMI	CMOS RAM Integer Parameter	00+C13_CR_CMI_10	20
CRT	Control ReseT		50
DCP1	Device Configuration Parameter	00_BR_DCP1	20
DCP2	Device Configuration Parameter	00_BR_DCP2	10
DIS2	Data Identification String PLC Program	00_CR_DIS2	70
DTG2	Distance To Go by log. Axis No	00_CR_DTG2_3_1	20
DTY1	Device Type	00_CR_DTY1	30
MAP1	Module Assign of Process	00_BR_MAP1_4	20
MCD1	Module Configuration: Device Information	00_BR_MCD1	10
MCM1	Module Configuration: Module Information	00_BR_MCM1	10
MCP1	Module Configuration: Process Information	00_BR_MCP1_1	20

Com.	**Description**	**Example**	**[ms]**
MCS1	Module Configuration: SFC- Information	00_BR_MCS1_1	20
PVF	PLC Variable Formatted	00_CR_PVF_VAR1	40
PVS	PLC Variable Single	00_CR_PVS_ErrorFlg	40
PVT	PLC Variable Type	00_BR_PVT_VAR1	10
SID1	Software Installation Data	00_BR_SID1	20

Fig. 2-22: Command execution times of the MWAX device group

Command Execution Times for the MSYX Device Group

Com.	**Description**	**Example**	**[ms]**
ASE	Actual SERCANS Error	00_CR_ASE	
CSE	Clear SERCANS Error	00_CW_CSE No Value	
DTY1	Device Type	00_CR_DTY1	
SID1	Software Installation Data	00_BR_SID1	
SPA1	Sercos Parameter	00_BR_SPA1_1_S-0-0001_40	
SPH	Sercos Phase	00_CW_SPH_1 Value: 2	

Fig. 2-23: Command execution times of the MSYX device group

3 FI Commands - MPCX Device Group (PC)

The following FI commands are valid for the MPCX device group. Always make sure to place device address "XX" before the FI command, e.g. XX_BR_CCP1 (also refer to the chapter "Elements of the FI command").

3.1 Reading of the Currently Set Device Address: ADA

MPCX device group

Designation **ADA** Active **D**evice **A**ddress

Explanation This command is used to read out the currently set device address. The following information is returned:

Type of information	**Status**	**Statement**
Currently set local/far device address		Device address
Local device		Yes/No
Local device address		Device address
System error information		Yes/No
Mechanism error information		Yes/No
Machine key information		4 Byte HEX
Machine key information	valid	Yes/No
Machine status information		4 Byte HEX
Sercans information		4 Byte HEX
Parameter download	running	Yes/No
PLC download	running	Yes/No
Firmware download	running	Yes/No
Offline/Online information		Yes/No
Device simulation	switched on	Yes/No
Device status information		ON/ OFF
Communication channel defined		Yes/No
PLC components available		Yes/No
Monitor mode	active	Yes/No

FI command Read out the currently set device address.

XX_BR_ADA1 **(Single Read)**

XX_BC_ADA1 **(Cyclic Read)**

XX_BB_ADA1 **(Break Cyclic Read)**

Response Structure The following table shows the general structure of the response to the FI command "ADA1".

Line 1	**Column 1**	**...**	**Column 18**

Rexroth
Bosch Group

Value Range/Meaning of Columns

1 =	Currently set local/far device address	[00...99]
2 =	Local device	[YES = local device NO = far device
3 =	Local device address	[00..63]
4 =	System error information	[0 = there is no system error 1 = there is a system error]
5 =	Mechanism error information	[0 = there is no mechanism error 1 = there is a mechanism error]
6 =	Machine key information	[4 byte in HEX coding]
7 =	Machine key information valid?	[0 = not valid, 1=valid]
8 =	Machine status information	[4 byte in HEX coding]
9 =	Sercans information	[4 byte in HEX coding]
10 =	Is parameter download active?	[0 = parameter download not running 1 = parameter download running]
11 =	Is PLC download active?	[0 = PLC download not running 1 = PLC download running]
12 =	Is firmware download active?	[0 = PLC download not running 1 = PLC download running]
13 =	Offline/Online information	[0 = device connection interrupted 1 = device connection O.K.]
14 =	Device simulation switched on?	[0 = NO Simulation mode 1 = simulation mode]
15 =	Current device status information	[0 = Device status=OFF 1 = Device status=ON]
16 =	Communication channel defined	[0 = NO communication channel 1 = Communication channel defined]
17 =	PLC components available	[0 = NO PLC component 1 = PLC component (DOS-Pcl) 2 = PLC component (WIN-Pcl)]
18 =	Monitor mode	[0 = NO monitor mode active 1 = Monitor mode active]

Example ADA1 Read the currently set device address.

FI command		**XX_BR_ADA1**
Line	**Column**	**Answer**
1	1	00
	2	YES
	3	00
	4	0
	5	0
	6	00000000
	7	0
	8	00000000
	9	00000000
	10	0
	11	0
	12	0
	13	1
	14	0
	15	1
	16	1
	17	2
	18	0

3.2 Parameterization of the Bus Configuration: BCI

MPCX device group

Designation **BCI** **B**us **C**onfiguration **I**nfo

Explanation This command only configures the two supported bus configurations Phoenix CMD Tool (**ONLY** for Interbus) and Hilscher Configurator (Interbus, Profibus, DeviceNet, ASI). During this process, the device address, the address of the interface module, and the bus system are selected.

FI command The FI command "**BW_BCI1"** is used to configure the Phoenix CMD tool (**ONLY** for Interbus).

XX_BW_BCI1_(1)_(2)_(3)	**(Single Write)**
(1) = Device address the Phoenix CMD tool is to communicate with	[00..63]
(2) = Address of the interface module	[0..255]
(3) = Selected bus system	[Here ALWAYS 1 = Interbus]

Response Structure The response to the "BW_BCI1" FI command consists of one line with one column.

Line 1	**Column 1**

Value Range/Meaning of Columns 1 = Status message (P_ACK) (P_ACK)

Example BCI1

The Phoenix CMD tool for device 0 is to be configured.

FI command		**XX_BW_BCI1_0_1_0**
Line	**Column**	**Answer**
1	1	(P_ACK)

FI command

The FI command "**BR_BCI1"** is used to read the configuration of the Phoenix CMD tool (**ONLY** for Interbus).

XX_BR_BCI1 **(Single Read)**

Response Structure

The response to the "BR_BCI1" FI command consists of one line with three columns.

Line 1	**Column 1**	**Column 2**	**Column 3**

Value Range/Meaning of Columns

(1) = Device address the Phoenix CMD tool is to communicate with	[0..63]
(2) = Address of the interface module	[0..255]
(3) = Selected bus system	[Here ALWAYS 1 = Interbus]

Example BCI1

The Phoenix CMD tool is configured for device 0.

FI command		**XX_BR_BCI1**
Line	**Column**	**Answer**
1	1	0
	2	0
	3	1

FI command

The FI command "**BW_BCI2"** is used to exit the Phoenix CMD tool (**ONLY** for Interbus) with a defined process ID.

XX_BW_BCI2_(1) **(Single Write)**

(1) = Process ID of the Phoenix CMD tool

Response Structure

The response to the "BW_BCI2" FI command consists of one line with one column.

Line 1	**Column 1**

Value Range/Meaning of Columns

1 = Status message (P_ACK) (P_ACK)

Example BCI2

The Phoenix CMD tool with process ID 212 is to be exited.

FI command		**XX_BW_BCI2_212**
Line	**Column**	**Answer**
1	1	(P_ACK)

FI command

The FI command "**BW_BCI13** is used to configure the Hilscher configurator (Interbus, Profibus, DeviceNet, ASI).

FI command

BW_BCI3_(1)_(2)_(3) **(Single Write)**

(1) = Device address the Hilscher configurator is to communicate with	[00..63]
(2) = Address of the interface module	[0..254 = Address of the selected interface module 255 = Selection of the interface module in the Hilscher configurator]
(3) = Selected bus system	[1 = Interbus 2 = Profibus 3 = DeviceNet 4 = ASI]

Response Structure

The response to the "BW_BCI3" FI command consists of one line with one column.

Line 1	Column 1

Value Range/Meaning of Columns

1 = Status message (P_ACK) (P_ACK)

Example BCI3

The Hilscher configurator is to be set for configuration of the Profibus for device 0. The respective interface module is selected via a menu in the Hilscher configurator.

FI command		**XX_BW_BCI3_0_255_2**
Line	**Column**	**Answer**
1	1	(P_ACK)

FI command

The FI command "**BW_BCI3** is used to read the configuration of the Hilscher configurator (Interbus, Profibus, DeviceNet, ASI).

XX_BR_BCI3 **(Single Read)**

Response Structure

The response to the "BW_BCI3" FI command consists of one line with three columns.

Line 1	Column 1	Column 2	Column 3

Value Range/Meaning of Columns

(1) = Device address the Hilscher configurator is to communicate with	[0..63]
(2) = Address of the interface module	[0..255]
(3) = Selected bus system	[1 = Interbus 2 = Profibus 3 = DeviceNet 4 = ASI]

Example BCI3

The Hilscher configurator is configured to ASI for device 3. Furthermore, the selection of the interface module is to be made by means of a menu in the Hilscher configurator.

FI command		**XX_BR_BCI3**
Line	**Column**	**Answer**
1	1	3
	2	255
	3	4

3.3 Interrupting Function Interface Jobs: BFJ

MPCX Device Group

Designation

BFJ **B**reak-**F**unction-Interface **J**obs

Explanation

This is a means for interrupting tasks or FI jobs. The FI command "BFJ1" interrupts all interface jobs, "BFJ2" interrupts the selected job.

Note: Not all FI jobs can be interrupted with the BFJ command!

FI command

Interrupt all FI jobs that are running.

XX_BR_BFJ1 **(Single Read)**

Response Structure

The following table shows the general structure of the response to the FI command "BFJ1". If FI jobs are running, the response consists of one to n lines (n = the number of FI jobs running), each with two columns.

Line 1...n:	Column 1	Column 2

Value Range/Meaning of Columns

1 = ID of job to be interrupted [01...20]

2 = FI command string.

Example BFJ1

Interrupts all FI jobs that are running.

Note:
The processing of ALL FI jobs that are currently running and that it is possible to interrupt is stopped by this FI command.

Assumption:
The two FI jobs with the job IDs 1 and 2 are running.

FI command		**XX_BR_BFJ1**
Line	**Column**	**Answer**
1	1	01
	2	02_BW_PAA2_C:\DOWNLOAD1.PDL /3
2	1	02
	2	01_BW_PAA2_C:\DOWNLOAD2.PDL/3\/3

FI command

Interrupt the selected FI job.

XX_BR_BFJ2_(1) **(Single Read)**

(1) = ID of job to be interrupted [01...20]

Response Structure

The following table shows the general structure of the response to the FI command "BFJ2". The response consists of a line with two columns.

Line 1	Column 1	Column 2

Value Range/Meaning of Columns

1 = ID of job to be interrupted [01...20]

2 = FI command [String of the FI Command]

Example BFJ2

Interrupts the FI job 01.

Note:
A parameter download job is currently running with the job ID 01 for the device 00.

FI command		**XX_BR_BFJ2_01**
Line	**Column**	**Answer**
1	1	01
	2	00_BW_PAA2_C:\DOWNLOAD1.PDL/3

3.4 Exiting Applications Together with MTGUI: CAM

MPCX device group

Designation **CAM** **C**lose **A**pplication **M**anager

Explanation This command is used for defined exiting of external applications entered in the configuration file IND_DEV.INI under the [KillManager]-entry with the key KillCloseTaskMtguiX=YES (with X as a run-time parameter). In this context, also refer to the description of the SDM1 command.

XX_BW_CAM1 **(Single Write)**

Response Structure The following table shows the general structure of the response to the FI command "BW_CAM1". The number of lines depends on the number of external applications to be exited, 1 line with 3 columns being provided for each external application.

Line 1...n:	**Column 1**	**Column 2**	**Column 3**

Value Range/Meaning of Columns

1 =	Status information on whether the external application has closed correctly in the preset time	[YES = External application closed correctly NO = External application could NOT be closed correctly in the preset time: External application has been "killed"]
2 =	Name of the external application at FI (see log-in name at FI)	[max. 20 ASCII characters]
3 =	Preset time for correct closing in ms	

Example CAM1 Those external applications are to be exited which are entered into configuration file IND_DEV.INI under the [KillManager] entry with the key KillCloseTaskMtguiX=YES (with X=run-time parameter von 1..20).

Assumption: The two external applications "VBDEMO.EXE" and "PRISMA" are to be exited together with the MTGUI.

FI command		**XX_BW_CAM1**
Line	**Column**	**Answer**
1	1	YES
	2	VBDEMO.EXE
	3	8000
2	1	YES
	2	PRISMA
	3	10000

3.5 Outputting the Local Device Configuration: CCP

MPCX Device Group

Designation

CCP **C**ell **C**onfiguration **P**arameter

Explanation

The configuration settings are read in from the "IND_DEV.INI" file. The configuration of the individual local communication addresses and the settings of the various Bosch Rexroth devices are determined in this file (see Chapter 5 "Installation").

FI command

Output the configuration settings of all devices defined in the "IND_DEV.INI" file.

XX_BR_CCP1 **(Single Read)**

XX_BC_CCP1 **(Cyclic Read)**

Response Structure

The following table shows the general structure of the response to the FI command "CCP1". The response consists of a maximum of n=16 lines (n=16 configurable devices), each with 15 columns.

Line 1...n:	Column 1	...	Column 15

Value Range/Meaning of Columns

1 =	Local device address	IND_DEV.INI entry: [DeviceAddrX]
2 =	Device name	IND_DEV.INI entry: DeviceName=
3 =	Device type	IND_DEV.INI entry: DeviceTyp=
4 =	PLC support	IND_DEV.INI entry: PLC=
5 =	Device status	IND_DEV.INI entry: DeviceStatus=
6 =	Assignment of a simulation pair	IND_DEV.INI entry: DeviceAssign=
7 =	Device mode	IND_DEV.INI entry: MtvncMode=
8 =	Communication channel	IND_DEV.INI entry: : [CommAddrX]
9 =	Description of the communication channel	IND_DEV.INI entry: CommStr=
10 =	Timeout value	IND_DEV.INI entry: Timeout=
11 =	Device group	(see Chapter 6.1 "Identifier")
12 =	PLC component type	IND_DEV.INI entry: Component type1=
13 =	PLC component type	IND_DEV.INI entry: Component type2=
14 =	Device log	IND_DEV.INI entry: DeviceProtocol=
15 =	Device simulation	IND_DEV.INI entry: DeviceSimulation=

Example CCP1 Read the configuration settings of all devices defined in the "IND_DEV.INI" file.

Assumption:
The following device types have been defined:

- Local device address 00: SERCANS-A
- Local device address 15: MTC200-P

FI command		**XX_BR_CCP1**
Line	**Column**	**Answer**
1	1	00
	2	Pressure barrel drive
	3	SERCANS-A
	4	NO
	5	ON
	6	NO
	7	OFF
	8	4
	9	V24,COM2,19200,EVEN,RS232,TCOFF
	10	3500
	11	MSCX
	12	NONE
	13	NONE
	14	CNC
	15	OFF
2	1	15
	2	Transport unit
	3	MTC200-P
	4	YES
	5	ON
	6	NO
	7	OFF
	8	1
	9	DPR,$D000,$0000,$2000,RAM0,TCON
	10	3500
	11	MTCX
	12	MTS-P01.02
	13	MTC-P
	14	CNC
	15	OFF

FI command

Output the configuration settings of the selected device type.

XX_BR_CCP2	**(Single Read)**
XX_BC_CCP1	**(Cyclic Read)**
(1)= Device type	[MTC200-P-G2, MTC200-R-G2, MTVNC, SERCANS-A, SERCANS-P, ISP200-P-G2, ISP200-R-G2, TRA200-P, TRA200-R, MTA200-P]

Response Structure

The following table shows the general structure of the response to the FI command "CCP2". The response consists of a maximum of n=16 lines (n=16 configurable devices), each with 15 columns.

Line 1...n:	Column 1	...	Column 15

Value Range/Meaning of Columns

1 =	Local device address	IND_DEV.INI entry: [DeviceAddrX]
2 =	Device name	IND_DEV.INI entry: [DeviceName=
3 =	Device type	IND_DEV.INI entry: [DeviceTyp=
4 =	PLC support	IND_DEV.INI entry: PLC=
5 =	Device status	IND_DEV.INI entry: DeviceStatus=
6 =	Assignment of a simulation pair	IND_DEV.INI entry: DeviceAssign=
7 =	Device mode	IND_DEV.INI entry: MtvncMode=
8 =	Communication channel	IND_DEV.INI entry: : [CommAddrX]
9 =	Description of the communication channel	IND_DEV.INI entry: CommStr=
10 =	Timeout value	IND_DEV.INI entry: Timeout=
11 =	Device group	(see Chapter 6.1 "Identifier")
12 =	PLC component type	IND_DEV.INI entry: Component type1=
13 =	CNC component type	IND_DEV.INI entry: Component type2=
14 =	Device log	IND_DEV.INI entry: DeviceProtocol=
15 =	Device simulation	IND_DEV.INI entry: DeviceSimulation=

Example CCP2

Read the configuration settings of the defined devices of type SERCANS-A.

Assumption:
The following device types have been defined:

- Local device address 00: SERCANS-A
- Local device address 03: MTA200-P
- Local device address 15: MTC200-P

FI command		XX_BR_CCP2_SERCANS-A
Line	**Column**	**Answer**
1	1	00
	2	Pressure barrel drive
	3	SERCANS-A
	4	NO
	5	ON
	6	NO
	7	OFF
	8	4
	9	V24,COM2,19200,EVEN,RS232,TCOFF
	10	3500
	11	MSCX
	12	NONE
	13	NONE
	14	CNC
	15	OFF

FI command

Output the configuration data of the devices that are addressed via the stipulated communication channel.

XX_BR_CCP3_(1) **(Single Read)**

XX_BC_CCP3_(1) **(Cyclic Read)**

(1) = Communication channel IND_DEV.INI entry: : [CommAddrX]

Response Structure

The following table shows the general structure of the response to the FI command "CCP3". The response consists of a maximum of n=16 lines (n=16 configurable devices), each with 15 columns.

Line 1...n:	Column 1	...	Column 15

Rexroth
Bosch Group

Value Range/Meaning of Columns

1 =	Local device address	IND_DEV.INI entry: [DeviceAddrX]
2 =	Device name	IND_DEV.INI entry: [DeviceName=
3 =	Device type	IND_DEV.INI entry: [DeviceTyp=
4 =	PLC support	IND_DEV.INI entry: PLC=
5 =	Device status	IND_DEV.INI entry: DeviceStatus=
6 =	Assignment of a simulation pair	IND_DEV.INI entry: DeviceAssign=
7 =	Device mode	IND_DEV.INI entry: MtvncMode=
8 =	Communication channel	IND_DEV.INI entry: : [CommAddrX]
9 =	Description of the communication channel	IND_DEV.INI entry: CommStr=
10 =	Timeout value	IND_DEV.INI entry: Timeout=
11 =	Device group	(see Chapter 6.1 "Identifier")
12 =	PLC component type	IND_DEV.INI entry: Component type1=
13 =	CNC component type	IND_DEV.INI entry: Component type2=
14 =	Device log	IND_DEV.INI entry: DeviceProtocol=
15 =	Device simulation	IND_DEV.INI entry: DeviceSimulation=

Example CCP3

Read the configuration data of the devices that are addressed via communication channel 1.

Assumption:
The following device types have been defined:

- Communication channel 4: SERCANS-A
- Communication channel 5: MTA200-P
- Communication channel 1: MTC200-P

FI command		XX_BR_CCP3_1
Line	**Column**	**Answer**
1	1	15
	2	Transport unit
	3	MTC200-P
	4	YES
	5	ON
	6	NO
	7	OFF
	8	1
	9	DPR,$D000,$0000,$2000,RAM0,TCON
	10	3500
	11	MTCX
	12	MTS-P01.2
	13	MTC-P
	14	CNC
	15	OFF

FI command

Output the configuration data of the devices that are addressed via the stipulated communication channel.

XX_BR_CCP4_(1)	**(Single Read)**
XX_BC_CCP4_(1)	**(Cyclic Read)**
(1) = Device group	[MTCX, MSCX, MISX, MTRX, MTAX] (see Chapter 6.1 "Identifier")

Response Structure

The following table shows the general structure of the response to the FI command "CCP4". The response consists of a maximum of n=16 lines (n=16 configurable devices), each with 15 columns.

Line 1...n:	**Column 1**	**...**	**Column 15**

Value Range/Meaning of Columns

1 =	Local device address	IND_DEV.INI entry: [DeviceAddrX]
2 =	Device name	IND_DEV.INI entry: [DeviceName=
3 =	Device type	IND_DEV.INI entry: [DeviceTyp=
4 =	PLC support	IND_DEV.INI entry: PLC=
5 =	Device status	IND_DEV.INI entry: DeviceStatus=
6 =	Assignment of a simulation pair	IND_DEV.INI entry: DeviceAssign=
7 =	Device mode	IND_DEV.INI entry: MtvncMode=
8 =	Communication channel	IND_DEV.INI entry: : [CommAddrX]
9 =	Description of the communication channel	IND_DEV.INI entry: CommStr=
10 =	Timeout value	IND_DEV.INI entry: Timeout=
11 =	Device group	(see Chapter "Identifier")
12 =	PLC component type	IND_DEV.INI entry: Component type1=
13 =	CNC component type	IND_DEV.INI entry: Component type2=
14 =	Device log	IND_DEV.INI entry: DeviceProtocol=
15 =	Device simulation	IND_DEV.INI entry: DeviceSimulation=

Example CCP4

Read the configuration settings of the defined MSCX devices.

Assumption:
The following device groups have been defined:

- Local device address 00: MSCX
- Local device address 03: MTCX

FI command		**XX_BR_CCP4_MSCX**
Line	**Column**	**Answer**
1	1	00
	2	Pressure barrel drive
	3	SERCANS-A
	4	NO
	5	ON
	6	NO
	7	OFF
	8	4
	9	V24,COM2,19200,EVEN,RS232,TCOFF
	10	3500
	11	MSCX
	12	NONE
	13	NONE
	14	CNC
	15	OFF

FI command

Output the configuration data of the device that is addressed via the stipulated device address.

XX_BR_CCP5_(1)	**(Single Read)**
XX_BC_CCP5_(1)	**(Cyclic Read)**
(1) = Device address	[00...63]

Response Structure

The following table shows the general structure of the response to the FI command "CCP5". The response consists of a line with 15 columns.

Line 1...n:	**Column 1**	**...**	**Column 15**

Value Range/Meaning of Columns

1 =	Local device address	IND_DEV.INI entry: [DeviceAddrX]
2 =	Device name	IND_DEV.INI entry: [DeviceName=
3 =	Device type	IND_DEV.INI entry: [DeviceTyp=
4 =	PLC support	IND_DEV.INI entry: PLC=
5 =	Device status	IND_DEV.INI entry: DeviceStatus=
6 =	Assignment of a simulation pair	IND_DEV.INI entry: DeviceAssign=
7 =	Device mode	IND_DEV.INI entry: MtvncMode=
8 =	Communication channel	IND_DEV.INI entry: : [CommAddrX]
9 =	Description of the communication channel	IND_DEV.INI entry: CommStr=
10 =	Timeout value	IND_DEV.INI entry: Timeout=
11 =	Device group	(see Chapter 6.1 "Identifier")
12 =	PLC component type	IND_DEV.INI entry: Component type1=
13 =	CNC component type	IND_DEV.INI entry: Component type2=
14 =	Device log	IND_DEV.INI entry: DeviceProtocol=
15 =	Device simulation	IND_DEV.INI entry: DeviceSimulation=

Example CCP5

Read the configuration settings of device address 00.

Assumption:
The following local device addresses have been defined:

- Local device address 00: MSCX
- Local device address 03: MTCX

FI command		**XX_BR_CCP5_00**
Line	**Column**	**Answer**
1	1	00
	2	Pressure barrel drive
	3	SERCANS-A
	4	NO
	5	ON
	6	NO
	7	OFF
	8	4
	9	V24,COM2,19200,EVEN,RS232,TCOFF
	10	3500
	11	MSCX
	12	NONE
	13	NONE
	14	CNC
	15	OFF

3.6 Setting and Reporting of a Device Change: CDA

MPCX device group

Designation

CDA **C**hange **D**evice **A**ddress

Explanation

This command is used to report a device change to other applications via the SYS message handling function of the Function Interface. At the same time, the newly set device address is saved for an FI restart.

The SYS message MSG_CHDEVADDR (define in INDIF000.H) with the additional information [device address WHEREFROM –DeviceAddress WHERETO] is triggered.

FI command

XX_BW_CDA1_(1)_{(2)} **(Single Write)**

(1) = Device address to which the change is to be made

(2) = Device address which has been set before (optional parameter)

Response Structure

The response to the "CAD1" FI command consists of one line with one column.

Line 1	**Column 1**

Value Range/Meaning of Columns

1 = Status message (P_ACK) (P_ACK)

Example CDA1

The device change from device 0 to device 5 is to be reported and saved.

FI command		**XX_BW_CDA1_5_0**
Line	**Column**	**Answer**
1	1	(P_ACK)

This command also triggers the SYS message MSG_CHDEVADDR with the additional information 00-05.

Note! If the optional second parameter is not entered, the former device address CANNOT be supplied.

Example CDA1

The device address is to be changed to device 5.

FI command		**XX_BW_CDA1_5**
Line	**Column**	**Answer**
1	1	(P_ACK)

This command also triggers the SYS message MSG_CHDEVADDR with the additional information XX-05.

3.7 Reading the FI Communication Error Counts: CEI

MPCX Device Group

Designation

CEI **C**ommunication **E**rror **I**nfo

FI command

Reading the counts for the communication errors recorded in the protocol.

XX_BR_CEI1 **(Single Read)**

Response Structure

The following table shows the general structure of the response to the FI command "CEI1". A line of 5 columns is output.

Line 1	**Column 1**	**...**	**Column 5**

Value Range/Meaning of Columns

1 = Error counter: PC side	Contains the communication error occurred until that time registered by the PC.
2 = Error counter: SIO side	Contains the communication error occurred until that time registered by the SIO.
3 = Error counter: internal timeout	Contains the internal timeouts occurred until that time – which are compensated through FI, if applicable.
4 = Error counter: number of dispatches of repeat telegrams	Contains the repeat telegrams occurred until that time.
5 = Error counter: timeout	Contains the timeouts occurred until that time – are signaled to the application.

Example: CEI1 Supply the current counts for communication errors.

FI command		XX_BR_CEI1
Line	**Column**	**Answer**
1	1	1
	2	0
	3	0
	4	1
	5	0

3.8 Controlling the FI Log-In Process: CFL

MPCX device group

Designation **CFL** **C**ontrol of **F**unction Interface **L**ogin

Explanation This command is used to define whether log-in at the Function Interface (FI) is permissible or not.

Note: As write value, a list of FI task names (separated by commas) can be transferred which are able to perform FI log-in despite a blocked FI log-in process.

FI command **XX_BW_CFL1_(1)** **(Single Write)**

(1) = Control info for the FI log-in process

[0 = BLOCK FI log-in process
1 = ENABLE FI log-in process]

Response Structure The response to the "CFL1" FI command consists of one line with one column.

Line 1	Column 1

Value Range/Meaning of Columns 1 = Status message (P_ACK) (P_ACK)

Example CFL1 The FI log-in process is to be blocked for **ALL** further applications. NO write value is transferred.

FI command		XX_BW_CFL1_0
Line	**Column**	**Answer**
1	1	(P_ACK)

Example CFL1 The FI log-in process is to be blocked for **ALL** further applications with the exception of VBDEMO.EXE.

FI command		Value to be written: VBDEMO.EXE XX_BW_CFL1_0
Line	**Column**	**Answer**
1	1	(P_ACK)

Explanation Here, it can be read out whether the log-in process at the Function Interface (FI) is currently permissible or not. Additionally, a list of the FI task names is provided which are permitted to log into the FI despite blocking of the FI log-in process.

FI command **BR_CFL1** **(Single Read)**

Response Structure

The response to the "CFL1" FI command consists of one line with two columns.

Line 1	Column 1	Column 2

Value Range/Meaning of Columns

1 =	Control info for the FI log-in process	[0 = currently, NO FI log-in process is permitted 1 = FI log-in process is permitted
2 =	If applicable, a list of the FI task names which are permitted to log into the FI despite blocking of the FI log-in process.	[-- = There is no FI task name list]

Example CFL1

The current control information for the FI log-in process is to be read out. There is NO FI task list.

FI command		**XX_BR_CFL1**
Line	**Column**	**Answer**
1	1	1
	2	--

Example CFL1

The current control information for the FI log-in process is to be read out. There is an FI task list.

FI command		**XX_BR_CFL1**
Line	**Column**	**Answer**
1	1	1
	2	VBDEMO.EXE,IND400T1.EXE

3.9 Commands for Executing WIN32 Applications: CPR

MPCX Device Group

Designation **CPR** **C**reate **PR**ocess

Explanation

WIN32 applications can be executed with this FI command. These applications may or may not be logged in the FI.

FI command

Execute a WIN32 application that is logged on in the FI.

XX_BW_CPR1_(1)_(2)_(3)	**(Single Write)**
(1) = Complete EXE name	Complete physical path name for the WIN32 application that is to be executed
(2) = Min Info	Control information as to whether or not the current screen window (output window) is to be minimized. The following applies: 0 = do not minimize 1 = minimize
(3) = Wait Info	Control information as to how the output window is focused; here, the following applies: 0 = Re-focussing with the SFW2 command 1 = Automatic re-focussing on termination of the WIN32 application

Response Structure

As this concerns a command there is no response data.

Example CPR1

The WIN32 application "VBDEMO.EXE" is executed via the FI. The output window is minimized and automatically focused again after "VBDEMO.EXE" has ended.

Assumption:
The VBDEMO.EXE program is in the subdirectory D:\Programs\Indramat\Mtgui\bin.

FI command	**XX_BW_CPR1_D:\Programs\Indramat\Mtgui\bin\VBDEMO.EXE_1_1**

FI command

Execute a WIN32 application, that is NOT logged on in the FI.

XX_BW_CPR2_(1)_(2)	**(Single Write)**
(1) = Complete EXE name	Complete physical path name for the WIN32 application that is to be executed.
(2) = Min Info	Control information as to whether or not the current screen window (output window) is to be minimized. The following applies: 0 = do not minimize 1 = minimize

Response Structure

As this concerns a command there is no response data.

Example CPR2

To start Windows Task Manager via the function interface. The output window is minimized and automatically focused again after "TASKMGR.EXE" has ended.

Assumption:
The program "TASKMGR.EXE" is in the subdirectory C:\Winnt\System32.

FI command	**XX_BW_CPR2_C:\Winnt\System32\Taskmgr.exe_1**

FI command

Execute a WIN32 application, that is NOT logged on in the FI.

XX_BW_CPR3_(1)_(2)	**(Single Write)**
(1) = Complete EXE name	Complete physical path name for the WIN32 application that is to be executed.
(2) = Min Info	Control information as to whether or not the current screen window (output window) is to be minimized. The following applies: 0 = do not minimize 1 = minimize

Response Structure

As this is a command, a result with one line and one column is returned.

Line 1	**Column 1**

Value Range/Meaning of Columns

1 = Process ID of the operating system

Example CPR3

To start Windows Task Manager via the function interface. The output window is minimized and automatically focused again after TASKMGR.EXE has ended.

Assumption:
The program TASKMGR.EXE is in the subdirectory C:\Winnt\System32.

FI command		**XX_BW_CPR3 C:\Winnt\System32\Taskmgr.exe_1**
Line	**Column**	**Answer**
1	1	179

3.10 List of Currently Available Window Names: CWL

MPCX device group

Designation **CWL** **C**urrent **W**indow **L**ist

Explanation This command is used to read out the list of the currently available window names. Additionally, the information on the process ID, process name and thread ID of each respective window is returned. If the process name CANNOT be defined, the process ID is given once more instead of the process name.

FI command **XX_BR_CWL1_{(1)}** **(Single Read)**

(1) = Optional parameter; it defines whether the window names are to be returned in alphabetic order or not. The following applies:

0 = NOT in alphabetic order

1 = In alphabetic order

If NO parameter has been passed, alphabetic order is NOT set as a default.

Response Structure The response to the "BR_CWL1" FI command consists of n lines with four columns. For each currently available window, 1 line is returned.

Line n	Column 1	Column 2	Column 3	Column 4

Value Range/Meaning of Columns

1 =	Window name	ASCII character string
2 =	Process ID	According to the window operating system (see Task manager)
3 =	Process name	According to the window operating system (see Task manager)
4 =	Thread ID	According to the window operating system (see Task manager)

Example CWL1 The currently available window names are read out in alphabetical order. Here, only the first 5 window names are shown to give an overview.

FI command		**XX_BR_CWL1_1**
Line	**Column**	**Answer**
1	1	COMINTFC /C=+G10 07.31 MAY 5 2003
1	2	140
1	3	COMINTFC.exe
1	4	139
2	1	DISINTFC
2	2	40
2	3	DISINTFC.EXE
2	4	56
3	1	FUNCTION INTERFACE STARTUP
3	2	40
3	3	DISINTFC.EXE
3	4	47
4	1	FUNCTION INTERFACE DEMO PROGRAM
4	2	122
4	3	VBDEMO.EXE
4	4	123
5	1	LIST OF ACTIONS
5	2	40
5	3	DISINTFC.EXE
5	4	47

3.11 Indication of a List Box of Selection Options: DCN

MPCX device group

Designation **DCN** **D**ialog **C**o**N**trol List box

Explanation This command is used to open a user-defined dialog list box to select a certain entry. As a return value, the line index for the selected dialog list box entry is provided. The dialog list box is parameterized by means of the write value. The following parameters are configured through the write value:

1 = Number of data lines (max. 50 lines)
2 = Text for the dialog heading (max. 80 characters)
3 = Text for the table heading (max. 80 characters)
4 = List of the individual data lines (max. 50 lines)

There are 2 ways to return the write value:

1. In INDIFX00.h, the data structure "dcn1_write_data_type" is defined which is used as a BINARY write value.

2. As ASCII/ANSI write value, the single data elements are separated by "|" (0x7C Pipe).

FI command **XX_BW_DCN1** **(Single Write)**

Response Structure

The response to the "DCN1" FI command consists of one line with one column.

Line 1	Column 1

Value Range/Meaning of Columns

1 =	Selected dialog list box entry	[-1 = Dialog list box with <Cancel> NO selection made > -1 = Selected dialog list box Entry starting with index 0]

Example DCN1

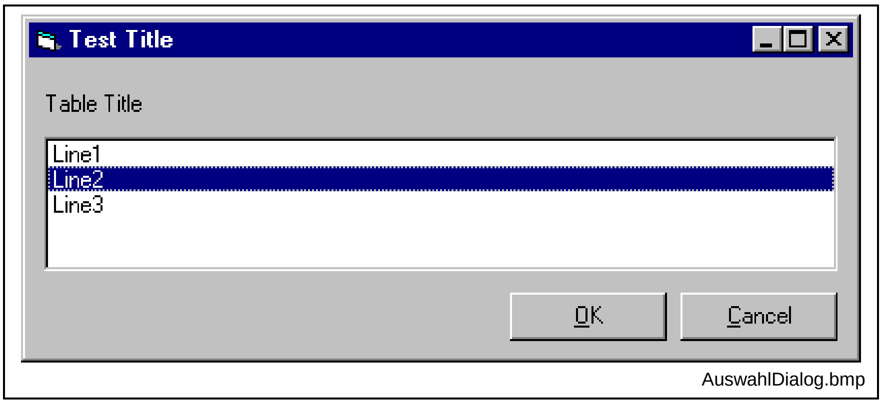
A dialog list box with the following parameters is to be opened:

- 3 data lines
- Dialog heading (test heading)
- Table heading (table heading)
- Data lines (line1,line2,line3)

Result:"Line2" has been selected (index 1).

<table>
<tr><td colspan="2">FI command</td><td>Value to be written:
3|Test Title|Table Title|Line1|Line2|Line3
XX_BW_DCN1</td></tr>
<tr><td>Line</td><td>Column</td><td>Answer</td></tr>
<tr><td>1</td><td>1</td><td>1</td></tr>
</table>

See below for an illustration of the dialog box configured before. The selection bar is on line 2, corresponding to Index 1 in the answer.



AuswahlDialog.bmp

Fig. 3-1: Selection dialog

3.12 Removing Function Interface Jobs: DFJ

MPCX Device Group

Designation **DFJ** **D**elete **F**unction-Interface **J**obs

Explanation

Jobs, also referred to as FI jobs, are removed from the management structure of the function interface. These are jobs that have either the status "READY" or "ERROR" . The FI command "DFJ1" removes all interface jobs; "DFJ2" removes the selected job.

FI command

Remove all FI jobs from the management structure of the function interface.

XX_BR_DFJ1	**(Single Read)**
XX_BC_DFJ1	**(Cyclic Read)**

Response Structure

The following table shows the general structure of the response to the FI command "DFJ1". The response consists of a maximum of n=19 lines (n=19 maximum number of FI jobs), each with two columns.

Line 1...n:	Column 1	Column 2

Value Range/Meaning of Columns

1 = Deleted job ID [01...20]

2 = FI command

Example DFJ1

Delete all FI jobs.

Assumption:
An NC program has been transferred successfully into the device (control unit) using the FI command "NCA1" (see FI commands for the MWCX device group).

- Job ID of the NC download program: 01

FI command		**XX_BR_DFJ1**
Line	**Column**	**Answer**
1	1	01
	2	02_BR_NCA1_"D:\Download.ini" /3

FI command

Remove the selected FI job from the management structure of the function interface.

XX_BR_DFJ2_(1) **(Single Read)**

XX_BC_DFJ1 **(Cyclic Read)**

(1) = Job ID [01...20]

Response Structure

The following table shows the general structure of the response to the FI command "DFJ2". The response consists of one line with 13 columns.

Line 1	Column 1	Column 2

Value Range/Meaning of Columns

1 = Deleted job ID [01...20]

2 = FI command [string, in accordance to chapter entitled "Elements of the FI Command"]

Example DFJ2

Delete the FI job 01.

Assumption:
An NC program has been transferred successfully into the device (control unit) using the FI command "NCA1" (see FI commands for the MWCX device group).

- Job ID of the NC download program: 01

FI command		**XX_BR_DFJ2_01**
Line	**Column**	**Answer**
1	1	01_BR_NCA1_"D:\Download.ini" /3

3.13 Deleting of the FI Command Stack Administration: DFS

MPCX Device Group

Designation **DFS** **D**elete **IF** Command **S**tack

Explanation

This FI command deletes the FI command stack administration. As a write value, a reference information string must be transmitted in the DataTransfer() function which is supplied as reference information with the SYS message "MSG_MESSAGECH".

FI command **XX_BW_DFS1** **(Single Write)**

Response Structure The response to the "DFS1" FI command consists of one line with one column.

Line 1	Column 1

Value Range/Meaning of Columns

1 = Status message (P_ACK) (P_ACK)

Example DFS1 Deleting of the FI Command Stack Administration. As a write value, a reference information string must be transmitted in the DataTransfer() function which is supplied as reference information with the SYS message "MSG_MESSAGECH".

FI command		Value to be written: Reference information string XX_BW_DFS1
Line	**Column**	**Answer**
1	1	(P_ACK)

3.14 Static Device Information: DIF

MPCX device group

Designation **DIF** **D**evice **I**n**F**ormation

Explanation Static device information and network information is read according to the "IND_DEV.INI" and "FAR_DEV.INI" files.

FI command Reading of the static device information and network information of a selected device.

BR_DIF1 **(Single Read)**

BC_DIF1 **(Cyclic Read)**

BB_DIF1 **(Break Cyclic Read)**

Response Structure The following table shows the general structure of the response to the "DIF1" FI command. The response consists of one line with 24 columns.

Line 1	Column 1	...	Column 24

Value Range/Meaning of Columns

1 =	Local/far device address	[00..63]
2 =	Device name	IND_DEV.INI entry: DeviceName=
3 =	Device type	IND_DEV.INI entry: DeviceType=
4 =	PLC support	IND_DEV.INI entry: PLC=
5 =	Device status	IND_DEV.INI entry: DeviceStatus=
6 =	Assignment of a simulation pair	IND_DEV.INI entry: DeviceAssign=
7 =	Device mode	IND_DEV.INI entry: MtvncMode=
8 =	Communication channel	IND_DEV.INI entry: [CommAddrX]
9 =	Description of the communication channel	IND_DEV.INI entry: CommStr=
10 =	Timeout value	IND_DEV.INI entry: Timeout=
11 =	Device group	(see Chapter 6.1 "Identifier")
12 =	PLC component type	IND_DEV.INI entry: Component type1=
13 =	CNC component type	IND_DEV.INI entry: Component type2=

14 =	Device log	IND_DEV.INI entry: DeviceProtocol=
15 =	Device simulation	IND_DEV.INI entry: DeviceSimulation=
16 =	Not yet assigned	[--]
17 =	Not yet assigned	[--]
18 =	Not yet assigned	[--]
19 =	Not yet assigned	[--]
20 =	Network ON/OFF	[ON = Network active OFF = No network active]
21 =	Network name	Max. 28 ASCII characters
22 =	PC number	[00..99,XX]
23 =	PC name	Max. 255 ASCII characters
24 =	Local device address	[00..63]

Example DIF1 Read the static device information of device 1.

FI command		**XX_BR_DIF1**
Line	**Column**	**Answer**
1	1	00
	2	Pressure barrel drive
	3	PC
	4	NO
	5	ON
	6	NO
	7	OFF
	8	4
	9	V24,COM2,19200,EVEN,RS232,TCOFF
	10	3500
	11	MPCX
	12	NONE
	13	NONE
	14	CNC
	15	OFF
	16	--
	17	--
	18	--
	19	--
	20	ON
	21	PC network 1
	22	29
	23	BTV20-STATION-LINKS
	24	00

3.15 Command for Terminating WIN32 Applications: DPR

MPCX Device Group

Designation **DPR** **D**elete **PR**ocess

Explanation WIN32 applications that have been logged in the FI **AND** that are processing the termination event can be terminated with this FI command. See LogInIf() description.

FI command Terminate a WIN32 application that is logged on in the FI **AND** that is processing a termination event.

XX_BW_DPR1_(1)	**(Single Write)**
(1) = LogInIf Login name	This refers to the login name entered at the FI (LogInIf()) during the login procedure

Response Structure As this concerns a command there is no response data.

Example DPR1 The WIN32 application "VBDEMO.EXE" that is running is terminated via the function interface. The FI command is carried out by any WIN32 application logged in the FI.

FI command	**XX_BW_DPR1_VBDEMO.EXE**

Explanation WIN32 applications can be exited with this FI command. To exit a process, the process ID is required.

FI command Exiting of a WIN32 application with a known process ID.

Note: Process ID: see FI command CPR3

XX_BW_DPR2_(1)	**(Single Write)**
(1) = Process ID	This is the process ID the operating system has assigned to the WIN32 application (as can be seen from the Task manager "PID")

Response Structure A result with one line and one column is returned.

Line 1	**Column 1**

Value Range/Meaning of Columns 1 = Status message (P_ACK) (P_ACK)

Example DPR2 The current WIN32 application with process ID 179 is exited.

FI command		**XX_BW_DPR2_179**
Line	**Column**	**Answer**
1	1	(P_ACK)

3.16 Error Information: ERI

MPCX Device Group

Designation **ERI** **ER**ror **I**nformation

Explanation Returns the error text and the additional text of an FI error code or a Windows NT error code.

FI command Read error text and additional text.

XX_BR_ERI1_(1)_(2)	**(Single Read)**
(1) = Error class	[1 = NACK error number, 2 = FI error code 3 = reserved 4 = Windows NT error code 5 = Sercos error text 6 = Visual Motion error text 7 = SIS-Header error text]
(2) = Error number	[LONG]

Response Structure The following table shows the general structure of the response to the FI command "ERI". Two lines, each with one column, are outputted. Line 1 contains the error text and line 2 contains the additional text.

Lines 1..2	Column 1

Meaning of the Column

1 = Error text	[language-dependent]
2 = Additional text	[language-dependent]

Example ERI

- Read the error text including the additional error text with error number 26.

FI command		**XX_BR_ERI1_1_26**
Line	**Column**	**Answer**
1	1	Error in mathematical expression.
2	1	Check mathematical expression. Correct NC program and re-transmit

3.17 Commands for Processing of Files: FCD

MPCX device group

Designation **FCD** File Comman**D**

FI command This command is used to delete a file. The complete file name must be returned.

XX_BW_FCD1_(1) **F**ile**C**omman**d**

(1) = Complete file name

Response Structure One line with 1 column is output, the full file name of the deleted file being returned.

Line 1	Column 1

Value Range/Meaning of Columns 1 = Complete file name of the deleted file

Note: File and path details must be enclosed in inverted commas.

Example FCD1

Delete the file D:\PROGRAM FILES\TEMP\DATA.TXT.

FI command		**XX_BW_FCD1_"D:\PROGRAM FILES\TEMP\DATA.TXT"**
Line	**Column**	**Answer**
1	1	D:\PROGRAM FILES\TEMP\DATA.TXT

3.18 Far Configuration Parameter: FCP

MPCX Device Group

Designation

FCP Far Device **C**onfiguration **P**arameter

Explanation

The FI command "FCP" returns the list of the addressable devices on the PC. Differentiation is made between two cases (A and B):

- PC is in the PC network and
- PC is standalone.

Case A
PC is in the PC Network

The list of FarDevices defined in the network configuration data on the PC is read (see "FAR_DEV.INI" file). In addition the local devices that are not defined as FarDevices are output.

Case B
PC stand-alone

The list of local devices is outputted if one or more of the following points apply:

- There is no network configuration data on the PC (see "FAR_DEV.INI" file).
- The PC has been disabled in the network configuration data, or
- The "PC Network Active" option is not switched on in the system configurator.

FI command

Read out the addressable devices on the PC.

XX_BR_FCP1{_(1))	**(Single Read)**
(1) = Device selection	[L= only local, F= only FAR] ! Optional !

Read-out of the addressable devices on the PC; but only applies to devices from the stipulated device groups:

XX_BR_FCP2_(1){_(2)}	**(Single Read)**
(1) = Device group	[MPCX, MTCX, MISC, MTAX, MTRX]
(2) = Device selection	[L= only local, F= only FAR] ! Optional !

Read-out of the addressable devices on the PC; however, only applies to devices of the stipulated device type:

XX_BR_FCP3_(1){_(2)}	**(Single Read)**
(1)= Device type	[MTC200-P-G2, MTC200-R-G2, MTVNC, SERCANS-A, SERCANS-P, ISP200-P-G2, ISP200-R-G2, TRA200-P, TRA200-R, MTA200-P]
(2) = Device selection	[L= only local, F= only FAR] ! Optional !

Response Structure

The following table shown the general structure of the FI commands "FCP1", "FCP2" and "FCP3". The number of lines depends on the actual configuration.

Result when network configuration data is available:

Line 1...n:	**Column 1**	**...**	**Column 10**

Value Range/Meaning of the Columns

1 = FarDevice address	[00...99]
2 = Device name	[max. 28 ASCII characters]
3 = Device type	[MTC200-P-G2, MTC200-R-G2, MTVNC, SERCANS-A, SERCANS-P, ISP200-P-G2, ISP200-R-G2, TRA200-P, TRA200-R, MTA200-P]
4 = Local device address	[00...63]
5 = PC No.	[00...99, XX]
6 = Local device	[YES, NO, --]
7 = Device status	ON, OFF
8 =Assignment of a simulation pair.	[00...63, NO]
9 = Device group	[MPCX, MTCX, MISC, ...]
10 = Online	[YES, NO, --]

Explanation of Column 1 FarDevice Address

The contents of Column 1 can always be used to address the local as well as the far (remote) devices. A generic application must have the value of a device address within the FI command.

Explanation of Column 7 Device Status

In case A, the "Disable" entry from the "FAR_DEV.INI" file is evaluated. The following assignment applies:

- ON if "Disable = NO" or if the "Disable" entry is missing
- OFF if "Disable = YES" or
- OFF if the PC is disabled.

	FarDevice, Disable = YES	**FarDevice, Disable = NO**
PC, Disable = YES	OFF	OFF
PC, Disable = NO	OFF	ON

Note: If a PC is disabled then its corresponding devices are also in the "Disable" status.

Explanation of Column 10 Online?

This column indicates whether there is currently a connection to the PC via which the device can be addressed. Differentiation is made between 3 possible cases:

- YES = The network connection to the PC is active
- NO = The network connection is down (interrupted).
- -- = The network connection has not yet been completely checked.

Note: YES is always output for B.

Example FCP1
Case A

Read the network configuration of all devices defined in the "FAR_DEV.INI" and "IND_DEV.INI" files.

Assumption:
The following device types have been defined:

- Device address 15: MTCNC
- Device address 11: MTVNC
- Device address 12: MTVNC

FI command		XX_BR_FCP1
Line	**Column**	**Answer**
1	1	15
	2	Drill left
	3	MTCNC
	4	05
	5	02
	6	YES
	7	ON
	8	11
	9	MTCX
	10	YES
2	1	11
	2	Drill left
	3	MTVNC
	4	01
	5	02
	6	YES
	7	ON
	8	15
	9	MTCX
	10	YES
3	1	12
	2	Drill right
	3	MTVNC
	4	02
	5	03
	6	NO
	7	OFF
	8	NO
	9	MTCX
	10	NO

Note: If there is an entry [DeviceOrder] in the "IND_DEV.INI" file or in the "FAR_DEV.INI" file, then these entries (lines) are output in the order in which they are listed there. If no entry [DeviceOrder] is given, then the devices are outputted according to the order of the sections in the file.

Example FCP1 Case B

Read the network configuration of all devices defined in the "IND_DEV.INI" file. (Case B)

Assumption:
The following device types have been defined but there is no network configuration data:

- Device address 05: MTC200-P-G2
- Device address 01: MTVNC

Note: No configuration data is available or the local PC is not active in the network or the PC has been disabled in the network configuration data (see the explanation for Case B).

FI command		**XX_BR_FCP1**
Line	**Column**	**Answer**
1	1	05
	2	Drill left
	3	MTC200-P-G2
	4	05
	5	XX
	6	YES
	7	ON
	8	NO
	9	MTCX
	10	YES
2	1	01
	2	Drill left
	3	MTVNC
	4	01
	5	XX
	6	YES
	7	ON
	8	NO
	9	MTCX
	10	YES

Note: If there is an entry [DeviceOrder] in the "IND_DEV.INI" file or in the "FAR_DEV.INI" file, then these entries (lines) are output in the order in which they are listed there. If no entry [DeviceOrder] is given, then the devices are outputted according to the order of the sections in the file.

3.19 Far Device Configuration Parameter: FDC

MPCX Device Group

Designation **FDC** Far **D**evice **C**onfiguration

Explanation FI command "FDC" returns the general data of the PC network. Differentiation is made between two cases (A and B):

- PC is in the PC network and
- PC is stand-alone.

Case A
PC is in the PC Network

The FI command returns the general data of the PC network. Furthermore, additional data such as the hostname and IP address of the PC is also outputted.

Case B
PC stand-alone

The data of the local PC is outputted if one or more of the following points apply:

- There is no network configuration data on the PC (see "FAR_DEV.INI" file).
- The PC has been disabled in the network configuration data or
- The "PC Network Active" option is not switched on in the system configurator.

FI command **XX_BR_FDC1** **(Single Read)**

Response Structure The following table shows the general structure of the response to the FI command "FDC1".

Line 1	Column 1	...	Column 4
Line 2	Column 1		
Line 3	Column 1		
Line 4	Column 1		
Line 5	Column 1		
Line 6	Column 1	...	Column 4

Value Range/Meaning of Columns

Line 1:	
1 = PC network exists?	[YES, NO]
2 = Name of the PC network	[max. 28 ASCII characters]
3 = Max. number of PCs	(Integer)
4 = Max. number of devices	(Integer)
Line 2:	
1 = PC No.	[00...99, XX]
Line 3:	
1 = Hostname/ Ethernet hostname possibly expanded by name of domain	(string)
Line 4:	
1 = Computer name/ NETBIOS name of computer	(string)
Line 5:	
1 = IP address of network card 1	(string)
...	...

4 = IP address of network card 4 (string)

Line 6:

1 = Master PC? [YES = PC is Master PC (Head PC), NO]

Example FDC1 Case A

Read the general data of the PC network.

Assumption:
A PC with two network cards has been defined:

- IP address of the 1st network card: 172.16.0.1
- IP address of the 2nd network card: 172.16.1.1

FI command		**XX_BR_FDC1**
Line	**Column**	**Answer**
1	1	YES
	2	Operation10
	3	20
	4	64
2	1	XX
3	1	Machine1.Cell1
4	1	MACHINE1
5	1	172.16.0.1
	2	172.16.1.1
6	1	YES

Example FDC1 Case B

Read the general data of the PC network.

Assumption:
No PC is active or defined within the network.

FI command		**XX_BR_FDC1**
Line	**Column**	**Answer**
1	1	NO
	2	--
	3	1
	4	16
2	1	XX
3	1	Machine1.Cell1
4	1	MACHINE1
5	1	172.16.0.1
6	1	--

3.20 Further Info Text: FIT

MPCX Device Group

Designation **FIT** Further **I**nfo **T**ext

Explanation Returns the additional text of an FI error code or a NACK error number.

FI command Read additional (further) text.

XX_BR_FIT1_(1)_(2) **(Single Read)**

XX_BC_FIT1_(1)_(2) **(Cyclic Read)**

(1) = Error class [1 = NACK error number,
2 = FI error code]

(2) = Error number [LONG]

Response Structure One line with one column is outputted for the additional text.

Line	Column

Meaning of the Column Additional text [language-dependent]

Example FIT Read the additional general error text with the number 26.

FI command		**XX_BR_FIT1_1_26**
Line	**Column**	**Answer**
1	1	Check mathematical expression. Correct NC program and re-transmit

Note: The general error result line contains an "X" in Column 5 when there is an additional text, otherwise simply "--". You can obtain the additional error text by calling up the "XX_BR_FIT1" command with the 1st and 2nd partial result.

3.21 Far PC Configuration Parameters: FPC

MPCX Device Group

Designation **FPC** **F**ar **PC** **C**onfiguration Parameter

Explanation The FI-Command "FPC" outputs the list of PCs that are defined in the network. Differentiation is made between two cases (A and B):

- PC is in the PC network and
- PC Stand-Alone.

Case A
PC is in the PC Network The list of PCs defined in the network configuration files on the PC (see "FAR_DEV.INI" file) is outputted.

Fall B
PC Stand-Alone The data of the local PC is outputted if one or more of the following points apply:

- There is no network configuration data on the PC (see "FAR_DEV.INI" file).
- The PC has been disabled in the network configuration data or
- The "PC Network Active" option is not switched on in the system configurator.

FI command **XX_BR_FPC1** **(Single Read)**

Response Structure

The following table shows the general structure of the response to the FI command "FPC1". The number of lines depends on the actual configuration. Result when network configuration data is available:

Line 1...n:	Column 1	...	Column 7

Value Range/Meaning of the Columns

1 = PC No.	[00...99, XX]
2 = Port	[IP address, host name]
3 = Name of PC	[max. 28 ASCII characters]
4 = Local device	[YES = PC is the local PC, NO = PC is a remote PC]
5 = Device status	[OFF = PC is disabled, ON = PC is enabled] corresponds to the "Disable" entry of section "PC<pcnr>"
6 = Master?	[YES = PC is Master PC (Head PC), NO] corresponds to the "Master PC" entry of section "PC<pcnr>"
7 = Online?	[YES, NO, --]

Explanation of Column 7 Online?

This column indicates whether there is currently a connection to the PC via which the device can be addressed. Differentiation is made between 3 possible cases:

- YES = The network connection to the PC is active
- NO = The network connection is down (interrupted).
- -- = The network connection has not yet been completely checked.

Note: YES is always output for B.

Example FPC1 Case A

Read the list of PCs that are defined in the function interface.

Assumption:
Two PCs are defined:

- PC1 with the IP address: 192.4.4.91
- PC2 with the name: st100103

FI command		**XX_BR_FPC1**
Line	**Column**	**Answer**
1	1	10
	2	192.4.4.91
	3	Drill station 02
	4	YES
	5	OFF
	6	NO
	7	YES
2	1	20
	2	st100103
	3	Drill station 03
	4	NO
	5	ON

	6	YES
	7	NO

Note: If there is an entry [DeviceOrder] in the "IND_DEV.INI" file or in the "FAR_DEV.INI" file, then these entries (lines) are output in the order in which they are listed there. If no entry [DeviceOrder] is given, then the devices are outputted according to the order of the sections in the file.

Example FPC1 Case B

Read the list of PCs that are defined in the function interface.
Assumption:
No PCs are defined:

FI command		**XX_BR_FPC1**
Line	**Column**	**Answer**
1	1	XX
	2	--
	3	--
	4	YES
	5	ON
	6	--
	7	YES

3.22 Writing/Reading the General FI Data Buffer: GDB

MPCX Device Group

Designation **GDB** **G**lobal **D**ata **B**uffer

Explanation

Writes/reads data for the general FI data buffer. A maximum of 100 byte can be transported in this FI data buffer.

Note: As much information as wished (max. 100 byte) can be exchanged between WIN32 applications by using the general FI data buffer. Data is identified by means of the relevant buffer ID.

Note!
The buffer IDs 686 to 695 are available for external applications!

FI command

Write data into an FI data buffer.

XX_BW_GDB1_(1) **(Single Write)**

(1) = Buffer ID [686-695]

Value to be written
Data to be transported (max. 100 byte)

Response Structure

(P_ACK) is returned following successful transmission.

Line 1	**Column 1**

Value Range/Meaning of Columns

1 = Successfully completed (P_ACK)

Example GDB1 Binary data (max. 100 byte) to be transferred to the general FI data buffer as a write value are written with the Buffer ID 686.

FI command		**XX_BW_GDB1_686**
Line	**Column**	**Answer**
1	1	(P_ACK)

FI command Read data from an FI data buffer.

XX_BR_GDB1_(1) **(Single Write)**

(1) = Buffer ID [686-695]

Response Structure The contents of the addressed FI data buffer are returned following successful transmission.

Line 1	**Column 1**	**Column 2**

Value Range/Meaning of Columns

1 =	Number of data	[BYTES]
2 =	Contents of the addressed FI data buffer	[Data of the addressed FI data buffer (max. 100 byte]

Example GDB1 Read the general FI data buffer using the buffer ID 686.

FI command		**XX_BR_GDB1_686**
Line	**Column**	**Answer**
1	1	15
	2	[123456789012345] [Data of the addressed FI data buffer (max. 100 byte]

3.23 Initialization of a Communication Address: ICA

MPCX Device Group

Designation **ICA** **I**nitialization **C**ommunication **A**ddress

Explanation By means of this command, a defined communication address of type V24 (that has been created by the system configurator – CommAddr entry in the configuration file IND_DEV.INI) is initialized with new parameters.

FI command **XX_BW_ICA1_(1)_(2)** **(Single Write)**

(1) = selected communication address of type V24 (CommAddr entry in the configuration file IND_DEV.INI)

(2) = initialization string according to specification (CommAddr entry in the configuration file IND_DEV.INI)

Response Structure The response to the "ICA1" FI command consists of one line with one column.

Line 1	**Column 1**

Value Range/Meaning of Columns

1 =	Status message (P_ACK)	(P_ACK)

Example ICA1 The defined communication address 1 (type: V25) is initialized with the following parameters:

COM-PORT:	1
BAUD RATE:	38400

PARITY:	NONE
MODE:	RS232
PC-COUNTER:	TCON

FI command		**XX_BW_ICA1_1_V24,COM1,38400,NONE,RS232,TCON**
Line	**Column**	**Answer**
1	1	(P_ACK)

By means of this command, a defined communication address of the TCP type (that has been created by the system configurator – CommAddr entry in the configuration file IND_DEV.INI) is initialized with new parameters.

FI command **XX_BW_ICA2_(1)_(2)** **(Single Write)**

(1) = selected communication address of the TCP type (CommAddr entry in the configuration file IND_DEV.INI)

(2) = initialization string according to specification (CommAddr entry in the configuration file IND_DEV.INI)

Response Structure The response to the "ICA2" FI command consists of one line with one column.

Line 1	**Column 1**

Value Range/Meaning of Columns

1 =	Status message (P_ACK)	(P_ACK)

Example ICA2 The defined communication address 1 (type: TCP) is re-initialized.

FI command		**XX_BW_ICA2_1_TCP,10.104.81.227,5002,TCON**
Line	**Column**	**Answer**
1	1	(P_ACK)

3.24 Information Regarding Function Interface Jobs: IFJ

MPCX Device Group

Designation **IFJ** **I**nformation about **F**unction-Interface **J**obs

Explanation Status information regarding active FI jobs can be read out. This status prompt allows, for instance, the basis for implementing a progress report (in the form of a display) during NC download as this can be run in the background for some time depending on the size of the NC program.

FI command Return status information on all active FI jobs.

XX_BR_IFJ1 **(Single Read)**

Response Structure The following table shows the general structure of the response to the FI command "IFJ1". The answer consists of a maximum of n=19 lines (n=19 maximum number of FI jobs), each with 16 columns.

Line 1...n:	**Column 1**	**...**	**Column 16**

Value Range/Meaning of Columns

1 =	Job ID	[01...20]
2 =	FI command	[string, in accordance to chapter 6.1 "Elements of the FI Command"]

3 = Job classification [1 = NC download,
2 = compile NC program package]
3 = Parameter down-/upload,
4 = Firmware down-/upload,
5 = Delete controller memory (CPU, PLC)
6 = Machine data down-/upload,
7 = Message texts (small control panels) down-/upload,
8 = PLC program download,
9 = Drive parameter down-/upload]

4 = Job status [RUN, READY, ERROR]

5 = Number of error lines in the error info buffer

6 = Max. processing time [ms] until TIMEOUT

7 = Start time of the job [hh:mm:ss:ms]

8 = Processing time up to now in ms

9 = Function interface connection (login) name of the application

10 = Progress type [1 = details of progress in %,
2 = details of absolute progress]

11 = Details of progress as percentage value [Value, --], depends on Column 10 "Progress type"

12 = Information on absolute progress [Value, --], depends on Column 10 "Progress type"

13 = Absolute end value [Value, --], depends on Column 10 "Progress type"

14 = Progress info buffer; contains display information, e.g., NC program line currently being transmitted.

15 = FI Job Error Code (see chapter entitled "Error Codes")

16 = Error info buffer

17 = FI error class [0=No error
1=NACK message of the controller,
2=FI error code,
3=General error text,
4=NT error texts,
5=SERCOS error texts,
6=Visual Motion error texts,
7=General SIS error texts]

Note: The results of the columns depend on the FI job that has been started.

Example IFJ1

Read the status information for all active FI jobs.

Assumption:

- The job with ID 01 has been started by the "NCA1" FI command and has been successfully completed with a READY message.

FI command		**XX_BR_IFJ1**
Line	**Column**	**Answer**
1	1	01
	2	02_BR_NCA1_"D:\Download.ini" /3
	3	2
	4	READY
	5	0

	6	600000
	7	16:15:22:123
	8	120
	9	VBDemo
	10	1
	11	100 %
	12	--
	13	--
	14	--
	15	0
	16	--
	17	0

FI command Return information regarding the selected and active FI job.

XX_BR_IFJ2_(1) **(Single Read)**

(1) = Job ID [01...20]

Note: Information regarding the structure of the response is available in the FI command "XX_BR_IFJ1" described above.

3.25 Reading of the FI Command Stack Administration: IFS

MPCX Device Group

Designation **IFS** **IF** Command **S**tack Info

Explanation By means of this command, the current allocation status of the FI command stack management can be read.

FI command **XX_BR_IFS1** **(Single Write)**

Response Structure The response to the "IFS1"I command consists of n lines, each with 4 columns.

Line 1...n	Column 1	...	Column 4

Value Range/Meaning of Columns

1 =	IF command stack index	[1..40]
2 =	IF command request string	[max. 500 ASCII characters]
3 =	Task name (corresponds to LOGINIF name of the application issued by the IF command	[max. 20 ASCII characters]
4 =	Access counter reading	[LONG value]

Example IFS1

Reads the current data of the IF command stack management, with 3 FI request strings being currently in the management.

FI command		**XX_BR_IFS1**
Line	**Column**	**Answer**
1	1	1
	2	00_BR_ASM2
	3	VBDEMO.EXE
	4	5467
2	1	2
	2	00_BR_AMM2_0
	3	IND400T.EXE
	4	456234
3	1	3
	2	02_BR_ASM2
	3	VBDEMO.EXE
	4	534892

3.26 Reading and Writing of PC Date and PC Time: LDT

MPCX Device Group

Designation

LDT PC **L**ocal **D**ate **T**ime

FI command

With this command, PC date and time are read. At the same time, the local date time information of the PC is supplied.

XX_BR_LDT1 **(Single Read)**

Response Structure

The following table shows the general structure of the response to the FI command "LDT1". A line of 1 column is output.

Line 1	**Column 1**	**...**	**Column 3**

Value Range/Meaning of Columns

1 = Information on date	[Day.month.year]
2 = Information on time	[hour:minute:second]
3 = LONG value	[internal LONG coding]

Example LDT11

Read the current date and time of the PC.

FI command		**XX_BR_LDT1**
Line	**Column**	**Answer**
1	1	25.04.2002
	2	07:26:06
	3	7192241

Explanation

This FI command sets PC date and time.

FI command

XX_BW_LDT1 **(Single Write)**

Value to be written

Date and time information to be written	[day.month.year hour:minute:second]

Note: The value to be written is passed to the "acValue" parameter as an ASCII value in the "DataTransfer" routine.

Response Structure

The following table shows the general structure of the response to the FI command "BW_LDT1". A line of 1 column is output.

Line 1	Column 1

Value Range/Meaning of Columns

1 = Status message (P_ACK)

Example LDT1

Set the PC clock to the data: 25.04.2002 07:31:33.

Write value: 25.04.2002 07:31:33

FI command		Value to be written: 25.04.2002 07:31:33 XX_BW_LDT1
Line	**Column**	**Answer**
1	1	(P_ACK)

3.27 Activated Language of the Bosch Rexroth GUI: LNG

MPCX Device Group

Designation

LNG Activated **L**a**NG**uage

Explanation

The country code of the activated language for the Bosch Rexroth GUI is output.

FI command

XX_BR_LNG **(Single Read)**

Response Structure

The response to the FI command "LNG" consists of one line with one column for the country code of the activated language.

Value Range of the Column

1 = Country code of the activated language [2 ASCII characters]

Country code	Language
EN	English (US/GB)
DE	German
ES	Spanish
IT	Italian
FR	French
HU	Hungarian
PT	Portuguese
SE	Swedish
CS	Czech

Example LNG

Read the country code of the activated language in the Bosch Rexroth GUI.

FI command		XX_BR_LNG
Line	**Column**	**Answer**
1	1	SE

Explanation

This command is used to define the language settings of the selected device.

FI command **XX_BW_LNG_(1)** **(Single Read)**

(1) = Language to be set =[see the Windows language extension, e.g. DE, EN, …]

Response Structure The following table shows the general structure of the response to the FI command "BW_LNG". A line of 1 column is output.

Line 1	Column 1

Value Range/Meaning of Columns 1 = Status message (P_ACK) (P_ACK)

Example LNG Set the language setting of the device 0 to English.

FI command		XX_BW_LNG_EN
Line	**Column**	**Answer**
1	1	(P_ACK)

3.28 Read System Messages: MSG

MPCX Device Group

Designation **MSG** **M**e**S**sa**G**e

Explanation Reading of system messages

FI command Message

XX_CC_MSG_(1) **(Cyclic Read)**

(1) = SYS-Message number

Note: Exists only as a cyclic command

Response Structure The response of the FI command 'MSG' consists of the system message data.

Example MSG 00_CC_MSG_64 (64 = MSG_SYSERRGEN)

FI command		00_CC_MSG_64/3
Line	**Column**	**Answer**
1	1	00

Limitation The following system messages:

SYS Message	SYS Message number
MSG_PCLUPDBEG	52
MSG_PARUPDBEG	24
MSG_FWAUPDBEG	82

These commands cannot be used with the following programs:

- Bosch Rexroth OPC server
- Bosch Rexroth DDE server

3.29 NT Shutdown Functions: NST

MPCX Device Group

Designation **NST** **N**T-**S**hu**T**-Down

Explanation This allows the NT operating system to be shut down.

FI command Triggers NT-Shut-Down (**WITHOUT** shutdown boxes).

XX_BW_NST1 **(Single Write)**

Response Structure As this concerns a command there is no response data.

Example NST1 Triggers NT-Shut-Down (**WITHOUT** shutdown boxes).

FI command	**XX_BW_NST1**

FI command Trigger NT-Shut-Down (**WITH** shutdown boxes).

XX_BW_NST2_(1)	**(Single Write)**
(1) = Time in seconds	The shutdown box appears for the input time

Response Structure As this concerns a command there is no response data.

Example NST2 NT-Shut-Down **WITH** shutdown boxes; the shutdown boxes appear for 30 seconds.

FI command	**XX_BW_NST2_30**

3.30 Formatting a Parameter Download Data File: PAF

MPCX Device Group

Designation **PAF** **PA**rameter **F**ile Converted

FI command By means of this FI command, an existing parameter download file can be re-formatted in such a way that the key names correspond to the parameter numbers. The structure of the download file corresponds to that of a Windows Ini file. Bosch Rexroth's own description in the document V20_Param_08_Definitions_Parameter_Download_01.doc is recommended for a more detailed account of the structure of the parameter download file.

XX_BW_PAF1_(1)_(2)	**(Single Write)**
(1) = Complete parameter download file name **unsorted**	[input file] must be available
(2) = Complete parameter download file name **sorted**	[output file] is generated.

Note: A valid device address can also be passed as a write value. If no write value is passed, this command requires the MTCNC parameter download format.

Response Structure The following table shows the general structure of the response to the FI command "PAF1". The response consists of a line with one column.

Line 1	Column 1

Value Range/Meaning of Columns

1 =	Status message (P_ACK)	(P_ACK)

Example PAF1

Generate on the basis of the unsorted parameter download file C:\TEMP\PARDATA.DAT the sorted parameter download file C:\TEMP\PARDATA.SOR.

FI command		**XX_BW_PAF1_"C:\TEMP\PARDATA.DAT"_"C:\TEMP\PARDATA.SOR"**
Line	**Column**	**Answer**
1	1	(P_ACK)

3.31 Generating Physical Directory Names: PHD

MPCX Device Group

Designation **PHD** **PH**ysical **D**irectory

Explanation

Generates physical directory names according to the BDI data written.

Note: This is based on BDI philosophy.

FI command

Generate physical directory names.

XX_BR_PHD1_(1)_(2)_(3)_(4)_(5)_(6) **(Single Write)**

(1) = Project ID	[-1= PROJECT_NEUTRAL -2= PROJECT_DEFAULT]
(2) = Section ID	[0= SECT_NEUTRAL 1= SECT_BIN 2= SECT_BASIC_DATA 3=SECT_OEM_DATA 4=SECT_CUSTOM_DATA 5=SECT_PROG_DATA]
(3) = Device address	[-1= DEVADDR_NEUTRAL otherwise the required device address]
(4) = Process ID	[-1= PROCESS_NEUTRAL otherwise the required process number]
(5) =Data type ID	[possible write values see BDI documentation (BDI_DEFINITIONS.H)]
(6) = Language ID	[possible write values see BDI documentation (WINNT.H)]

Response Structure

The following table shows the general structure of the response to the FI command "PHD1".

Line 1	**Column 1**

Value Range/Meaning of Columns

1 =	Physical directory name	[complete physical directory name in accordance with the BDI data written]

Example PHD1

Requesting the physical directory name for:
PROJECT_NEUTRAL
SECT_BIN
DEVADDR_NEUTRAL
PROCESS_NEUTRAL
DATATYPE_NEUTRAL
LANG_NEUTRAL

FI command		**XX_BR_PHD1_-1_0_-1_-1_0_0**
Line	**Column**	**Answer**
1	1	D:\Program Files\Indramat\Mtgui\Bin

3.32 Writing and Reading of a PC Port Address (Byte Access): POB

MPCX Device Group

Designation **POB** **PO**rt **B**yte Access

FI command This command is used for writing a PC port address (byte access).

XX_BW_POB1_(1)_(2)	**(Single Read)**
(1) = requested PC port address	Declaration format: 0x port address
(2) = PC port value to be written	Declaration format: 0x port value

Response Structure The following table shows the general structure of the response to the FI command "POB1". A line of 1 column is output.

Line 1	Column 1

Value Range/Meaning of Columns

1 = Status message	(P_ACK)

Example POW1 Write the value 0x0000 into the PC port address 0x31C.

FI command		**XX_BW_POW1_0x31C_0x0000**
Line	**Column**	**Answer**
1	1	(P_ACK)

FI command This command is used for reading a PC port address (byte access).

XX_BR_POB1_(1)	**(Single Read)**
(1) = requested PC port address	Declaration format: 0x port address

Response Structure The following table shows the general structure of the response to the FI command "POB1". A line of 1 column is output.

Line 1	Column 1

Value Range/Meaning of Columns 1= PC port value read

Example POB1 Read the PC port address 0x31C.

FI command		**00_BR_POB1_0x31C**
Line	**Column**	**Answer**
1	1	0x00

3.33 Writing and Reading of a PC Port Address (Word Access): POW

MPCX Device Group

Designation **POW** **PO**rt **W**ord Access

FI command This command is used for writing a PC port address (word access).

XX_BW_POW1_(1)_(2)	**(Single Read)**
(1) = requested PC port address	Declaration format: 0x port address
(2) = PC port value to be written	Declaration format: 0x port value

Response Structure The following table shows the general structure of the response to the FI command "POW1". A line of 1 column is output.

Line 1	Column 1

Value Range/Meaning of Columns 1 = Status message (P_ACK) (P_ACK)

Example POW1 Write the value 0x0000 into the PC port address 0x31C.

FI command		**XX_BW_POW1_0x31C_0x0000**
Line	**Column**	**Answer**
1	1	(P_ACK)

FI command This command is used for reading a PC port address (word access).

XX_BR_POW1_(1)	**(Single Read)**
(1) = requested PC port address	Declaration format: 0x port address

Response Structure The following table shows the general structure of the response to the FI command "POW1". A line of 1 column is output.

Line 1	Column 1

Value Range/Meaning of Columns 1= PC port value read

Example POW1 Read the PC port address 0x31C.

FI command		**XX_BW_POW1_0x31C**
Line	**Column**	**Answer**
1	1	0x0000

3.34 Ready Message for a WIN32 Application: RPR

MPCX Device Group R

Designation **RPR** **R**eady **PR**ocess

Explanation Using this FI command, WIN32 applications logged in the FI can inform the initiating program that they are ready for operation.

Note!
The process for the WIN32 application was generated by means of the FI command: "XX_BW_CPR1_(1)_(2)_(3)".

FI command

Inform the initiating program that the WIN32 program invoked is ready for operation.

XX_BW_RPR1 **(Single Write)**

Response Structure

As this concerns a command there is no response data.

3.35 Shutdown Manager: SDM

MPCX device group

Designation

SDM **S**hut**D**own **M**anager

Explanation

This command is used to read the configuration data for the Shutdown manager. It is defined in IND_DEV.INI under the [KillManager] entry.

XX_BR_SDM1 **(Single Read)**

Response Structure

The following table shows the general structure of the response to the FI command "BR_SDM1". The number of lines depends on the current configuration data, each line consisting of one column.

Line 1...n:	Column 1

Value Range/Meaning of the Columns

1 = Designation of the configuration date and its value	[ASCII characters]

Brief description for the individual configuration files:

KillMessageBox:

YES = A dialog box is to appear when the Shutdown manager is called up

NO = NO dialog box is to appear when the Shutdown manager is called up

KillMessageBoxTime:

Indicates in seconds for which time the dialog box is to appear.

KillShutDown:

YES = The operating system is to be shut down

NO = The operating system is NOT to be shut down

KillReboot:

YES = The computer is to be rebooted automatically

NO = The computer is NOT rebooted

KillRemotePc:

YES = The computers in the network are to be shut down

NO = The computers in the network are NOT shut down

KillAutoStart:

YES = The Shutdown manager is to be activated time-controlled

NO = NO time-control function of the Shutdown manager

KillTimePeriod:

Indicates the time period in hours. After expiry of the given number of hours since the start of the FI the Shutdown manager will be started automatically.

KillTimeAbsolute:

Indicates the time at which the Shutdown manager is started automatically.

Format: hour:minute:second

KillCloseTaskNameX: (with X = run-time parameter of 1..20)

Indicates the task name (= name of the external application at the FI) which is passed a "Closing request message" on execution of the Shutdown manager. This provides for correct closing of the respective external application.

If the external application is NOT logged in at the FI, the window message WM_CLOSE is sent. If the external application fails to close after the preset time, it will be "killed".

KillCloseTaskTimeX: (with X = run-time parameter of 1..20)

Indicates the time in ms allowed for correct closing of the external application. After expiry of this time period the external application will be "killed".

KillCloseTaskMtguiX: (with X = run-time parameter of 1..20)

YES = The external application is to close when the MTGUI is exited

NO = The external application is NOT closed when the MTGUI is exited

KillAutoStartDelayTime:

Indicates the waiting period in ms before the requested applications are restarted (after a shutdown process). This waiting period is necessary to ensure that the exited applications have been removed from the user memory before they are restarted.

KillAutoStartApplicationX: (with X = run-time parameter of 1..20)

Indicates the complete call-up path of the application to be restarted.

Note: This call-up path should always be given in inverted commas.

Example SDM1

Read the current configuration data of the Shutdown manager.

FI command		**XX_BR_SDM1**
Line	**Column**	**Answer**
1	1	KillMessageBox=YES
2	1	KillMessageBoxTime=30 sec
3	1	KillShutdown=YES
4	1	KillReboot=YES
5	1	KillRemotePc=NO
6	1	KillAutoStart=YES
7	1	KillTimePeriod=10 h
8	1	KillTimeAbsolute=--
9	1	KillCloseTaskName1=VBDEMO.EXE
10	1	KillCloseTaskTime1=8000 ms
11	1	KillCloseTaskMtgui1=NO
12	1	KillCloseTaskName2=PRISMA
13	1	KillCloseTaskTime2=10000 ms
14	1	KillCloseTaskMtgui2=YES

15	1	KillAutoStartDelayTime=8000 ms
16	1	"D:\Program Files\Indramat\Mtgui\bin\comdesk.exe"
17	1	"D:\Program Files\Indramat\Mtgui\bin\PRISMA.EXE"
18	1	"D:\Program Files\Indramat\Mtgui\bin\VBDEMO.EXE"

Explanation Thus command starts the Shutdown manager on **all** computers in the network.

FI command **BW_SDM1** **(Single Write)**

Response Structure The response to the "BW_SDM1" FI command consists of n lines with three columns. For each computer in the network, 1 line is returned.

Line 1...n	Column 1	Column 2	Column 3

Value Range/Meaning of Columns

1 =	Address of the computer in the network	[00.0.15,XX]
2 =	Name of the computer	
3 =	Station name (see FAR_DEV.INI	[max. 28 ASCII characters]

Example SDM1 The Shutdown manager is to be started on all computers of the network (here, it is assumed that there are 3 computers). The answer consists of 3 lines – 1 line for each computer.

FI command		**XX_BW_SDM1**
Line	**Column**	**Answer**
1	1	01
	2	BTV20-R1
	3	Station_1_L
2	1	02
	2	BTV20_R2
	3	Station_1_R
3	1	03
	2	BV20-R3
	3	Loading station

Explanation This command is used to start the Shutdown manager on the local computer.

FI command **BW_SDM2** **(Single Write)**

Response Structure The response to the "BW_SDM2" FI command consists of one line with three columns.

Line 1	Column 1	Column 2	Column 3

Value Range/Meaning of Columns

1 =	Address of the computer	[00.0.15,XX]
2 =	Name of the computer	
3 =	Station name (see FAR_DEV.INI	[max. 28 ASCII characters]

Example SDM2 The Shutdown manager is to be started on the local computer. Assumption: NO active network!

FI command		**XX_BW_SDM2**
Line	**Column**	**Answer**
1	1	XX
	2	BTV20-MTC200-P
	3	--

3.36 Triggering an FI Device Polling Cycle: SDP

MPCX Device Group

Designation **SDP** **S**tart **D**evice **P**olling

Explanation This FI command triggers an FI device polling cycle.

FI command **XX_BW_SDP1** **(Single Write)**

Response Structure The response to the "SDP1" FI command consists of one line with one column.

Line 1	**Column 1**

Value Range/Meaning of Columns

1 = Status report (P_ACK)

Example SDP1 Triggering the FI device polling cycle.

FI command		**XX_BW_SDP1**
Line	**Column**	**Answer**
1	1	(P_ACK)

3.37 Focusing Commands: SFW

MPCX Device Group

Designation **SFW** **S**et **F**ocus to **W**indow

Explanation The screen can be focused with these FI commands.

FI command Focus the screen on the DOS-BOF user interface.

XX_BW_SFW1 **(Single Write)**

Response Structure As this concerns a command there is no response data.

Example SFW1 Focus on the DOS-BOF user interface screen.

Assumption:
This FI command is set from a WIN32 application and used to focus the display window of the DOS-BOF user interface currently running.

FI command	**XX_BW_SFW1**

FI command Focus the screen (target window) of a WIN32 application connected to the FI via LogInIf().

XX_BW_SFW2_(1)_(2)_(3) **(Single Write)**

(1) = LogInIf Login names	This refers to the login name entered at the FI (LogInIf()) during the login procedure.

(2) = Min Info	Control information as to whether or not the current screen window (output window) is to be minimized. The following applies: 0 = do not minimize 1 = minimize
(3) = Wait Info	Control information as to how the output window is focused; here, the following applies: 0 = Re-focusing with the SFW2 command 1 = Automatic re-focusing on termination of the WIN32 application

Response Structure As this concerns a command there is no response data.

Example SFW2 Focus on the WIN32 application that has logged on in the FI with the login name "VBDEMO.EXE". The current screen window (output window) is minimized and refocusing takes place automatically at the end of the WIN32 application (VBDEMO.EXE).

FI command	**XX_BW_SFW2_VBDEMO.EXE_1_1**

3.38 Software Installation Data: SID

MPCX device group

Designation **SID** **S**oftware **I**nstallation **D**ata

Explanation Information is returned regarding installation. This information includes installation paths, the software versions used, DLL mode, context information, plus service pack and release information.

FI command The installation data and/or software version data is read in.

XX_BR_SID1 **(Single Read)**

Response Structure One line with 16 columns is output for the returned values.

Line 1	**Column 1**	**...**	**Column 16**

Meaning of the Columns

1 = Basic directory	[EXE files of the BOF]
2 = FI installation directory	[FI directory]
3 = Data directory	[in accordance with BOF]
4 = GBO version	[from INDRAMAT.ini]
5 = IF-DLL mode	[from INDRAMAT.ini]
6 = IF version	[from INDRAMAT.ini from DLL mode 400]
7 = Service pack info	[from INDRAMAT.ini from DLL mode 420]
8 = Release info	[from INDRAMAT.ini from DLL mode 420]
9 = IF-Build-Info	[in accordance with Build process]
10 = Current context name	[in accordance with the installation]
11 = Physical installation path	[in accordance with the installation]
12 = Complete IF version indication string	
13 = WinPCL build number	[in accordance with WinPCL]
14 = Version number of the	[in accordance with WinPCL]

PLC compiler

15 = Version number of the PLC linker	[in accordance with WinPCL]
16 = Version number of the PLC data basis	[in accordance with WinPCL]

17 = Platform version

Example SID1 Return information on the current installation.

FI command		00_BR_SID1
Line	**Column**	**Answer**
1	1	--
	2	D:\Program Files\Indramat\MTGUI\Bin
	3	--
	4	005-22Vxx
	5	07.20
	6	07V00
	7	--
	8	--
	9	Build 3124 Mar 6 2003 08:53:55
	10	MTGUI_0-23T01 B3327
	11	D:\Program Files\Indramat\MTGUI\
	12	FI: 07V00 DLL-Mode: 07.20 Build 3124 Mar 6 2003 08:53:55
	13	347.15.4.11
	14	771
	15	515
	16	78
	17	Platform: 02V01 Build: 3214

Note: Refer to FI command "PHD" for working with absolute paths.

3.39 Issuing a SYS Message: SSM

MPCX Device Group

Designation **SSM** **S**et **S**ys **M**essage

Explanation This allows SYS messages to be issued.

Note: The SYS message handling of the FI **MUST** BE known!

FI command This allows SYS messages to be issued. Acknowledgement must first be received from the WIN32 applications that want to receive the issued SYS message.

Note!
Any additional information with a maximum length of 200 characters can be transmitted simultaneously as a write value.

XX_BW_SSM1_(1)_(2) (Single Write)

(1) = SYS message number (ALWAYS an even number)	Value range: 2..4000 Note: The SYS message number is ALWAYS even, while the acknowledgement number associated with it is always an odd number!
(2) = input acknowledgement time in msec	Input acknowledgement time – the WIN32 applications that want to receive the SYS message MUST acknowledge the fact within this time.

Response Structure

The following table shows the general structure of the response to the FI command "SSM1".

Line 1	Column 1	...	Column 8

Value Range/Meaning of Columns

1 =	Status report	[READY=SYS message has been correctly acknowledged by the WIN32 applications] [ERROR=SYS message has NOT been acknowledged by a WIN32 application within the pre-set time]
2 =	Task name (LogInIf name)	[Task name that has triggered the SYS message]
3 =	SYS message number	[contains the issued SYS message number]
4 =	Acknowledgement time	[contains the pre-set acknowledgement time]
5 =	Reference information	[contains, where applicable, the additional information transferred as a write value]
6 =	Length of additional information	[0 where NO additional information has been transferred]
7 =	Where applicable, LOG channel of the FI that has NOT acknowledged	[-- = acknowledgements have been completed in time or the LOG channel number of the WIN32 application that has NOT acknowledged in time]
8 =	Where applicable, task name that has NOT acknowledged in time.	[-- = acknowledgements have been completed in time or the task name that has NOT acknowledged in time]

Example SSM1

Issues SYS message 3302 with a pre-set acknowledgement time of 20000 msec. The additional information, device address 00, is also transferred as a write value.

FI command		Value to be written: 00 XX_BW_SSM1_3302_20000
Line	**Column**	**Answer**
1	1	READY
	2	VBDEMO.EXE
	3	3302
	4	20000
	5	00
	6	2
	7	--
	8	--

FI command

This allows SYS messages to be issued **WITHOUT** the necessary acknowledgements.

Note!
Any reference information with a maximum length of 200 characters can be transmitted simultaneously as a write value.

XX_BW_SSM2_(1)	**(Single Write)**
(1) = SYS message number (ALWAYS an even number)	Value range: 2..4000 Note: The SYS message number is ALWAYS even, while the acknowledgement number associated with it is always an odd number!

Value to be written
Reference information

Response Structure

The following table shows the general structure of the response to the FI command "SSM2".

Line 1	Column 1	...	Column 8

Value Range/Meaning of Columns

1 =	Status report	[READY=SYS message has been issued correctly]
2 =	Task name (LogInIf name)	[Task name that has triggered the SYS message]
3 =	SYS message number	[contains the issued SYS message number]
4 =	Acknowledgement time	[0]
5 =	Reference information	[contains, where applicable, the additional information transferred as a write value]
6 =	Length of additional information	[0 where NO additional information has been transferred]
7 =	Where applicable, LOG channel of the FI that has NOT acknowledged	[--]
8 =	Where applicable, task name that has NOT acknowledged in time.	[--]

Example SSM2

Issues SYS message 3302 **WITHOUT** acknowledgement. The additional information, device address 00, is also transferred as a write value.

FI command		**Value to write: 00** **XX_BW_SSM2_3302**
Line	**Column**	**Answer**
1	1	READY
	2	VBDEMO.EXE
	3	3302
	4	0
	5	00
	6	2
	7	--
	8	--

4 FI Commands - MTCX Device Group (MTVNC)

The FI Commands described in this chapter are valid for the MTCX device group. In this device group, the following types are listed as well as possible device addresses:

Group	Accompanying Types	Address
MTCX	MTVNC	[00...63]

Note: Please note that the device address must be set before the respective FI command, e.g. 00_BR_ASM1 (refer also here to Chapter 2.1 "Elements of the FI Command").

With a few exceptions, commands for the MWCX device group are also valid for the MTCX device group. The chapter 2.3 "Overview of FI Commands", "Overview of the MTCX Device Group" contains a summary of the possible FI commands for the MTCX device group.

4.1 Static/Dynamic Device Information: DIF

MTCX Device Group

Designation **DIF** **D**evice **I**n**F**ormation

Explanation Static device information and network information is read according to the "IND_DEV.INI" and "FAR_DEV.INI" files.

FI command Reading of the static device information and network information of a selected device.

BR_DIF1 **(Single Read)**

BC_DIF1 **(Cyclic Read)**

BB_DIF1 **(Break Cyclic Read)**

Response Structure The following table shows the general structure of the response to the "DIF1" FI command. The response consists of one line with 24 columns.

Line 1	Column 1	...	Column 24

Value Range/Meaning of Columns

1 =	Local/far device address	[00..63]
2 =	Device name	IND_DEV.INI entry: DeviceName=
3 =	Device type	IND_DEV.INI entry: DeviceType=
4 =	PLC support	IND_DEV.INI entry: PLC=
5 =	Device status	IND_DEV.INI entry: DeviceStatus=
6 =	Assignment of a simulation pair	IND_DEV.INI entry: DeviceAssign=
7 =	Device mode	IND_DEV.INI entry: MtvncMode=
8 =	Communication channel	IND_DEV.INI entry: [CommAddrX]
9 =	Description of the communication channel	IND_DEV.INI entry: CommStr=
10 =	Timeout value	IND_DEV.INI entry: Timeout=
11 =	Device group	(see Chapter 6.1 "Identifier")
12 =	PLC component type	IND_DEV.INI entry: Component type1=

13 =	CNC component type	IND_DEV.INI entry: Component type2=
14 =	Device log	IND_DEV.INI entry: DeviceProtocol=
15 =	Device simulation	IND_DEV.INI entry: DeviceSimulation=
16 =	Not yet assigned	[--]
17 =	Not yet assigned	[--]
18 =	Not yet assigned	[--]
19 =	Not yet assigned	[--]
20 =	Network ON/OFF	[ON = Network active OFF = No network active]
21 =	Network name	Max. 28 ASCII characters
22 =	PC number	[00..99,XX]
23 =	PC name	Max. 255 ASCII characters
24 =	Local device address	[00..63]

Example DIF1 Read the static device information and network information of device 1 while the network is active.

FI command		01_BR_DIF1
Line	**Column**	**Answer**
1	1	01
	2	Simulation system 01
	3	MTVNC
	4	YES
	5	ON
	6	NO
	7	OFF
	8	4
	9	SHM,1,TCON
	10	3500
	11	MWCX
	12	MTS-P
	13	MTC-P
	14	CNC
	15	OFF
	16	--
	17	--
	18	--
	19	--
	20	ON
	21	PC network 1
	22	29
	23	BTV20-STATION-LINKS
	24	01

Explanation The dynamic device information and network information is read. The current data identifications are made available from the selected controller.

FI command

Reading of the dynamic device information and network information of a selected device.

BR_DIF2	**(Single Read)**
BC_DIF2	**(Cyclic Read)**
BB_DIF2	**(Break Cyclic Read)**

Response Structure

The answer consists of 23 lines, each line having a specific meaning.

Line 1	Static device information
Line 2	Firmware information
Line 3	Current parameter set
Line 4	Current PLC program
Line 5	Current machine data set
Line 6	Current NC package for memory A
Line 7	Current NC package for memory B
Line 8	Current cycle package
Line 9	Current NC program name for process 0
Line 10	Current NC program name for process 1
Line 11	Current NC program name for process 2
Line 12	Current NC program name for process 3
Line 13	Current NC program name for process 4
Line 14	Current NC program name for process 5
Line 15	Current NC program name for process 6
Line 16	Current tool list for process 0
Line 17	Current tool list for process 1
Line 18	Current tool list for process 2
Line 19	Current tool list for process 3
Line 20	Current tool list for process 4
Line 21	Current tool list for process 5
Line 22	Current tool list for process 6
Line 23	Current I/O configuration table

Meaning of line 1

Line 1 returns the most significant static device information and network information and consists of 18 columns.

Line 1	Column 1	...	Column 18

Value Range/Meaning of Columns

1 =	Line number	[1]
2 =	Status information	Contains the information whether or not the subsequent data in this line is valid; the following applies: [0 = Data is invalid - further column results [--] 1 = Data is valid]
3 =	Local/far device address	[00..63]
4 =	Device name	According to device configuration
5 =	Device type	According to device configuration
6 =	PLC Components	According to device configuration
7 =	CNC components	According to device configuration
8 =	Device group	(see Chapter 2.1 "Elements of the FI Command", "Identifier")

9 =	Device status	According to device configuration ON = DeviceStatus ON OFF = DeviceStatus OFF
10 =	Current device status	ON = Device ONLINE OFF = Device OFFLINE
11 =	Not yet assigned	[--]
12 =	Not yet assigned	[--]
13 =	Not yet assigned	[--]
14 =	Network ON/OFF	[ON = Network active OFF = No network active]
15 =	Network name	Max. 28 ASCII characters
16 =	PC number	[00..99,XX]
17 =	PC name	Max. 255 ASCII characters
18 =	Local device address	[00..63]

Meaning of line 2

Returns the firmware versions of the existing controller components. Each line consists of 8 columns.

Line 2	Column 1	...	Column 8

Value Range/Meaning of Columns

1 =	Line number	[2]
2 =	Status information	Contains the information whether or not the subsequent data in this line is valid; the following applies: [0 = Data is invalid – further column results [--] 1 = Data is valid]
3 =	Firmware version of the CNC component	Designation according to convention
4 =	Firmware version of the PLC component	Designation according to convention
5 =	Firmware version of the 1.APR component	Designation according to convention
6 =	Firmware version of the 2.APR component	Designation according to convention
7 =	Firmware version of the 3.APR component	Designation according to convention
8 =	Firmware version of the 4.APR component	Designation according to convention

Meaning of line 3

Returns the identification of the current parameter set and consists of 6 columns.

Line 3	Column 1	...	Column 6

Value Range/Meaning of Columns

1 =	Line number	[3]
2 =	Status information	Contains the information whether or not the subsequent data in this line is valid; the following applies: [0 = Data is invalid – further column results [--] 1 = Data is valid]
3 =	Index of the parameter set	[01..99]
4 =	Designation of the parameter set	Max. 32 ASCII characters
5 =	Date string	Date of generation/modification

6 =	Time string	Time of generation/modification

Meaning of line 4

Returns the identification of the current PLC program and consists of 6 columns.

Line 4	Column 1	...	Column 6

Value Range/Meaning of Columns

1 =	Line number	[4]
2 =	Status information	Contains the information whether or not the subsequent data in this line is valid; the following applies: [0 = Data is invalid – further column results [--] 1 = Data is valid]
3 =	Index of the PLC program	Always [00]
4 =	PLC resource name = PLC program name	Max. 32 ASCII characters
5 =	Date string	Date of generation/modification
6 =	Time string	Time of generation/modification

Meaning of line 5

Returns the identification of the current machine data set and consists of 6 columns.

Line 5	Column 1	...	Column 6

Value Range/Meaning of Columns

1 =	Line number	[5]
2 =	Status information	Contains the information whether or not the subsequent data in this line is valid; the following applies: [0 = Data is invalid – further column results [--] 1 = Data is valid]
3 =	Index of the machine data set	[01..99]
4 =	Designation of the machine data set	Max. 32 ASCII characters
5 =	Date string	Date of generation/modification
6 =	Time string	Time of generation/modification

Meaning of line 6

Returns the identification of the current NC package in memory A and consists of 6 columns.

Line 6	Column 1	...	Column 6

Value Range/Meaning of Columns

1 =	Line number	[6]
2 =	Status information	Contains the information whether or not the subsequent data in this line is valid; the following applies: [0 = Data is invalid – further column results [--] 1 = Data is valid]
3 =	Index of the NC package in memory A	[01..99]
4 =	Designation of the NC package in memory A	Max. 32 ASCII characters
5 =	Date string	Date of generation/modification
6 =	Time string	Time of generation/modification

Meaning of line 7

Returns the identification of the current NC package in memory B and consists of 6 columns.

Line 7	Column 1	...	Column 6

Value Range/Meaning of Columns

1 =	Line number	[7]
2 =	Status information	Contains the information whether or not the subsequent data in this line is valid; the following applies: [0 = Data is invalid – further column results [--] 1 = Data is valid]
3 =	Index of the NC package in memory B	[01..99]
4 =	Designation of the NC package in memory B	Max. 32 ASCII characters
5 =	Date string	Date of generation/modification
6 =	Time string	Time of generation/modification

Meaning of line 8

Returns the identification of the current cycle package and consists of 6 columns.

Line 8	Column 1	...	Column 6

Value Range/Meaning of Columns

1 =	Line number	[8]
2 =	Status information	Contains the information whether or not the subsequent data in this line is valid; the following applies: [0 = Data is invalid – further column results [--] 1 = Data is valid]
3 =	Index of the cycle package	[01..99]
4 =	Designation of the cycle package	Max. 32 ASCII characters
5 =	Date string	Date of generation/modification
6 =	Time string	Time of generation/modification

Meaning of the lines 9 - 15

These lines return information on the current NC program for the processes 0..6 and consist of 8 columns each.

Line 9..15	Column 1	...	Column 8

Value Range/Meaning of Columns

1 =	Line number	[9..15]
2 =	Status information	Contains the information whether or not the subsequent data in this line is valid; the following applies: [0 = Data is invalid – further column results [--] 1 = Data is valid]
3 =	Process number	[00..06]
4 =	Process name	Max. 40 ASCII characters
5 =	Current NC memory	[A,B]
6 =	Current NC program number	[01..99]
7 =	Current NC program name	Max. 32 ASCII characters
8 =	Current NC block	

Meaning of the lines 16 - 22

These lines return information on the current tool lists for the processes 0..6 and consist of 12 columns each.

Line 16.0.22	Column 1	...	Column 12

Value Range/Meaning of Columns

1 =	Line number	[16..22]
2 =	Status information	Contains the information whether or not the subsequent data in this line is valid; the following applies: [0 = Data is invalid – further column results [--] 1 = Data is valid]
3 =	Process number	[00..06]
4 =	Process name	Max. 40 ASCII characters
5 =	Tool list index	Always [00]
6 =	Name of the tool list	Max. 32 ASCII characters
7 =	Date string	Date of generation/modification
8 =	Time string	Time of generation/modification
9 =	Tool magazine type	[MAGAZINE] [TURRET]
10 =	Number of spindles	[0..4]
11 =	Number of grippers	[0..4]
12 =	Number of magazine locations	[0..999]

Meaning of line 23

Returns the identification of the current I/O configuration list and consists of 6 columns.

Line 23	Column 1	...	Column 6

Value Range/Meaning of Columns

1 =	Line number	[23]
2 =	Status information	Contains the information whether or not the subsequent data in this line is valid; the following applies: [0 = Data is invalid – further column results [--] 1 = Data is valid]
3 =	Index of the I/O configuration list	[01..99]
4 =	Designation of the I/O configuration list	Max. 32 ASCII characters
5 =	Date string	Date of generation/modification
6 =	Time string	Time of generation/modification

Example DIF2

Read the dynamic device information of device 1. This is an MTVNC, and the network is active.

FI command		**01_BR_DIF2**
Line	**Column**	**Answer**
1	1	1
	2	1
	3	01
	4	Simulation system
	5	MTVNC
	6	NONE

	7	NONE
	8	MTCX
	9	ON
	10	ON
	11	--
	12	--
	13	--
	14	ON
	15	PC network 1
	16	29
	17	BTV20-RIGHT
	18	01
2	1	2
	2	1
	3	VNC01/0004-23T09
	4	--
	5	--
	6	--
	7	--
	8	--
3	1	3
	2	1
	3	01
	4	Config_3_Process
	5	07.04.03
	6	09:45:42
4	1	4
	2	1
	3	00
	4	Prg_3_Process
	5	30.04.03
	6	09:03:45
5	1	5
	2	1
	3	01
	4	Machine_Data_3_Process
	5	07.04.03
	6	09:48:34
6	1	6
	2	1
	3	30
	4	Gearbox 2

	5	15.10.02
	6	13:47:34
7	1	7
	2	0
	3	--
	4	--
	5	--
	6	--
8	1	8
	2	1
	3	03
	4	Cycle package_3_Process
	5	15.10.02
	6	14:48:56
9	1	9
	2	1
	3	00
	4	Master process 0
	5	A-
	6	01
	7	NcPrg_01
	8	N0000 G00 X10.0
10	1	10
	2	1
	3	01
	4	Process 1
	5	A-
	6	--
	7	--
	8	--
11	1	11
	2	1
	3	02
	4	Process 2
	5	A-
	6	--
	7	--
	8	--
12	1	12
	2	0
	3	--
	4	--

	5	--
	6	--
	7	--
	8	--
13	1	13
	2	0
	3	--
	4	--
	5	--
	6	--
	7	--
	8	--
14	1	14
	2	0
	3	--
	4	--
	5	--
	6	--
	7	--
	8	--
15	1	15
	2	0
	3	--
	4	--
	5	--
	6	--
	7	--
	8	--
16	1	16
	2	1
	3	00
	4	Master process 0
	5	00
	6	Tool list gearbox 2
	7	14.10.02
	8	15:34:23
	9	[MAGAZINE]
	10	1
	11	1
	12	200
17	1	17
	2	1

	3	01
	4	Process 1
	5	--
	6	--
	7	--
	8	--
	9	[MAGAZINE]
	10	1
	11	2
	12	100
18	1	18
	2	1
	3	02
	4	Process 2
	5	--
	6	--
	7	--
	8	--
	9	[MAGAZINE]
	10	1
	11	1
	12	50
19	1	19
	2	0
	3	--
	4	--
	5	--
	6	--
	7	--
	8	--
	9	--
	10	--
	11	--
	12	--
20	1	20
	2	0
	3	--
	4	--
	5	--
	6	--
	7	--
	8	--

	9	--
	10	--
	11	--
	12	--
21	1	21
	2	0
	3	--
	4	--
	5	--
	6	--
	7	--
	8	--
	9	--
	10	--
	11	--
	12	--
22	1	22
	2	0
	3	--
	4	--
	5	--
	6	--
	7	--
	8	--
	9	--
	10	--
	11	--
	12	--
23	1	23
	2	0
	3	--
	4	--
	5	--
	6	--

FI command Returns information on which controller data is currently available in the selected device.

BR_DIF3	**(Single Read)**
BC_DIF3	**(Cyclic Read)**

Response Structure

The answer consists of 1 line with 14 columns, each column having a specific meaning.

1 = Active parameter set available Yes/No
2 = NC package memory A available Yes/No
3 = NC package memory B available Yes/No
4 = NC zero points memory A available Yes/No
5 = NC zero points memory B available Yes/No
6 = NC events available Yes/No
7 = NC variables available Yes/No
8 = NC D-corrections available Yes/No
9 = NC cycles available Yes/No
10 = Active machine data record available Yes/No
11 = PLC retain variables available Yes/No
12 = Tool lists available Yes/No
13 = Drive parameters available Yes/No
14 = I/O configuration list available Yes/No

Line 1	**Column 1**	**...**	**Column 14**

Value Range/Meaning of Columns

1 = Controller data available Yes/NO [YES,NO]

Return information on which controller data is currently available in the selected device.

FI command		**01_BR_DIF3**
Line	**Column**	**Answer**
1	1	YES
	2	YES
	3	NO
	4	YES
	5	YES
	6	YES
	7	YES
	8	YES
	9	YES
	10	YES
	11	YES
	12	NO
	13	YES
	14	NO

4.2 Reading the Firmware/Monitor Identification: MTC

MTCX Device Group

Designation **MTC** **MT-C**NC Slot Software Version

FI command This command is used to read the firmware identification from the various control components (slot numbers).

Note: For the time this FI command is executed, the internal FI communication interlocks (fast timeout monitoring, offline operation, etc.) are switched off.

FI command

BR_MTC_(1)	**(Single Read)**
BC_MTC_(1)	**(Cyclic Read)**
(1) = Slot number	[1=CNC, 2=SIO, 3=PLC, 4=APR1 5=APR2, 6=APR3, 7=APR4]

The controller sends a line with 1 column for the firmware version of the requested slot. If a slot is not equipped, the controller returns the answer N_ACK 224 (time monitoring with internal communication)

Response Structure The following table shows the general structure of the response to the FI command "MTC". A line of 1 column is output.

Line 1	Column 1

Value Range/Meaning of Columns

1 = Firmware identification string [max. 16 ASCII characters]

Example 1 MTC Read the firmware identification of slot number 1 (CPU) of device 00.

FI command		**00_BR_MTC_1**
Line	**Column**	**Answer**
1	1	VNC01/0004-23T03

Example 2 MTC Reading of the firmware of an empty slot:

FI command		**00_BR_MTC_3**
Line	**Column**	**Answer**
1	1	1
	2	224
	3	0x00000000
	4	Time error with internal communication
	5	X

Explanation With the FI command "BR_MTC1", the monitor versions of the various components (CNC, PLC, APR) can be read out.

FI command **BR_MTC1** **(Single Read)**

Response Structure The response to the "BR_MTC1" FI command consists of six lines with four columns. One line is returned for each potential component (CNC, PLC, APR1-4).

Line 1.0.6	Column 1	Column 2	Column 3	Column 4

Value Range/Meaning of Columns

1 =	Line number	[1..6]
2 =	Component information	[CNC=NC component

SPS=PLC component
APR=APR component

3 = Monitor version – "old" format

4 = Monitor version – "new" format

Example MTC1 Read the monitor versions for device 0.

FI command		**00_BR_MTC1**
Line	**Column**	**Answer**
1	1	1
	2	CNC
	3	--
	4	--
2	1	2
	2	PLC
	3	--
	4	--
3	1	3
	2	APR
	3	--
	4	--
4	1	4
	2	APR
	3	--
	4	--
5	1	5
	2	APR
	3	--
	4	--
6	1	6
	2	APR
	3	--
	4	--

5 FI Commands - MWCX Device Group (MTC 200)

The FI Commands described in this chapter are valid for the MWCX device group. In this device group, the following types are listed as well as possible device addresses:

Group	Accompanying Types	Address
MWCX	MTC200-P-G2, MTC200-R-G2	[00...63]

Note: Please note that the device address must be set before the respective FI command, e.g. 00_CR_AAC_0 (refer also here to the chapter 2.1 "Elements of the FI Command").

5.1 Active Acceleration Value: AAC

MWCX device group

Designation **AAC** Active **AC**celeration

Explanation The current acceleration value of an NC process is read out. Within an NC program, an acceleration limit can be programmed by means of the "programmable acceleration ACC" function. This is the case when, for instance, the axes of the workpiece carrier is to be moved depending on the weight of the workpiece.

FI Command Output the active acceleration value of an CNC process of the selected device from the MWCX device group.

CR_AAC1_(1)	**(Single Read)**
CC_AAC1_(1)	**(Cyclic Read)**
CB_AAC1_(1)	**(Break Cyclic Read)**
(1) = NC process number	[0...6]

Response Structure The following table shows the general structure of the response to the FI command "AAC". One line with three columns is output for the NC command, the acceleration value and the unit.

Line 1	Column 1	....	Column 3

Value Range of the Columns

1 = NC command	[ACC]
2 = Acceleration value	[0...100]
3 = Unit	[%]

Example AAC1 Reads the active acceleration value in NC process 0 of device address 00.

FI command	00_CR_AAC1_0		
Answer			
Line	**Column 1**	**Column 2**	**Column 3**
1	ACC	50	[%]

Reference to Literature See chapter entitled "Literature" [1].

5.2 Active Angle Dimension (RAD/DEG): AAD

MWCX device group

Designation **AAD** Active **A**ngle **D**imension

Explanation The active angle dimension of an NC process is read out. The arguments of the angle functions SIN, COS, TAN and the results of the inverse functions of the angle functions ASIN, ACOS, ATAN can be specified or calculated both in "radiants" (RAD) as well as in "degrees" (DEG).

FI command Output the active angle dimension of an NC process of the selected device from the MWCX device group.

CR_AAD_(1)	**(Single Read)**
CC_AAD_(1)	**(Cyclic Read)**
CB_AAD_(1)	**(Break Cyclic Read)**
(1) = NC process number	[0...6]

Response Structure The response to the FI command "AAD" consists of one line with one column for the unit [RAD/DEG].

Line 1	Column 1

Example AAD Reads the active angle dimension in NC process 0 of device address 00.

FI command		**00_CR_AAD_0**
Line	**Column**	**Answer**
1	1	RAD

Reference to Literature See chapter entitled "Literature" [2].

5.3 Actual (Current) Axis Speed (Spindle Speed): AAS

MWCX device group

Designation **AAS** **A**ctual **A**xis **S**peed

Explanation The current axis speed and spindle speed of an NC process for the selected device are read out. The FI command "AAS1" refers to the NC process number and to the code of the axis meaning, whereas the FI command "AAS2" allows the current speed to be queried in relation to the physical axis number.

FI command Output the current axis speed related to the NC process number and to the code of the meaning of the axis.

Using the optional third parameter it is possible to pre-select conversion of the result into mm/min or inch/min. If, however, a spindle is selected as an axis, indicating a measurement system serves no purpose.

CR_AAS1_(1)_(2){_(3)}	**(Single Read)**
CC_AAS1_(1)_(2){_(3)}	**(Cyclic Read)**
CB_AAS1_(1)_(2){_(3)}	**(Break Cyclic Read)**
(1) = NC process number	[0...6]
(2) = Axis meaning	[0...11] (see Chapter "Data Tables")
(3) = Required measurement system (opt.)	[mm, inch]

FI command

Output the current axis speed of the selected device related to the physical axis number.

Using the optional second parameter it is possible to pre-select conversion of the result into mm/min or inch/min. If, however, a spindle is selected as an axis, indicating a measurement system serves no purpose.

CR_AAS2_(1){_(2)}	**(Single Read)**
CC_AAS2_(1){_(2)}	**(Cyclic Read)**
CB_AAS2_(1){_(2)}	**(Break Cyclic Read)**
(1) = Physical axis number	[1...32, according to settings of the system parameters]
(2) = Required measurement system (opt.)	[mm, inch]

Response Structure

The following table shows the general structure of the response to the FI command "AASx" . One line is output with 4 columns for the axis designation, axis speed, unit and the axis speed limited to "indicated decimal places".

Line 1	Column 1	Column 2	Column 3	Column 4

Meaning of the Columns

1 = Axis name	[according to settings of axis parameters]
2 = Speed	[according to settings of axis parameters]
3 = Unit	[according to settings of process parameters and required measurement system]
4 = Speed	[as Column 2, but rounded up or down according to the parameter "indicated decimal places"]

Note: If the selected axis is not defined then the response in all columns is [--].

Example AAS1

Reads the current axis speed of the Z axis in NC process of device address 00.

FI command	**00_CR_AAS1_0_2**			
Answer				
Line	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	Z1	-1.2345	[mm/min]	-1.235

Example AAS1

Reads the current axis speed of the Z axis in NC process of device address 00. Output of values in inch/min.

FI command	**00_CR_AAS1_0_2_inch**			
Answer				
Line	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	Z1	-0.0486	[inch/min]	-0.049

Example AAS2

Reads the current speed of spindle S (e.g., physical axis number 4) of device address 00.

FI command	**00_CR_AAS2_4**			
Answer				
Line	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	S1	4000.0	1/min	4000.0

Reference to Literature

See chapter entitled "Literature" [3].

5.4 Active NC Block: ABI

MWCX device group

Designation **ABI** **A**ctive NC-**B**lock **I**nformation

Explanation The active NC record or a user-defined NC block is read out. This allows an NC record display to be constructed with an active NC record as well as the number of the previous and following NC records.

FI command Output the active NC block as well as the previous and following NC blocks of an NC process for the selected device from the MWCX device group.

BR_ABI_(1){_(2)_(3)}	**(Single Read)**
BC_ABI_(1){_(2)_(3)}	**(Cyclic Read)**
BB_ABI_(1){_(2)_(3)}	**(Break Cyclic Read)**
(1) = NC process number	[0...6]
(2) = Number of previous NC blocks	[1..4] ! Optional !
(3) = Number of following NC blocks	[1..4] ! Optional !

Note: If the optional parameters are not specified then only the current NC record is output.

Response Structure The number of lines (1...n = 9) in the response depends on the number of NC records requested. Each line consists of a column containing the respective NC record.

Note: If there is no valid NC program in the device then the value of all columns is [--] .

Example ABI Reads the active NC record and the two previous and two following NC records of NC process 0 of device address 00.

FI command		**00_BR_ABI_0_2_2**
Line	**Column**	**Answer**
1	1	N0000 .START
2	1	N0001 T13 BSR .M6
3	1	N0002 G90 G41 G54 G17 F2000.0 S3200.00 M003
4	1	N0003 G00 X 60.0000 Y -30.0000
5	1	N0004 Z -6.0000

Reference to Literature See chapter entitled "Literature" [4].

5.5 Active Cutting Speed of the Reference Spindle: ACS

MWCX device group

Designation **ACS** Active **C**utting **S**peed

Explanation Output of the active cutting speed of the reference spindle of an NC process for the selected device from the MWCX device group.

FI command

CR_ACS_(1)	**(Single Read)**
CC_ACS_(1)	**(Cyclic Read)**
CB_ACS_(1)	**(Break Cyclic Read)**
(1) = NC process number	[0...6]

Response Structure The following table shows the general structure of the response to the FI command "ACS". One line with three columns is output for the S number of the reference spindle, the cutting speed and the unit according to the settings of the system parameters.

Line 1	Column 1	...	Column 3

Value Range/Meaning of Columns

1 = S number of reference spindle	S1, S2, S3
2 = Cutting speed	[format according to settings of the parameters]
3 = Unit	[according to settings of the system parameters]

Note: If no reference spindle is defined in the selected NC process then the value of Column 1 is [*S]; Columns 2 and 3 are given the value [--].

Example ACS Reads the active cutting speed in NC process 0 of device address 00.

FI command	**00_CR_ACS_0**		
Answer			
Line	**Column 1**	**Column 2**	**Column 3**
1	S1	200	[m/min]

Reference to Literature See chapter entitled "Literature" [5].

5.6 Active D-Correction Number: ADN

MWCX device group

Designation **ADN** Active **D**-Correction **N**umber

Explanation The active D-correction number of an NC process of the MWCX device group is output. The D-corrections are cumulative to the tool-geometry data of the register effecting the tool management.

FI command Output the active D-correction numbers of an NC process of the selected device from the MWCX device group.

FI command

CR_ADN1_(1)	**(Single Read)**
CC_ADN1_(1)	**(Cyclic Read)**
CB_ADN1_(1)	**(Break Cyclic Read)**
(1) = NC process number	[0...6]

Response Structure One line with two columns is output for the active D-correction number of the indicated NC process. The meaning of the elements is as follows:

1 = Identifier	[D]
2 = D-correction number	[0] =De-selection of D-correction [1..99] = Selection of D-correction

Example ADN Read the active D-correction number of NC process 0 of device address 00.

FI command		**00_CR_ADN1_0**
Line	**Column**	**Answer**
1	1	D
	2	0

Reference to Literature See chapter entitled "Literature" [6].

5.7 Active Diagnosis Window ADW

MWCX device group

Designation **ADW** Active **D**iagnosis **W**indow

Explanation Indicates the window types for which data is required.

For improved performance and with some diagnosis windows, not all the data is called up each time a diagnosis is performed, but only when the data is actually required.

Through this FI command, the diagnosis server can be informed that the data of the respective window type is required.

This command has to be issued at least every 3 seconds while the data is required. If this command is not issued any more, calling-up of all data will stop.

FI command Indicates the window types for which data is required.

BW_ADW1_(1){_(2)}	**(Single Write)**
(1) = Type of diagnosis window	[1 = CNC error, 2 = sequence errors, 3 = general errors, 4 = messages, 10 = start requirements, 11 = warnings, 12 = setup diagnosis]
(2) = Module number	[1...99] ! only for window type 1 -4!

Example ADW1 Call up data for CNC error in controller 0 module 1.

FI command	**00_BW_ADW1_1_1**

5.8 Active Event Monitoring: AEM

MWCX device group

Designation **AEM** Active **E**vent **M**onitoring

Explanation The status of the event monitoring of the specified NC process of the MWCX device group is output. Events are binary variables that can be used by the NC program; these variables represent any status defined by the programmer just like flags in the PLC program. Waiting for a defined status of an event therefore allows the possibility of process synchronization.

FI command Output the status of the event monitoring of an NC process of the selected device from the MWCX device group.

CR_AEM_(1)	**(Single Read)**
CC_AEM_(1)	**(Cyclic Read)**
CB_AEM_(1)	**(Break Cyclic Read)**
(1) = NC process number	[0...6]

Response Structure One line and one column are output for the status of the event monitoring. The meaning of the elements is as follows:

EEV = Activation of event monitoring

DEV = Suppressing of event monitoring

Example AEM Read the status of the event monitoring of NC process 0 of device address 00.

FI command		**00_CR_AEM_0**
Line	**Column**	**Answer**
1	1	EEV

Reference to Literature See chapter entitled "Literature" [7].

5.9 Active Edge Number: AEN

MWCX device group

Designation **AEN** Active **E**dge **N**umber

Explanation The active edge number of an NC process is output. Changing the active cutter in the NC program results in the provision of the corresponding correction and tool life data which the tool management then accesses during subsequent processing.

FI command Output the active edge number of an NC process of the selected device from the MWCX device group.

CR_AEN_(1)	**(Single Read)**
CC_AEN_(1)	**(Cyclic Read)**
CB_AEN_(1)	**(Break Cyclic Read)**
(1) = NC process number	[0...6]

Response Structure One line with two columns is output for the identifier "E = Edge" and for the active edge number. The active cutter corresponds to the single-digit decimal number [1...9] that is assigned the address letter "E".

Example AEN Read the active edge number of NC process 0 of device address 00.

FI command		**00_CR_AEN_0**
Line	**Column**	**Answer**
1	1	E
	2	1

Reference to Literature See chapter entitled "Literature" [8].

5.10 Active Feedrate Override: AFO

MWCX device group

Designation **AFO** Active **F**eedrate **O**verride

Explanation The current value of the feedrate override of an NC process is outputted. Override is interpreted in the NC, irrespective of the mode; it has an effect on any axis movement (except on homing digital axes).

FI command Output the current value of the feedrate override of an NC process of the selected device from the MWCX device group.

CR_AFO1_(1)	**(Single Read)**
CC_AFO1_(1)	**(Cyclic Read)**
CB_AFO1_(1)	**(Break Cyclic Read)**
(1) = NC process number	[0...6]

Response Structure

The following table shows the general structure of the response to the FI command "AFO". One line with three columns is output for the identifier, the current value of the feedrate override and the unit [%].

Line 1	Column 1	....	Column 3

Value Range/Meaning of Columns

1 = Identifier	[OVR=Override]
2 = Current value of the feedrate override	[0...255]
3 = Unit	[%]

Note: The valid range of override weighting by the PLC program is between 0 and 255%. The NC limits the axis and/or processor speed to the maximum values set in the parameters if an override value is set that is too large.

Example AFO1

Reads the current value of the feedrate override in NC process 0 of device address 00.

FI command	**00_CR_AFO1_0**		
Answer			
Line	**Column 1**	**Column 2**	**Column 3**
1	OVR	60	[%]

Reference to Literature

See chapter entitled "Literature" [9].

5.11 Actual (Current) Feedrate: AFR

MWCX device group

Designation

AFR Actual **F**eed **R**ate

Explanation

The current value of the feedrate of an NC process is output. The details of the feedrate in an NC program are expressed by means of a feedrate value with the address letter "F" and a feedrate that is input directly as a constant or by means of an expression.

FI command

Output the current value of the feedrate of an NC process of the selected device from the MWCX device group.

Using the optional second parameter it is possible to pre-select conversion of the result into mm or inches.

CR_AFR_(1){_(2)}	**(Single Read)**
CC_AFR_(1){_(2)}	**(Cyclic Read)**
CB_AFR_(1){_(2)}	**(Break Cyclic Read)**
(1) = NC Process number	[0...6]
(2) = Required measurement system (opt.)	[mm, inch]

Response Structure

The following table shows the general structure of the response to the FI command "AFR". One line with three columns is output for the identifier, the current value of the feedrate and the unit.

Line 1	Column 1	...	Column 3

Value Range/Meaning of Columns

1 = Identifier	[F = feedrate]
2 = Value	[format according to settings of the parameters]
3 = Unit	[according to settings of the process parameters]

Example AFR

Reads the current feedrate in NC process 0 of device address 00.

FI command	**00_CR_AFR_0**		
Answer			
Line	**Column 1**	**Column 2**	**Column 3**
1	F	30000.0	[mm/min]

Example AFR

Reads the current feedrate in NC process 0 of device address 00. The displayed value is to be converted into inch/min:

FI command	**00_CR_AFR_0_inch**		
Answer			
Line	**Column 1**	**Column 2**	**Column 3**
1	F	1181.1	[inch/min]

Reference to Literature

see chapter entitled "Literature" [!)].

5.12 Active G Functions: AGF

MWCX device group

Designation

AGF Active **G**-**F**unction

Explanation

The active G functions of an NC process of the selected device from the MTCX device group are read out.

FI command

CR_AGF_(1){_(2)}	**(Single Read)**
CC_AGF_(1){_(2)}	**(Cyclic Read)**
CB_AGF_(1){_(2)}	**(Break Cyclic Read)**
(1) = NC process number	[0...6]
(2) = G code group	[1...21] ! Optional !

Note: If the optional parameter is not specified, then all active G codes are output for all G code groups.

Response Structure

One line is output, whereby the number of columns depends on the number of G code groups that are requested. If the optional parameter has not been specified, the response consists of one line with 21 columns. If the optional parameter has been specified then the response consists of one line with one column which contains the active G function of the selected G code group.

Note: In cases where no G function of the selected G code group is active, the response consists of the characters [--].

Example AGF

Reads the active G function of G code group 17 in the NC process 0 of device address 00.

FI command		**00_CR_AGF_0_17**
Line	**Column**	**Answer**
1	1	G30

Reference to Literature

See chapter entitled "Literature" [11].

5.13 Active M Functions: AMF

MWCX device group

Designation **AMF** Active **M**-**F**unction

Explanation The active M functions of an NC process of the selected device from the MWCX device group are read out.

FI command

CR_AMF_(1){_(2)}	**(Single Read)**
CC_AMF_(1){_(2)}	**(Cyclic Read)**
CB_AMF_(1){_(2)}	**(Break Cyclic Read)**
(1) = NC process number	[0...6]
(2) = M function Group	[1...16] ! Optional !

Note: If the optional parameter is not specified then all active M functions of all M function groups are output.

Response Structure One line is output, whereby the number of columns depends on the number of M function groups that are requested. When the optional parameter has not been specified, the response consists of one line with 16 columns. If the optional parameter has been specified then the response consists of one line with one column which contains the active M function of the selected M function group.

Note: In cases where no M function of the selected M function group is active, the answer consists of the characters [--].

Example AMF Read the active M function of M function group 2 in NC process 0 of device address 00.

FI command		**00_CR_AMF_0_2**
Line	**Column**	**Answer**
1	1	M005

Reference to Literature See chapter entitled "Literature" [12].

5.14 Active Mechanism Messages: AMM

MWCX device group

Designation **AMM** Active **M**echanism **M**essages

Explanation Messages regarding active mechanism errors and mechanism diagnostics are output. These messages are assigned to a particular mechanism or process. Depending on the FI command, the device address, device name, mechanism number, mechanism name, type of message, message source, type of message (2), message number, short text and reference text are all output.

FI command Output mechanism messages currently pending for all active devices.

BR_AMM1 **(Single Read)**

BC_AMM1 **(Cyclic Read)**

BB_AMM1 **(Break Cyclic Read)**

Note: The "AMM1" FI command refers to all devices within this device group. Therefore, any valid device address can be indicated in the command line (see Example AMM1).

Response Structure The following table shows the general structure of the response to the FI command "AMM1". The response consists of a maximum of n=512 lines (n=16 devices x 32 mechanisms = 512), each with 12 columns.

Line 1...n	Column 1	...	Column 12

Value Range/Meaning of Columns

1 =	Device address	[00...63]
2 =	Device name	[max. 32 ASCII characters]
3 =	Mechanism number	[0...31]
4 =	Mechanism name	[max. 28 ASCII characters]
5 =	Type of message	[F = fault/error, D = diagnosis]
6 =	Message source	[CNC, PLC]
7 =	Type of message (2)	[S = Status, O = Operator, E = External, I = Internal]
8 =	Message number	[0...600]
9 =	Message text	[max. 54 ASCII characters]
10 =	Reference text	[x= exists, -- = does not exist]
11 =	2 bytes of additional information for the message number	is required to resolve the information "@" (see AMM5)
12 =	Filename for additional information for message text	e.g. in HTML format

Example AMM1 Read the current mechanism messages of all active devices.
Assumption:
The following device addresses and mechanisms are defined:

- Device address 01 with 2 mechanisms 0 and 1, and
- Device address 03 with one mechanism 0.

FI command		**03_BR_AMM1**
Line	**Column**	**Answer**
1	1	01
	2	Drill center
	3	0
	4	Station 1
	5	D
	6	CNC
	7	S:
	8	79
	9	Station waiting until tool-change command has ended.
	10	x
	11	0
	12	
2	1	01
	2	Drill center
	3	1
	4	Station 2
	5	F
	6	CNC
	7	O
	8	1
	9	No external 24V supply.
	10	x
	11	0
	12	
3	1	03
	2	Milling center
	3	0
	4	Camshaft 30.40.25.0S
	5	D
	6	CNC
	7	S:
	8	71
	9	Circular interpolation
	10	x
	11	0
	12	

FI command

Output the currently pending mechanism messages of the selected device.

BR_AMM2	**(Single Read)**
BC_AMM2	**(Cyclic Read)**
BB_AMM2	**(Break Cyclic Read)**

Response Structure

The following table shows the general structure of the response to the FI command "AMM2". The response consists of up to a maximum of n=31 lines, each with 12 columns.

Line 1...n	Column 1	...	Column 12

Value Range/Meaning of Columns

1 =	Device address	[00...63]
2 =	Device name	[32 ASCII characters]
3 =	Mechanism number	[0...31]
4 =	Mechanism name	[max. 28 ASCII characters]
5 =	Type of message	[F = fault/error, D = diagnosis]
6 =	Message source	[CNC, PLC]
7 =	Type of message (2)	[S = Status, O = Operator, E = External, I = Internal]
8 =	Message number	[0...600]
9 =	Message text	[max. 54 ASCII characters]
10 =	Reference text	[x= exists, -- = does not exist]
11 =	2 byte additional information for the message number	is required to resolve the information "@" (see AMM5)
12 =	Filename for additional information for message text	e.g. in HTML format

Example AMM2

Reads the current mechanism messages of device address 01.
<u>Assumption:</u>
Device address 01 with 2 defined mechanisms 0 and 1.

FI command		**01_BR_AMM2**
Line	**Column**	**Answer**
1	1	01
	2	Drill center
	3	0
	4	Station 1
	5	D
	6	CNC
	7	S:
	8	79
	9	Station waiting until tool-change command has ended.
	10	x
	11	0
	12	

2	1	01
	2	Drill center
	3	1
	4	Station 2
	5	F
	6	CNC
	7	O
	8	1
	9	No external 24V supply.
	10	x
	11	0
	12	

Reference to Literature

See chapter entitled "Literature" [13].

FI command

Output the currently pending messages of the mechanisms listed for the selected device of the MWCX device group.

BR_AMM3_(1) **(Single Read)**

BC_AMM3_(1) **(Cyclic Read)**

BB_AMM3_(1) **(Break Cyclic Read)**

(1) = Selection list for a max. of 10 mechanisms [0_1_2_ ... _31]

Response Structure

The following table shows the general structure of the response to the FI command "AMM3". The number of lines (1 .. n=32) depends on the number of requested mechanism messages. Each line in turn consists of 12 columns.

Line 1...n	Column 1	...	Column 12

Value Range/Meaning of Columns

1 =	Device address	[00...63]
2 =	Device name	[max. 32 ASCII characters]
3 =	Mechanism number	[0...31]
4 =	Mechanism name	[max. 28 ASCII characters]
5 =	Type of message	[F = fault/error, D = diagnosis]
6 =	Message source	[NC, PLC]
7 =	Type of message (2)	[S = Status, O = Operator, E = External, I = Internal]
8 =	Message number	[0...600]
9 =	Message text	[max. 54 ASCII characters]
10 =	Reference text	[x= exists, -- = does not exist]
11 =	2 bytes of additional information for the message number	is required to resolve the information "@" (see AMM5)
12 =	Filename for additional information for message text	e.g. in HTML format

Reference to Literature

See chapter entitled "Literature" [13].

Example AMM3

Reads the current messages of mechanisms 0 and 1 of device address 01.

Assumption:
Device address 01 with 2 defined mechanisms 0 and 1.

FI command		**01_BR_AMM3_0_1**
Line	**Column**	**Answer**
1	1	01
	2	Drill center
	3	0
	4	Station 1
	5	D
	6	CNC
	7	S:
	8	79
	9	Station waits until tool-change command has ended.
	10	x
	11	0
	12	
2	1	01
	2	Drill center
	3	1
	4	Station 2
	5	F
	6	CNC
	7	O
	8	1
	9	No external 24V supply.
	10	x
	11	0
	12	

FI command

Output the currently pending messages of the mechanisms listed for the devices of the MWCX device group.

BR_AMM4_(1)	**(Single Read)**
BC_AMM4_(1)	**(Cyclic Read)**
BB_AMM4_(1)	**(Break Cyclic Read)**
(1) = Selection list for a max. of 10 mechanisms	[Format: x.y]

Format x.y	**Value Range**
X	Device address [00..0.63]
Y	Mechanism number [0...31]

Response Structure

The following table shows the general structure of the response to the FI command "AMM4". The number of lines (n=10 mechanisms, maximum) depends on the number of requested mechanism messages. Each line in turn consists of 12 columns.

Line 1...n	Column 1	...	Column 12

Value Range/Meaning of Columns

1 =	Device address	[00...63]
2 =	Device name	[max. 32 ASCII characters]
3 =	Mechanism number	[0...31]
4 =	Mechanism name	[max. 28 ASCII characters]
5 =	Type of message	[F = fault/error, D = diagnosis]
6 =	Message source	[CNC, PLC]
7 =	Type of message (2)	[S = Status, O = Operator, E = External, I = Internal]
8 =	Message number	[0...600]
9 =	Message text	[max. 54 ASCII characters]
10 =	Reference text	[x= exists, -- = does not exist]
11 =	2 byte additional information for the message number	is required to resolve the information "@" (see AMM5)
12 =	Filename for additional information for message text	e.g. in HTML format

Reference to Literature See chapter entitled "Literature" [13].

Example AMM4 Reads the current messages of mechanisms 0 and 1 of device address 01 as well as the messages of mechanism 0 of device address 03.

<u>Assumption:</u>
The following device addresses and mechanisms are defined:

- Device address 01 with 2 mechanisms 0 and 1, and
- Device address 03 with one mechanism 0.

FI command		**01_BR_AMM4_01.0_01.1_03.0**
Line	**Column**	**Answer**
1	1	01
	2	Drill center
	3	0
	4	Station 1
	5	D
	6	CNC
	7	S:
	8	79
	9	Station waits until tool-change command has ended.
	10	x
	11	0
	12	
2	1	01
	2	Drill center
	3	1
	4	Station 2
	5	F
	6	CNC
	7	O

	8	1
	9	No external 24V supply.
	10	x
	11	0
	12	
3	1	03
	2	Milling center
	3	0
	4	Camshaft 30.40.25.0S
	5	D
	6	CNC
	7	S:
	8	71
	9	Circular interpolation
	10	x
	11	0
	12	

FI command

Device and mechanism related output of the reference text for the selected message number for the devices of the MWCX device group.

BR_AMM5_(1)_(2)_(3) **(Single Read)**

1 = Mechanism number [0...31]

(2) = Message number [0...600]

(3) = 2 bytes of additional information for the message number

---

Note: The third parameter of AMM5 is given as the 11th partial result of commands AMM1 ... AMM4

Response Structure

The following table shows the general structure of the response to the FI command "AMM5". The number of lines n=512 lines (n=16 devices x 32 mechanisms = 512) depends on the number of requested mechanism messages. Each line in turn consists of 10 columns.

Line 1...n	Column 1	...	Column 10

Value Range/Meaning of Columns

1 =	device address	[00...63]
2 =	Device name	[max. 32 ASCII characters]
3 =	Mechanism number	[0...31]
4 =	Mechanism name	[max. 28 ASCII characters]
5 =	Type of message	[F = fault/error, D = diagnosis]
6 =	Message source	[CNC, PLC]
7 =	Type of message (2)	[S = Status, O = Operator, E = External, I = Internal]
8 =	Message number	[0...600]
9 =	Reference text	[max. [max. 14 lines with a max. 78 characters/line]
10 =	Filename for additional information for reference text	e.g. in HTML format

Example AMM5

Reads the reference text for the required message number 79 of selected mechanism 0 for selected device 01.

FI command		**01_BR_AMM5_0_79_0**
Line	**Column**	**Answer**
1	1	01
	2	Drill center
	3	0
	4	Station 1
	5	D
	6	CNC
	7	--
	8	79
	9	Station waits for completed execution of the active tool change command.
	10	

Reference to Literature

See chapter entitled "Literature" [13].

5.15 Active NC Memory Size: ANM

MWCX device group

Designation **ANM** Active **N**C **M**emory Size

Explanation The size of the active NC memory is read out.

FI command **CR_ANM** **(Single Read)**

Response Structure The following table shows the general structure of the response to the FI command "ANM". A line with 3 columns is output for identification, size of the total NC memory, and largest free block:

Line 1	Column 1	Column 2	Column 3

Value Range/Meaning of Columns

1 = NC Memory Size	[string]
2 = Total size of the NC memory	[long]
3 = Largest free block of the NC memory	[long]

Example: ANM Read the size of the active NC memory.

FI command	**00_CR_ANM**		
Answer			
Line	**Column 1**	**Column 2**	**Column 3**
1	NC Memory Size	654321	234567

5.16 Active Machine Parameter Index: API

MWCX device group

Designation **API** Active Machine-**P**arameter **I**ndex

Explanation Information regarding the active machine-parameter records of all defined devices of the MWCX device group are output. The following are output: the device addresses, index, GUI display, name, size, date and time of creation or of the last change and details of the defined processes of the active machine parameter record.

FI command

BR_API1	**(Single Read)**
BC_API1	**(Cyclic Read)**
BB_API1	**(Break Cyclic Read)**

Note: The "API1" FI command refers to all devices within this device group. Therefore, any valid device address can be indicated in the command line (see Example "API1").

Response Structure The following table shows the general structure of the response to the FI command "API1". The response consists of up to a maximum of n=16 lines, each with 8 columns.

Line 1...n:	Column 1	...	Column 8

Value Range/Meaning of Columns

1 =	device address	[00...63]
2 =	Index of active parameter record	[0]
3 =	Identification string of the parameter record	[max. 84 ASCII characters]
4 =	Name of parameter record	[max. 32 ASCII characters]
5 =	Size of parameter record [byte]	[max. 8 ASCII characters]
6 =	Date of creation or of the last change in the parameter record.	[8 ASCII characters in format: DD.MM.YY]
7 =	Time of creation or of the last change in the parameter record.	[8 ASCII characters in format: HH:MM:SS]
8 =	Additional information (e.g. details of defined processes).	[max. 24 ASCII characters]

Note: In cases where there is no active machine parameter record in the device or where the active machine parameter record has been changed, Column 1 is given the device address and Columns 2 to 8 the value [--].

Example API1 Reads the information on the active machine parameter records of all defined devices.
Assumption:
The following device addresses of the MWCX device group have been defined:

- Device address 00: MTC200-P,
- Device address 01: MTC200-P and
- Device address 02: MTVNC.

FI command		**01_BR_API1**
Line	**Column**	**Answer**
1	1	00
	2	0
	3	00MSD 0209-15 15625 28.01.9913:29:10M123456
	4	MSD 0209-15
	5	15625
	6	28.01.99
	7	13:29:10
	8	M123456
2	1	01
	2	0
	3	88PCI 12.45.12.34 10584 11.11.9811:11:11M12
	4	PCI 12.45.12.34
	5	10584
	6	11.11.98
	7	11:11:11
	8	M12

3	1	02
	2	0
	3	11Lab 5 DRV 24464 01.03.9914:25:10M13456
	4	Lab 5 DRV
	5	24464
	6	01.03.99
	7	14:25:10
	8	M13456

Reference to Literature

See chapter entitled "Literature" [14].

FI command

BR_API2 (Single Read)
BC_API2 (Cyclic Read)
BB_API2 (Break Cyclic Read)

Response Structure

The following table shows the general structure of the response to the FI command "API2". The response consists of a line with eight columns.

Line 1	Column 1	...	Column 8

Value Range/Meaning of Columns

1 =	Device address	[00...63]
2 =	Index of active parameter record	[0]
3 =	Identification string of the parameter record.	[max. 84 ASCII characters]
4 =	Name of parameter record	[max. 32 ASCII characters]
5 =	Size of parameter record [byte]	[max. 8 ASCII characters]
6 =	Date of creation or of the last change in the parameter record.	[8 ASCII characters in format: DD.MM.YY]
7 =	Time of creation or of the last change in the parameter record.	[8 ASCII characters in format: HH:MM:SS]
8 =	Additional information (e.g. details of defined processes).	[max. 24 ASCII characters]

Note: In cases where there is no active machine parameter record in the device or where the active machine parameter record has been changed, Column 1 is given the device address and Columns 2 to 8 the value [--].

Example API2

Reads the information on the active machine parameter record of device address 02.

Assumption:
The following device addresses of the MWCX device group have been defined:

- Device address 00: MTC200-P,
- Device address 01: MTC200-R and
- Device address 02: MTVNC.

FI command		**02_BR_API2**
Line	**Column**	**Answer**
1	1	02
	2	0
	3	00MSD 0209-15 15625 28.01.9913:29:10M123456
	4	MSD 0209-15
	5	15625
	6	28.01.99
	7	13:29:10
	8	M123456

Reference to Literature See chapter entitled "Literature" [14].

5.17 Active Note in NC Program (Note and NC Record Number): APM

MWCX device group

Designation **APM** Active **P**art-Program **M**essage

Explanation The active note of the NC record as well as the NC record number of an NC process of the MWCX device group is output. Every NC record can contain a note that is displayed in the diagnostics menu of the Bosch Rexroth GUI after the NC record has been processed. The note in the diagnostics line remains active until it is overwritten by a new note (also refer to "Active Note in NC Program (only NC Record Number): APN").

FI command

CR_APM_(1)	**(Single Read)**
CC_APM_(1)	**(Cyclic Read)**
CB_APM_(1)	**(Break Cyclic Read)**
(1) = NC process number	[0...6]

Response Structure The following table shows the general structure of the response to the FI command "APM". One line with two columns is output for the NC record number and the NC note is output.

Line 1	**Column 1**	**Column 2**

Value Range/Meaning of Columns

1 = NC record number of the note [0000...9999]

2 = Note [max. 48 ASCII characters]

Note: If the current NC program does not contain a note, then the result of Column 1 is [0000] and that of Column 2 is [--].

Example APM Read the active note in the NC process of device address 00.

FI command		**00_CR_APM_0**
Line	**Column**	**Answer**
1	1	0002-{}-
	2	Technological instructions

Reference to Literature See chapter entitled "Literature" [15].

5.18 Active Note in NC Program (only NC Record Number): APN

MWCX device group

Designation **APN** **A**ctive **P**art-Program Message-**N**umber

Explanation The NC record number of the active note of an NC process of the MWCX device group is output. Every NC record can contain a note that is displayed in the diagnostics menu of the Bosch Rexroth GUI after the NC record has been processed. The note in the diagnostics line remains active until it is overwritten by a new note (also refer to "Active Note in NC Program (Note and NC Record Number): APM").

FI command

CR_APN_(1)	**(Single Read)**
CC_APN_(1)	**(Cyclic Read)**
CB_APN_(1)	**(Break Cyclic Read)**
(1) = NC process number	[0...6]

Response Structure One line with one column is output for the NC record number of the active note.

Line 1	Column 1

Value Range/Meaning of Columns

1 = NC record number of the note	[0000...9999]

Note: If the current NC program does not contain a note, then the result of Column 1 is [0000].

Example APN Read the NC record number of the active note in NC process 0 of device address 00.

FI command		00_CR_APN_0
Line	**Column**	**Answer**
1	1	0002

Reference to Literature See chapter entitled "Literature" [15].

5.19 Actual (Current) Position Value of an Axis: APO

MWCX device group

Designation **APO** Actual Axis **PO**sition

Explanation The actual position of a selected axis is read out. The FI command "APO1" returns the position of an axis, related to the code of the axis meaning. On the other hand, the FI command "APO2" returns the position of an axis, related to the physical axis number.

FI command Output the position of the selected axis of the device specified, related to the code of the axis meaning.

Using the optional fourth parameter it is possible to pre-select conversion of the result into mm or inches. If, however, a spindle is selected as an axis, indicating a measurement system serves no purpose.

CR_APO1_(1)_(2)_(3){_(4)}	**(Single Read)**
CC_APO1_(1)_(2)_(3){_(4)}	**(Cyclic Read)**
CB_APO1_(1)_(2)_(3){_(4)}	**(Break Cyclic Read)**
(1) = NC process number	[0...6]
(2) = Axis meaning	[0...11] (see Chapter "Data Tables")
(3) = System of coordinates	[1 = machine coordinates 2 = program coordinates]
(4) = Required measurement system (opt.)	[mm, inch]

FI command Output the position of the selected axis of the device specified, related to the physical axis number.

Using the optional third parameter it is possible to pre-select conversion of the result into mm or inches. If, however, a spindle is selected as an axis, indicating a measurement system serves no purpose.

CR_APO2_(1)_(2){_(3)}	**(Single Read)**
CC_APO2_(1)_(2){_(3)}	**(Cyclic Read)**
CB_APO2_(1)_(2){_(3)}	**(Break Cyclic Read)**
(1) = Physical axis number	[1...32, according to settings of the system parameters]
(2) = System of coordinates	[1 = machine coordinates 2 = program coordinates]
(3) = Required measurement system (opt.)	[mm, inch]

Response Structure The following table shows the general structure of the response to the FI commands "APO1" and "APO2". One line is output with 4 columns for the axis designation, position, unit and the position limited to "indicated decimal places".

Line 1	Column 1	Column 2	Column 3	Column 4

Value Range/Meaning of Columns

1 = Axis name	[according to settings of axis parameters]
2 = Position	[according to settings of process parameters]
3 = Unit	[according to settings of process parameters: mm, inch]
4 = Position	[as Column 2, but rounded up or down according to the parameter "indicated decimal places"]

Note: If the selected axis is not defined then the response in all columns is "--".

Example APO1 Read the current position of the Z axis in machine coordinates in NC process 0 of device address 00. Values are displayed in the basic measurement system.

FI command	**00_CR_APO1_0_2_1**			
Answer				
Line	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	Z1	-1.2345	[mm]	-1.235

Example APO1

Read the current position of the Z axis in machine coordinates in NC process 0 of device address 00. Values are displayed in inches.

FI command	**00_CR_APO1_0_2_1_inch**			
Answer				
Line	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	Z1	-0.0486	[inch]	-0.049

Example APO2

Reads the current position of the Z axis (physical axis number = 3) in machine coordinates for the device address 00. The values are displayed in the basic measuring system.

FI command	**00_CR_APO2_3_1**			
Answer				
Line	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	Z1	-1.2345	[mm]	-1.235

Reference to Literature

See chapter entitled "Literature" [16].

5.20 Active NC Program Number: APP

MWCX device group

Designation

APP Active **P**art-**P**rogram Number

Explanation

The active NC program number of an NC process is read out.

FI command

CR_APP_(1)	**(Single Read)**
CC_APP_(1)	**(Cyclic Read)**
CB_APP_(1)	**(Break Cyclic Read)**
(1) = NC process number	[0...6]

Response Structure

The following table shows the general structure of the response to the FI command "APP". One line with 2 columns is output for the NC memory and the NC program number.

Line 1	**Column 1**	**Column 2**

Value Range/Meaning of Columns

1 = NC memory	[A = memory A; B = memory B]
2 = NC program number	[01...99]

or in setup mode:

1 = MDI	(instead of NC memory)
2 = --	(instead of NC program number)

Example APP Read the active NC program number in NC process 0.

FI command		**00_CR_APP_0**
Line	**Column**	**Answer**
1	1	A
	2	01

or in setup mode:

FI command		**00_CR_APP_0**
Line	**Column**	**Answer**
1	1	MDI

Note! To reach the active zero-offset bank or the preselected NC memory, the **CR_SPP** command (= preselected NC program) must be called up.

Reference to Literature See chapter entitled "Literature" [17].

5.21 Axis Reference Flags: ARF

MWCX device group

Designation **ARF** **A**xis **R**eference **F**lags

Explanation The reference flags for a process are to be displayed. These flags exist for the interpolation axes {X,Y,Z,U,V,W,A,B,C}

FI command

CR_ARF_(1)	**(Single Read)**
CC_ARF_(1)	**(Cyclic Read)**
CB_ARF_(1)	**(Cyclic Break)**
(1) = NC process number	[0...6]

Response Structure A line with 9 columns is output, each for the axis meaning: X, Y, Z, U, V, W, A, B, C axis.

An axis reference flag can have the following three values:

0	Axis not in reference
1	Axis in reference
--	Axis not present

Example ARF Displays the axis reference flags for process 0

Assumption:

- X, Y, Z axes are in reference,
- U, V, W axes are not in reference,
- A, B, C axes are not present

FI command		**00_CR_ARF_0**							
Answer									
Line	**Col.1**	**Col.2**	**Col.3**	**Col.4**	**.5**	**.6**	**.7**	**.8**	**.9**
1	1	1	1	0	0	0	--	--	--

5.22 Actual (Current) Rapid Override: ARO

MWCX device group

Designation **ARO** Actual **R**apid **O**verride

Explanation The current value of the rapid override of an NC process of the MWCX device group is output. This value is evaluated by the NC for all axis movements that are executed with "G00". The valid range of override weighting by the PLC program is between 0 and 255%.

FI command Output the current value of the feedrate override of an NC process of the selected device from the MWCX device group.

CR_ARO1_(1)	**(Single Read)**
CC_ARO1_(1)	**(Cyclic Read)**
CB_ARO1_(1)	**(Break Cyclic Read)**
(1) = NC process number	[0...6]

Response Structure The following table shows the general structure of the response to the FI command "ARO". One line with three columns is output for the identifier, the current value of the rapid override and the unit [%].

Line 1	Column 1	...	Column 3

Value Range/Meaning of Columns

1 = Identifier	[ROV= rapid override]
2 = Current value of the rapid override	[0...255]
3 = Unit	[%]

Note: The valid range of override weighting by the PLC program is between 0 and 255%. The NC limits the axis and/or processor speed to the maximum values set in the parameters if an override value is set that is too large.

Example ARO1 Read the current value of the rapid override in NC process 0 of device address 00.

FI command	**00_CR_AFO1_0**		
Answer			
Line	**Column 1**	**Column 2**	**Column 3**
1	ROV	100	[%]

Reference to Literature See chapter entitled "Literature" [18].

5.23 Axis Reference Table: ART

MWCX device group

Designation **ART** **A**xis **R**eference **T**able

Explanation The complete axis reference tables for a system are requested. They can be used to determine to which process the physical axes are assigned and with which axis meaning. They can also determine in which process dynamically assigned axes are possible.

FI command Read the axis reference tables of all processes 0-6. If the optional parameter is input, the output is limited to one process.

CR_ART1{_(1)}	**(Single Read)**
CC_ART1{_(1)}	**(Cyclic Read)**
(1) = NC process number (optional)	[0...6]

Response Structure Seven lines are output (1 optional line) each with 12 columns and each having the axis number of the assigned axis. The first to twelfth column receives the physical axis number corresponding to the axis meaning X,Y,Z,U,V,W, A,B,C,S1,S2,S3. If a process does not have an axis for an axis meaning, then the result in this column is [- -].

If an axis can be assigned dynamically during operation, then the result in the first column is "*X" and the result in the other columns is "*Y" to "*S". If there is no process present at all, then the result in all columns for this line is [- -].

Line1...7	Column 1	Column 2	...	Column 12

With the following meaning:

Line 1, 2...7:	Axis reference table for process 0, 1...6
Column 1, 2...12:	Axis number for axis meaning X, Y...S3

Example ART1 Reads the complete axis reference table for device 00

Assumption:

Processes 0, 1 and 4 are present
and the axes:

1 (X axis in process 0),

2 (Y axis in process 0 or process 1),

3 (X axis in process 4),

4 (S axis in process 0 or process 4),

<table>
<tr><td>FI command</td><td colspan="4">00_CR_ART1</td></tr>
<tr><td colspan="5">Answer</td></tr>
<tr><td>Line</td><td>Column 1</td><td>Column 2</td><td>...</td><td>Column 12</td></tr>
<tr><td>1</td><td>1</td><td>2</td><td>...</td><td>--</td></tr>
<tr><td>2</td><td>--</td><td>*Y</td><td>...</td><td>4</td></tr>
<tr><td>3</td><td>--</td><td>--</td><td>...</td><td>--</td></tr>
<tr><td>4</td><td>--</td><td>--</td><td>...</td><td>--</td></tr>
<tr><td>5</td><td>3</td><td>--</td><td>...</td><td>*S</td></tr>
<tr><td>6</td><td>--</td><td>--</td><td>...</td><td>--</td></tr>
<tr><td>7</td><td>--</td><td>--</td><td>...</td><td>--</td></tr>
</table>

Example ART1

Reads the axis reference table for process 1 of device 00:

FI command	00_CR_ART1_1			
Answer				
Line	**Column 1**	**Column 2**	**...**	**Column 12**
1	--	*Y	...	4

FI command

For a device which is offline (DeviceStatus=OFF), the axis reference data is simulated according to the current parameter record.

BR_ART **(Single Read)**

Response Structure

The following table shows the general structure of the response to the FI command "ART". A line of 1 column is output.

Line 1	Column 1

Value Range/Meaning of Columns

1 = Binary axis reference table data

Example: ART

Read the binary axis reference table data of the device 00.

FI command		00_BR_ART
Line	**Column**	**Answer**
1	1	Read axis reference table data

5.24 Actual (Current) Spindle Data: ASD

MWCX device group

Designation

ASD **A**ctual **S**pindle **D**ata

Explanation

The current spindle data of an NC process of the selected device from the MWCX device group is read out.

This command is a compilation of PSS, ASS, MSS, ASO, MSO and ASG.

FI command

Output the current axis data of an NC process related to the spindle number.

CR_ASD_(1)_(2) **(Single Read)**

CC_ASD_(1)_(2) **(Cyclic Read)**

CB_ASD_(1)_(2) **(Break Cyclic Read)**

(1) = NC process number [0...6]

(2) = Spindle number [1...3]

Response Structure

The following table shows the general structure of the response to the FI command "ASD". A line with 9 columns is output for axis denomination, current spindle speed, programmed spindle speed, maximum spindle speed, and the unit according to settings of the process parameters, current spindle override, maximum spindle override, and the current gear level.

Line 1	Column 1	...	Column 9

Value Range/Meaning of Columns

1 = Axis name	[S, S1, S2, S3]
2 = Current spindle speed	[according to settings of axis parameters]
3=programmed Spindle speed	[according to settings of axis parameters]
4= max. spindle speed	[according to settings of axis parameters]
5 = Unit	1/min
6 = Current spindle override	[0 ... MAXSOVR]
7= max. spindle override	[according to settings of axis parameters]
8 = Gear identifier	[g]
9 = Current gear level	[1 ... 3, --]

Note: If the selected spindle is not defined in the selected NC process, the result contains the value "--" in all the columns; if it is presently not assigned, "*S" will appear in the first, and "--" in the remaining columns.

Example: ASD

Read the current data of the 1st spindle in NC process 0 of device address 00.

FI command	**00_CR_ASD_0_1**		
Answer			
Line	**Column**	**Value**	Meaning
1	1	S1	Axis Designation
	2	3000.0	Progr. Spindle speed
	3	2999.9	Current spindle speed
	4	5000.0	Maximum spindle speed
	5	1/min	Unit
	6	100%	Current spindle override
	7	120%	Maximum spindle override
	8	g	Gear level identifier
	9	1	Current gear level

Reference to Literature

See chapter entitled "Literature" [21].

See chapter entitled "Literature" [22].

5.25 Determining the Current System Error: ASE

MWCX device group

Designation **ASE** Actual **S**ystem **E**rror

Explanation The current system error is read out, whereby the response 0x0000 indicates that the SERCANS card is functioning correctly.

FI command

CR_ASE **(Single Read)**

CC_ASE **(Cyclic Read)**

CB_ASE **(Break Cyclic Read)**

Response Structure The following table shows the general structure of the response to the FI command ASE. In line 1, column 4, the number of the drive is output that reports the current system error. Not all current system errors can be directly allocated to a drive. In this case, the single result "Drive No." is set to 0x0000.

Line 1	Column 1	...	Column 4

Value Range/Meaning of Columns

1 = 0x0000

2 = 0x0000

3 = Actual system error

4 = Drive No.

Example ASE Reading the current system error returns LWL ring interrupted.

FI command		00_CR_ASE
Line	**Column**	**Answer**
1	1	0x0000
	2	0x0000
	3	0x8009
	4	0x0000

Reference to Literature See chapter entitled "Literature" [42].

5.26 Active Spindle for Process: ASF

MWCX device group

Designation **ASF** Active **S**pindle **F**or Process

Explanation The active (selected) spindle of the selected NC process is output. As there can be several spindles in an NC process, it is necessary for certain NC functions such as G96 (constant cutting speed), that these are active on another spindle as well as on the first spindle. The following NC functions are dependent on the selected main spindle:

- G33 thread cutting
- G63/G64 tapping
- G65 tapping; spindle serves as leading axis
- G95 feed per turn and
- G96 constant cutting speed.

FI command

CR_ASF_(1)	**(Single Read)**
CC_ASF_(1)	**(Cyclic Read)**
CB_ASF_(1)	**(Break Cyclic Read)**
(1) = NC process number	[0...6]

Response Structure The response to the FI-Command "ASF" consists of one line with one column for the selected active spindle.

Active Spindle for Process:	[S1, S2, S3, *S]

Note: If no active spindle is selected in the NC process, then the response for Column 1 is [*S].

Example ASF Reads the selected active spindle in an NC process 0 of device address 00.
Assumption:

- A main circular-axis spindle (S1) has been defined in NC process 0,
- The spindle has been selected as active spindle by the NC command "SPF 1" and
- The G function "G96" is active in the NC program.

FI command		**00_CR_ASF_0**
Line	**Column**	**Answer**
1	1	S1

Reference to Literature See chapter entitled "Literature" [19].

5.27 Actual (Current) Spindle Gear Level: ASG

MWCX device group

Designation **ASG** Actual **S**pindle **G**ear

Explanation The current spindle gear level of an NC process of the selected device from the MWCX device group is read out. The control signals of the gear selection are only evaluated by the CNC when one gear with at least two gear levels has been entered within the axis parameters.

FI command

CR_ASG_(1)_(2)	**(Single Read)**
CC_ASG_(1)_(2)	**(Cyclic Read)**
CB_ASG_(1)_(2)	**(Break Cyclic Read)**
(1) = NC process number	[0...6]
(2) = Spindle number	[1...3]

Response Structure The response to the "ASG" FI command consists of one line with two columns for the identifier and for the current spindle gear level of the selected NC process.

Line 1	Column 1	Column 2

Value Range/Meaning of Columns

1 = Identifier	[g = gear]
2 = Current spindle gear level	[1...3, -]

Note: If no current spindle gear level is selected in the NC process or in the NC program then Column 1 receives the value [g] and Column 2 the value [-].

Example ASG Read the current spindle gear level of the 1st spindle in NC process 0 of device address 00.

FI command		**00_CR_ASG_0_1**
Line	**Column**	**Answer**
1	1	g
	2	1

Reference to Literature See chapter entitled "Literature" [20].

5.28 Active System Error Messages: ASM

MWCX device group

Designation **ASM** Active **S**ystem **M**essages

Explanation The current device information (system errors, device conditions) that effect the functioning of the entire electrical device are output Depending on the FI command, the device address, device name, message number, type of message, short text and reference text are all output.

FI command Output of the current device information (system errors, device conditions) that is pending for all active devices from the MWCX device group.

BR_ASM1	**(Single Read)**
BC_ASM1	**(Cyclic Read)**
BB_ASM1	**(Break Cyclic Read)**

Note: The "ASM1" FI command refers to all devices within this device group. This means that any valid device address can be indicated in the command line (see Example ASM1).

Response Structure The following table shows the general structure of the response to the FI command "ASM1". The number of lines (1 .. n=15) depends on the number of defined devices. Each line consists of 8 columns for the device address, device name, message number, message status, short text and indication of whether there is an reference text for this error message.

Line 1...n	Column 1	...	Column 8

Value Range/Meaning of Columns

1 =	Device address	[00...63]
2 =	Device name	[max. 32 ASCII characters]
3 =	Message number	[0...150]
4 =	Type of message	[F = fault/error, D = diagnosis]
5 =	Message text	[max. 54 ASCII characters]
6 =	Reference text	[x= exists, -- = does not exist]
7 =	2 bytes of additional information for the message number	is required to resolve the information "@" (see ASM5)
8 =	Filename for additional information for message text	e.g. in HTML format

Example ASM1 Reads the current device information (system errors, device conditions) of all defined devices of the MWCX device group.

<u>Assumption:</u>
The following three devices are defined:

- Device address 01
- Device address 07 and
- Device address 10.

FI command		07_BR_ASM1
Line	**Column**	**Answer**
1	1	01
	2	Drill center
	3	71
	4	F
	5	PLC battery voltage too low.
	6	X
	7	0
	8	
2	1	07
	2	Milling center 1
	3	74
	4	F
	5	SLM time monitoring
	6	X
	7	0
	8	
3	1	10
	2	Milling center 2
	3	1
	4	D
	5	Error has been corrected.
	6	X
	7	0
	8	

FI command

Output of the current device information (system errors, device conditions) that is pending for the selected device from the MWCX device group.

BR_ASM2	**(Single Read)**
BC_ASM2	**(Cyclic Read)**
BB_ASM2	**(Break Cyclic Read)**

Response Structure

The following table shows the general structure of the response to the FI command "ASM2". The answer consists of a line of 8 columns for the device address, device name, message number, message status, short text and indication of whether there is an reference text for this device information (system errors, device conditions).

Line 1...n	Column 1	...	Column 8

Value Range/Meaning of Columns

1 =	Device address	[00...63]
2 =	Device name	[max. 32 ASCII characters]
3 =	Message number	[0...150]
4 =	Type of message	[F = fault/error, D = diagnosis]
5 =	Message text	[max. 54 ASCII characters]
6 =	Reference text	[x= exists, -- = does not exist]
7 =	2 bytes of additional information for the message number	is required to resolve the information "@" (see ASM5)
8 =	Filename for additional information for message text	e.g. in HTML format

Example ASM2

Read the current device information (system errors, device conditions) of device address 01.

Assumption:
The following three devices are defined:

- Device address 01
- Device address 07 and
- Device address 10

FI command		**01_BR_ASM2**
Line	**Column**	**Answer**
1	1	01
	2	Drill center
	3	71
	4	F
	5	PLC battery voltage too low.
	6	X
	7	0
	8	

FI command

Output of the current device information (system errors, device conditions) of the device listed from the MWCX device group.

BR_ASM3_(1) **(Single Read)**

BC_ASM3_(1) **(Cyclic Read)**

BB_ASM3_(1) **(Break Cyclic Read)**

(1) = Selection list for a max. of 10 MWCX devices [00_01_ ... _15]

Response Structure

The following table shows the general structure of the response to the FI command "ASM3". The number of lines (1 .. n=15) depends on the number of MWCX devices listed. Each line consists of 8 columns for the device address, device name, message number, message status, short text and indication of whether there is an reference text for this device information (system errors, device conditions).

Line 1...n	**Column 1**	**...**	**Column 8**

Value Range/Meaning of Columns

1 =	Device address	[00...63]
2 =	Device name	[max. 32 ASCII characters]
3 =	Message number	[0...150]
4 =	Type of message	[F = fault/error, D = diagnosis]
5 =	Message text	[max. 54 ASCII characters]
6 =	Reference text	[x= exists, -- = does not exist]
7 =	2 byte additional information for the message number	is required to resolve the information "@" (see ASM5)
8 =	Filename for additional information for message text	e.g. in HTML format

Example ASM3

Reads the current device information (system errors, device conditions) of the selected MWCX devices.

Assumption:
The following devices addresses are defined:

- Device address 01,
- Device address 07 and
- Device address 10.

FI command		**01_BR_ASM3_01_10**
Line	**Column**	**Answer**
1	1	01
	2	Drill center
	3	71
	4	F
	5	PLC battery voltage too low.
	6	X
	7	0
	8	
2	1	10
	2	Milling center 2
	3	1
	4	D
	5	Error has been corrected.
	6	X
	7	0
	8	

FI command

Output of the current device information (system errors, device conditions) of all defined devices (in accordance with the system configuration) from the MWCX device group.

BR_ASM4_(1)	**(Single Read)**
BC_ASM4_(1)	**(Cyclic Read)**
BB_ASM4_(1)	**(Break Cyclic Read)**
(1) = Device group	[MWCX, MISX]

Response Structure

The following table shows the general structure of the response to the FI command "ASM4". The number of lines (1 .. n=15) depends on the number of MWCX devices defined. Each line consists of 8 columns for the device address, device name, message number, message status, short text and indication of whether there is an reference text for this device information (system errors, device conditions)

Line 1...n	Column 1	...	Column 8

Value Range/Meaning of Columns

1 =	Device address	[00...63]
2 =	Device name	[max. 32 ASCII characters]
3 =	Message number	[0...150]
4 =	Type of message	[F = fault/error, D = diagnosis]
5 =	Message text	[max. 54 ASCII characters]
6 =	Reference text	[x= exists, -- = does not exist]
7 =	2 bytes of additional information for the message number	is required to resolve the information "@" (see ASM5)
8 =	Filename for additional information for message text	e.g. in HTML format

Example ASM4

Reads the current device information (system errors, device conditions) of all defined devices of the MWCX device group.
Assumption:
The following devices addresses are defined:

- Device address 01 and
- Device address 10.

FI command		**01_BR_ASM4_MWCX**
Line	**Column**	**Answer**
1	1	01
	2	Drill center
	3	71
	4	F
	5	PLC battery voltage too low.
	6	X
	7	0
	8	
2	1	10
	2	Milling center 2
	3	1
	4	D
	5	Error has been corrected.
	6	X
	7	0
	8	

FI command

Output the reference text for the currently pending device information (system errors, device conditions), related to the device and the message number.

BR_ASM5_(1)_(2)	**(Single Read)**
(1) = Message number	[0...150]
(2) = 2 bytes of additional information for the message number	

Response Structure

The following table shows the general structure of the response to the FI command "ASM5". The response consists of a line with 6 columns for device address, device name, message number and reference text.

Line 1...n	Column 1	...	Column 6

Value Range/Meaning of Columns

1 =	Device address	[00...63]
2 =	Device name	[max. 32 ASCII characters]
3 =	Message number	[0...150]
4 =	Type of message	[F = fault/error, D = diagnosis]
5 =	Reference text	[max. [max. 14 lines with a max. 78 characters/line]
6 =	Filename for additional information for reference text	e.g. in HTML format

Example ASM5

Read the reference text relating to the device information (system errors, device conditions) with message number 74 of device address 01.

<table>
<tr><th colspan="2">FI command</th><th>01_BR_ASM5_74_0</th></tr>
<tr><th>Line</th><th>Column</th><th>Answer</th></tr>
<tr><td>1</td><td>1</td><td>01</td></tr>
<tr><td></td><td>2</td><td>Drill center</td></tr>
<tr><td></td><td>3</td><td>74</td></tr>
<tr><td></td><td>4</td><td>F</td></tr>
<tr><td></td><td>5</td><td>Replace the SLM module on the PLC card (slot 3).</td></tr>
<tr><td></td><td>6</td><td></td></tr>
</table>

Reference to Literature

See chapter entitled "Literature" [13].

5.29 Actual (Current) NC Sequence Number: ASN

MWCX device group

Designation **ASN** Actual **S**equence **N**umber

Explanation The active NC sequence number of an NC process of the selected device from the MWCX device group is output.

FI command

CR_ASN_(1)	**(Single Read)**
CC_ASN_(1)	**(Cyclic Read)**
CB_ASN_(1)	**(Break Cyclic Read)**
(1) = NC process number	[0...6]

Response Structure The response to the "ASN" FI command consists of one line with one column for the active NC sequence number [N0000...N9999].

Line 1	Column 1

Note: If no valid NC program exists then Column 1 receives the value [N0000].

Example ASN Read the active NC sequence number of NC process 0 of device address 00.

FI command		00_CR_ASN_0
Line	**Column**	**Answer**
1	1	N0002

Reference to Literature See chapter entitled "Literature" [4].

5.30 Actual (Current) Spindle Override: ASO

MWCX device group

Designation **ASO** Actual **S**pindle **O**verride

Explanation The current value of the spindle override of an NC process of the MWCX device group is output. Override is valid for all non-interpolating axes (i.e. for spindle axes and magazine axes). Override is interpreted in the NC, irrespective of the mode; it has an effect on any axis movement (except on homing digital axes).

FI command Output the current value of the override of the selected device of the MWCX device group related to the NC process and the spindle number.

CR_ASO1_(1)_(2)	**(Single Read)**
CC_ASO1_(1)_(2)	**(Cyclic Read)**
CB_ASO1_(1)_(2)	**(Break Cyclic Read)**
(1) = NC process number	[0...6]
(2) = Spindle number	[1...3]

Response Structure The following table shows the general structure of the response to the FI command "ASO1". One line with three columns is output for the identifier, the current value of the override and the unit [%].

Line 1	Column 1	...	Column 3

Value Range/Meaning of Columns

1 = Identifier	[S= Spindle]
2 = Current value of the override with unit	[0...255]
3 = Unit	[%]

Note: The valid range of override weighting by the PLC program is between 0 and 255 %. The NC limits the axis and/or processor speed to the maximum values set in the parameters if an override value is set that is too large.

If the spindle number is not defined within the selected process then the result in Column 1 is [--].

Example ASO1 Read the current value of the override of Spindle 1 in NC process 0 of device address 00.

FI command	**00_CR_ASO1_0_1**		
Answer			
Line	**Column 1**	**Column 2**	**Column 3**
1	S:	60	[%]

Reference to Literature See chapter entitled "Literature" [21].

5.31 Actual (Current) Spindle Speed: ASS

MWCX device group

Designation **ASS** Actual **S**pindle **S**peed

Explanation The current spindle speed (axis velocity) of an NC process of the selected device from the MWCX device group is read out.

FI command Output the current axis speed of an NC process related to the spindle number.

CR_ASS_(1)_(2)	**(Single Read)**
CC_ASS_(1)_(2)	**(Cyclic Read)**
CB_ASS_(1)_(2)	**(Break Cyclic Read)**
(1) = NC process number	[0...6]
(2) = Spindle number	[1...3]

Response Structure The following table shows the general structure of the response to the FI command "ASS". One line with three columns for the name of the axis, the axis speed and the unit is output in accordance with the settings of the process parameters.

Line 1	**Column 1**	**...**	**Column 3**

Value Range/Meaning of Columns

1 = Axis name	[S, S1, S2, S3]
2 = Spindle speed	[according to settings of axis parameters]
3 = Unit	[1/min; according to parameter setting]

Note: If the spindle number is not defined in the selected NC process, then the result in Column 1 is [--], the result in Column 2 is [0.0] and that in Column 3 is [1/min].

Example ASS Read the current axis speed of the 1st spindle in NC process 0 of device address 00.

FI command	**00_CR_ASS_0_1**		
	Answer		
Line	**Column 1**	**Column 2**	**Column 3**
1	S1	4000.0	1/min

Reference to Literature See chapter entitled "Literature" [22].

5.32 Active Tool Number: ATN

MWCX device group

Designation **ATN** **A**ctive **T**ool-**N**umber

Explanation The active tool number of an NC process of the selected device from the MWCX device group is read out.

FI command

CR_ATN_(1)	**(Single Read)**
CC_ATN_(1)	**(Cyclic Read)**
CB_ATN_(1)	**(Break Cyclic Read)**
(1) = NC process number	[0...6]

Response Structure The response for the "ATN" FI command consists of one line with two columns for the identifier and the number of the active tool.

Line 1	**Column 1**	**Column 2**

Value Range/Meaning of Columns

1 = Identifier	[T = tool]
2 = Number of active tool	[1...9999999]

Note: If no tool is active in the selected NC process then Column 1 receives the value [T] and Column 2 the value [0].

Example ATN Read the number of the active tool in NC process 0 of device address 00.

FI command		**00_CR_ATN_0**
Line	**Column**	**Answer**
1	1	T
	2	4

Reference to Literature See chapter entitled "Literature" [23].

5.33 Reading Actual (Current) Tool Place Information: ATP

MWCX device group

Designation **ATP** Actual **T**ool **P**lace Information

Explanation Information regarding the tool place and the current edge of the pre-selected tool is returned by the "ATP" command. The control unit response telegram also returns information on the current position of the tool magazine. For this reason, the "ATP" access has 3 filter options. The following information is returned by the control unit upon the FI command "ATP":

- ATP1 Set and actual position of the tool magazine and edge place information for the active tool
- ATP2 Edge and place information for the active tool.
- ATP3 Set and actual position of the tool magazine.

The FI command refers to the indicated NC process. If the control is not able to return any data, then the corresponding partial result [--] is transmitted.

FI command Set and actual position of the tool magazine and edge place information of the active tool

CR_ATP1_(1)	**(Single Read)**
CC_ATP1_(1)	**(Cyclic Read)**
CB_ATP1_(1)	**(Break_Cyclic Read)**
(1) = NC process number	[0...6]

Response Structure The following table shows the general structure of the response to the "ATP1" FI command . One line with 4 columns is output for the returned values.

Line 1	Column 1	...	Column 4

Value Range/Meaning of the Columns

1 =	Set position of magazine	[1...999]	
2 =	Actual position of magazine	[1...999]	
3 =	Active tool edge number	[1...9]	
4 =	Tool place (type + place number)	[Mx= magazine/turret Sx = spindle Gx = gripper	[x=1...999] [x=1...4] [x=1...4]]

Note: Details of the current command and actual position of the tool magazine refer to the reference point of the magazine controller.

Example ATP1 Read the position of the tool magazine plus edge and tool-place information for the active tool from NC process 0 of device 00.

FI command		**00_CR_ATP1_0**
Line	**Column**	**Answer**
1	1	3
	2	3
	3	1
	4	S1

FI command

Edge and place information for the active tool.

CR_ATP2_(1)	**(Single Read)**
CC_ATP2_(1)	**(Cyclic Read)**
CB_ATP2_(1)	**(Break_Cyclic Read)**
(1) = NC process number	[0...6]

Response Structure

The following table shows the general structure of the response to the "ATP2" FI command. One line with 2 columns is output for the returned values.

Line 1	Column 1	Column 2

Value Range/Meaning of the Columns

1 =	Active tool edge number	[1...9]	
2 =	Tool place (type + place number)	[Mx= magazine/turret Sx = spindle Gx = gripper	[x=1...999] [x=1...4] [x=1...4]]

Example ATP2

Reads the edge and tool place information of the active tool from NC process 0 of device 00.

FI command		**00_CR_ATP2_0**
Line	**Column**	**Answer**
1	1	1
	2	S1

FI command

Output of the position information of the tool magazine.

CR_ATP3_(1)	**(Single Read)**
CC_ATP3_(1)	**(Cyclic Read)**
CB_ATP3_(1)	**(Break_Cyclic Read)**
(1) = NC process number	[0...6]

Response Structure

The following table shows the general structure of the response to the "ATP3" FI command. One line with 2 columns is output for the returned values.

Line 1	Column 1	Column 2

Value Range/Meaning of the Columns

1 = Command position of magazine	[1...999]
2 = Actual position of magazine	[1...999]

Note: Details of the current command and actual position of the tool magazine refer to the reference point of the magazine controller.

Example ATP3

Read the command and actual position of the tool magazine from NC process 0 of device 00.

FI command		**00_CR_ATP3_0**
Line	**Column**	**Answer**
1	1	3
	2	3

Reference to Literature

See chapter entitled "Literature" [8].

5.34 Access to Actual (Current) Tool Data Record: ATR

MWCX device group

Designation **ATR** Actual **T**ool data **R**ecord

Explanation Returns a complete basic data record and/or cutter data record of the current processing tool.

FI command Read basic data record or cutter data record of the current tool.

CR_ATR_(1)_(2)	**(Single Read)**
CC_ATR_(1)_(2)	**(Cyclic Read)**
CB_ATR_(1)_(2)	**(Break Cyclic Read)**
(1) = NC process number	[0...6]
(2) = Data record	[0 = base tool data, 1...9 = cutter data]

Response Structure The following table shows the general structure of the response to the "CR_ATR" FI command. One line is output with 28 (basic data) or 40 (cutter data) columns for the returned values.

Line 1	Column 1	...	Column 28/40

Value Range/Meaning of the Columns

1...28 = Requested base tool data	[max. 28 data elements] (see basic value range data)
1...40 = Requested tool cutter data	[max. 40 data elements] (see value range of cutter data)

Example TDR1 Read the base tool data record in NC process 0 of the tool currently processing.

FI command		00_CR_ATR_0_0
Line	**Column**	**Answer**
1	1	928
	2	Miller D20
	3	S:
	4	1
	5	1234567
	6	1234
	7	2
	8	1
	9	..............+..p............
	10	0
	11	M1
	12	M
	13	--
	14	M
	15	--
	16	[cycl]
	17	[mm]
	18	4

	19	102
	20	0.000000
	21	0.000000
	22	0.000000
	23	0.000000
	24	0.000000
	25	0.000000
	26	0.000000
	27	0.000000
	28	0.000000

Reference to Literature See chapter entitled "Literature" [8].

5.35 Accepting the Data Record for the Current Tool: ATU

MWCX device group

Designation **ATU** **A**ctual **T**ool data **U**pdate

Explanation The current tool data record that has been changed following editing is accepted by the CNC.

FI command **CR_ATU_(1)** **(Single Read)**

(1) = NC process number [0...6]

Response Structure One line is output with a column for acknowledgement of whether or not the FI command has been executed successfully.

(P_ACK) = **P**ositive **ACK**nowledge New data record of current tool has been accepted

Example ATU Accept the changed data record of the current tool in NC process 0 of device address 00.

FI command		**00_CR_ATU_0**
Line	**Column**	**Answer**
1	1	(P_ACK)

5.36 Active Zero-Offset Bank: AZB

MWCX device group

Designation **AZB** Active **Z**ero-**O**ffset Bank

Explanation The number of the active zero-offset bank of an NC process of the selected device from the MWCX device group is read out.

The zero offsets allow the origin of a coordinate axis to be shifted (offset) by a set value, related to the original position of the machine. A record of these shifts is held in the zero-offset banks.

FI command

CR_AZB1_(1)	**(Single Read)**
CC_AZB1_(1)	**(Cyclic Read)**
CB_AZB1_(1)	**(Break Cyclic Read)**
(1) = NC process number	[0...6]

Response Structure The response to the "AZB1" FI command consists of one line with two columns for the identifier (O = offset) and the number of the active zero-offset bank [0...2].

Line 1	Column 1	Column 2

Example AZB Read the number of the active zero-offset bank in NC process 0 of device address 00.

FI command		00_CR_AZB1_0
Line	**Column**	**Answer**
1	1	O
	2	2

Reference to Literature See chapter entitled "Literature" [24].

5.37 NC Cycle Download: CCA

MWCX device group

Designation **CCA** N**C**-**C**ycle **A**ccess

Explanation NC cycles are downloaded by means of the download file and NC cycle files via all active processes.

FI command NC Cycle Download.

BW_CCA1_(1)	**(Single Write)**

(1) = Download file with path details.

Note: File and path details must be enclosed in inverted commas.

Response Structure The response to the "CCA1" FI command consists of three lines, each with one column. The meaning of the elements is as follows:

Line 1 = Job ID [01...20]
(see Chapter "FI Commands for the MPCX Device Group: IFJ").

Line 2 = FI command
[String, in accordance with Chapter "Elements of the FI Command"]

Line 3 = FI Job Error Code
(see Chapter "Error Codes")

Example CCA1 00_BW_CCA1_"D:\Program Files\Indramat\Mtgui\Temp\download.ini"/3

FI command		00_BW_CCA1_"D:\Program Files\Indramat\Mtgui\Temp\download.ini"/3
Line	**Column**	**Answer**
1	1	01
2	1	00_BW_CCA1_"D:\Program Files\Indramat\Mtgui\Temp\download.ini"/3
3	1	0

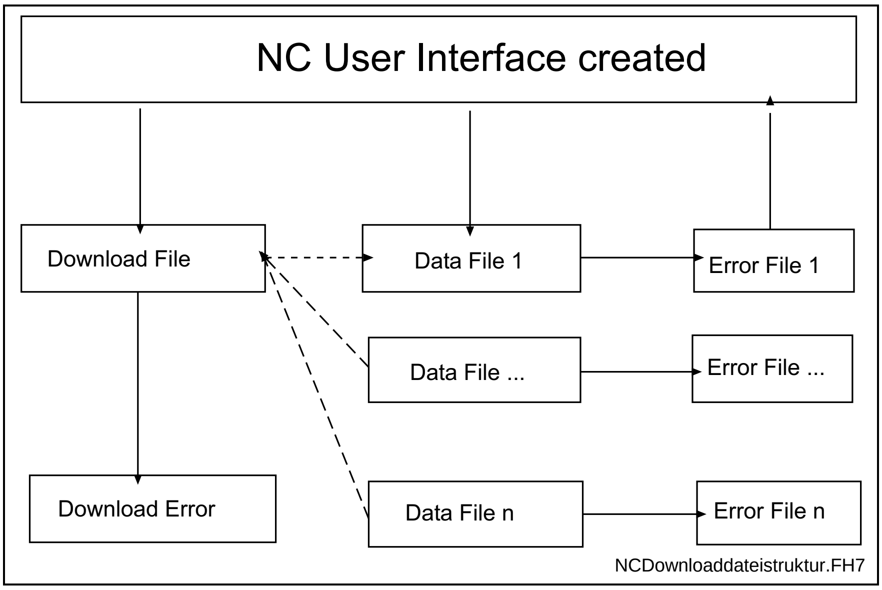


Fig. 5-1: File structure of the download file

Structure of Download File The structure of the "download.ini" file used in this example corresponds to an Ini file in Windows.

Note: Care must be taken in the use of upper and lower case letters.

Section [Common]
General information is stored in the "Common" section.

Key **Max_Compiler**
Number of compilers to be called. The compiler contains the control file as a pass parameter and translates the data into the respective data files. A pass value of zero signifies no compiler call.

This key is an optional value. If this value is not present, no compiler is active.

Key **DownloadError**
Indicates whether or not an error has occurred during downloading. This value is only set in the event of an error.

Example:

```
[Common]
DownloadError = YES   ; Error
Max_Compiler = 2
```

Section [CompilerXX]
This section contains information regarding the compiler. There is a separate section for each compiler. The name of the section consists of the "Compiler" text and a two digit number.

XX: is a two digit index which begins at 1 and has a maximum size of Max_Compiler.

Section [CycPackage_Info]

Key **Cycle package information**
The package identification is compiled from several keys. The total length of all package identifications must not exceed **a maximum of 84** characters. The length of the individual identifications is described below:

Package number	"**PackageNo**"	max. 2 characters
Package name	"**PackageName**"	max. 32 characters
Package size:	"**PackageSize**"	max. 8 characters left-justified
Package time:	"**PackageTime**"	max. 8 characters
Package date:	"**PackageDate**"	max. 8 characters
Package default:	"**PackageDefault**"	max. 26 characters (optional)

Total:		max. 84 characters

Information on date and time is given in the format

Date :	dd.mm.yy
Time:	hh:mm:ss

Example:

```
[CycPackage_Info]
PackageNo =         1
PackageName =       NC program package
PackageSize =       1234
PackageTime =       13:10:10
PackageDate =       24.12.00
```

Section list of NC cycle programs [ListOfCycPrograms]
The list of the NC cycle programs to be transferred is stored in the section "ListOfCycPrograms".

Key **Max_Index_Data**
Corresponds to the number of NC cycle programs to be transferred.

Key **consecutive index of the NC cycle programs**
Four-digit number starting with 1, identifies with a value the full file name of the NC cycle programs:

zzzzzzz	Data type (CYC-PRG)
xx	Process number
yyy	Program number of the cycle programs (1 – 255)

The file extension can be freely selected. ".dat" has been used in the following example.

Examples:

CYC-PRG-00-086	Cycle program for process 0 program 86
CYC-PRG-01-001	Cycle program for process 1 program 1

Example:

```
[ListOfCycPrograms]

Max_Index_Data=50

0001=K:\Program Files\Indramat\Mtgui\Project_000\CYC-PRG-00-01.dat
0002=K:\Program Files\Indramat\Mtgui\Project_000\CYC-PRG-01-01.dat
...
0050=K:\Program Files\Indramat\Mtgui\Project_000\CYC-PRG-06-99.dat
```

Data File Structure

These files contain the data for downloading and for the compiler. Their structure corresponds to the Windows "Ini" structure. The compiler uses this file for the input and output data.

Note: Care must be taken in the use of upper and lower case letters.

Data for the NC program is stored in the respective files as a section. It is composed of general data and the actual program.

Section [Common]

Program version:	**Version**	
Process:	**Process**	[0..6]
Program number:	**No**	[0..255]
Program name:	**Name**	max. 32 characters
Program size:	**Size**	
Program time:	**Time**	max. 8 characters
Program date:	**Date**	max. 8 characters
Program short identification:	**ShortID**	max. 8 characters
Program status:	**Status, (optional)**	

Information on date and time is given in the format
Date : dd.mm.yy
Time: hh:mm:ss

Status flag	Description
C	Compiled
E	Error
N	Not compiled
No details	No compiler call

Fig. 5-2: Status flags

Section Data

Key **Max_Index_Data**
Corresponds to the number of NC blocks to be transmitted

Key **consecutive index of NC records**
Five-digit number starting with 00001.

Note: An NC block should not contain any unnecessary blank spaces or NC comments. Equally, "PROGRAM END" may not occur as it is language-dependent.

Example:

```
[Data]
Max_Index_Data=25
00001=N0000 G0 X0 Y0 Z0
...
00025=N0024 .Start
```

5.38 NC Cycle Upload: CCA

MWCX device group

Designation **CCA** N**C**-**C**ycle **A**ccess

Explanation NC cycles are uploaded via all active processes. During upload, a basic file (upload file) and an NC cycle file are created.

FI command NC cycle upload.

BR_CCA1_(1) **(Single Read)**

(1) = Upload file with path details

Note: Enclose file and path details in inverted commas.

Response Structure The response to the CCA1 FI command consists of three lines, each with one column. The meaning of the elements is as follows:

- Line 1 = Job ID [01...20]
  (see Chapter "FI Commands for the MPCX Device Group", IFJ).
- Line 2 = FI command
  [String, in accordance with Chapter "Elements of the FI Command"]
- Line 3 = FI Job Error Code
  (see Chapter "Error Codes")

Example: CCA 00_BR_CCA1_"D:\Program Files\Indramat\Mtgui\Temp\Upload.ini"/3

FI command		**00_BR_CCA1_"D:\Program Files\Indramat\Mtgui\ Temp\upload.ini"/3**
Line	**Column**	**Answer**
1	1	01
2	1	00_BR_CCA1_"D:\Program Files\Indramat\Mtgui\ Temp\upload.ini" /3
3	1	0

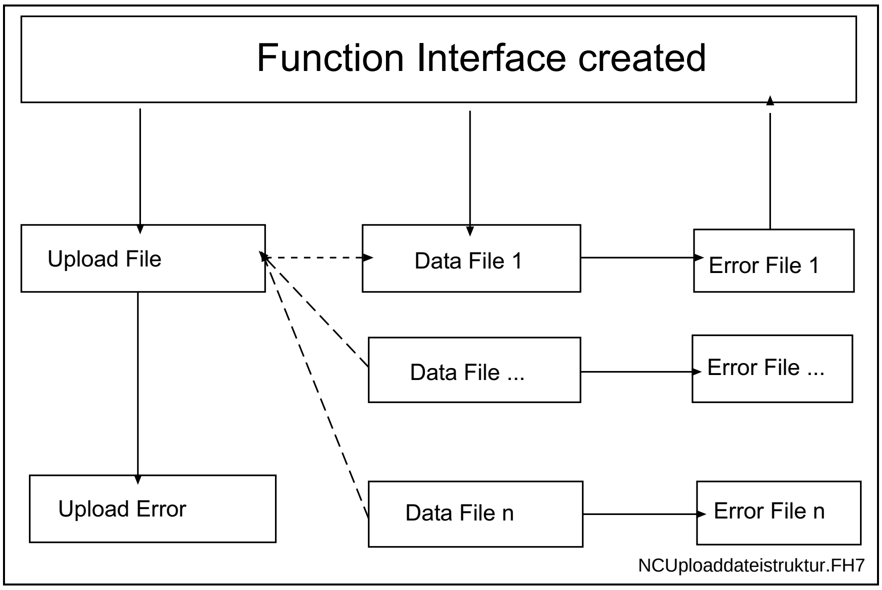


Fig. 5-3: File structure of the upload file

Structure of Upload File

The upload file is structured in the Windows – "Ini" format structure.

Note: Care must be taken in the use of upper and lower case letters.

Section [Common]
General information is stored in the "Common" section.

Key **UploadError**
Indicates whether or not an error has occurred during uploading. This value is only set in the event of an error.

Example:

[Common]
UploadError = YES ; error

Section NC cycles package information [CycPackage_Info]

Key **Cycle package information**
The package identification is compiled from several keys. The total length of all package identifications must not exceed **a maximum of 84** characters. The length of the individual identifications is described below:

Package number"**PackageNo**" max. 2 characters
Package name "**PackageName**" max. 32 characters
Package size: "**PackageSize**" max. 8 characters left-justified
Package time: "**PackageTime**" max. 8 characters
Package date: "**PackageDate**" max. 8 characters
Package default:"**PackageDefault**" max. 26 characters (optional)

Total: max. 84 characters

Information on date and time is given in the format
Date : dd.mm.yy
Time: hh:mm:ss

Example:

[CycPackage_Info]
PackageNo = 1
PackageName = NC program package
PackageSize = 1234
PackageTime = 13:10:10
PackageDate = 24.12.00

Section list of NC cycle programs [ListOfCycPrograms]
The list of the NC cycle programs to be transferred is stored in the section "ListOfCycPrograms".

Key **Max_Index_Data**
Corresponds to the number of NC cycle programs to be transferred.

Key **consecutive index of the NC cycle programs**
Four-digit number starting with 1, identifies with a value the full file name of the NC cycle programs:

zzzzzzz	Data type (CYC-PRG)
xx	Process number
yyy	Program number of the cycle programs (1 – 255)

The file extension can be freely selected. ".dat" has been used in the following example.

Examples:

CYC-PRG-00-086	Cycle program for process 0 program 86
CYC-PRG-01-001	Cycle program for process 1 program 1

Example:

```
[ListOfCycPrograms]
Max_Index_Data=50
0001=K:\Program Files\Indramat\Mtgui\Project_000\CYC-PRG-00-001.dat
0002=K:\Program Files\Indramat\Mtgui\Project_000\CYC-PRG-01-001.dat
...
0050=K:\Program Files\Indramat\Mtgui\Project_000\CYC-PRG-06-099.dat
```

Data File Structure

Contains the actual data for the upload. Their structure corresponds to the Windows "Ini" structure.

Note: Care must be taken in the use of upper and lower case letters.

Data for the cycle program is stored in the respective files as a section. It is composed of general data and the actual program.

Section [Common]

Program version:	**Version**	
Process:	**Process**	[0..6]
Program number:	**No**	[0..255]
Program name:	**Name**	max. 32 characters
Program size:	**Size**	
Program time:	**Time**	max. 8 characters
Program date:	**Date**	max. 8 characters
Program short identification:	**ShortID**	max. 8 characters
Program status:	**Status, (optional)**	

Information on date and time is given in the format

Date : dd.mm.yy

Time: hh:mm:ss

Status flag	Description
C	Compiled
E	Error
N	Not compiled
No details	No compiler call

Fig. 5-4: Status flags

Section [Data]

Key **Max_Index_Data**
Corresponds to the number of NC blocks to be transmitted

Key **consecutive index of NC records**
Five-digit number starting with 1.

Example:

```
[Data]
Max_Index_Data=25
00001=N0000 G0 X0 Y0 Z0
...
00025=N0024 .Start
```

5.39 Create MI Import Data: CMD

MWCX device group

Designation **CMD** **C**reate **M**I Import **D**ata

Explanation Creation of the data the Message Integrator requires for data import.

FI command Creation of the data of all ProVi messages.

BR_CMD1 **(Single Read)**

Response Structure The command does not return any answer. If no error is signaled, the respective files have been generated.

FI command Creation of the data of a certain ProVi message type.

BR_CMD2_(1){_(2)} **(Single Read)**

(1) = Message type	[1 = error, 2 = messages, 10 = warnings, 11 = start requirements, 12 = setup diagnosis]
(2) = Module number	[1...99] ! only for message type 1 -2!

Response Structure The command does not return any answer. If no error is signaled, the respective files have been generated.

FI command Creation of the data of all step chain messages.

BR_CMD3 **(Single Read)**

Response Structure The command does not return any answer. If no error is signaled, the respective files have been generated.

FI command Creation of the data of a certain step chain register.

BR_CMD4!(1) **(Single Read)**

(1) = Register name [ASCII characters]

Note: The separator "!" is used in this command.

Response Structure The command does not return any answer. If no error is signaled, the respective files have been generated.

5.40 Current NC Data Information: CNI

MWCX device group

Designation **CNI** **C**urrent **N**C **I**nformation

Explanation The following NC data information is returned for the selected process:

1 = Current NC memory
2 = NC package number, or cycle package number
3 = NC package identification string, or cycle package identification string
4 = Current NC program number
5 = Current NC program name
6 = Current NC block number
7 = Current NC block in original format
8 = Current NC block in display format
9 = Current NC operation mode
10 = Next NC program number
11 = Next NC block number
12 = Next NC block in original format
13 = Next NC block in display format

FI command The current NC data information is to be returned for the selected process.

Note: For the partial results

- Current operating mode
- Next NC program number
- Next NC block number
- Next NC block in original format
- Next NC record in display format

the "CNI1" command always returns the partial result [- -]. For new applications, it is recommended to use the "CNI2" command!

BR_CNI1_(1) **(Single Read)**
BR_CNI1_(1) **(Cyclic Read)**
BB_CNI1_(1) **(Break Cyclic Read)**

(1) = Process number [0..6]

Response Structure The following table shows the general structure of the response to the FI command "CNI1". The response consists of one line with 13 columns.

Line 1	Column 1	...	Column 13

Value Range/Meaning of Columns

1 =	NC memory	[A = NC memory A; B = NC memory B MDI = MDI mode]
2 =	NC package number	[01..99 = valid NC package number -1 = no NC package in the controller=
3 =	NC package identification string	[Result empty if there is NO NC package in the NC memory]
4 =	Current NC program number	[Result empty if MDI mode is active; otherwise, the current NC program number]
5 =	Current NC program name	[- Result empty, if MDI mode is active or if there is NO NC program in the controller - Cycle, if cycle program is active - Current NC program name]
6 =	Current NC block number	[N0000....N9999]], if there is NO valid NC program, [N0000] is returned
7 =	Current NC block in original format	[[--], if there is NO current NC block]
8 =	Current NC block in display format	[[--], if there is NO current NC block]
9 =	Current NC operation mode	[--]
10 =	Next NC program number	[--]
11 =	Next NC block number	[--]
12 =	Next NC block in original format	[--]
13 =	Next NC block in display format	[--]

Example CNI1 Read the NC data information of process 0 of device 03

FI command		**03_BR_CNI1_0**
Line	**Column**	**Answer**
1	1	A
	2	1
	3	01Free_NC-Programs& 4807 08.11.0213:44:22
	4	00
	5	Cycle
	6	N0040
	7	N0040 N0039:0000:0001 G00 X-100.0 REV .R_4.1
	8	N0039:0000 G00 X-100.0 REV .R_4.1
	9	--
	10	--
	11	--
	12	--
	13	--

FI command

The current NC data information is to be returned for the selected process. Here, the single NC data information can be requested individually (see controller info)

BR_CNI2_(1)_(2) **(Single Read)**

BC_CNI2_(1)_(2) **(Cyclic Read)**

BB_CNI2_(1)_(2) **(Break Cyclic Read)**

(1) = Process number	[0..6]
(2) =Controller info (1 Byte) bit-coded	[Meaning of the individual bits: **Bit 0 set:** Request MDI or correction block **Bit 1 set:** Request current NC block **Bit 2 set:** Request next NC block **Bit 3 set:** Request NC program name **Bit 4** to **Bit 7** have net been set yet The individual bits can be "or'd"].

Note: The controller info is available so that the controller will **NOT** be bothered with conversion of NC block unless this is necessary.

Response Structure

The following table shows the general structure of the response to the FI command "CNI2". The response consists of one line with 11 columns.

Line 1	Column 1	...	Column 11

Value Range/Meaning of Columns

1 =	NC memory	[A = NC memory A; B = NC memory B MDI = MDI mode]
2 =	NC package number, or cycle package number	[01..99 = valid NC package number or cycle package number -1 = no NC package in the controller]
3 =	NC package identification string	[result empty if there is NO NC package or cycle package in the controller]
4 =	Current NC program number	[Result empty if MDI mode is active; otherwise, the current NC program number]
5 =	Current NC program name	[- Result empty, if MDI mode is active or if there is NO NC program in the controller - Cycle, if cycle program is active - Current NC program name]
6 =	Current NC block number	[N0000....N9999]], if there is NO valid NC program, [N0000] is returned
7 =	Current NC block in original format	[[--], if there is NO current NC block]
8 =	Current NC block in display format	[[--], if there is NO current NC block]
9 =	Current NC operation mode	[0 = NO NC program in the controller 1 = NC program active 2 = Cycle program active 3 = Supplementary block active 4 = MDI active]
10 =	Next NC program number	[Result empty if MDI mode is active; otherwise, the next NC

		program number]
11 =	Next NC block number	[N0000....N9999], if there is NO valid NC program, [N0000] is returned
12 =	Next NC block in original format	[[--], if there is NO next NC block]
13 =	Next NC block in display format	[[--], if there is NO next NC block]

Note: If the system is in cycle program, the cycle package number (2nd partial result) and the cycle package identification string (3rd partial result) is returned; otherwise, the NC package number or the NC package identification string is provided.

Example CNI2

Read **ALL** potential NC data information of process 0 of device 03-

FI command		**03_BR_CNI2_0_0x0F**
Line	**Column**	**Answer**
1	1	A
	2	1
	3	01drilling cycles 01 4807 08.11.0213:44:22
	4	00
	5	Cycle
	6	N0040
	7	N0040 N0039:0000:0001 G00 X-100.0 REV .R_4.1
	8	N0039:0000 G00 X-100.0 REV .R_4.1
	9	3
	10	0
	11	N0041
	12	N0041 N0040:0000:0001 G01 Y100.0
	13	N0040:0000:0001 G01 Y100.0

5.41 Current Process Information: CPI

MWCX device group

Designation **CPI** **C**urrent **P**rocess **I**nformation

Explanation The following information is returned for the selected process:

1 = Device address
2 = Device name
3 = Process number
4 = Process name
5 = Type of message
6 = Message source
7 = Type of message (2)
8 = Message number
9 = Message text
10 = Additional text available Yes/No
11 = 2 byte additional information
12 = HTML file name
13 = NC notification number
14 = NC notification

FI command The current process information is to be returned for the selected process.

BR_CPI1_(1) **(Single Read)**
BC_CPI1_(1) **(Cyclic Read)**
BB_CPI1_(1) **(Break Cyclic Read)**
(1) = Process number [0..6]

Response Structure The following table shows the general structure of the response to the FI command "CPI1". The response consists of one line with 14 columns.

Line 1	Column 1	...	Column 14

Value Range/Meaning of Columns

1 =	Device address	[00...63]
2 =	Device name	[max. 32 ASCII characters]
3 =	Process number	[0...6]
4 =	Process name	[max. 28 ASCII characters]
5 =	Type of message	[F = fault/error, D = diagnosis]
6 =	Message source	[CNC, PLC]
7 =	Type of message	[S = Status, O = Operator, E = External, I = Internal]
8 =	Message number	[0...600]
9 =	Message text	[max. 54 ASCII characters]
10 =	Reference text	[x= exists, -- = does not exist]
11 =	2 byte additional information for the message number	is required to resolve the information "@" (see AMM5)
12 =	File name for additional information for notification text	e.g. in HTML format

13 = NC notification number [0000...9999]

14 = NC notification [max. 50 characters]

Note: If the current NC program does not contain a note, then the result of Column 13 is [0000] and that of Column 14 is [--].

Example CPI1

Read the process information of process 0 of device 03

FI command		**03_BR_CPI1_0**
Line	**Column**	**Answer**
1	1	03
	2	Drill center
	3	0
	4	Station 1
	5	D
	6	CNC
	7	S
	8	79
	9	Station waiting until tool-change command has ended.
	10	X
	11	0
	12	
	13	0002
	14	Technological instructions

5.42 Position Set Point of an Axis: CPO

MWCX device group

Designation **CPO** **C**ommand **PO**sition

Explanation

The actual position set point of a selected axis is read out. The "CPO1" FI command returns the command value of an axis related to the code of the axis meaning. The "CPO2" FI command, on the other hand, returns the command value of an axis related to the physical axis number.

FI command

Output the command value of the selected axis of the device specified, related to the code of the axis meaning.

Using the optional fourth parameter it is possible to pre-select conversion of the result into mm or inches. If, however, a spindle is selected as an axis, indicating a measurement system serves no purpose.

CR_CPO1_(1)_(2)_(3){_(4)}	**(Single Read)**
CC_CPO1_(1)_(2)_(3){_(4)}	**(Cyclic Read)**
CB_CPO1_(1)_(2)_(3){_(4)}	**(Break Cyclic Read)**
(1) = NC process number	[0...6]
(2) = Axis meaning	[0...11] (see Chapter "Data Tables")
(3) = System of coordinates	[1 = machine coordinates 2 = program coordinates]
(4) = Required measurement system	[mm, inch]

(opt.)

FI command

Output the command position of an axis of the device specified, related to the physical axis number.

Using the optional third parameter it is possible to pre-select conversion of the result into mm or inches. If, however, a spindle is selected as an axis, indicating a measurement system serves no purpose.

CR_CPO2_(1)_(2){_(3)}	**(Single Read)**
CC_CPO2_(1)_(2){_(3)}	**(Cyclic Read)**
CB_CPO2_(1)_(2){_(3)}	**(Break Cyclic Read)**
(1) = Physical axis number	[1...32, according to settings of the system parameters]
(2) = System of coordinates	[1 = machine coordinates 2 = program coordinates]
(3) = Required measurement system (opt.)	[mm, inch]

Response Structure

The following table shows the general structure of the response to the FI commands "CPO1" and "CPO2". One line is output with 4 columns for the axis designation, position, unit and the position limited to "indicated decimal places".

Line 1	Column 1	Column 2	Column 3	Column 4

Value Range/Meaning of Columns

1 = Axis name	[according to settings of axis parameters]
2 = Position	[according to settings of process parameters]
3 = Unit	[according to the settings of process parameters: mm, inch]
4 = Position	[as Column 2, but rounded up or down according to the parameter "indicated decimal places"]

Note: If the specified axis is not defined in the selected NC process then the response in all columns is [--].

Example CPO1

Read the current position of the Z axis in machine coordinates in NC process 0 of device address 00.

FI command	**00_CR_CPO1_0_2_1**			
Answer				
Line	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	Z1	-5.98975	[mm]	-5.990

Example CPO1

Read the current position of the Z axis in machine coordinates in NC process 0 of device address 00. The result is to be output in inches.

FI command	**00_CR_CPO1_0_2_1_inch**			
Answer				
Line	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	Z1	-0.23582	[inch]	-0.236

Example CPO2

Read the current position of the Z axis in machine coordinates in NC process 0 of device address 00 (e.g. physical axis number = 3) in machine coordinates.

FI command	**00_CR_CPO2_3_1**			
	Answer			
Line	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	Z1	-5.98975	[mm]	-5.990

Reference to Literature

See chapter entitled "Literature" [25].

5.43 Trigger Control Reset: CRT

MWCX device group

Designation

CRT **Control-Reset**

Explanation

The control reset allows the selected device to be reset during an active system error. If there is no system error at the selected device then the job is ignored.

CAUTION

Carrying out a reset completely re-initializes the device.
During initialization, communication is temporarily interrupted (inherent to design).

FI command

BW_CRT **(Single Write)**

Value to be written

Trigger reset 0

Note: The value to be written is passed to the "acValue" parameter in the "DataTransfer" routine.

Response Structure

The return value of the "DataTransfer" routine is [0] if the write procedure has been successfully completed. In the event of an error, more information in the form of a general error result line can be requested by the routine "ReadGroupItem" (refer here to chapter 8, "Error Codes" and "General Error Result Line").

Example CRT

Triggers a control reset on the selected device.

FI command		**Value to be written: 0** **00_BW_CRT**
Lines	**Column**	**Answer**
1	1	(P_ACK)

Reference to Literature

See chapter entitled "Literature" [26].

5.44 Checking of the Virtual Axis: CVA

MWCX device group

Designation **CVA** **C**heck **V**irtual **A**xis

Explanation This command is used to check whether the requested drive address is a virtual axis.

FI command **BR_CVA1_(1)** **(Single Read)**

(1) = Requested drive address [1..32] with MTC systems
[1..16] with MTA systems

Response Structure The response to the "BR_CVA1" FI command consists of one line with one column.

Line 1	Column 1

Value Range/Meaning of Columns

1 =	Information on whether the selected drive is a virtual axis	[0 = Virtual axis 1 = Real axis

Example CVA1 Check whether drive 1 at device 0 is a real or a virtual axis. The axis in question is a real axis.

FI command		**00_BR_CVA1_1**
Line	**Column**	**Answer**
1	1	1

5.45 Device Axis Configuration Parameter: DAC

MWCX device group

Designation **DAC** **D**evice **A**xis **C**onfiguration Parameter

Explanation The configuration of the device axes that are configured in the active machine parameter record is read out. The following belong to the configuration data of the device axes: axis number, corresponding process, assigned processes, type of axis, APR number, APR axis number, main axis meaning, secondary axis meaning, main axis name, secondary axis name and corresponding axis number.

FI command Output the current parameters of all configured device axes.

BR_DAC1 **(Single Read)**

Response Structure The following table shows the general structure of the response to the "DAC1" FI command. The number of answer lines [1...32 per NC process] depends on the number of configured device axes. Each line consists of 11 columns.

Line 1...n:	Column 1	...	Column 11

Note: If there is no active machine parameter record in the device then the columns [1...11] are not applicable.

Value Range/Meaning of Columns

1 =	Physical axis number	[1...32]
2 =	NC process number	[0...6]
3 =	Assigned processes	[0...6,--]
4 =	Type of axis	[see Chapter "Data Tables"]
5 =	APR number	[1...5]
6 =	APR axis number	[1...8]
7 =	Main axis meaning	[see the chapter entitled "Data Tables"]
8 =	Secondary axis meaning	[see the chapter entitled "Data Tables"]
9 =	Main axis name	[Xi, Yi, Zi, Ui, Vi, Wi, Ai, Bi ,Ci, Si, --] (i=[ ], [1...3])
10 =	Secondary axis name	[Xi, Yi, Zi, Ui, Vi, Wi, Ai, Bi ,Ci, Si, --] (i=[ ], [1...3])
11 =	Assigned axis number	[1...32, --]

Reference to Literature

See chapter entitled "Literature" [27].

Example DAC1

Reads the current parameters of all configured device axes of the active machine parameter record of device address 00.

Assumption:
The following three device axes have been defined:

- Digital linear axis (axis number 1)
- Digital linear axis (axis number 2)
- Main circular axis spindle (axis number 3).

FI command		**00_BR_DAC1**
Line	**Column**	**Answer**
1	1	1
	2	0
	3	--
	4	81
	5	1
	6	1
	7	0
	8	--
	9	X
	10	--
	11	--

2	1	2
	2	0
	3	--
	4	81
	5	1
	6	2
	7	1
	8	--
	9	Y
	10	--
	11	--
3	1	3
	2	0
	3	--
	4	85
	5	1
	6	4
	7	8
	8	--
	9	S:
	10	--
	11	--

FI command

Output the current parameters of the selected device axis type.

BR_DAC2_(1)	**(Single Read)**
(1) = axis type	[1 = only digital axes, 2 = only analog axes]

Response Structure

The following table shows the general structure of the response to the "DAC2" FI command. The number of answer lines [1...32] depends on the number of configured device axes. Each line consists of 11 columns.

Line 1...n	**Column 1**	**...**	**Column 11**

Note: If there is no active machine parameter record in the device then the columns [1...11] are not applicable.

Value Range/Meaning of Columns

1 =	Axis number	[1...32]
2 =	NC process number	[0...6]
3 =	Assigned processes	[0...6,--]
4 =	Type of axis	[see chapter entitled "Data Tables", Axis Types]
5 =	APR number	[1...5]
6 =	APR axis number	[1...8]
7 =	Main axis meaning	[see chapter entitled "Data Tables", Axis Meanings]
8 =	Secondary axis meaning	[see chapter entitled "Data Tables", Axis Meanings]
9 =	Main axis name	[Xi, Yi, Zi, Ui, Vi, Wi, Ai, Bi ,Ci, Si, --]

(i=[ ], [1...3])

10 =	Secondary axis name	[Xi, Yi, Zi, Ui, Vi, Wi, Ai, Bi ,Ci, Si, --] (i=[ ], [1...3])
11 =	Assigned axis number	[1...32, --]

Example DAC2 Reads the current parameters of all configured digital device axes of the active machine parameter record of device address 00.

Assumption:
A digital, linear axis with axis number 1 has been defined.

FI command		**00_BR_DAC2**
Line	**Column**	**Answer**
1	1	1
	2	0
	3	--
	4	81
	5	1
	6	1
	7	0
	8	--
	9	X
	10	--
	11	--

5.46 NC D-Correction Download: DCA

MWCX device group

Designation **DCA** NC-**D**-**C**orrection **A**ccess

Explanation D-corrections are downloaded by means of the download file via all active processes.

FI command NC D-correction download.

BW_DCA1_(1) **(Single Write)**

(1) = Download file with path details.

Note: File and path details must be enclosed in inverted commas.

Response Structure The response to the "DCA1" FI command consists of three lines, each with one column. The meaning of the elements is as follows:

Line 1 = Job ID [01...20]
(see Chapter "FI Commands for the MPCX Device Group", IFJ).

Line 2 = FI command
[String, in accordance with Chapter "Elements of the FI Command"]

Line 3 = FI Job Error Code
(see Chapter "Error Codes")

Example DCA1

00_BW_DCA1_"D:\Program Files\Indramat\Mtgui\Temp\download.ini"/3

FI command		**00_BW_DCA1_"D:\Program Files\Indramat\Mtgui\Temp\download.ini"/3**
Line	**Column**	**Answer**
1	1	01
2	1	00_BW_DCA1_"D:\Program Files\Indramat\Mtgui\Temp\download.ini"/3
3	1	0

Structure of the download file

The structure of the "download.ini" file used in this example corresponds to an Ini file in Windows.

Note: Care must be taken in the use of upper and lower case letters.

Section [Common]

This is currently only used for error processing, i.e., if an error is detected during a process, then the *DownloadError* key is written with "YES" within this section.

Example:

```
[Common]
DownloadError = YES   ; error
```

Section [DCorrectionPackage_Info]

The package identification is compiled from several keys. The total length of all package identifications must not exceed **a maximum of 84** characters. The length of the individual identifications is described below:

Package number"**PackageNo**" max. 2 characters
Package name "**PackageName**" max. 32 characters
Package size: "**PackageSize**" max. 8 characters left-justified
Package time: "**PackageTime**" max. 8 characters
Package date: "**PackageDate**" max. 8 characters
Package default:"**PackageDefault**" max. 26 characters (optional)

Total: max. 84 characters

Information on date and time is given in the format
Date : dd.mm.yy
Time: hh:mm:ss

Example:

```
[DCorrectionPackage_Info]
PackageNo =          1
PackageName =        D correction
PackageSize =        1234
PackageTime =        13:10:10
PackageDate =        24.12.00
```

Section [DCorrection_A]

A: corresponds to a process number [0..6]

A section entry ([DCorrection_A]) is an optional entry, i.e., if a section for a process is absent it is not regarded as an error.

The key values correspond to the D-correction numbers [1..99] and the values are the write values of D-corrections (L1, L2, L3, R, unit optional). Missing key values are not regarded as errors.

```
[DCorrection_0]
001=L1 1.0 L2 2.0 L3 3.0 R 4.0
...
099=L1 10.0 L2 20.0 L3 30.0 R 40.0
```

```
[DCorrection_1]
001=L1 1.0 L2 2.0 L3 3.0 R 4.0 mm
...
050=L1 10.0 L2 20.0 L3 30.0 R 40.0 mm

[DCorrection_6]
001=L1 1.0 L2 2.0 L3 3.0 R 4.0
...
099=L1 10.0 L2 20.0 L3 30.0 R 40.0
```

5.47 NC D-Correction Upload: DCA

MWCX device group

Designation **DCA** NC-**D**-**C**orrection **A**ccess

Explanation D corrections are uploaded via all active processes.

FI command D corrections upload.

BR_DCA1_(1) **(Single Read)**

(1) = Upload file with path details

Note: Enclose file and path details in inverted commas.

In this command, the progress information is implemented in %. It can be interrogated via the command IFJ of the MPCX device group.

Response Structure The response to the DCA1 FI command consists of three lines, each with one column. The meaning of the elements is as follows:

- Line 1 = Job ID [01...20]
  (see Chapter "FI Commands for the MPCX Device Group", IFJ).
- Line 2 = FI command
  [String, in accordance with Chapter "Elements of the FI Command"]
- Line 3 = FI Job Error Code
  (see Chapter "Error Codes")

Example DCA1 00_BR_DCA1_"D:\Program Files\Indramat\Mtgui\Temp\Upload.ini"/3

FI command		**00_BR_DCA1_"D:\Program Files\Indramat\Mtgui\Temp\upload.ini"/3**
Line	**Column**	**Answer**
1	1	01
2	1	00_BR_DCA1_"D:\Program Files\Indramat\Mtgui\Temp\upload.ini"/3
3	1	0

Structure of Upload File The upload file is structured in the Windows – "Ini" format structure.

Section [Common]
General information is stored in the COMMON section.

Key **UploadError**
Indicates whether or not an error has occurred during uploading. This value is only set in the event of an error.

Example:

```
[Common]
UploadError = YES        ; error
```

Section NC variables information [DCorrectionPackage_Info]
Key **program package information**
The package identification is compiled from several keys. The total length of all package identifications must not exceed **a maximum of 84** characters. The length of the individual identifcations is described in the following:

Package number"**PackageNo**" max. 2 characters
Package name "**PackageName**" max. 32 characters
Package size: "**PackageSize**" max. 8 characters left-justified
Package time: "**PackageTime**" max. 8 characters
Package date: "**PackageDate**" max. 8 characters
Package default:"**PackageDefault**" max. 26 characters (optional)

Total: max. 84 characters

Information on date and time is given in the format
Date : dd.mm.yy
Time: hh:mm:ss

Example:

```
[DCorrectionPackage_Info]
PackageNo =             1
PackageName =           D correction
PackageSize =           1234
PackageTime =           13:10:10
PackageDate =           24.12.00
```

Section NC variables download [DCorrection_A]
A: corresponds to a process number [0..6]

The key values correspond to the D-correction numbers [1..99] and the values are L1, L2, L3, R, and the unit.

```
[DCorrection_0]
001= L1 1.0 L2 2.0 L3 3.0 R 4.0 mm
...
099= L1 1.0 L2 2.0 L3 3.0 R 4.0 mm

[DCorrection_1]
001= L1 1.0 L2 2.0 L3 3.0 R 4.0 mm
...
099= L1 1.0 L2 2.0 L3 3.0 R 4.0 mm

 [DCorrection_6]
001= L1 1.0 L2 2.0 L3 3.0 R 4.0 mm
...
099= L1 1.0 L2 2.0 L3 3.0 R 4.0 mm
```

5.48 Reading D-Correction Data: DCD

MWCX device group

Designation **DCD** **D**-**C**orrection **D**ata

Explanation The values of a D-correction register of the selected NC process are read out.

The D-corrections are additive to the tool geometry data of the register that effects the tool management, i.e. they are additive to the existing geometry registers L1, L2, L3 and R.

There are 99 D-correction numbers available for each of the seven NC processes. Each D-correction number therefore contains the registers L1, L2, L3 and R. Value assignment of the D-correction register is via the Bosch Rexroth GUI or via the function interface.

FI command Reading of a D-correction register value of an NC process of the selected device.

CR_DCD1_(1)_(2)_(3){_(4)}	**(Single Read)**
CC_DCD1_(1)_(2)_(3){_(4)}	**(Cyclic Read)**
CB_DCD1_(1)_(2)_(3){_(4)}	**(Break Cyclic Read)**
(1) = NC process number	[0...6]
(2) = D-correction number	[1...99]
(3) = Number of the D-correction register:	[1=L1, 2=L2, 3=L3, 4=R]
(4) = Required measurement system (opt.)	[mm, inch]

Using the optional fourth parameter it is possible to pre-select conversion of the result into mm or inches.

Response Structure The response consists of one line with three columns for the identifier (length correction L1 to L3 and radius correction R), the value of the requested D-correction register, and the unit in accordance with the settings of the process parameters.

Line	Column 1	Column 2	Column 3

Value Range/Meaning of Columns

1 =	Identifier	[L1, L2, L3, R]
2 =	Value of D-correction	[formatting according to setting of the process parameters]
3 =	Unit	[mm, inch; according to settings of the process parameters]

Note: If the requested D-correction number or the D-correction register is not assigned a value then the value 0 is output as response, formatted according to the settings in the process parameters.

Example DCD1 Read the value of the D-correction register 4 at device address 00 in NC process 0 of the D-correction number 1 (radius correction R).

FI command	**00_CR_DCD1_0_1_4**		
Answer			
Line	**Column 1**	**Column 2**	**Column 3**
1	R	0.0860	[mm]

Reference to Literature See chapter entitled "Literature" [6].

5.49 Read Device Component Information: DCI

MWCX device group

Designation **DCI** **D**evice **C**omponent **I**nformation

Explanation The current device component information is read out of the device. From the device component information, the user is provided with information on the components the addressed device is equipped with, and the firmware each component contains. The command will not file if no access to firmware is possible (e.g. while the device is in monitor mode). Instead, the failed access is reported through the firmware access status.

FI command Read the device component information.

BR_DCI1 **(Single Read)**

Response Structure The following table shows the general structure of the response to the FI command "BR_DCI1". For each device component available in the device, one line is returned. Each line consists of 11 columns.

FI command		**00_BR_DCI1**
Line	**Column**	**Answer**
1	1	PCB type
	2	Configured component type
	3	Detected component type
	4	Firmware access status; i.e. has an error occurred accessing the firmware, Yes/No? Valid range of values [YES/NO] In case of an error, the error cause can be defined from one the two following columns.
	5	Error class on accessing firmware identification: (see Error Class Definition under General Error Result Line)
	6	Error code on accessing firmware identification: (see Error Code Definition under Error Codes)
	7	Firmware identification
	8	Firmware version
	9	Firmware release
	10	Is the component address in column 11 a sub-address, Yes/No? Valid range of values [YES/NO]
	11	Component address
2	1	PCB type
	..	..
	11	Component address

Example DCI1 At device address 00, read out the current device component information.

FI command		**00_BR_DCI1**
Line	**Column**	**Answer**
1	1	CPU
	2	MTC-P
	3	MTC-P
	4	NO
	5	0
	6	0
	7	CPU06/0006-23V10
	8	23
	9	10
	10	NO
	11	1
2	1	PLC
	2	MTS-P01.2
	3	MTS-P01.2
	4	NO
	5	0
	6	0
	7	PLC06S-M05-06V05
	8	06
	9	05
	10	NO
	11	3
3	1	APR
	2	--
	3	APR-P
	4	NO
	5	0
	6	0
	7	APR06/0003-23T06
	8	23
	9	06
	10	NO
	11	4

Example DCI1 while booting is blocked

While booting is blocked (i.e. while the device is in monitor mode), read out the current device component information at device address 00.

FI command		00_BR_DCI1
Line	**Column**	**Answer**
1	1	CPU
	2	MTC-P
	3	MTC-P
	4	YES
	5	1
	6	2082
	7	--
	8	--
	9	--
	10	NO
	11	1
2	1	PLC
	2	MTS-P01.2
	3	MTS-P01.2
	4	YES
	5	1
	6	2082
	7	--
	8	--
	9	--
	10	NO
	11	3
3	1	APR
	2	--
	3	APR-P
	4	YES
	5	1
	6	2082
	7	--
	8	--
	9	--
	10	NO
	11	4

5.50 Device Configuration Parameter: DCP

MWCX device group

Designation **DCP** **D**evice **C**onfiguration **P**arameter

Explanation The device configuration parameters that are entered in the active machine parameter record as well as in the "IND_DEV.INI" file are output. The configuration parameters of the device include the device address, the device name, device type, mechanism number, mechanism name, and the process types.

FI command Output the configuration parameters of all defined devices.

BR_DCP1 **(Single Read)**

Note: The "DCP1" FI command refers to all defined devices. Therefore, any valid device address can be indicated in the command line (see example DCP1).

Response Structure The following table shows the general structure of the response to the "DCP1" FI command. The response consists of a maximum of n=512 lines (n=16 devices x 32 mechanisms = 512), each with 7 columns.

Line 1...n:	Column 1	...	Column 7

Note: If no active machine parameter record exists in the device, then the columns [1...7] for the respective device are not applicable.

Value Range/Meaning of Columns

1 =	device address	[00...63]
2 =	Device name	[max. 32 ASCII characters]
3 =	Device Type	[MTC200-P-G2, MTC200-R-G2, MTVNC]
4 =	Mechanism number	[0...31]
5 =	Mechanism name	[max. 28 ASCII characters]
6 =	Process type	[1= internal, 2 = external process]
7 =	Process type	[1 = NC Process, 2 = PLC Process]

Example DCP1 Read the device configuration parameters of all defined devices.

Assumption:
Three devices have been defined

- Device address 00 (MTC200-P-G2)
- Device address 01 (MTC200-R-G2) and
- Device address 02 (MTC200-P-G2)

FI command		**00_BR_DCP1**
Line	**Column**	**Answer**
1	1	00
	2	Rotary transfer machine
	3	MTC200-P-G2
	4	1
	5	Master
	6	1
	7	2
2	1	01
	2	0
	3	MTC200-R-G2
	4	0
	5	Milling machine 01
	6	1
	7	1
3	1	02
	2	0
	3	MTC200-P-G2
	4	1
	5	Milling machine 02
	6	1
	7	1

FI command

Output the configuration parameters of the selected device.

BR_DCP2 **(Single Read)**

Response Structure

The following table shows the general structure of the response to the "DCP2" FI command. The response consists of a line with 7 columns.

Line 1	**Column 1**	**...**	**Column 7**

Note: If no active machine parameter record exists in the device, then the columns [1...7] for the respective device are not applicable.

Value Range/Meaning of Columns

1 =	device address	[00...63]
2 =	Device name	[max. 32 ASCII characters]
3 =	Device Type	[MTC200-P-G2, MTC200-R-G2, MTVNC, MTRA-P, MTRA-R]
4 =	Mechanism number	[0...31]
5 =	Mechanism name	[max. 28 ASCII characters]
6 =	Process type	[1= internal, 2 = external process]
7 =	Process type	[1 = NC Process, 2 = PLC Process]

Example DCP2

Read the device configuration parameter of the selected device (device address 01).

Assumption:
Three devices have been defined

- Device address 00 (MTC200-G2)
- Device address 01 (MTC200-G2)
- Device address 02 (MTC200-G2)

FI command		**01_BR_DCP2**
Line	**Column**	**Answer**
1	1	01
	2	0
	3	MTC200-G2
	4	0
	5	Milling machine 01
	6	1
	7	1

Reference to Literature

See chapter entitled "Literature" [28].

5.51 D Correction Register: DCR

MWCX device group

Designation **DCR** **D**-**C**orrection **R**ecord

Explanation

The values of a D-correction record of the selected NC process are read out or written.

The D-corrections are additive to the tool geometry data of the register that effects the tool management, i.e. they are additive to the existing geometry registers L1, L2, L3 and R.

There are 99 D-correction numbers available for each of the seven NC processes. Each D-correction number therefore contains the registers L1, L2, L3 and R. Value assignment of the D-correction register is via the Bosch Rexroth GUI or via the function interface.

FI command

Reading of a D-correction record of an NC process of the selected device.

CR_DCR1_(1)_(2){_(3)}	**(Single Read)**
CC_DCR1_(1)_(2){_(3)}	**(Cyclic Read)**
CB_DCR1_(1)_(2){_(3)}	**(Break Cyclic Read)**
(1) = NC process number	[0...6]
(2) = D-correction number	[1...99]
(3) = Required measurement system (opt.)	[mm, inch]

Using the optional third parameter it is possible to pre-select conversion of the result into mm or inches.

Note: If the value of a single D-correction register is to be read then the command "CR_DCD1" should be used.

Response Structure

The response consists of four lines, each with three columns for the identifier (length correction L1 to L3 and radius correction R), the value of the requested D-correction register, and the unit in accordance with the settings of the process parameters.

Line 1..0.4	Column 1	...	Column 3

Value Range/Meaning of Columns

1 =	Identifier	[L1, L2, L3, R]
2 =	Value of D-correction	[Formatting of values according to the settings of process parameters]
3 =	Unit	[mm, inch; according to parameter settings]

Note: If the requested D-correction number or the D-correction register is not assigned a value then the value 0 is output as response, formatted according to the settings in the process parameters.

Example DCR1 without optional Parameters

Read the value of all D-correction registers at device address 00 of NC process 0 of D-correction number 1.

FI command	**00_CR_DCR1_0_1**		
Answer			
Line	**Column 1**	**Column 2**	**Column 3**
1	L1	1.2586	[mm]
2	L2	3.5892	[mm]
3	L3	0.0000	[mm]
4	R	0.0860	[mm]

FI command

Write all D-correction register values of an NC process of the selected device.

CW_DCR_(1)_(2)	**(Single Write)**
(1) = NC process number	[0...6]
(2) = D-correction number	[1...99]

Value to be written

D correction register	[L1<value> L2<value> L3<value> R<value>] opt. unit]

If there is no optional information for the unit {mm, inch}, then the values refer to the base programming unit of the process. If the unit entered differs from the basic programming unit then the values entered are converted into the values of the base programming unit.

Note: In the conversion from mm → inch, rounding errors are unavoidable, as precision is lost!

The single values are separated by a space, whereby the formatting should be carried out according to the settings of the process parameters. (see example DCR1: write D correction register).

Response Structure

One line is output with a column for acknowledgement of whether or not the FI command has been executed successfully.

(P_ACK) = **P**ositive **ACK**nowledge Value has been written

Example DCR
Write D-Correction Register

Write all D-correction registers at device address 00 of NC process 0 of D-correction number 1 with the following 5 values:

1. Value L1: 1.2586
2. Value L2: 3.5892
3. Value L3: 0.0000 and
4. Value R: 0.0860
5. Unit of the values: mm (optional)

Note: The values to be written are passed in the "Data Transfer" routine to the "acValue" parameter and must be separated from each other by a space " ".

FI command		**Value to write:** **L1 1.2586 L2 3.5892 L3 0 R 0.086 mm** **00_CW_DCR_0_1**
Line	**Column**	**Answer**
1	1	(P_ACK)

Reference to Literature

See chapter entitled "Literature" [6].

5.52 Setting of the Communication Timeout: DCT

MWCX device group

Designation **DCT** **D**evice **C**ommunication **T**imeout

Explanation By means of this command, the timeout time for the selected device is set dynamically (timeout time in ms).

FI command **BW_DCT1_(1)** **(Single Write)**

(1) = requested timeout time in ms

Response Structure The response to the "DCT1" FI command consists of one line with one column.

Line 1	Column 1

Value Range/Meaning of Columns 1 = Status message (P_ACK) (P_ACK)

Example DCT1 For the device 00, the timeout time is set 1500 ms.

FI command		**00_BW_DCT1_1500**
Line	**Column**	**Answer**
1	1	(P_ACK)

FI command With this command, the timeout time for the selected device can be reset to default value.

BW_DCT2 **(Single Write)**

Response Structure The response to the "DCT2" FI command consists of one line with one column.

Line 1	Column 1

Value Range/Meaning of Columns 1 = Status message (P_ACK) (P_ACK)

Example DCP2 For the device 00, the timeout time is reset to the default value.

FI command		**00_BW_DCT2**
Line	**Column**	**Answer**
1	1	(P_ACK)

5.53 Deleting the FI Exclusive Mode: DEM

MWCX device group

Designation **DEM** **D**elete FI **E**xclusive **M**ode

Explanation This command is used to deactivate FI Exclusive mode for the selected device address.

FI Exclusive mode: In this mode, **ALL** the processes logged in at the FI – **with the exception** of the process issuing the SEM command – are blocked from data communication with this device address. This mode is used for example for data communication which must NOT be interrupted by other data requests. However, it is **imperative** that this FI Exclusive mode is deleted once more through the DEM command.

FI command **BW_DEM1** **(Single Write)**

Response Structure

The following table shows the general structure of the response to the FI command "BW_DEM1". A line of 1 column is output.

Line 1	Column 1

Value Range/Meaning of Columns

1 = Status message (P_ACK) (P_ACK)

Example DEM1

Deactivate the FI Exclusive mode for device address 0. The FI Exclusive mode for device address 0 has previously been activated by the SEM command.

FI command		00_BW_DEM1
Line	Column	Answer
1	1	(P_ACK)

5.54 Static/Dynamic Device Information: DIF

MWCX device group

Designation

DIF **D**evice **I**n**F**ormation

Explanation

Static device information and network information is read according to the "IND_DEV.INI" and „FAR_DEV.INI" files.

FI command

Reading of the static device information and network information of a selected device.

BR_DIF1 **(Single Read)**

BC_DIF1 **(Cyclic Read)**

BB_DIF1 **(Break Cyclic Read)**

Response Structure

The following table shows the general structure of the response to the "DIF1" FI command. The response consists of one line with 24 columns.

Line 1	Column 1	...	Column 24

Value Range/Meaning of Columns

1 =	Local/far device address	[00..63]
2 =	Device name	IND_DEV.INI entry: DeviceName=
3 =	Device type	IND_DEV.INI entry: DeviceType=
4 =	PLC support	IND_DEV.INI entry: PLC=
5 =	Device status	IND_DEV.INI entry: DeviceStatus=
6 =	Assignment of a simulation pair	IND_DEV.INI entry: DeviceAssign=
7 =	Device mode	IND_DEV.INI entry: MtvncMode=
8 =	Communication channel	IND_DEV.INI entry: [CommAddrX]
9 =	Description of the communication channel	IND_DEV.INI entry: CommStr=
10 =	Timeout value	IND_DEV.INI entry: Timeout=
11 =	Device group	(see Chapter 6.1 "Identifier")
12 =	PLC component type	IND_DEV.INI entry: Component type1=
13 =	CNC component type	IND_DEV.INI entry: Component type2=
14 =	Device log	IND_DEV.INI entry: DeviceProtocol=
15 =	Device simulation	IND_DEV.INI entry: DeviceSimulation=
16 =	Not yet assigned	[--]

17 =	Not yet assigned	[--]
18 =	Not yet assigned	[--]
19 =	Not yet assigned	[--]
20 =	Network ON/OFF	[ON = Network active OFF = No network active]
21 =	Network name	Max. 28 ASCII characters
22 =	PC number	[00..99,XX]
23 =	PC name	Max. 255 ASCII characters
24 =	Local device address	[00..63]

Example DIF1 Read the static device information and network information of device 1 while the network is active.

FI command		**01_BR_DIF1**
Line	**Column**	**Answer**
1	1	01
	2	Drilling station right
	3	MTC200-P-G2
	4	NO
	5	ON
	6	NO
	7	RUN
	8	4
	9	SHM,1,TCON
	10	3500
	11	MTCX
	12	NONE
	13	NONE
	14	CNC
	15	OFF
	16	--
	17	--
	18	--
	19	--
	20	ON
	21	PC network 1
	22	29
	23	BTV20-STATION-LEFT
	24	01

Explanation The dynamic device information and network information is read. The current data identifications are made available from the selected controller.

FI command

Reading of the dynamic device information and network information of a selected device.

BR_DIF2	**(Single Read)**
BC_DIF2	**(Cyclic Read)**
BB_DIF2	**(Break Cyclic Read)**

Response Structure

The answer consists of 23 lines, each line having a specific meaning.

Line 1	Static device information
Line 2	Firmware information
Line 3	Current parameter set
Line 4	Current PLC program
Line 5	Current machine data set
Line 6	Current NC package for memory A
Line 7	Current NC package for memory B
Line 8	Current cycle package
Line 9	Current NC program name for process 0
Line 10	Current NC program name for process 1
Line 11	Current NC program name for process 2
Line 12	Current NC program name for process 3
Line 13	Current NC program name for process 4
Line 14	Current NC program name for process 5
Line 15	Current NC program name for process 6
Line 16	Current tool list for process 0
Line 17	Current tool list for process 1
Line 18	Current tool list for process 2
Line 19	Current tool list for process 3
Line 20	Current tool list for process 4
Line 21	Current tool list for process 5
Line 22	Current tool list for process 6
Line 23	Current I/O configuration table

Meaning of line 1

Line 1 returns the most significant static device information and network information and consists of 18 columns.

Line 1	Column 1	...	Column 18

Value Range/Meaning of Columns

1 =	Line number	[1]
2 =	Status information	Contains the information whether or not the subsequent data in this line is valid; the following applies: [0 = Data is invalid – further column results [--] 1 = Data is valid]
3 =	Local/far device address	[00..63]
4 =	Device name	According to device configuration
5 =	Device type	According to device configuration
6 =	PLC Components	According to device configuration
7 =	CNC components	According to device configuration
8 =	Device group	(see Chapter "Identifier")
9 =	Device status	According to device configuration ON = DeviceStatus ON

		OFF = DeviceStatus OFF
10 =	Current device status	ON = Device ONLINE OFF = Device OFFLINE
11 =	Not yet assigned	[--]
12 =	Not yet assigned	[--]
13 =	Not yet assigned	[--]
14 =	Network ON/OFF	[ON = Network active OFF = No network active]
15 =	Network name	Max. 28 ASCII characters
16 =	PC number	[00..99,XX]
17 =	PC name	Max. 255 ASCII characters
18 =	Local device address	[00..63]

Meaning of line 2

Returns the firmware versions of the existing controller components. Each line consists of 8 columns.

Line 2	Column 1	...	Column 8

Value Range/Meaning of Columns

1 =	Line number	[2]
2 =	Status information	Contains the information whether or not the subsequent data in this line is valid; the following applies: [0 = Data is invalid – further column results [--] 1 = Data is valid]
3 =	Firmware version of the CNC component	Designation according to convention
4 =	Firmware version of the PLC component	Designation according to convention
5 =	Firmware version of the 1.APR component	Designation according to convention
6 =	Firmware version of the 2.APR component	Designation according to convention
7 =	Firmware version of the 3.APR component	Designation according to convention
8 =	Firmware version of the 4.APR component	Designation according to convention

Meaning of line 3

Returns the identification of the current parameter set and consists of 6 columns.

Line 3	Column 1	...	Column 6

Value Range/Meaning of Columns

1 =	Line number	[3]
2 =	Status information	Contains the information whether or not the subsequent data in this line is valid; the following applies: [0 = Data is invalid – further column results [--] 1 = Data is valid]
3 =	Index of the parameter set	[01..99]
4 =	Designation of the parameter set	Max. 32 ASCII characters
5 =	Date string	Date of generation/modification
6 =	Time string	Time of generation/modification

Meaning of line 4 Returns the identification of the current PLC program and consists of 6 columns.

Line 4	Column 1	...	Column 6

Value Range/Meaning of Columns

1 =	Line number	[4]
2 =	Status information	Contains the information whether or not the subsequent data in this line is valid; the following applies: [0 = Data is invalid – further column results [--] 1 = Data is valid]
3 =	Index of the PLC program	Always [00]
4 =	PLC resource name = PLC program name	Max. 32 ASCII characters
5 =	Date string	Date of generation/modification
6 =	Time string	Time of generation/modification

Meaning of line 5 Returns the identification of the current machine data set and consists of 6 columns.

Line 5	Column 1	...	Column 6

Value Range/Meaning of Columns

1 =	Line number	[5]
2 =	Status information	Contains the information whether or not the subsequent data in this line is valid; the following applies: [0 = Data is invalid – further column results [--] 1 = Data is valid]
3 =	Index of the machine data set	[01..99]
4 =	Designation of the machine data set	Max. 32 ASCII characters
5 =	Date string	Date of generation/modification
6 =	Time string	Time of generation/modification

Meaning of line 6 Returns the identification of the current NC package in memory A and consists of 6 columns.

Line 6	Column 1	...	Column 6

Value Range/Meaning of Columns

1 =	Line number	[6]
2 =	Status information	Contains the information whether or not the subsequent data in this line is valid; the following applies: [0 = Data is invalid – further column results [--] 1 = Data is valid]
3 =	Index of the NC package in memory A	[01..99]
4 =	Designation of the NC package in memory A	Max. 32 ASCII characters
5 =	Date string	Date of generation/modification
6 =	Time string	Time of generation/modification

Meaning of line 7

Returns the identification of the current NC package in memory B and consists of 6 columns.

Line 7	Column 1	...	Column 6

Value Range/Meaning of Columns

1 =	Line number	[7]
2 =	Status information	Contains the information whether or not the subsequent data in this line is valid; the following applies: [0 = Data is invalid – further column results [--] 1 = Data is valid]
3 =	Index of the NC package in memory B	[01..99]
4 =	Designation of the NC package in memory B	Max. 32 ASCII characters
5 =	Date string	Date of generation/modification
6 =	Time string	Time of generation/modification

Meaning of line 8

Returns the identification of the current cycle package and consists of 6 columns.

Line 8	Column 1	...	Column 6

Value Range/Meaning of Columns

1 =	Line number	[8]
2 =	Status information	Contains the information whether or not the subsequent data in this line is valid; the following applies: [0 = Data is invalid – further column results [--] 1 = Data is valid]
3 =	Index of the cycle package	[01..99]
4 =	Designation of the cycle package	Max. 32 ASCII characters
5 =	Date string	Date of generation/modification
6 =	Time string	Time of generation/modification

Meaning of the lines 9 - 15

These lines return information on the current NC program for the processes 0..6 and consist of 8 columns each.

Line 9..15	Column 1	...	Column 8

Value Range/Meaning of Columns

1 =	Line number	[9..15]
2 =	Status information	Contains the information whether or not the subsequent data in this line is valid; the following applies: [0 = Data is invalid – further column results [--] 1 = Data is valid]
3 =	Process number	[00..06]
4 =	Process name	Max. 40 ASCII characters
5 =	Current NC memory	[A,B]
6 =	Current NC program number	[01..99]
7 =	Current NC program name	Max. 32 ASCII characters
8 =	Current NC block	

Meaning of the lines 16 -22

These lines return information on the current tool lists for the processes 0..6 and consist of 12 columns each.

Line 16.0.22	Column 1	...	Column 12

Value Range/Meaning of Columns

1 =	Line number	[16..22]
2 =	Status information	Contains the information whether or not the subsequent data in this line is valid; the following applies: [0 = Data is invalid – further column results [--] 1 = Data is valid]
3 =	Process number	[00..06]
4 =	Process name	Max. 40 ASCII characters
5 =	Tool list index	Always [00]
6 =	Name of the tool list	Max. 32 ASCII characters
7 =	Date string	Date of generation/modification
8 =	Time string	Time of generation/modification
9 =	Tool magazine type	[MAGAZINE] [TURRET]
10 =	Number of spindles	[0..4]
11 =	Number of grippers	[0..4]
12 =	Number of magazine locations	[0..999]

Meaning of line 23

Returns the identification of the current I/O configuration list and consists of 6 columns.

Line 23	Column 1	...	Column 6

Value Range/Meaning of Columns

1 =	Line number	[23]
2 =	Status information	Contains the information whether or not the subsequent data in this line is valid; the following applies: [0 = Data is invalid – further column results [--] 1 = Data is valid]
3 =	Index of the I/O configuration list	[01..99]
4 =	Designation of the I/O configuration list	Max. 32 ASCII characters
5 =	Date string	Date of generation/modification
6 =	Time string	Time of generation/modification

Example DIF2

Reads the dynamic device information of device 1. It is either an MTC200-P-G2, or it is active in a network.

FI command		**01_BR_DIF2**
Line	**Column**	**Answer**
1	1	1
	2	1
	3	01
	4	Drilling station left
	5	MTC200-P-G2
	6	MTS-P
	7	MTC-P
	8	MWCX
	9	ON
	10	ON
	11	--
	12	--
	13	--
	14	ON
	15	PC network 1
	16	29
	17	BTV20-RIGHT
	18	01
2	1	2
	2	1
	3	CPU06/0006-23T06
	4	PLC06S-M05-06T03
	5	APR06/0003-23T03
	6	--
	7	--
	8	--
3	1	3
	2	1
	3	01
	4	Config_3_Process
	5	07.04.03
	6	09:45:42
4	1	4
	2	1
	3	00
	4	Prg_3_Process
	5	30.04.03
	6	09:03:45
5	1	5

	2	1
	3	01
	4	Machine_Data_3_Process
	5	07.04.03
	6	09:48:34
6	1	6
	2	1
	3	30
	4	Gearbox 2
	5	15.10.02
	6	13:47:34
7	1	7
	2	0
	3	--
	4	--
	5	--
	6	--
8	1	8
	2	1
	3	03
	4	Cycle package_3_Process
	5	15.10.02
	6	14:48:56
9	1	9
	2	1
	3	00
	4	Master process 0
	5	A
	6	01
	7	NcPrg_01
	8	N0000 G00 X10.0
10	1	10
	2	1
	3	01
	4	Process 1
	5	A
	6	--
	7	--
	8	--
11	1	11
	2	1
	3	02

	4	Process 2
	5	A
	6	--
	7	--
	8	--
12	1	12
	2	0
	3	--
	4	--
	5	--
	6	--
	7	--
	8	--
13	1	13
	2	0
	3	--
	4	--
	5	--
	6	--
	7	--
	8	--
14	1	14
	2	0
	3	--
	4	--
	5	--
	6	--
	7	--
	8	--
15	1	15
	2	0
	3	--
	4	--
	5	--
	6	--
	7	--
	8	--
16	1	16
	2	1
	3	00
	4	Master process 0
	5	00

	6	Tool list gearbox 2
	7	14.10.02
	8	15:34:23
	9	[MAGAZINE]
	10	1
	11	3
	12	45
17	1	17
	2	1
	3	01
	4	Process 1
	5	--
	6	--
	7	--
	8	--
	9	[MAGAZINE]
	10	1
	11	1
	12	60
18	1	18
	2	1
	3	02
	4	Process 2
	5	--
	6	--
	7	--
	8	--
	9	[MAGAZINE]
	10	1
	11	1
	12	60
19	1	19
	2	0
	3	--
	4	--
	5	--
	6	--
	7	--
	8	--
	9	--
	10	--
	11	--

	12	--
20	1	20
	2	0
	3	--
	4	--
	5	--
	6	--
	7	--
	8	--
	9	--
	10	--
	11	--
	12	--
21	1	21
	2	0
	3	--
	4	--
	5	--
	6	--
	7	--
	8	--
	9	--
	10	--
	11	--
	12	--
22	1	22
	2	0
	3	--
	4	--
	5	--
	6	--
	7	--
	8	--
	9	--
	10	--
	11	--
	12	--
23	1	23
	2	0
	3	--
	4	--
	5	--

	6	--

FI command

Returns information on which controller data is currently available in the selected device.

BR_DIF3 **(Single Read)**

BC_DIF3 **(Cyclic Read)**

Response Structure

The answer consists of 1 line with 14 columns, each column having a specific meaning.

1 = Active parameter set available Yes/No
2 = NC package memory A available Yes/No
3 = NC package memory B available Yes/No
4 = NC zero points memory A available Yes/No
5 = NC zero points memory B available Yes/No
6 = NC events available Yes/No
7 = NC variables available Yes/No
8 = NC D-corrections available Yes/No
9 = NC cycles available Yes/No
10 = Active machine data record available Yes/No
11 = PLC retain variables available Yes/No
12 = Tool lists available Yes/No
13 = Drive parameters available Yes/No
14 = I/O configuration list available Yes/No

Line 1	**Column 1**	**...**	**Column 14**

Value Range/Meaning of Columns

1 = Controller data available Yes/NO [YES,NO]

Example DIF3

Return information on which controller data is currently available in the selected device.

FI command		**01_BR_DIF3**
Line	**Column**	**Answer**
1	1	YES
	2	YES
	3	NO
	4	YES
	5	YES
	6	YES
	7	YES
	8	YES
	9	YES
	10	YES
	11	YES
	12	NO
	13	YES
	14	NO

5.55 Long Identification of NC/PLC Data Records and Data Cycles: DIS

MWCX device group

Designation **DIS** **D**ata **I**dentification **S**tring

Explanation Reads the long identification (directory entries) of NC/PLC data records. Included in the directory entries are the number of the entry in the directory, the name, length and date and time of creation and/or details of the last time the respective data record was changed. The long identifications of the following NC/PLC data records are output:

NC parameter record (FI command: DIS1)

PLC program (FI command: DIS2)

NC package (FI command: DIS3)

Tool list (FI command): DIS4)

Machine data (FI command): DIS5) and

NC program (FI command: DIS6)

FI command Output the directory entries of the valid NC parameter record in the selected device.

BR_DIS1 **(Single Read)**

BC_DIS1 **(Cyclic Read)**

BB_DIS1 **(Break Cyclic Read)**

Response Structure The following table shows the general structure of the response to the "DIS1" FI command. The response consists of a line with five columns.

Line 1	Column 1	...	Column 5

Value Range/Meaning of Columns

1 =	Number in NC parameter directory	[01...99]
2 =	Name of the NC parameter record	[max. 32 ASCII characters]
3 =	Length of the NC parameter record	[byte]
4 =	Date of creation/last change to NC parameter record	[DD.MM.YY]
5 =	Time of creation/last change to NC parameter record	[HH:MM:SS]

Note: If there is no valid NC parameter record in the selected device then all columns contain [--] . This command can also be used when the selected device is in OFFLINE mode (DeviceStatus=OFF).

Example DIS1 Read the directory entries of the NC parameter record at device address 00.
Assumption:

There is a valid NC parameter record in the selected device.

FI command		**00_BR_DIS1**
Line	**Column**	**Answer**
1	1	01
	2	KEY1
	3	3579
	4	16.05.99
	5	10:41:08

Reference to Literature See chapter entitled "Literature" [29].

FI command Output the directory entries of the valid PLC program in the selected device.

BR_DIS2 **(Single Read)**

BC_DIS2 **(Cyclic Read)**

BB_DIS2 **(Break Cyclic Read)**

Response Structure The following table shows the general structure of the response to the "DIS2" FI command. The response consists of a line with six columns.

Line 1	**Column 1**	**...**	**Column 6**

Value Range/Meaning of Columns

1 =	Number in PLC directory	[01...99]
2 =	Name of the PLC program	[max. 8 ASCII characters]
3 =	Length of the PLC program	[byte]
4 =	Date of creation/last change to PLC program	[DD.MM.YY]
5 =	Time of creation/last change to the PLC program	[HH:MM:SS]
6 =	Date of creation/last change to PLC program	[DD.MM.YYYY]

Note: If there is no valid PLC program in the selected device then all columns contain [--] .

Example DIS2 Read the directory entries of the PLC program at address 00.
Assumption:
There is a valid PLC program in the selected device.

FI command		**00_BR_DIS2**
Line	**Column**	**Answer**
1	1	01
	2	KEY1
	3	20018
	4	10.05.99
	5	12:42:00
	6	10.05.1999

Reference to Literature See chapter entitled "Literature" [30].

FI command

Output the directory entries of the valid NC package of the selected NC memory.

BR_DIS3_(1)	**(Single Read)**
BC_DIS3_(1)	**(Cyclic Read)**
BB_DIS3_(1)	**(Break Cyclic Read)**
(1) = NC memory	[1 = NC memory A; 2 = NC memory B]

Response Structure

The following table shows the general structure of the response to the "DIS3" FI command. The response consists of a line with five columns.

Line 1	Column 1	...	Column 5

Value Range/Meaning of Columns

1 =	Number in NC package directory	[01...99]
2 =	Name of the NC package	[max. 32 ASCII characters]
3 =	Length of the NC package	[byte]
4 =	Date of creation/last change to NC package	[DD.MM.YY]
5 =	Time of creation/last change to NC package	[HH:MM:SS]

Note: If there is no valid NC package in the selected NC memory then all columns contain [--] .

Example DIS3

Read the directory entries of the NC package in NC memory A at device address 00.
Assumption:
There is a valid NC package in memory A of the selected device.

FI command		**00_BR_DIS3_1**
Line	**Column**	**Answer**
1	1	01
	2	KEY1
	3	3579
	4	16.05.99
	5	10:41:08

Reference to Literature

See chapter entitled "Literature" [31].

FI command

Output the directory entries of the valid tool list of the selected NC process.

BR_DIS4_(1)	**(Single Read)**
BC_DIS4_(1)	**(Cyclic Read)**
BR_DIS4_(1)	**(Break Cyclic Read)**
(1) = NC process number	[0...6]

Response Structure

The following table shows the general structure of the response to the "DIS4" FI command. The response consists of a line with five columns.

Line 1	Column 1	...	Column 5

Value Range/Meaning of Columns

1 =	Number in the tool list index	[01...99]
2 =	Name of the tool list	[max. 32 ASCII characters]
3 =	Length of the tool list	[byte]
4 =	Date of creation/last change to the tool list	[DD.MM.YY]
5 =	Time of creation/last change to the tool list	[HH:MM:SS]

Note: If there is no valid tool list in the selected NC process then all columns contain [--] .

Example DIS4

Read the directory entries of the tool list of NC process 0 at device address 00.

Assumption:

There is a valid tool list in NC process 0 of the selected device.

FI command		**00_BR_DIS4_0**
Line	**Column**	**Answer**
1	1	01
	2	KEY1
	3	2048
	4	17.09.99
	5	10:45:08

Reference to Literature

See chapter entitled "Literature" [32].

FI command

Output the directory entries of the valid machine data record in the selected device.

BR_DIS5	**(Single Read)**
BC_DIS5	**(Cyclic Read)**
BB_DIS5	**(Break Cyclic Read)**

Response Structure

The following table shows the general structure of the response to the "DIS5" FI command. The response consists of a line with five columns.

Line 1	**Column 1**	**...**	**Column 5**

Value Range/Meaning of Columns

1 =	Number in machine data directory	[01...99]
2 =	Name of the data record	[max. 32 ASCII characters]
3 =	Length of data record	[byte]
4 =	Date of creation/last change to the data record	[DD.MM.YY]
5 =	Time of creation/last change to the data record	[HH:MM:SS]

Note: If there is no valid machine data in the selected device then all columns contain [--] .

Example DIS5

Read the directory entries of the machine data record in device address 00.

Assumption:

There is valid machine data in the selected device

FI command		**00_BR_DIS5**
Line	**Column**	**Answer**
1	1	01
	2	KEY1
	3	3180
	4	18.12.98
	5	21:20:02

Reference to Literature

See chapter entitled "Literature" [32].

FI command

Output the directory entries of the valid NC program.

BR_DIS6_(1)_(2)_(3)	**(Single Read)**
BC_DIS6_(1)_(2)_(3)	**(Cyclic Read)**
BB_DIS6_(1)_(2)_(3)	**(Break Cyclic Read)**
(1) = Cycle package or NC memory	[0 = Package string of the cycle package, 1 = NC memory A, 2 = NC memory B]
(2) = NC process number	[0...6]
(3) = NC program number or NC cycle package number	[1...99] [0]

Response Structure

The following table shows the general structure of the response to the "DIS6" FI command. The response consists of a line with six columns.

Line 1	**Column 1**	**...**	**Column 6**

Value Range/Meaning of Columns

1 =	Package number	[01...99]
2 =	Number of the NC program/cycle program	[01...99]
3 =	Name of the NC program/cycle program	[max. 32 ASCII characters]
4 =	Length of the NC program/cycle program	[byte]
5 =	Date of creation/last change to NC program/cycle program	[DD.MM.YY]
6 =	Time of creation/last change to NC program/cycle program	[HH:MM:SS]

Note: If there is no valid NC package in the selected NC process, then all columns contain [--] .

Example DIS6

Read the directory entries of the third NC program (NC package number 2, NC memory A, NC process 0) at device address 00.
Assumption:
There is valid data in the selected device.

FI command		**00_BR_DIS6_1_0_3**
Line	**Column**	**Answer**
1	1	03
	2	Audi A4
	3	3579
	4	16.05.99
	5	10:41:08

Reference to Literature See chapter entitled "Literature" [17].

5.56 Download/ Upload of Drive Parameters: DPA

MWCX device group

Designation **DPA** **D**rive **P**arameter **A**ccess

Explanation Drive parameter data records are downloaded by means of a download file. This download command is an FI job.

Structure of Download File The structure of the download file corresponds to that of a Windows Ini file.

Summary:

Section [COMMONDATA]

Contains general information on the generation of this file.

DeviceAddr=Device address for which the drive parameters have been collected.
DeviceName=Device name
DeviceType=Device type
DriveAddrList=List of the drive addresses contained in this file.
MTGUIVersion=GUI version used to generate this download file.
SaveDate=Date when this file was generated (e.g. through an UPLOAD process).
SaveTime=Time of the day when this file was generated.
SaveElementCode=Contains bit-coded information on which Sercos data elements (see SPA commands) are available in this file.
SaveType=Contains the information on which Sercos parameters are available in this file. This concerns the following Sercos parameters:
- according to the list from S-0-0017
- according to the list from S-0-0192

Section [DESCRIPTION]

Contains a brief description of the keys under the section [DRIVExx:X-Y-ZZZZ].
xx=Drive address
X=Sercos data type (S=standard data,P=product data)
Y=Parameter set (0..7)
ZZZZ=Data block no. (0..4095)

Section [DRIVExx]

Contains the required drive data.

DriveType=Drive types
Max_P_Number=Max. data block number for the product data
Max_P_Set=Max.parameter set for the product data
Max_S_Number=Max. data block number for the standard data
Max_S_Set=Max.parameter set for the standard data

Section [DRIVExx:X-Y-ZZZZ]

Contains the SERCOS parameter data.

001=Number of data lines for the SERCOS operating date
002=Name of the SERCOS parameter
003=Attributes of the SERCOS parameter
004=Unit of the SERCOS parameter
005=Min. input value of the SERCOS parameter
006= Max. input value of the SERCOS parameter
007.001-007.XXX=Data lines for the SERCOS operating date
008=Data status of the SERCOS parameter

BW_DPA1_(1)_(2)_{(3)}	**(Single Write)**
(1) = Defines which drives are to be downloaded	0 = Drives according to the current parameter set > 0 = Requested drive address [1..32]
(2) = Complete download file name	Download file according to the preset structure
(3) = Optional parameter; defines bit-coded controller information. Format: WORD in HEX code 0xYYZZ If this parameter does not exist, the following default setting is used: - Writing according to list S-0-0192 - Write only operating date - Attribute comparison is performed	**High-Byte (0xYY)** defines according to which list the SERCOS parameters are written; the following applies: 0x00 = acc. to S-0-0192 0x01 = acc. to S-0-0017 0x02 = acc. to INI-File (not yet implemented !!) **Low-Byte (0xZZ)** defines which SERCOS elements are to be written; here, only the attributes (0x04) and the operating date (0x40) can be written. **EXCEPTION:** If the highest bit has been set in this word, attribute comparison is switched off during download!!

Response Structure

The response to the "DPA1" FI command consists of three lines, each with one column. The meaning of the elements is as follows:

- Line 1 = Job ID [01...20]
  (see Chapter "FI Commands for the MPCX Device Group", IFJ).
- Line 2 = FI command
  [String, in accordance with Chapter "Elements of the FI Command"]
- Line 3 = FI Job Error Code
  (see Chapter "Error Codes")

Note: File and path details must be enclosed in inverted commas.

Example DPA1

The SERCOS parameters stored in the download file D:\SERCPAR.DAT are to be transferred into the parameterized drives of device 0.

As there is no optional 3rd parameter, the default setting is used.

Writing according to list S-0-0192

Write only operating date

Attribute comparison is performed

FI command		**00_BW_DPA1_0_"D:\SERCPAR.DAT"**
Line	**Column**	**Answer**
1	1	01
2	1	00_BW_ DPA1_0_"D:\SERCPAR.DAT"
3	1	0

Explanation

Reads the SERCOS parameters from the drives and saves them to the upload file. This upload command is an FI job.

Structure of upload file

The structure of the upload file corresponds to that of a Windows Ini file. The structure is identical with that of a download file.

BR_DPA1_(1)_(2)_{(3)}	**(Single Read)**
(1) = defines which drives are to be saved	0 = Drives according to the current parameter set > 0 = Requested drive address [1..32]
(2) = Complete upload file name	
(3) = Optional parameter; defines bit-coded controller information. Format: WORD in HEX code 0xYYZZ If this parameter does not exist, the following default setting is used: - Reading according to list S-0-0192 - Read attribute and operating date	**High-Byte (0xYY)** defines according to which list the SERCOS parameters are read; the following applies: 0x00 = acc. to S-0-0192 0x01 = acc. to S-0-0017 0x02 = acc. to INI-File (not yet implemented !!) **Low-Byte (0xZZ)** defines which SERCOS elements are to be read; the following applies: 0x01 = Date status 0x02 = Name 0x03 = Attribute 0x08 = Unit 0x10 = Min. input value 0x20 = Max. input value 0x40 = Operating date The corresponding bits can be OR'd, e.g. operating date (0x40) and unit (0x08) produces OR'd 0x48.

Response Structure

The response to the "DPA1" FI command consists of three lines, each with one column. The meaning of the elements is as follows:

- Line 1 = Job ID [01...20]
  (see Chapter "FI Commands for the MPCX Device Group", IFJ).
- Line 2 = FI command
  [String, in accordance with Chapter "Elements of the FI Command"]
- Line 3 = FI Job Error Code
  (see Chapter "Error Codes")

Note: File and path details must be enclosed in inverted commas.

Example DPA1

The SERCOS parameters which are then to be stored in the upload file D:\SERCPAR.DAT, are to be saved from the parameterized drives of device 0. Data storage is to be performed according to list S-0-0017.

During this process, the SERCOS elements:

Data status

Name

Attribute

Unit

Min. input value

Max. input value

Operating date

are to be saved.

FI command		**00_BR_DPA1_0_"D:\ SERCPAR .DAT_0x017F"**
Line	**Column**	**Answer**
1	1	01
2	1	00_BR_DPA1_0_"D:\SERCPAR.DAT_0x017F"
3	1	0

Explanation Reads the log file generated during download of the drive parameters. With the "BR_DPA2" FI command described in the following, this file is read out subsequently to indicate download errors.

Note: File and path details must be enclosed in inverted commas.

BR_DPA2_(1) (Single Read)

(1) = Complete download file name

Response Structure The response to the "DPA2" command consists of n lines, each with 9 columns. One line is provided for each drive.

Line n	**Column 1**	**...**	**Column 9**

Value Range/Meaning of Columns

1 =	Drive address	[1..32]
2 =	Download status	[READY] = Successful download for the drive [WARNING] = Download failed for at least 1 SERCOS parameter [ERROR] = Download failed for the drive
3 =	Error text	[--] = No error text available; otherwise, the error text for the failed drive download
4 =	Current drive firmware – acc. to S-0-0030	
5 =	Drive firmware – acc. to the download file	
6 =	Number of SERCOS parameters missing in the download file	[0] = All required SERCOS parameters are available (> 0] = Number of missing SERCOS parameters
7 =	List of missing SERCOS parameters	[--] = No missing SERCOS parameters; otherwise, the list of missing SERCOS parameters, the character ‚|' (0x7C) being used as a separator, e.g.:S-0-0009|P-0-0096.
8 =	Number of SERCOS parameters which could NOT be loaded	[0] = All required SERCOS parameters could be loaded (> 0] = Number of missing SERCOS parameters which could NOT be loaded
9 =	List of SERCOS parameters which could NOT be loaded	[--] = No unloadable SERCOS parameters; otherwise, the list of unloadable SERCOS parameters, the character ‚|' (0x7C) being used as a separator, e.g.:S-0-0006|S-0-0359.

Example DPA2

SERCOS parameters have been transferred into drives 1 and 2 of device 0. In drive 2, the SERCOS parameters S-0-0006 and S-0-0359 are missing, and attribute comparison has failed for the SERCOS parameters S-0-0393, P-0-0099. and P-0-0260.

FI command		**00_BR_DPA2_"D:\SERCPAR.DAT"**
Line	**Column**	**Answer**
1	1	1
	2	READY
	3	--
	4	HSM1.1-SSE-03V25
	5	HSM1.1-SSE-03V22
	6	0
	7	--
	8	0
	9	--
2	1	2
	2	WARNING
	3	--
	4	HSM1.1-SSE-03V25
	5	HSM1.1-SSE-03V22
	6	2
	7	S-0-0006|S-0-0359
	8	3
	9	S-0-0393|P-0-0099|P-0-0260

Explanation

Reads the detailed information from the log file generated during download of the drive parameters. The "BR_DPA3" FI command described in the following is used to supply the detailed error information of the individual drive in plain text.

Note: File and path details must be enclosed in inverted commas.

BR_DPA3_(1)_(2)_(3)	**(Single Read)**
(1) = Requested drive address	[1..32]
(2) = Controller information on whether the detailed information on missing or unloadable SERCOS parameters is requested	[0] = Information on the missing SERCOS parameters [1] = Information on the unloadable SERCOS parameters
(3) = Complete download file name	

Response Structure

The response to the "DPA3" command consists of n lines, each with 9 columns. There is one line for each missing or unloadable SERCOS parameter.

Line n	**Column 1**	**Column 2**

Value Range/Meaning of Columns

1 =	SERCOS parameter designation	According to SERCOS specification, e.g.: S-0-0009
2 =	Error text	

Example DPA3

Detailed information on the missing SERCOS parameters of drive 2 (device 0) is to be requested.

FI command		**00_BR_DPA3_2_0_"D:\SERCPAR.DAT"**
Line	**Column**	**Answer**
1	1	S-0-0006
	2	The SERCOS operating date is NOT available in the drive parameter download file.
2	1	S-0-0359
	2	The SERCOS operating date is NOT available in the drive parameter download file.

Explanation

This command is used to read out drive addresses and the respective axis type available in the transferred drive data download file.

Note: File and path details must be enclosed in inverted commas.

BR_DPA4_(1) **(Single Read)**

(1) = Complete download file name

Response Structure

The response to the "DPA4" command consists of n lines, each with 4 columns. One line is provided for each drive.

Line n	**Column 1**	**...**	**Column 4**

Value Range/Meaning of Columns

1 =	Drive address	[1..32]
2 =	Type of axis	[see Chapter 6.2 "Data Tables"]
3 =	Date of data generation	Contains the date of generation of the drive data
4 =	Time of data generation	Contains the time of generation of the drive data

Example DPA4

In the drive data download file, the SERCOS data for drives 1,2,5,8 are stored.

FI command		**00_BR_DPA4_"D:\SERCPAR.DAT"**
Line	**Column**	**Answer**
1	1	1
	2	81
	3	04.06.2003
	4	14:16:23
2	1	2
	2	82
	3	04.06.2003
	4	14:16:23
3	1	5
	2	81
	3	04.06.2003
	4	14:16:23
4	1	8
	2	83

	3	04.06.2003
	4	14:16:23

5.57 Delete NC Program: DPN

MWCX device group

Designation **DPN** **D**elete **P**rogram **NC**

Explanation An NC program located in an NC package directory is deleted.

FI command

BW_DPN_(1)_(2)_(3)_(4)	**(Single Write)**
(1) = NC package directory number	[1...99]
(2) = NC process number	[0 6]
(3) = NC program number	[1...99]
(4) = with check / without check	[1 / 0]

Response Structure One line with one column is output to acknowledge the FI command issued. The following meanings then apply, depending on parameter 4 (check):

With check (1)	
(BOF_C_NCPROG_CREATED)	NC program not deleted.

Without check (0)	
(BOF_FCT_OK) = **BOF_F**un**CT**ion**_OK**	NC program has been deleted.

Example DPN The NC program numbered 1 in NC package directory 3 of process 2 is to be deleted.

FI command		**00_BW_DPN_1_2_3_0**
Line	**Column**	**Answer**
1	1	(BOF_FCT_OK)

Reference to Literature See chapter entitled "Literature" [31].

5.58 Delete NC Program Package: DPP

MWCX device group

Designation **DPP** **D**elete **P**rogram **P**ackage

Explanation An NC program package is deleted in the NC package directory of the selected MWCX device group.

FI command

BW_DPP_(1)	**(Single Write)**
(1) = NC program package	[1...99]

Response Structure One line with one column is output to acknowledge the FI command issued. The meaning of the elements is as follows:

(BOF_FCT_OK) = **BOF_F**un**CT**ion**_OK**	Program package has been deleted.

Example DPP The NC program package numbered 1 in the NC package directory is to be deleted.

FI command		**00_BW_DPN_1_2_3_0**
Line	**Column**	**Answer**
1	1	(BOF_FCT_OK)

Reference to Literature See chapter entitled "Literature" [17].

5.59 Reading the Device Status Information: DSI

MWCX device group

Designation **DSI** **D**evice **S**tatus **I**nformation

Explanation This allows the most important device status information to be read out. The following information is returned:

Type of information	**Status**	**Statement**
System error information		Yes/No
Information on mechanism error		Yes/No
Machine key information		4 Byte HEX
Machine key information	valid	Yes/No
Machine status information		4 Byte HEX
Sercans information		4 Byte HEX
Parameter download	running	Yes/No
PLC download	running	Yes/No
Firmware download	running	Yes/No
Offline/Online information		Yes/No
Device simulation	switched on	Yes/No
Device status information		ON, OFF
PLC components available		Yes/No
Monitor mode	active	Yes/No

FI command Read out device status information for ALL defined devices.

BR_DSI1	**(Single Read)**
BC_DSI1	**(Cyclic Read)**
BB_DSI1	**(Break Cyclic Read)**

Note: The "DSI1" FI command refers to all devices within this device group. Therefore, any valid device address can be indicated in the command line (see example DSI1). The FI device polling mechanism **MUST** be switched on (see system configurator)!

Response Structure

The following table shows the general structure of the response to the "DSI1" FI command.

Line 1...n	Column 1	...	Column 13

Value Range/Meaning of Columns

1 =	device address	[00...63]
2 =	System error information	[0 = there is no system error 1 = there is a system error]
3 =	Information on mechanism error	[0 = there is no mechanism error 1 = there is a mechanism error]
4 =	Machine key information	[4 byte in HEX coding]
5 =	Is machine key information valid?	[0 = not valid, 1=valid]
6 =	Machine status information	[4 byte in HEX coding]
7 =	Sercans information	[4 byte in HEX coding]
8 =	Is parameter download active?	[0 = parameter download not running 1 = parameter download running]
9 =	Is PLC download active?	[0 = PLC download not running 1 = PLC download running]
10 =	Is firmware download active?	[0 = PLC download not running 1 = PLC download running]
11 =	Offline/Online information	[0 = device connection interrupted 1 = device connection O.K.]
12 =	Device simulation switched on?	[0 = NO Simulation mode 1 = simulation mode]
13 =	Current device status information	[0 = DeviceStatus=OFF 1 = DeviceStatus=ON]
14 =	Communication channel defined?	[0 = NO communication channel 1 = Communication channel defined]
15 =	PLC components available?	[0 = NO PLC component 1 = PLC component (DOS-Pcl) 2 = PLC component (WIN-Pcl)]
16 =	Monitor mode	[0 = NO monitor mode active 1 = Monitor mode active]

Example DSI1

Read the current device status information.
Assumption:
The following devices addresses are defined:

- Device address 01 (MWCX device)
- Device address 03 (MWSX device)

FI command		**01_BR_DSI1**
Line	**Column**	**Answer**
1	1	01
	2	0
	3	0
	4	00000000
	5	0
	6	00000000
	7	00000000
	8	0
	9	0
	10	0
	11	1
	12	0
	13	1
2	1	03
	2	1
	3	0
	4	00000000
	5	0
	6	00000000
	7	00000000
	8	0
	9	0
	10	0
	11	1
	12	0
	13	1
	14	1
	15	2
	16	0

FI command

Read out device status information for a selected device.

BR_DSI2	**(Single Read)**
BC_DSI2	**(Cyclic Read)**
BB_DSI2	**(Break Cyclic Read)**

Response Structure

The following table shows the general structure of the response to the "DSI2" FI command.

Line 1...n	Column 1	...	Column 13

Value Range/Meaning of Columns

1 =	device address	[00...63]
2 =	System error information	[0 = there is no system error 1 = there is a system error]
3 =	Information on mechanism error	[0 = there is no mechanism error 1 = there is a mechanism error]
4 =	Machine key information	[4 byte in HEX coding]
5 =	Is machine key information valid?	[0 = not valid, 1=valid]
6 =	Machine status information	[4 byte in HEX coding]
7 =	Sercans information	[4 byte in HEX coding]
8 =	Is parameter download active?	[0 = parameter download not running 1 = parameter download running]
9 =	Is PLC download active?	[0 = PLC download not running 1 = PLC download running]
10 =	Is firmware download active?	[0 = PLC download not running 1 = PLC download running]
11 =	Offline/Online information	[0 = device connection interrupted 1 = device connection O.K.]
12 =	Device simulation switched on?	[0 = NO Simulation mode 1 = simulation mode]
13 =	Current device status information	[0 = DeviceStatus=OFF 1 = DeviceStatus=ON]
14 =	Communication channel defined?	[0 = NO communication channel 1 = Communication channel defined]
15 =	PLC components available ?	[0 = NO PLC component 1 = PLC component (DOS-Pcl) 2 = PLC component (WIN-Pcl)]
16 =	Monitor mode	[0 = NO monitor mode active 1 = Monitor mode active]

Example DSI2 Read the current device status information for the selected device.

FI command		00_BR_DSI2
Line	**Column**	**Answer**
1	1	00
	2	0
	3	0
	4	00000000
	5	0
	6	00000000
	7	00000000
	8	0
	9	0
	10	0
	11	1
	12	0
	13	1
	14	1

	15	2
	16	0

5.60 Tool Management Configuration Data: DTC

MWCX device group

Designation **DTC** **D**evice **T**ool Management **C**onfiguration

Explanation Supplies the most important system parameter data for tool management.

FI command Read tool management data.

BR_DTC1 **(Single Read)**

BC_DTC1 **(Cyclic Read)**

Response Structure One line with 10 columns is output for the returned values.

Line 1	Column 1	...	Column 10

Value Range/Meaning of the Columns

1 =	Tool Management	[YES, NO]
2 =	Setup list	[[STATION], [PROGRAM]]
3 =	Max. number of tool edges	[1...9]
4 =	Wear register	[YES, NO]
5 =	Offset register	[YES, NO]
6 =	Comment	[YES, NO]
7 =	Wear factors	[YES, NO]
8 =	Tool life	[YES, NO]
9 =	Geometry limit values	[YES, NO]
10 =	Tool technology	[[TURN./MILL.], [GRINDING]]

Note: If there is no tool management (Column 1: NO), then all partial results from Column 2 are marked as [--].

Example DTC1 Returns the system parameter data from the tool management

FI command		**00_BR_DTC1**
Line	**Column**	**Answer**
1	1	YES
	2	[STATION]
	3	4
	4	YES
	5	YES
	6	NO
	7	NO
	8	YES
	9	YES
	10	[TURN./MILL.]

FI command Data is read from tool management, as e.g. basic user data and tool edge user data.

BR_DTC2 **(Single Read)**

BC_DTC2 **(Cyclic Read)**

Response Structure

One line with 48 columns is output for the returned values.

Line 1	Column 1	...	Column 48

Value Range/Meaning of the Columns

1 =	Tool Management	[YES, NO]
2 =	Setup list	[[STATION], [PROGRAM]]
3 =	Max. number of tool edges	[1...9]
4 =	Wear register	[YES, NO]
5 =	Offset register	[YES, NO]
6 =	Comment	[YES, NO]
7 =	Wear factors	[YES, NO]
8 =	Tool life	[YES, NO]
9 =	Geometry limit values	[YES, NO]
10 =	Tool technology	[[TURN./MILL.], [GRINDING]]
11 =	Tool user date 1	[YES, NO]
12 =	Tool user date 1	[Tool user date,--]
13 =	Tool user date 2	[YES, NO]
14 =	Tool user date 2	[Tool user date,--]
15 =	Tool user date 3	[YES, NO]
16 =	Tool user date 3	[Tool user date,--]
17 =	Tool user date 4	[YES, NO]
18 =	Tool user date 4	[Tool user date,--]
19 =	Tool user date 5	[YES, NO]
20 =	Tool user date 5	[Tool user date,--]
21 =	Tool user date 6	[YES, NO]
22 =	Tool user date 6	[Tool user date,--]
23 =	Tool user date 7	[YES, NO]
24 =	Tool user date 7	[Tool user date,--]
25 =	Tool user date 8	[YES, NO]
26 =	Tool user date 8	[Tool user date,--]
27 =	Tool user date 9	[YES, NO]
28 =	Tool user date 9	[Tool user date,--]
29 =	Cutter user date 1	[YES, NO]
30 =	Cutter user date 1	[Cutter user date,--]
31 =	Cutter user date 2	[YES, NO]
32 =	Cutter user date 2	[Cutter user date,--]
33 =	Cutter user date 3	[YES, NO]
34 =	Cutter user date 3	[Cutter user date,--]
35 =	Cutter user date 4	[YES, NO]
36 =	Cutter user date 4	[Cutter user date,--]
37 =	Cutter user date 5	[YES, NO]
38 =	Cutter user date 5	[Cutter user date,--]
39 =	Cutter user date 6	[YES, NO]
40 =	Cutter user date 6	[Cutter user date,--]
41 =	Cutter user date 7	[YES, NO]

42 =	Cutter user date 7	[Cutter user date,--]
43 =	Cutter user date 8	[YES, NO]
44 =	Cutter user date 8	[Cutter user date,--]
45 =	Cutter user date 9	[YES, NO]
46 =	Cutter user date 9	[Cutter user date,--]
47 =	Cutter user date 10	[YES, NO]
48 =	Cutter user date 10	[Cutter user date,--]

Note: If there is no tool management (Column 1: NO), then all partial results from Column 2 are marked as [--].

Example DTC2

Supply the system parameter data from the tool management.

FI command		**00_BR_DTC2**
Line	**Column**	**Answer**
1	1	YES
	2	[STATION]
	3	4
	4	YES
	5	YES
	6	NO
	7	NO
	8	YES
	9	YES
	10	[TURN./MILL.]
	11	NO
	12	--
	13	NO
	14	--
	15	NO
	16	--
	17	NO
	18	--
	19	NO
	20	--
	21	NO
	22	--
	23	NO
	24	--
	25	NO
	26	--
	27	NO
	28	--
	29	NO
	30	--

	31	NO
	32	--
	33	NO
	34	--
	35	NO
	36	--
	37	NO
	38	--
	39	NO
	40	--
	41	NO
	42	--
	43	NO
	44	--
	45	NO
	46	--
	47	NO
	48	--

Reference to Literature See chapter entitled "Literature" [8].

5.61 Distance to Go of Axis Movement: DTG

MWCX device group

Designation **DTG** **D**istance **T**o **G**o

Explanation The distance to go of the movement of a selected axis is output. The FI command "DTG1" returns the distance to go of an axis, related to the code of the axis meaning. The FI command "DTG2", on the other hand, returns the distance to go of an axis, related to the physical axis number.

FI command Output the distance to go of the selected axis of the device specified, related to the code of the axis meaning.

Using the optional fourth parameter it is possible to pre-select conversion of the result into mm or inches. If, however, a spindle is selected as an axis, indicating a measurement system serves no purpose.

CR_DTG1_(1)_(2)_(3){_(4)}	**(Single Read)**
CC_DTG1_(1)_(2)_(3){_(4)}	**(Cyclic Read)**
CB_DTG1_(1)_(2)_(3){_(4)}	**(Break Cyclic Read)**
(1) = NC process number	[0...6]
(2) = Axis meaning	[0...11; 20]; (see chapter "Data Tables")
(3) = System of coordinates	[1 = machine coordinates 2 = program coordinates]
(4) = Required measurement system (opt.)	[mm, inch]

FI command Output the distance to go of the movement of the selected axis of the device specified related to the physical axis number.

Using the optional third parameter it is possible to pre-select conversion of the result into mm or inches. If, however, a spindle is selected as an axis, indicating a measurement system serves no purpose.

CR_DTG2_(1)_(2){_(3)}	**(Single Read)**
CC_DTG2_(1)_(2){_(3)}	**(Cyclic Read)**
CB_DTG2_(1)_(2){_(3)}	**(Break Cyclic Read)**
(1) = Physical axis number	[1...32, according to settings of the system parameters]
(2) = System of coordinates	[1 = machine coordinates 2 = program coordinates]
(3) = Required measurement system (opt.)	[mm, inch]

Response Structure

The following table shows the general structure of the response to the FI commands "DTG1" and "DTG2". One line is output with 4 columns for the axis designation, distance to go, unit and the distance to go limited to "indicated decimal places".

Line 1	Column 1	...	Column 4

Value Range/Meaning of Columns

1 = Axis name	[according to settings of axis parameters]
2 = Distance to go	[according to settings of process parameters]
3 = Unit	[mm, inch]
4 = Distance to go	[as Column 2, but rounded up or down according to the parameter "indicated decimal places"]

Note: If the specified axis or a spindle is not defined in the selected NC process then the answer in all columns is [--].

Example DTG1

Read the distance to go of the movement of the Z axis in machine coordinates in NC process 0 of device address 00.

FI command	**00_CR_DTG1_0_2_1**			
Answer				
Line 1	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	Z1	-1.2345	[mm]	-1.235

Example DTG1

Read the distance to go of the movement of the Z axis in machine coordinates in NC process 0 of device address 00. Values are displayed in inches.

FI command	**00_CR_DTG1_0_2_1_inch**			
Answer				
Line 1	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	Z1	-0.0486	[inch]	-0.049

Example DTG2

Read the distance to go of the movement of the Z axis (physical axis number = 3) in machine coordinates at the device address 00.

FI command	**00_CR_DTG2_3_1**			
Answer				
Line 1	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	Z1	-1.2345	[mm]	-1.235

Reference to Literature

See chapter entitled "Literature" [16].

5.62 Device Type and Accompanying Components: DTY

MWCX device group

Designation **DTY** **D**evice **TY**pe

Explanation The device type and the accompanying components of the selected device address are output.

FI command **BR_DTY1** **(Single Read)**

Response Structure The following table shows the general structure of the response to the "DTY1" FI command. A line with three columns is output for the device type, as well as the names of the first device component and the name of the second device component.

Line 1	Column 1	...	Column 3

Value Range/Meaning of Columns

1 =	Device Type	(see chapter entitled "Elements of the FI Command", and "Identifier")
2 =	Component type1	IND_DEV.INI entry: Componenttype1=
3 =	Component type 2	IND_DEV.INI entry: Componenttype2=

Example DTY1 Output the device type and the accompanying components of device address 00.

FI command	00_BR_DTY1		
Answer			
Line	Column 1	Column 2	Column 3
1	MTC200-P-G2	MTS-P	MTC-P

5.63 Diagnosis Window Data: DWD

MWCX device group

Designation **DWD** **D**iagnosis **W**indow **D**ata

Explanation Diagnostic messages are output. The data is edited in such a way that it can be output directly in the diagnosis overview, i.e., where applicable, different types of diagnosis, such as a ProVi message and a process message, are returned simultaneously.

FI command Output all diagnostic messages.

For reasons of optimization, not all data is called up. Accordingly, the Diagnosis server must be informed that the data is required (see ADW).

BR_DWD1_(1){_(2)}	**(Single Read)**
BC_DWD1_(1){_(2)}	**(Cyclic Read)**
(1) = Type of diagnosis window	[1 = NC error, 2 = sequence errors, 3 = general errors, 4 = messages, 10 = start requirements, 11 = warnings, 12 = setup diagnosis]
(2) = Module number	[1...99] ! only for window type 1 -4 !

Output first diagnostic messages.

BR_DWD2_(1){_(2)}	**(Single Read)**
BC_DWD2_(1){_(2)}	**(Cyclic Read)**

(1) = Type of diagnosis window	[1 = NC error, 2 = sequence errors, 3 = general errors, 4 = messages, 10 = start requirements, 11 = warnings, 12 = setup diagnosis]
(2) = Module number	[1...99] ! only for window type 1 -4 !

Response Structure

The following table shows the general structure of the "DWD1" and "DWD2" FI commands. The number of lines depends on the number of messages pending. Different columns are valid according to the type of diagnosis. If there are no messages, the number of lines is 0.

Line 1...n	Column 1	...	Column 14

Meaning of the Columns

1 =	Message text	[ASCII characters]
2 =	Time stamp day	[mm.dd.yyyy]
3 =	Time stamp hour	[hh:mm:ss]
4 =	Reference text available	[YES, NO]
5 =	Type of diagnosis	[1 = ProVi, 2 = SFC, 3 = MTC-NC, 4 = MTA-NC]
6 =	Message number	[ASCII characters]
7 =	Message ID	[ASCII characters] (DWORD, decimal) (ProVi)
8 =	Mechanism number	[0..31] (MTC-NC) [0] (MTA-NC)
9 =	2 byte additional information	[ASCII characters] (MTC NC)
10 =	Message group	[1...9999] (MTA-NC)
11 =	SFC entity name	[ASCII characters]
12 =	NC note	[ASCII characters] (MTC NC)
13 =	Analysis of criteria available	[YES, NO] (ProVi, SFC)
14 =	Message HTML file	[ASCII characters] (ProVi, MTC-NC)

Example DWD1

All diagnostic messages from Module 3 in Control unit 0.
There are two messages present:

FI command		**00_BR_DWD1_4_3**
Line	**Column**	**Answer**
1	1	Guard not closed
	2	01.27.2000
	3	14:56:32
	4	YES
	5	1
	6	34
	7	43923028
	8	
	9	
	10	
	11	
	12	
	13	YES
	14	

FI command		00_BR_DWD1_4_3
Line	**Column**	**Answer**
2	1	Station waiting until tool-change command has ended.
	2	01.27.2000
	3	15:03:10
	4	YES
	5	3
	6	79
	7	
	8	1
	9	0
	10	
	11	
	12	
	13	NO
	14	

Example DWD2

The first diagnostic message from Module 3 in Control unit 0.
There are two messages present:

FI command		00_BR_DWD2_4_3
Line	**Column**	**Answer**
1	1	Guard not closed
	2	01.27.2000
	3	14:56:32
	4	YES
	5	1
	6	34
	7	43923028
	8	
	9	
	10	
	11	
	12	
	13	YES
	14	

Reference to Literature

See chapter entitled "Literature" [13].

5.64 Component Information for a System Error: ECI

MWCX device group

Designation **ECI** **E**rror **C**omponent **I**nformation

Explanation When a system error is present, this command is used to define which controller component is causing the error. Here, PLC components are differentiated from general controller components (e.g. CNC, Synax, MTA, …).

FI command **BR_ECI1** **(Single Read)**

Response Structure The response to the "ECI1" FI command consists of one line with two columns.

Line 1	Column 1	Column 2

Value Range/Meaning of Columns

1 =	PLC component information	[0 = There is NO system error at the PLC) 1 = There is a system error at the PLC]
2 =	General information on controller components	[0 = There is NO system error at the general control component [0 = There is a system error at the general control component; the following applies: 2 = CNC component 3 = SYNAX component 4 = VISUAL-MOTION component 5 = MTA component 6 = TRANS 200 component]

Example ECI1 There is a system error present in device 0 (MTC200-P-G2) which is caused by the PLC component.

FI command		**00_BR_ECI1**
Line	**Column**	**Answer**
1	1	1
1	2	0

5.65 Existing Errors: EDE

MWCX device group

Designation **EDE** **E**xisting **D**iagnosis **E**rror

Explanation Whether or not errors exist in a control unit or in a module is queried. These can be step chain errors, NC errors or ProVi errors.

FI command Query whether there are errors in this control unit.

BR_EDE1 **(Single Read)**

BC_EDE1 **(Cyclic Read)**

Response Structure The following table shows the general structure of the "EDE1" FI command.

Line 1	Column 1

Meaning of the Columns

1 = Error exists [YES, NO]

Example EDE1

Do errors exist in control unit 0?

FI command		**00_BR_EDE1**
Line	**Column**	**Answer**
1	1	YES

FI command

Query whether or not errors exist in a specific module.

BR_EDE2_(1) **(Single Read)**

BC_EDE2_(1) **(Cyclic Read)**

(1) = Module number [1...99]

Response Structure

The following table shows the general structure of the "EDE2" FI command.

Line 1	**Column 1**

Meaning of the Columns

1 = Error exists [YES, NO]

Example EDE2

Do errors exist in Module 1 on Control unit 0?

FI command		**00_BR_EDE2_2**
Line	**Column**	**Answer**
1	1	NO

Reference to Literature

See chapter entitled "Literature" [13].

5.66 Existing Diagnosis Window: EDW

MWCX device group

Designation

EDW **E**xisting **D**iagnosis **W**indow

Explanation

Which types of diagnosis window exist is queried.

FI command

Output all types of diagnosis window.

BR_EDW1 **(Single Read)**

Response Structure

The following table shows the general structure of the "EDW1" FI command. The number of lines depends on the number of types of window existing.

Line 0...n	**Column 1**	**Column 2**

Meaning of the Columns

1 = Type of diagnosis window	[1 = NC error, 2 = sequence errors, 3 = general errors, 4 = messages, 10 = start requirements, 11 = warnings, 12 = setup diagnosis]
2 = Module number	[ASCII characters] 0 = Diagnosis window type does not belong to any module

Example EDW1

All types of diagnosis window in control unit 0.

There are three diagnosis windows.

FI command		**00_BR_EDW1**
Line	**Column**	**Answer**
1	1	10
	2	0
2	1	1
	2	3
3	1	2
	2	3

FI command

Output all types of diagnosis window for one module.

BR_EDW2_(1)	**(Single Read)**
(1) = Module number	[1...99]

Response Structure

The following table shows the general structure of the "EDW2" FI command. The number of lines depends on the number of types of window existing.

Line 0...n	Column 1	Column 2

Meaning of the Columns

1 = Type of diagnosis window	[1 = NC error, 2 = sequence errors, 3 = general errors, 4 = messages]
2 = Module number	[ASCII characters] 0 = Diagnosis window type does not belong to any module

Example EDW2

All types of diagnosis window in Module 3, Control unit 0.

There are two diagnosis windows.

FI command		**00_BR_EDW2_3**
Line	**Column**	**Answer**
1	1	1
	2	3
2	1	2
	2	3

FI command

Query a specific type of diagnosis window.

BR_EDW3_(1){_(2)}	**(Single Read)**
(1) = Type of diagnosis window	[1 = NC error, 2 = sequence errors, 3 = general errors, 4 = messages, 10 = start requirements, 11 = warnings, 12 = setup diagnosis]
(2) = Module number	[1...99] ! only for window type 1 -4 !

Response Structure

The following table shows the general structure of the "EDW3" FI command.

Line 1	Column 1

Meaning of the Columns

1 = Type of diagnosis window exists	[YES, NO]

Example EDW3

Query whether or not a NC error window exists in module 3, control unit 0.

FI command		**00_BR_EDW3_1_3**
Line	**Column**	**Answer**
1	1	YES

5.67 Existing NC Diagnoses: END

MWCX device group

Designation **END** **E**xisting **N**C **D**iagnosis

Explanation Which NC diagnostic types exist is queried. Depending on the FI command, specific types are queried or else the diagnostic types for one module are output together.

FI command Query which NC diagnostic types are available in a module.

BR_END1_(1) **(Single Read)**

(1) = Module number [1...99]

Response Structure The following table shows the general structure of the "END1" FI command.

Line 1	**Column 1-2**

Meaning of the Columns

1 = Messages exist [YES, NO]

2 = Errors exist [YES, NO]

Example END1 Query the NC diagnostic types in Module 2 on Control unit 0.

FI command		**00_BR_END1_2**
Line	**Column**	**Answer**
1	1	NO
	2	YES

FI command Query a specific NC diagnostic type.

BR_END2_(1)_(2) **(Single Read)**

(1) = Message type [1 = error, 2 = messages]

(2) = Module number [1...99]

Response Structure The following table shows the general structure of the "END2" FI command.

Line 1	**Column 1**

Meaning of the Columns

1 = Diagnosis type exists [YES, NO]

Example END2 Are there any messages in module 4 in control unit 0?

FI command		**00_BR_END2_2_4**
Line	**Column**	**Answer**
1	1	YES

5.68 Existing PLC Diagnoses: EPD

MWCX device group

Designation **EPD** **E**xisting **P**LC **D**iagnosis

Explanation Which PLC diagnostic types exist is queried. Depending on the FI command, specific types are queried or else the diagnostic types for a device or a module are output together.

FI command Query which PLC diagnostic types are available on a control unit.

BR_EPD1 **(Single Read)**

Response Structure The following table shows the general structure of the "EPD1" FI command.

Line 1	Column 1-3

Meaning of the Columns

1 = Start requirement exists	[YES, NO]
2 = Warning exists	[YES, NO]
3 = Setup diagnosis exists	[YES, NO]

Example EPD1 Query PLC diagnostic types in control unit 0.

FI command		**00_BR_EPD1**
Line	**Column**	**Answer**
1	1	YES
	2	NO
	3	YES

FI command Query which PLC diagnostic types are available in a module.

BR_EPD2_(1) **(Single Read)**

(1) = Module number [1...99]

Response Structure The following table shows the general structure of the "EPD2" FI command.

Line 1	Column 1-3

Meaning of the Columns

1 = Messages exist	[YES, NO]
2 = Errors exist	[YES, NO]
3 = Step chains exist	[YES, NO]

Example EPD2 Query the PLC diagnostic types in Module 2 on Control unit 0.

FI command		**00_BR_EPD2_2**
Line	**Column**	**Answer**
1	1	NO
	2	YES
	3	YES

FI command Query a specific PLC diagnostic type.

BR_EPD3_(1){_(2)} **(Single Read)**

(1) = Message type	[1 = error, 2 = messages, 3 = SFC, 10 = warnings, 11 = start requirements, 12 = setup diagnosis]
(2) = Module number	[1...99] ! only for message type 1 -3!

Response Structure The following table shows the general structure of the "EPD3" FI command.

Line 1	Column 1

Meaning of the Columns 1 = Diagnosis type exists [YES, NO]

Example EPD3 Are there any messages in module 4 in control unit 0?

FI command		**00_BR_EPD3_2_4**
Line	**Column**	**Answer**
1	1	YES

5.69 End Point of an Axis Movement: EPO

MWCX device group

Designation **EPO** **E**nd **PO**sition

Explanation The end point of the movement of a selected axis is output. The FI command "EPO1" returns the end point of the movement, related to the code of the axis meaning. The FI command "EPO2", on the other hand, returns the end point of the movement of an axis related to the physical axis number.

FI command Output the end point of the selected device related to the code of the axis meaning.

Using the optional fourth parameter it is possible to pre-select conversion of the result into mm or inches. If, however, a spindle is selected as an axis, indicating a measurement system serves no purpose.

CR_EPO1_(1)_(2)_(3){_(4)} **(Single Read)**

CC_EPO1_(1)_(2)_(3){_(4)} **(Cyclic Read)**

CB_EPO1_(1)_(2)_(3){_(4)} **(Break Cyclic Read)**

(1) = NC process number	[0...6]
(2) = Axis meaning	[0...11; 20]; (see chapter "Data Tables")
(3) = System of coordinates	[1 = machine coordinates 2 = program coordinates]
(4) = Required measurement system (opt.)	[mm, inch]

FI command Output the end point of the selected axis of the device specified, related to the physical axis number.

Using the optional third parameter it is possible to pre-select conversion of the result into mm or inches. If, however, a spindle is selected as an axis, indicating a measurement system serves no purpose.

CR_EPO2_(1)_(2){_(3)} **(Single Read)**

CC_EPO2_(1)_(2){_(3)} **(Cyclic Read)**

CB_EPO2_(1)_(2){_(3)} **(Break Cyclic Read)**

(1) = Physical axis number	[1...32, according to settings of the system parameters]
(2) = System of coordinates	[1 = machine coordinates 2 = program coordinates]
(3) = Required measurement system (opt.)	[mm, inch]

Response Structure

The following table shows the general structure of the response to the FI commands "EPO1" and "EPO2". One line is output with 4 columns for the axis designation, end point of the movement, unit and the position limited to "indicated decimal places".

Line 1	Column 1	Column 2	Column 3	Column 4

Value Range/Meaning of Columns

1 = Axis name	[according to settings of axis parameters]
2 = end point	[according to settings of process parameters]
3 = Unit	[mm, inch]
4 = end point	[as Column 2, but rounded up or down according to the parameter "indicated decimal places"]

Note: If the specified axis is not defined in the selected NC process then the response in all columns is [--].

Example EPO1

Read the end point of the movement of the Z axis in machine coordinates in NC process 0 of device address 00.

FI command	**00_CR_EPO1_0_2_1**			
Answer				
Line	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	Z1	-1.2345	[mm]	-1.235

Example EPO1

Read the end point of the movement of the Z axis in machine coordinates in NC process 0 of device address 00. Values are displayed in inches:

FI command	**00_CR_EPO1_0_2_1_inch**			
Answer				
Line	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	Z1	-0.0486	[inch]	-0.049

Example EPO2

Read the end point of the movement of the Z axis (physical axis number = 3) in machine coordinates at device address 00.

FI command	**00_CR_EPO2_3_1**			
Answer				
Line	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	Z1	-1.2345	[mm]	-1.235

Reference to Literature

See chapter entitled "Literature" [16].

5.70 Existing ProVi Types: EPT

MWCX device group

Designation **EPT** **E**xisting **P**roVi **T**ypes

Explanation Which ProVi types are programmed in the current PLC program is queried. The data is returned in a suitable form for the message texts of the small control panels. There is no need to define modules in Moduldef.ini.

FI command Output all ProVi types.

BR_EPT1 **(Single Read)**

Response Structure The following table shows the general structure of the "EPT1" FI command. The number of lines depends on the number of ProVi types existing.

Line 0...n	Column 1	Column 2

Meaning of the Columns

1 = Type [11 = Error, 12 = Messages, 20 = Start requirements, 21 = Warnings, 22 = Setup diagnosis]

2 = Index [ASCII characters]

Example EPT1 All ProVi types in control unit 0. There are three diagnosis windows:

FI command		00_BR_EPT1
Line	**Column**	**Answer**
1	1	20
	2	0
2	1	11
	2	3
3	1	12
	2	3

5.71 Error Status: EST

MWCX device group

Designation **EST** **E**rror **ST**ate

Explanation Queries the error state of a variable.

FI command Query the frozen error status of a variable.

BR_EST1!(1)!(2) **(Single Read)**

BC_EST1!(1)!(2) **(Cyclic Read)**

(1) = Error ID [ASCII characters] (DWORD, decimal)

(2) = Variable name [ASCII characters]

Note: The separator "!" is used in this command.

Response Structure The following table shows the general structure of the "EXD1" FI command.

Line 1	Column 1

Meaning of the Columns

1 = Error state

WinPCL - Example EST

Read the value of WinPCL variable "IB_EXT24" in WinPCL program "Prog", at device address 00.

Assumption:
The WinPCL variable "IB_EXT24" is declared as BOOL in WinPCL program "Prog" .

FI command		**00_BR_EST1!5892855!:Prog.IB_EXT24**
Line	**Column**	**Answer**
1	1	1

5.72 Execution Display: EXD

MWCX device group

Designation

EXD **EX**ecution **D**isplay

Explanation

Information for displaying the execution of a movement is output.

FI command

Query the execution of a step or of an action.

BR_EXD1!(1)!(2)!(3)	**(Single Read)**
BC_EXD1!(1)!(2)!(3)	**(Cyclic Read)**
(1) = SFC entity name	[ASCII characters]
(2) = Step or action name	[ASCII - characters]
(3) = Behaviour of mode	[1 – all modes, 2 – manual mode]

Note: The separator "!" is used in this command.

Response Structure

The following table shows the general structure of the "EXD1" FI command.

Line 1	Column 1

Meaning of the Columns

1 = Execution	[1 – can be executed, 0 – cannot be executed]

Example EXD1

Query the execution of the step "open" for the chain "clamp" in control unit 0 for all modes.

FI command		**00_BR_EXD1!Station03A.Clamp!Open!1**
Line	**Column**	**Answer**
1	1	1

FI command

Query whether the condition analysis (control image) of a step chain is enabled.

BR_EXD2!(1)	**(Single Read)**
(1) = SFC entity name	[ASCII characters]

Note: The separator "!" is used in this command.

Response Structure

The following table shows the general structure of the "EXD2" FI command.

	Line 1	Column 1
Meaning of the Columns	1 = Enabled	[1 - enabled, 0 – not enabled]

Example EXD2 Query whether the condition analysis of the "clamp" chain has been enabled.

FI command		**00_BR_EXD2!Station03A.Clamp**
Line	**Column**	**Answer**
1	1	1

5.73 Global Process Parameter Configuration: GPC

MWCX device group

Designation **GPC** **G**lobal **P**rocess **C**onfiguration

Explanation The configuration of the global process parameter of the active machine parameter record of the selected device from the MWCX device group is read out.

The following are all a part of the global process parameters: the programmable and actually displayed digits after the decimal point for the displacement, the name of the NC process, the base programming unit, the max. zero-point-data bank number, D-corrections, whether a basic setting is required, whether a reference is required, whether a transformation between Cartesian and polar coordinates is possible, jogging axis results in a reset and the re-positioning of the tool memory axis.

Note: The FI commands "GPPx" (see Global Process Parameters "GPP") should be preferred to the FI commands "GPCx" as the access speed there has been optimized.

FI command Output of the configuration of the global process parameters of all defined NC processes of the active machine parameter record.

BR_GPC1 **(Single Read)**

Response Structure

The following table shows the general structure of the response to the "GPC1" FI command. The response consists of between one and a maximum of n=7 lines (n= max. number of defined NC processes [0...6] = 7), each with 12 columns.

Line 1	Column 1	...	Column 12

Note: If there is no active machine parameter record in the device then the columns [1..0.12] are not applicable.

Value Range/Meaning of Columns

1 =	NC process number	[0...6]
2 =	Name of the NC process	[max. 20 ASCII characters]
3 =	Basic coordinate system	[mm, inch]
4 =	Programmed number of positions after decimal point	[4, 5]
5 =	Displayed positions after the decimal point	[0...4]
6 =	Max. zero-point-data bank number	[0...9]
7 =	D corrections	[YES, NO]
8 =	Home position required	[YES, NO]
9 =	Reference required	[YES, NO]
10 =	Cartesian-polar coordinate transformation	[YES, NO]
11 =	Manual axis jogging causes reset	[YES, NO]
12 =	Tool storage axis repositioning	[YES, NO]

Example GPC1

Read the configuration of the global process parameters of all defined NC processes of the active machine parameter record of device address 00.
Assumption:
The following three NC processes are defined:

Sled 1 (NC process number 0),

Turret 1 (NC process number 1) and

Turret 2 (NC process number 3).

FI command		**00_BR_GPC1**
Line	**Column**	**Answer**
1	1	0
	2	Sled 1
	3	[mm]
	4	4
	5	3
	6	0
	7	YES
	8	NO
	9	NO
	10	NO
	11	YES
	12	NO
2	1	1
	2	Turret 1
	3	[mm]

FI command		**00_BR_GPC1**
Line	**Column**	**Answer**
	4	4
	5	3
	6	0
	7	NO
	8	YES
	9	YES
	10	NO
	11	YES
	12	NO
3	1	3
	2	Turret 2
	3	[mm]
	4	4
	5	3
	6	0
	7	NO
	8	YES
	9	NO
	10	NO
	11	YES
	12	NO

Reference to Literature See chapter entitled "Literature" [34].

FI command Output the configuration of the global process parameters of the active machine parameter record of the selected device related to the NC process.

BR_GPC2_(1)	**(Single Read)**
(1) = NC process number	[0...6]

Response Structure The following table shows the general structure of the response to the "GPC2" FI command. The response consists of a line with 12 columns.

Line 1	**Column 1**	**...**	**Column 12**

Note: If there is no active machine parameter record in the device or the selected NC process is not defined then the columns [1...12] are not applicable.

Value Range/Meaning of Columns

1 =	NC process number	[0...6]
2 =	Name of the NC process	[max. 20 ASCII characters]
3 =	Basic coordinate system	[mm, inch]
4 =	Programmed number of positions after decimal point	[4, 5]
5 =	Displayed positions after the decimal point	[0...4]

Rexroth
Bosch Group

6 =	Max. zero-point-data bank number	[0...9]
7 =	D corrections	[YES, NO]
8 =	Home position required	[YES, NO]
9 =	Reference required	[YES, NO]
10 =	Cartesian-polar coordinate transformation	[YES, NO]
11 =	Manual axis jogging causes reset	[YES, NO]
12 =	Tool storage axis repositioning	[YES, NO]

Example GPC2

Read the global process parameter in NC process 0 of the active machine parameter record of device address 00.

Assumption:

The following three NC processes are defined:

Sled 1 (NC process number 0),

Turret 1 (NC process number 1) and

Turret 2 (NC process number 3).

FI command		**00_BR_GPC2_0**
Line	**Column**	**Answer**
1	1	0
	2	Sled 1
	3	[mm]
	4	4
	5	3
	6	0
	7	YES
	8	NO
	9	NO
	10	NO
	11	YES
	12	NO

Reference to Literature

See chapter entitled "Literature" [34].

5.74 Global Process Parameters: GPP

MWCX device group

Designation **GPP** **G**lobal **P**rocess **P**arameter

Explanation

The global process parameters of the active machine parameter record of the selected device from the MWCX device group is read out. This includes the programmable and actually displayed digits after the decimal point for the displacement, the name of the CC process, the basic coordinate system and the max. zero-point-data bank number.

Note: The FI commands "GPPx" should be preferred to the FI commands "GPCx" as the access speed has been optimized.

FI command

Output of the configuration of the global process parameters of all defined NC processes of the active machine parameter record.

BR_GPP1 **(Single Read)**

Response Structure

The following table shows the general structure of the response to the "GPC1" FI command. The response consists of one up to a maximum of n=7 lines (n= max. number of defined NC processes [0...6] = 7), each with six columns.

Line 1	Column 1	...	Column 6

Note: If there is no active machine parameter record in the device then the columns [1...6] are not applicable.

Value Range/Meaning of Columns

1 =	NC process number	[0...6]
2 =	Name of the NC process	[max. 20 ASCII characters]
3 =	Basic coordinate system	[mm, inch]
4 =	Programmed number of positions after decimal point	[4, 5]
5 =	Displayed positions after the decimal point	[0...4]
6 =	Max. zero-point-data bank number	[0...9]

Example GPP1

Read the global process parameters of all defined NC processes of the active machine parameter record of device address 00.
Assumption:
The following three NC processes are defined:
Sled 1 (NC process number 0),
Turret 1 (NC process number 1) and
Turret 2 (NC process number 3).

FI command		**00_BR_GPP1**
Line	**Column**	**Answer**
1	1	0
	2	Sled 1
	3	[mm]
	4	4
	5	3
	6	0
2	1	1
	2	Turret 1
	3	[mm]
	4	4
	5	3
	6	0
3	1	3
	2	Turret 2
	3	[mm]
	4	4
	5	3
	6	0

Reference to Literature

See chapter entitled "Literature" [35].

FI command

Output the configuration of the global process parameters of the active machine parameter record of the selected device related to the NC process.

BR_GPP2_(1)	**(Single Read)**
(1) = NC process number	[0...6]

Response Structure

The following table shows the general structure of the response to the "GPP2" FI command. The response consists of a line with six columns.

Line 1	**Column 1**	**...**	**Column 6**

Note: If there is no active machine parameter record in the device or the selected NC process is not defined then the columns [1...6] are not applicable.

Value Range/Meaning of Columns

1 =	NC process number	[0...6]
2 =	Name of the NC process	[max. 20 ASCII characters]
3 =	Basic coordinate system	[mm, inch]
4 =	Programmed number of positions after decimal point	[4, 5]
5 =	Displayed positions after the decimal point	[0...4]
6 =	Max. zero-point-data bank number	[0...9]

Example GPP2

Read the global process parameter in NC process 0 of the active machine parameter record of device address 00.

Assumption:
The following three NC processes are defined:
Sled 1 (NC process number 0),
Turret 1 (NC process number 1) and
Turret 2 (NC process number 3).

FI command		**00_BR_GPP2_0**
Line	**Column**	**Answer**
1	1	0
	2	Sled 1
	3	[mm]
	4	4
	5	3
	6	0

Reference to Literature

See chapter entitled "Literature" [35].

5.75 Insert NC Program Package: IPP

MWCX device group

Designation **IPP** **I**nsert NC-**P**rogram **P**ackage

Explanation Inserts an NC program package into the NC package directory.

FI command **BW_IPP_(1){_(2)}** **(Single Write)**

(1) =	Number in NC package directory	[1...99]
(2) =	Is the NC package directory entry empty?	[0 = without check (preset); 1 = with check] ! Optional !

Note: If an NC program package already exists at the selected number of the NC package directory, an error is signaled if execution of the check has been selected.

Value to be written Name of the NC package [max. 32 ASCII characters]

Note: The value to be written is passed to the "acValue" parameter as an ASCII value in the "DataTransfer" routine.

Response Structure No response is returned on this FI command!

Example IPP Enter the NC program package with the designation "KEY1" into number 1 of the NC package directory.

<u>Assumption:</u>

It is to be checked whether the selected entry in the NC package directory is empty.

FI command	**Value to be written: KEY1** **00_BW_IPP_1_1**

5.76 Module Assignment of a Process: MAP

MWCX device group

Designation **MAP** **M**odul **A**ssign of **P**rocess

Explanation The module to which a particular process is assigned is read out from the "Moduldef.ini" file. This file is located in the directory "[LW]:\Program Files\Indramat\MTGUI\CustomData\Resource" and contains the data for all module configurations.

The process data is located in three sections:

[DeviceAddrX\ModulY\Process]

whereby "X" stands for the device addressed and "Y" for the configured module numbers.

FI command Determine the module to which the process belongs. Information is read out from the module configuration of the MWCX device group.

BR_MAP1_(1) **(Single Read)**

BC_MAP1_(1) **(Cyclic Read)**

BB_MAP1_(1) **(Break Cyclic Read)**

1 = Mechanism number [0...31]

Response Structure

The following table shows the general structure of the response to the "MAP1" FI command. One line with one column is output for the module number that has been determined.

Line 1	Column 1

Value Range of the Column

1 = Module number [0...99]

Example MAP1

Read the module number to which NC process number 4 is assigned from the module configuration.

Assumption:
The module to which NC process 4 is assigned has module number 5.

FI command		00_BR_MAP1_4
Line	Column	Answer
1	1	5

Reference to Literature

See chapter entitled "Literature" [36].

5.77 Read Reference Name of a PLC Variable: MAR

MWCX device group

Designation

MAR **M**ap **A**bsolute PCL-**R**eference

PLC Explanation

The absolute reference name of a symbolic PLC variable is read out.

FI command

Read the absolute reference name of a PLC variable.

BR_MAR_(1) **(Single Read)**

BC_MAR_(1) **(Cyclic Read)**

(1) = Designator of PLC variables

Response Structure

The following table shows the general structure of the response to the FI command "MAR". One line with one column is output for the reference name that has been determined.

Line 1	Column 1

Meaning of the Column

1 = Identifier of the PLC variable

PLC – Example MAR

Read the absolute reference name of the PLC variable with the identifier "abref" at device address 00.

Assumption:
The PLC variable with the identifier "abref" is of the type "INTEGER".

FI command		00_BR_MAR_abref
Line	Column	Answer
1	1	%M100.0

WinPCL-Explanation

The absolute reference name of a symbolic WinPCL PLC variable with program entity is read out.

FI command

Read the absolute reference name of a WinPCL PLC variable.

BR_MAR1_(1) **(Single Read)**

(1) = Identifier of the PLC variable

WinPCL - Example MAR1

Read the absolute reference name of the WinPCL variable with the identifier "Prog.abref" at device address 00.

Assumption:
The WinPCL variable with the identifier "Prog.abref" is of the type "INTEGER" and is present in WinPCL program "Prog".

FI command		**00_BR_MAR_:Prog.abref**
Line	**Column**	**Answer**
1	1	%M100.0

Reference to Literature

See chapter entitled "Literature" [30].

5.78 Device Data of the Module Configuration: MCD

MWCX device group

Designation

MCD **M**odul **C**onfiguration: **D**evice Information

Explanation

All device data for module configuration is read out from the "Moduldef.ini" file which is stored in the "[LW]:\Program Files\Indramat\MTGUI\CustomData\Resource" directory following standard installation. The device data is in the sections [DeviceAddrX], whereby "X" stands for the configured device addresses.

FI command

Read out device data within the module configuration of the MWCX device groups.

BR_MCD1	**(Single Read)**
BC_MCD1	**(Cyclic Read)**
BB_MCD1	**(Break Cyclic Read)**

Note: The "MCD1" FI command refers to all devices within the MWCX device group. Therefore, any valid device address can be indicated in the command line (see example MCD1).

Response Structure

The following table shows the general structure of the response to the "MCD1" FI command. The number of lines depends on the number of configured devices. Each line consists of four columns for the device address as well as PLC-FB names for providing setup diagnostics, warning messages and start requirements.

Line 1	**Column 1**	**...**	**Column 4**

Value Range of the Columns

1 = Device address	[0...63]
2 = PLC-FB name for the setup diagnostics	[max. 9 ASCII characters]
3 = PLC-FB name for the warning messages	[max. 9 ASCII characters]
4 = PLC-FB name for the start requirements	[max. 9 ASCII characters]

Example MCD1

Read all device data of the module configuration.

Assumption:
The following devices have been configured in the MWCX device group:

- Device address 01 (MTC200-P)
- Device address 03 (MTC200-P)

FI command	**03_BR_MCD1**			
Answer				
Line	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	01	PVSetup_1	PVWarn_1	PVStart_1
2	03	PVSetup_3	PVWarn_3	PVStart_3

Reference to Literature See chapter entitled "Literature" [36].

5.79 Module Data of the Module Configuration: MCM

MWCX device group

Designation **MCM** **M**odul **C**onfiguration: **M**odul Information

Explanation All module data for module configuration is read out from the "Moduldef.ini" file which is stored in the "[LW]:\Program Files\Indramat\MTGUI\CustomData\Resource" directory following standard installation. This file contains all module configuration data. The module data is located in sections [DeviceAddrX\ModulY], whereby "X" stands for the device addressed and "Y" for the configured module numbers.

FI command Read out module data from the module configuration with respect to a device from the MWCX device group.

BR_MCM1 **(Single Read)**

BC_MCM1 **(Cyclic Read)**

BB_MCM1 **(Break Cyclic Read)**

Response Structure The following table shows the general structure of the response to the "MCM1" FI command. The number of lines depends on the number of configured modules of a device. Each line consists of four columns for the module number, module name and PLC-FB names for general module errors and module messages.

Line 1	**Column 1**	**...**	**Column 4**

Value Range of the Columns

1 = Module number	[0...99]
2 = Module name	[max. 28 ASCII characters]
3 = PLC-FB name for general module errors	[max. 9 ASCII characters]
4 = PLC-FB name for module messages	[max. 9 ASCII characters]

Example MCM1 Read the module data of device 03 from the module configuration:

Assumption:
The following modules have been defined:

- Module number 5
- Module number 7

FI command	**03_BR_MCM1**			
Answer				
Line	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	5	Module 5 – Milling	PVError_5	PVMsg_5
2	7	Module 7 - Drilling	PVError_7	PVMsg_7

Reference to Literature See chapter entitled "Literature" [36].

5.80 Process Data of the Module Configuration: MCP

MWCX device group

Designation **MCP** **M**odul **C**onfiguration: **P**rocess Information

Explanation All process data of a certain module is read out from the "Moduldef.ini" file which is stored in the "[LW]:\Program Files\Indramat\MTGUI\CustomData\Resource" directory following standard installation. This file contains all module configuration data. The process data is located in sections [DeviceAddrX\ModulY\Process], whereby "X" stands for the device addressed and "Y" for the selected module number.

FI command

BR_MCP1_(1)	**(Single Read)**
BC_MCP1_(1)	**(Cyclic Read)**
BB_MCP1_(1)	**(Break Cyclic Read)**
(1) = Module number	[0...99]

Response Structure The response to the FI command "MCP1" consists of one up to a maximum number of n=32 lines with 1 column for the number of the NC process or of the external mechanisms.

Line 1...32	Column 1

Value Range of the Column 1 = Mechanism number [0...31]

Example MCP1 Read the NC process number of module 5 of device 00 of the module configuration.

Assumption:
The following NC processes are defined:

NC process number 1

NC process number 4

FI command		**00_BR_MCP1_5**
Line	**Column**	**Answer**
1	1	1
2	1	4

Reference to Literature See chapter entitled "Literature" [36].

5.81 SFC Data of the Module Configuration: MCS

MWCX device group

Designation **MCS** **M**odul **C**onfiguration: **S**FC Information

Explanation All SFC data of a certain module is read out from the "Moduldef.ini" file which is stored in the "[LW]:\Program Files\Indramat\MTGUI\CustomData\Resource" directory following standard installation. This file contains all module configuration data. The SFC data is located in sections [DeviceAddrX\ModulY\Sfc], whereby "X" stands for the device addressed and "Y" for the selected module number.

FI command Read out the SFC data with respect to the module of a device from the module configuration of the MWCX device group.

BR_MCS1_(1)	**(Single Read)**
BC_MCS1_(1)	**(Cyclic Read)**
BB_MCS1_(1)	**(Break Cyclic Read)**
(1) = Module number	[0...99]

Response Structure The number of lines depends on the number of configured Indrastep step chains for a device. Each line contains a column for the name of the Indrastep step chains.

Value Range of the Column 1 = Name of the Indrastep step chain [format W.X.Y.Z]

Format W.X.Y.Z	Value Range
W	Max. 9 ASCII characters
X	Max. 9 ASCII characters ! OPTIONAL !
Y	Max. 9 ASCII characters ! OPTIONAL !
Z	Max. 9 ASCII characters ! OPTIONAL !

Example MCS1 Read the name of the Indrastep step chain of module 5 from device 03 of the module configuration.

Assumption:
The following Indrastep step chains have been defined:

ISFB_1

- FB_US.ISFB_3
- FB_US.ISFB_3.SW1
- FB_US.ISFB_3.SW1.ABBA

FI command		**03_BR_MCS1_5**
Line	**Column**	**Answer**
1	1	ISFB_1
2	1	FB_US.ISFB_3
3	1	FB_US.ISFB_3.SW1
4	1	FB_US.ISFB_3.SW1.ABBA

Reference to Literature See chapter entitled "Literature" [36].

5.82 Processing of Machine Data and Download: MDA

MWCX device group

Designation **MDA** **M**achine **D**ata **A**ccess

Explanation Complete machine data records are downloaded by means of a download file.

FI command Machine data record download command whereby two predefined functions are to be programmed by the user. These two functions concern:

1. Function for creating the download file itself:

```
Long MachineDataDownloadBegin( Long lProjectNumber,
                               Long lDeviceNumber,
                               Long lIndexNumber,
                               Char* pcMDLFileName,
                               Long lMaxLengthFileNameBuffer,
                               Char* pcErrorText,
                               Long lMaxLengthErrorTextBuffer)
```

Pass parameters:

lProjectNumber:	Currently selected project number
lDeviceNumber:	Currently selected device address
lIndexNumber:	Currently selected machine data record directory number [1..99]
pcMDLFileName:	Contains the complete file names for the created machine data record download file.
lMaxLengthFileNameBuffer:	Max. length of the buffer for the name of the machine data record download file.
pcErrorText:	Text of user error, if applicable
lMaxLengthErrorTextBuffer:	Max. length of the buffer for the user error text.

2. Function called up at the end of the parameter download:

```
Long MachineDataDownloadEnd( Char* pcMDLFileName,
                             Long lResult)
```

Pass parameters:

pcMDLFileName:	Contains the complete file names for the created machine data record download file.
lResult:	Contains the status message of the downloading process of the machine data record. Here, 0 = Machine data record download procedure O.K. > 0 = An error occurred

The two functions must be programmed in a DLL by the user and also exported from it.

BW_MDA1_(1)_{(2)} **(Single Write)**

(1) = Machine data record directory number; the two functions to be implemented are located in INDIF410.DLL.

(2) = Complete DLL name, if required, in which the two functions to be implemented are located.

Response Structure

The response to the "MDA1" FI command consists of three lines, each with one column. The meaning of the elements is as follows:

- Line 1 = Job ID [01...20]
  (see Chapter "FI Commands for the MPCX Device Group", IFJ).
- Line 2 = FI command
  [String, in accordance with Chapter "Elements of the FI Command"]
- Line 3 = FI Job Error Code
  (see Chapter "Error Codes")

Note: File and path details must be enclosed in inverted commas.

Example MDA1

00_BW_MDA1_3_"D:\UserDir\USER.DLL"

FI command		**00_BW_MDA1_3_D:\UserDir\USER.DLL**
Line	**Column**	**Answer**
1	1	01
2	1	00_BW_MDA1_3_D:\UserDir\USER.DLL
3	1	0

FI command

Machine data record download command whereby the machine data record download file is directly indicated.

BW_MDA2_(1) **(Single Write)**

(1) = Download file name of complete machine data record

Response Structure

The response to the "MDA2" FI command consists of three lines, each with one column. The meaning of the elements is as follows:

- Line 1 = Job ID [01...20]
  (see Chapter "FI Commands for the MPCX Device Group", IFJ).
- Line 2 = FI command
  [String, in accordance with Chapter "Elements of the FI Command"]
- Line 3 = FI Job Error Code
  (see Chapter "Error Codes")

Note: File and path details must be enclosed in inverted commas.

Example MDA2

00_BW_MDA2_"D:\DOWNLOAD.DAT"

FI command		**00_BW_MDA2_"D:\DOWNLOAD.DAT"**
Line	**Column**	**Answer**
1	1	01
2	1	00_BW_MDA2_"D:\DOWNLOAD.DAT"
3	1	0

Structure of Download File

The structure of the download file corresponds to that of a Windows Ini file. Bosch Rexroth's own description in V21_Madat_Download_Upload_01.doc, is recommended for a more detailed account of the structure of the download file.

Summary:

Section [ID_MACHINE_DATA]
Information concerning the identification of the machine data record.

Section [ID_TYPE_DEFINITION]
Information concerning the identification of the type definition.

Section [TYPE_DEFINITION_INFO]
Max. data type identification number

Section [TYPE_DEFINITION_XXX]
Data for the various type definitions.

Section [PAGE_INFO]
Max. defined page number.

Section ID_PAGE_DEFINITION_XXX]
Information concerning the identification of the page definition.

Section [PAGE_DEFINITION_XXX]
Data for the page definitions.

Section [PAGE_DESCRIPTION_XXX_YYY]
Data for writing the individual data elements of a page.

Section [PAGE_DATA_INFO]
Max. defined page number for writing of machine data.

Section [PAGE_DATA_ELEMENTS_XXX]
Information for the machine data that is to be written.

Section [PAGE_DATA_XXX]
Data for the machine data that is to be written.

Explanation

This command is used to delete the data values of **one** or **all** defined machine data pages. However, the machine data page definitions are **NOT** deleted.

FI command

BW_MDA3_(1) **(Single Write)**

(1) = Page selection [0 = ALL page
1..299 = Selected page]

Response Structure

The response to the "MDA3" FI command consists of one line with one column.

Line 1	Column 1

Value Range/Meaning of Columns

1 = Status message (P_ACK) (P_ACK)

Example MDA3

The data values of page 101 for device address 0 are to be deleted.

FI command		00_BW_MDA3_101
Line	**Column**	**Answer**
1	1	(P_ACK)

FI command

With this command, ALL machine data page definitions in the selected device are deleted.

BW_MDA4 **(Single Write)**

Response Structure

The response to the "MDA4" FI command consists of one line with one column.

Line 1	Column 1

Value Range/Meaning of Columns	1 =	List of the deleted machine data page numbers, or -- if NO machine data page numbers have been deleted.	List of page numbers separated by comma or by --

Example MDA4 The following machine data page definitions have been deleted for the device 00.

FI command		**00_BW_MDA4**
Line	**Column**	**Answer**
1	1	1,2,10,11,12,21,30,40,50,60,61,62,90,91,92,101,102,103,104

5.83 Machine Data Upload: MDA

MWCX device group

Designation **MDA** **M**achine **D**ata **A**ccess

Explanation Uploads complete machine data records from a selected device. The data read is written into an upload file with an identical structure to that of a download file.

FI command Machine data record upload command whereby two predefined functions are to be programmed by the user. These two functions concern:

1. The function supplies the complete name of the upload file:

```
Long MachineDataUploadBegin(  Long lProjectNumber,
                              Long lDeviceNumber,
                              Char* pcUploadFileName,
                              Long lMaxLengthFileNameBuffer,
                              Char* pcErrorText,
                              Long lMaxLengthErrorTextBuffer)
```

Pass parameters:

• lProjectNumber:	Currently selected project number
• lDeviceNumber:	Currently selected device address
• pcUploadFileName: the machine	Contains the complete file name for data record upload file to be created.
• lMaxLengthFileNameBuffer:	max. length of the buffer for the name of the machine data record upload file.
• pcErrorText:	If necessary, user error text
• lMaxLengthErrorTextBuffer:	Max. length of the buffer for the user error text.

2. Function called up at the end of the machine data record upload:

```
Long MachineDataUploadEnd(  Char* pcUploadFileName,
                            Long lResult)
```

Pass parameters:

pcUploadFileName:	Contains the complete file names for the created machine data record upload file.
lResult:	Contains the status message of the uploading process of the machine data record. Here, 0 = Parameter upload procedure O.K. > 0 = Error has occurred

The two functions must be programmed in a DLL by the user and also exported from it.

BR_MDA1_(1)_{(2)} **(Single Read)**

(1) = Machine data record directory number; the two functions to be implemented are located in INDIF410.DLL.

(2) = Complete DLL name, if required, in which the two functions to be implemented are located.

Response Structure

The response to the "MDA1" FI command consists of three lines, each with one column. The meaning of the elements is as follows:

- Line 1 = Job ID
  [01...20] (see Chapter "FI Commands for the MPCX Device Group: IFJ").
- Line 2 = FI command
  [String, in accordance with Chapter "Elements of the FI Command"]
- Line 3 = FI Job Error Code
  (see Chapter "Error Codes")

Note: File and path details must be enclosed in inverted commas.

Example MDA1

00_BR_MDA1_2_"D:\UserDir\USER.DLL"

FI command		**00_BR_MDA1_2_"D:\UserDir\USER.DLL"**
Line	**Column**	**Answer**
1	1	01
2	1	00_BR_MDA1_2_"D:\UserDir\USER.DLL"
3	1	0

FI command

Machine data record upload command whereby the machine data record upload file is directly indicated.

BR_MDA2_(1) **(Single Read)**

(1) = complete machine data record upload file name

Response Structure

The response to the "MDA2" FI command consists of three lines, each with one column. The meaning of the elements is as follows:

- Line 1 = Job ID
  [01...20] (see Chapter "FI Commands for the MPCX Device Group: IFJ").
- Line 2 = FI command
  [String, in accordance with Chapter "Elements of the FI Command"]
- Line 3 = FI Job Error Code
  (see Chapter "Error Codes")

Note: File and path details must be enclosed in inverted commas.

Example MDA2

00_BR_MDA2_"D:\UPLOAD.DAT"

FI command		00_BR_MDA2_"D:\UPLOAD.DAT"
Line	**Column**	**Answer**
1	1	01
2	1	00_BR_MDA2_"D:\UPLOAD.DAT"
3	1	0

Structure of Upload File

The structure of the upload file corresponds to that of a Windows Ini file. Bosch Rexroth‘s own description in "V21_Madat_Download_Upload_01.doc", is recommended for a more detailed account of the structure of the download file.

For a summary refer to the description under Machine Data Record Download Command.

5.84 Inputting an NC Record: MDI

MWCX device group

Designation

MDI **M**anual **D**ata **I**nput

FI command

Input an NC record for direct execution in manual mode.

CW_MDI_(1)	**(Single Write)**
(1) = NC process number	[0...6]

Value to be written

NC record	(see DOK-MTC200-NC**PRO*V..)

Note: The value to be written is passed to the "acValue" parameter as an ASCII string in the "DataTransfer" routine.

Response Structure

One line is output with a column for acknowledgement of whether or not the FI command has been executed successfully.

(P_ACK) = **P**ositive **ACK**nowledge	Data element has been set

Example MDI

Write an NC record for direct execution in NC process 0.

Conditions

The control unit must be in "Setup" ("Manual") mode. Axes X1 and Y1 exist.

Value to be written

G01 X1 50.45 Y1 35.456 F 1000

FI command		00_CW_MDI_0
Line	**Column**	**Answer**
1	1	(P_ACK)

or, if the process is not ready for the next NC record: Error 1014 = BOF_NEGATIVE_ACKNOWLEDGE (N_ACK):

Line	Column	Answer	
1	1	**1**	(=N_ACK)
	2	**37**	(=text number of N_ACK)
	3	**0x00000000**	(=additional information for some texts)
	4	**Process still active**	(=text of the N_ACK error)

or, if a syntax error is detected in the passed NC record: Error 1014 = BOF_NEGATIVE_ACKNOWLEDGE (N_ACK):

Line	Column	Answer
1	1	**1** (=N_ACK)
	2	**18** (=text number of N_ACK)
	3	**0xFFFFFFFC** (=additional information for some texts)
	4	**Unrecognized expression in the NC program**

Monitoring the MDI Status

During MDI operation the status of the process should be monitored by reading the diagnostic message:

BR_AMM3_(1) **(Single Read)**

.

(1) = NC process number [0...6]

Example 1

Before inputting an NC record:

FI command		**00_BR_AMM3_0**
Line	**Column**	**Answer**
1	1...8	(For process information see Documentation BR_AMM3)
	9	Ready to start: Operating mode 'setup"

Example 2

After inputting an NC record:

FI command		**00_BR_AMM3_0**
Line	**Column**	**Answer**
1	1...8	(For process information see documentation BR_AMM3)
	9	Ready to start for processing of MDI record

Possible error codes:
Example 3

Assumption:
It is not possible to process the NC record because of an erroneous expression.

FI command		**00_BR_AMM3_0**
Line	**Column**	**Answer**
1	1...8	(For process information see documentation BR_AMM3)
	9	Invalid axis

Example 4

Assumption:
External start conditions are missing for the process to execute the NC record.

FI command		**00_BR_AMM3_0**
Line	**Column**	**Answer**
1	1...8	(For process information see documentation BR_AMM3)
	9	Failure of external 24 Volt supply

Reference to Literature

See chapter entitled "Literature" [4].

5.85 Reading and Writing Machine Data: MDS

MWCX device group

Designation **MDS** **M**achine **D**ata **S**ingle

Explanation For reading and writing a list of machine data. A maximum of 10 machine data items can be written or read at the same time.

FI command Write machine data.

BW_MDS1_(1)_{(2)..(10)} **(Single Write)**

(1) = Info string for the machine data date to be written

Structure of the info strings:
Data type\page number\data element\
Travel variable1\travel variable2\data value\
data unit

Data type:
Value according to the machine data type definition (normal: 1..29)

Page number:
Value according to page definition

Data element:
Value according to page definition

Travel variable1:
Value according to page definition

Travel variable2:
Value according to page definition

Data value:
Value to be written

Data unit:
String of units

Response Structure The following table shows the general structure of the response to the "MDS1" FI command. There is one response line for each write value.

Line 1...n	Column 1	...	Column 5

Value Range/Meaning of Columns

1 =	Status message with regard to write procedure	1 = Write procedure has been executed successfully. 0 = Write procedure could NOT be executed.
2 =	Info string for the machine data date to be written	See syntax of the MDS command
3 =	Error class	0 = No error 1 = Communication error (NACK) 2 = FI error 3 = Error text
4 =	Short message text	If the write procedure has been executed successfully, then --, otherwise the short error text is given.
5 =	Reference information	If the write procedure has been executed successfully, then --, otherwise reference information is given.

Example MDS1

Two machine data values are written:

1. Value:

Data type: DREAL (ID number: 13)
Page number: 123
Data element: 1
Travel variable1: 0
Travel variable2: 0
Data value: 123.23
Data unit: NONE (encoded as – !)

2. Value:

Data type: POS (ID number: 14)
Page number: 103
Data element: 3
Travel variable1: 1
Travel variable2: 2
Data value: 100.00
Data unit: mm

FI command		**00_BW_MDS1_13\123\1\0\0\123.23_- \14\103\3\1\2\100.00\mm**
Line	**Column**	**Answer**
1	1	1
	2	13\123\1\0\0\123.23\-\
	3	0
	4	--
	5	--
2	1	1
	2	14\103\3\1\2\100.00\mm
	3	0
	4	--
	5	--

FI command

Read machine data.

BR_MDS1_(1)_{(2)..(10)}	**(Single Read)**
(1) = Info string for the machine data date to be read	Structure of the info strings: Data type\page number\data element\ Travel variable1\travel variable2 **Data type:** Value according to the machine data type definition (normal: 1..29) **Page number:** Value according to page definition **Data element:** Value according to page definition **Travel variable1:** Value according to page definition **Travel variable2:** Value according to page definition

Response Structure

The following table shows the general structure of the response to the "MDS1" FI command. There is one response line for each value read.

Line 1...n	Column 1	...	Column 7

Value Range/Meaning of Columns

1 =	Data value read as a string	If an error occurs during reading – is given.
2 =	Data unit read as a string	If an error occurs during reading – is given.
3 =	Number of places after the decimal point	If an error occurs during reading – is given.
4 =	Info string for the machine data date to be read	See syntax of the MDS command.
5 =	Error class	0 = No error 1 = Communication error (NACK) 2 = FI error 3 = Error text
6 =	Short message text	If the read procedure has been executed successfully, then --, otherwise the short error text is given.
7 =	Reference information	If the read procedure has been executed successfully, then --, otherwise reference information is given.

Example MDS1

Three machine data values are read:

1. Value:

Data type: DREAL (ID number: 13)
Page number: 101
Data element: 1
Travel variable1: 0
Travel variable2: 0

2. Value:

Data type: POS (ID number: 14)
Page number: 122
Data element: 3
Travel variable1: 1
Travel variable2: 2

3. Value:

Data type: WORD (ID number: 3)
Page number: 122
Data element: 4
Travel variable1: 1
Travel variable2: 2

FI command		00_BR_MDS1_13\101\1\0\0_14\122\3\1\2_3\122\4\1\2
Line	**Column**	**Answer**
1	1	111.11
	2	
	3	0
	4	13\101\1\0\0
	5	0
	6	--
	7	--
2	1	66.6666
	2	MM
	3	4
	4	14\122\3\1\2
	5	0
	6	--
	7	--
3	1	10
	2	
	3	0
	4	3\122\4\1\2
	5	0
	6	--
	7	--

Reference to Literature See chapter entitled "Literature" [33].

5.86 Downloading Message Texts: MFD

MWCX device group

Designation **MFD** **M**essage **F**iles **D**ownload

FI command This is used to load the message texts into the device indicated. These message texts are required for small devices. The following message texts are transmitted, depending on the type of device:

- system error messages
- transmission error messages
- mechanism messages

Note: This FI command is an FI job!

BW_MFD1 **(Single Write)**

Response Structure

The response to the "MFD1" FI command consists of three lines, each with one column. The meaning of the elements is as follows:

Line 1 = Job ID	[01...20] (refer to chapter entitled "FI Commands for the MPCX Device Group", IFJ).
Line 2 = FI command	[String] (in accordance with the chapter entitled "Elements of the FI Command")
Line 3 = FI job error code	(see chapter entitled "Error Codes")

Example MFD1

Load message texts into the device with device address 00.

FI command		**00_BW_MFD1**
Line	**Column**	**Answer**
1	1	01
2	1	00_BW_MFD1
3	1	0

5.87 Maximum Feedrate Override: MFO

MWCX device group

Designation **MFO** **M**aximal **F**eedrate **O**verride

Explanation

The value of the maximum feedrate override for the selected device of the MWCX device group is read out.

FI command

CR_MFO1_(1)	**(Single Read)**
CC_MFO1_(1)	**(Cyclic Read)**
CB_MFO1_(1)	**(Break Cyclic Read)**
(1) = NC process number	[0...6]

Response Structure

The following table shows the general structure of the response to the "MFO1" FI command. One line with three columns is output for the identifier, the current value of the maximum feedrate override and the unit [%].

Line 1	**Column 1**	**....**	**Column 3**

Value Range/Meaning of Columns

1 = Identifier	[MAX]
2 = Value of maximum feedrate override	[0...100]
3 = Unit	[%]

Example MFO1

Read the current value of the maximum feedrate override in NC process 0 of device address 00.

FI command	**00_CR_MFO1_0**		
Answer			
Line	**Column 1**	**Column 2**	**Column 3**
1	MAX	100	[%]

Reference to Literature

See chapter entitled "Literature" [9].

5.88 Maximum Feedrate: MFR

MWCX device group

Designation **MFR** **M**aximal **F**eed **R**ate

Explanation The value of the maximum feedrate of an NC process is output.

FI command Output the value of the maximum feedrate.

Using the optional second parameter it is possible to pre-select conversion of the result into mm or inches.

CR_MFR_(1){_(2)}	**(Single Read)**
CC_MFR_(1){_(2)}	**(Cyclic Read)**
CB_MFR_(1){_(2)}	**(Break Cyclic Read)**
(1) = NC process number	[0...6]
(2) = Required measurement system (opt.)	[mm, inch]

Response Structure The following table shows the general structure of the response to the FI command "MFR". One line with three columns is output for the identifier, the current value of the maximum feedrate and the unit.

Line 1	Column 1	....	Column 3

Value Range/Meaning of Columns

1 = Identifier	[F = feedrate]
2 = Feedrate overrides	[format, according to settings of the parameters,--]
3 = Unit	[according to settings of the parameters]

Example MFR Read the value of the maximum feedrate in NC process 0 of device address 00.

FI command	00_CR_MFR_0		
Answer			
Line	Column 1	Column 2	Column 3
1	F	120000.0	[mm/min]

Example MFR Read the value of the maximum feedrate in NC process 0 of device address 00. The displayed value is to be converted into inch/min:

FI command	00_CR_MFR_0_inch		
Answer			
Line	Column 1	Column 2	Column 3
1	F	4724.4	[inch/min]

Reference to Literature See chapter entitled "Literature" [18].

5.89 Reading Machine Key Information: MKS

MWCX device group

Designation **MKS** **M**achine **K**ey **S**tatus

Explanation Current machine key information can be read for the selected device.

FI command Read machine key information for selected device.

BR_MKS	**(Single Read)**
BC_MKS	**(Cyclic Read)**
BB_MKS	**(Break Cyclic Read)**

Response Structure The following table shows the general structure of the response to the FI command "MKS".

Line 1	Column 1	Column 2

Value Range/Meaning of Columns

1 =	Machine key information	[4 byte in HEX coding]
2 =	Information valid?	[0 = not valid, 1=valid]

Example MKS Read the current machine key information for device 0.

FI command		**00_BR_MKS**
Line	**Column**	**Answer**
1	1	00000000
	2	0

5.90 Writing the GUI-SK Block: MKT

MWCX device group

Designation **MKT** **M**achine **K**ey **T**able

Explanation Writes the GUI-SK16 block in the PLC.

FI command Write GUI-SK16 block.

BW_MKT1_(1)	**(Single Write)**
(1) = List of the 48 PLC variables for writing the GUI-SK16 block.	A distinction is made between the following cases: 1. Clear GUI-SK16 block. 2. Write the GUI-SK16 block with the 48 PLC variables, filling gaps with $SPACE.

Response Structure (P_ACK) is returned following successful transmission.

Line 1	Column 1

Value Range/Meaning of the Columns

1 =	Successfully completed	(P_ACK)

1. Example MKT1

Clear GUI-SK16 block:

FI command		**Value to write: $EMPTY** **00_BW_MKT1**
Line	**Column**	**Answer**
1	1	(P_ACK)

2. Example MKT1

Write GUI-SK16 block:

FI command		**Value to write:** **SPSVAR1,SPSVAR2,$SPACE,...** **00_BW_MKT1**
Line	**Column**	**Answer**
1	1	(P_ACK)

FI command

Write the GUI-SK16 block, writing only those PLC variables which are defined in the current PLC program. All undefined PLC variables are automatically replaced by $SPACE and returned as a partial result (column 2).

BW_MKT2_(1)

(1) = List of the 48 PLC variables for writing the GUI-SK16 block.

(Single Write)

A distinction is made between the following cases:

1. Clear GUI-SK16 block: BW_MKT2 $EMPTY
2. Write the GUI-SK16 block with the 48 PLC variables, filling gaps with $SPACE: BW_MKT1 SPSVAR1,SPSVAR2, $SPACE,$SPACE,....

Response Structure

After successful transmission, one line with two columns is returned.

Line 1	**Column 1**	**Column 2**

Value Range/Meaning of Columns

1 =	Status report	[0 = at least 1 PLC variable in the current PLC program is NOT defined 1 = ALL PLC variables could be written]
2 =	List of the NON-defined PLC variables in the current PLC program	[-- = ALL PLC variables could be written, or else list of the PLC variables that could not be written.] The individual PLC variables are separated by a comma.

Example MKT1

Write GUI-SK16 component with 48 PLC variables, while the PLC variables SPSVAR11 and SPSVAR12 are NOT defined in the current PLC program.

FI command		**Value to be written:** **SPSVAR1,SPSVAR2,...SPSVAR48** **00_BW_MKT2**
Line	**Column**	**Answer**
1	1	(P_ACK)

Extended information

The variables are divided into 3 groups of 16 variables each and have the following meaning:

1. Variables 1 - 16: Machine function keys
2. Variables 17 - 32: Status pressed
3. Variables 33 - 48: Status shining

Notes: When, for example, only the first 8 M keys are used, the telegram will contain only these 8 PLC variables. The other 40 variables need not be defined in the transmission parameter.

When certain areas, e.g. of M keys, are left unused, they must be filled up with '$SPACE' up to the next variable.

5.91 Maxim Rapid Override: MRO

MWCX device group

Designation **MRO** **M**aximal **R**apid **O**verride

Explanation The value of the maximum rapid override of the selected device of the MWCX device group is read.

FI command

CR_MRO1_(1)	**(Single Read)**
CC_MRO1_(1)	**(Cyclic Read)**
CB_MRO1_(1)	**(Break Cyclic Read)**
(1) = NC process number	[0...6]

Response Structure The following table shows the general structure of the response to the "MRO1" FI command. One line with three columns is output for the identifier, the current value of the maximum rapid override and the unit [%].

Line 1	Column 1	....	Column 3

Value Range/Meaning of Columns

1 = Identifier	[RMAX]
2 = Value of maximum rapid override	[0...100]
3 = Unit	[%]

Example MRO1 Read the maximum value of the rapid override in NC process 0 of device address 00.

FI command	**00_CR_MRO1_0**		
Answer			
Line	**Column 1**	**Column 2**	**Column 3**
1	RMAX	100	[%]

Reference to Literature See chapter entitled "Literature" [18].

5.92 Read System Messages: MSG

MWCX device group

Designation **MSG** **M**e**S**sa**G**e

Explanation Reading of system messages

FI command Message

CC_MSG_(1)	**(Cyclic Read)**

(1) = SYS-Message number

Note: Exists only as a cyclic command

Response Structure

The response of the FI command 'MSG' consists of the system message data.

Example MSG

00_CC_MSG_64 (64 = MSG_SYSERRGEN)

FI command		**00_CC_MSG_64/3**
Line	**Column**	**Answer**
1	1	00

Restriction

The following system messages:

SYS Message	SYS message numbers
MSG_PCLUPDBEG	52
MSG_PARUPDBEG	24
MSG_FWAUPDBEG	82

cannot be used with the following programs:

- Bosch Rexroth OPC Server
- Bosch Rexroth DDE Server

5.93 Maximum Spindle Override: MSO

MWCX device group

Designation

MSO **M**aximal **S**pindle **O**verride

Explanation

The value of the maximum spindle override of the selected device of the MWCX device group is read.

FI command

CR_MSO1_(1)_(2)	**(Single Read)**
CC_MSO1_(1)_(2)	**(Cyclic Read)**
CB_MSO1_(1)_(2)	**(Break Cyclic Read)**
(1) = NC process number	[0...6]
(2) = Number of spindle	[1...3]

Response Structure

The following table shows the general structure of the response to the "MSO1" FI command. One line with three columns is output for the identifier, the value of the maximum spindle override and the unit [%].

Line 1	**Column 1**	**....**	**Column 3**

Value Range/Meaning of Columns

1 = Identifier	[SMAX]
2 = Value of maximum rapid override	[0...100]
3 = Unit	[%]

Example MSO1

Read the maximum value of the spindle override in NC process 0 of device address 00.

FI command	**00_CR_MSO1_0_1**		
Answer			
Line	**Column 1**	**Column 2**	**Column 3**
1	SMAX	100	[%]

Reference to Literature

See chapter entitled "Literature" [21].

5.94 Maximum Spindle Speed: MSS

MWCX device group

Designation **MSS** **M**aximal **S**pindle **S**peed

Explanation The value of the maximum spindle speed of the selected device of the MWCX device group is read out.

FI command

CR_MSS_(1)_(2)	**(Single Read)**
CC_MSS_(1)_(2)	**(Cyclic Read)**
CB_MSS_(1)_(2)	**(Break Cyclic Read)**
(1) = NC process number	[0...6]
(2) = Number of spindle	[1...3]

Response Structure The following table shows the general structure of the response to the "MSS" FI command. One line with three columns is output for the identifier, the speed and the unit [1/min].

Line 1	Column 1	....	Column 3

Value Range/Meaning of Columns

1 = Identifier	[S = spindle]
2 = Speed	[format according to settings of the parameters]
3 = Unit	1/min

Example MSS Read the maximum value of the speed of the 1st spindle in NC process 0 of device address 00.

FI command	**00_CR_MSS_0_1**		
Answer			
Line	**Column 1**	**Column 2**	**Column 3**
1	S:	7500.0	1/min

5.95 Reading the Firmware/Monitor Identification: MTC

MWCX device group

Designation **MTC** **MT-C**NC Slot Software Version

FI command This command is used to read the firmware identification from the various control components (slot numbers).

Note: For the time this FI command is executed, the internal FI communication interlocks (fast timeout monitoring, offline operation, etc.) are switched off.

FI command

BR_MTC_(1)	**(Single Read)**
BC_MTC_(1)	**(Cyclic Read)**
(1) = Slot number	[1=CNC, 2=SIO, 3=PLC, 4=APR1 5=APR2, 6=APR3, 7=APR4]

The control sends a line with 1 column for the firmware version of the desired slot. If the slot is not occupied, the control answers with N_ACK 224 (time monitoring during internal communication).

Response Structure The following table shows the general structure of the response to the FI command "MTC". A line of 1 column is output.

Line 1	Column 1

Value Range/Meaning of Columns

1 = Firmware identification string [max. 16 ASCII characters]

1st Example MTC

Read the firmware identification of slot number 1 (CPU) of device 00.

FI command		**00_BR_MTC_1**
Line	**Column**	**Answer**
1	1	VNC01/0004-23T03

2nd Example MTC

Reading of the firmware of an empty slot:

FI command		**00_BR_MTC_3**
Line	**Column**	**Answer**
1	1	1
	2	224
	3	0x00000000
	4	Time error with internal communication
	5	X

Explanation

With the FI command "BR_MTC1", the monitor versions of the various components (CNC, PLC, APR) can be read out.

FI command

BR_MTC1 **(Single Read)**

Response Structure

The response to the "BR_MTC1" FI command consists of six lines with four columns. One line is returned for each potential component (CNC, PLC, APR1-4).

Line 1.0.6	Column 1	Column 2	Column 3	Column 4

Value Range/Meaning of Columns

1 = Line number [1..6]

2 = Component information [CNC = NC component
SPS = PLC component
APR = APR component

3 = Monitor version – "old" format

4 = Monitor version – "new" format

Example MTC1 Reads the monitor versions for device 0, 1 APR being available.

FI command		**00_BR_MTC1**
Line	**Column**	**Answer**
1	1	1
	2	CNC
	3	MON-PMK 09.05/0705.02.01
	4	FWC-CONTROL-MON-06V00-NN
2	1	2
	2	PLC
	3	MON-PMK 09.05/0705.02.01
	4	FWC-CONTROL-MON-06V00-NN
3	1	3
	2	APR
	3	MON-PMK 09.05/0705.02.01
	4	FWC-CONTROL-MON-06V00-NN
4	1	4
	2	APR
	3	--
	4	--
5	1	5
	2	APR
	3	--
	4	--
6	1	6
	2	APR
	3	--
	4	--

5.96 User Machine Data: MTD

MWCX device group

Designation **MTD** **M**achine **T**able **D**ata

FI command Output of user machine data.

CR_MTD1_(1)_(2)_(3)_(4)_(5) **(Single Read)**

CC_MTD1_(1)_(2)_(3)_(4)_(5) **(Cyclic Read)**

(1) = Page number	[1...299]
(2) = Run variable 1	[-1000 ... +1000]
(3) = Run variable 2	[-1000 ... +1000]
(4) = Element number	[1...1000]
(5) = Name	[1...13]

Answer

Data element
10110100
Read from the MD page 152 via LV1: 0 and LV2: 1 the 13th element of the type UDINT CR_MTD_152_0_1_13_8 150

FI command Write machine table data.

CW_MTD1_(1)_(2)_(3)_(4)_(5) **(Single Write)**

(1) = Page number	[1...299]
(2) = Run variable 1	[-1000 ... +1000]
(3) = Run variable 2	[-1000 ... +1000]
(4) = Element number	[1...1000]
(5) = Identifier code	[1...13]

Code	Identifier	Byte	Min. value	Max. value
1	BOOL	1	0	
2	BYTE	1	0	
3	WORD	2	0	
4	DWORD;	4	0	
5	STRING	max. 220 bytes		
6	SINT	1	- 128	127
7	INT	2	- 32768	32767
8	DINT	4	- 2147483648	2147483647
9	USINT	1	0	255
10	UINT	2	0	65535
11	UDINT	4	0	4294967295
12	REAL	4	-3.4 E38	3.4 E38
13	DREAL	8	-1.7 E308	1.7 E308

Value to be written variable value [acc. to the display format of the BOF]

Note: The value to be written is passed to the "acValue" parameter in the "DataTransfer" routine.

Response Structure One line with one column is output to acknowledge the FI command issued. The meaning of the elements is as follows:

(P_ACK) = **P**ositive **ACK**nowledge Value has been successfully transmitted

5.97 NC Program Compile and Download: NCA

MWCX device group

Designation **NCA** **NC** Program **A**ccess

Explanation NC programs are downloaded via a download file and NC program files and via all active processes.

FI command NC program download.

BW_NCA1_(1) **(Single Write)**

(1) = Download file with path details.

Note: Enclose file and path details in inverted commas.

Response Structure The response to the "NCA1" FI command consists of three lines, each with one column. The meaning of the elements is as follows:

- Line 1 = Job ID [01...20]
  (see Chapter "FI Commands for the MPCX Device Group: IFJ").
- Line 2 = FI command
  [String, in accordance with Chapter "Elements of the FI Command"]
- Line 3 = FI Job Error Code
  (see Chapter "Error Codes")

....

Example NCA1 00_BW_NCA1_"D:\Program Files\Indramat\Mtgui\Temp\download.ini"/3

FI command		**00_BW_NCA1_"D:\Program Files\Indramat\Mtgui\Temp\download.ini"/3**
Line	**Column**	**Answer**
1	1	01
2	1	00_BW_NCA1_"D:\Program Files\Indramat\Mtgui\Temp\download.ini"/3
3	1	0

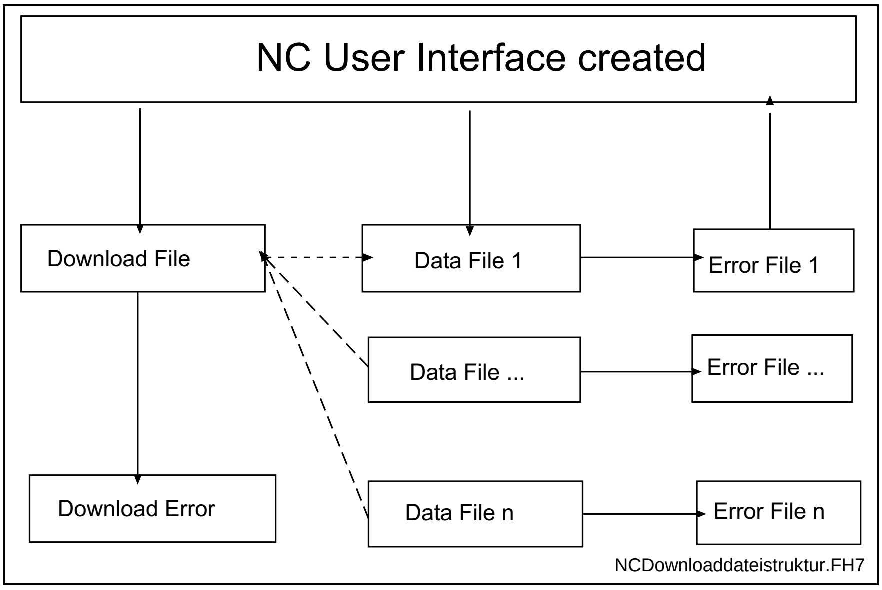


Fig. 5-5: File structure of the download file

Structure of Download File The structure of the "download.ini" file used in this example corresponds to an Ini file in Windows.

Note: Care must be taken in the use of upper and lower case letters.

Section [Common]
General information is stored in the COMMON section.

Key **Max_Compiler**
Number of compilers to be called. The compiler contains the control file as a pass parameter and translates the data into the respective data files. A pass value of zero signifies no compiler call.

This key is an optional value. If this value is not present, no compiler is active.

Key **DownloadError**
Indicates whether or not an error has occurred during downloading. This value is only set in the event of an error.

Example:

```
[Common]
DownloadError = YES   ; Error
Max_Compiler = 2
```

Section [CompilerXX]

This section contains information regarding the compiler. There is a separate section for each compiler. The name of the section consists of the "Compiler" text and a two digit number.

XX: is a two digit index which begins at 1 and has a maximum size of Max_Compiler.

Section [NCPackage_Info]

Key **Memory**
Indicates the memory into which the NC program package is loaded.
Memory=1 ;Memory A
Memory=2 ;Memory B

Key **Program package information**
The package identification is compiled from several keys. The total length of all package identifications must not exceed **a maximum of 84** characters. The length of the individual identifications is described below:

Package number	"**PackageNo**"	max. 2 characters
Package name	"**PackageName**"	max. 32 characters
Package size:	"**PackageSize**"	max. 8 characters left-justified
Package time:	"**PackageTime**"	max. 8 characters
Package date:	"**PackageDate**"	max. 8 characters
Package default:	"**PackageDefault**"	max. 26 characters (optional)
Total:		max. 84 characters

Information on date and time is given in the format
Date : dd.mm.yy
Time: hh:mm:ss

<u>Example:</u>

```
[NCPackage_Info]
Memory=
PackageNo =        1
PackageName =      NC program package
PackageSize =      1234
PackageTime =      13:10:10
PackageDate =      24.12.00
```

Section [ListOfNCPrograms]

The list of NC programs to be transferred is stored in the ListOfNCPrograms section.

Key **Max_Index_Data**
Corresponds to the number of NC programs to be transmitted.

Key **consecutive index of the NC programs**
Four-digit number starting with 1, identifies with a value the full file name of the NC programs including the setup lists. The names of the NC programs and setup lists are structured as follows:

zzzzzzz	Data type (NC-PRG or SetupList)
xx	Process number
yyy	Program number of the NC program (with free NC programs, the index number)

The file extension can be freely selected. ".dat" has been used in the following example.

<u>Examples:</u>

NC-PRG-00-86	N program for process 0 program 86
SETUPLIST-03-25	Setup list for process 3 program 25

Example:

```
[ListOfNCPrograms]
Max_Index_Data=50
0001=K:\Program Files\Indramat\Mtgui\Project_000\\NC-PRG-00-01.Dat
0002=K:\Program Files\Indramat\Mtgui\Project_000\\NC-PRG-01-01.Dat

0050=K:\Program Files\Indramat\Mtgui\Project_000\\NC-PRG-06-99.Dat
```

Data File Structure

These contain the actual data for downloading or for the compiler. The structure corresponds to that of the Windows "Ini" structure. The compiler uses this file for the input and output data.

Note: Care must be taken in the use of upper and lower case letters.

Data for the NC program is stored in the respective files as a section. It is composed of general data and the actual program.

Section [Common]

Program version:	**Version**	
Process:	**Process**	[0..6]
Program number:	**No**	[0.0.99]
Program name:	**Name**	max. 32 characters
Program size:	**Size**	
Program time:	**Time**	max. 8 characters
Program date:	**Date**	max. 8 characters
Program short identification:	**ShortID**	max. 8 characters
Program status:	**Status, (optional)**	

Information on date and time is given in the format
Date: dd.mm.yy
Time: hh:mm:ss

Status flag	Description
C	Compiled
E	Error
The marked section is then printed out.	Not compiled
No details	No compiler call

Fig. 5-6: Description of the status flags

Section Data
Key **Max_Index_Data**
Corresponds to the number of NC blocks to be transferred.

Key **consecutive index of NC records**
Five-digit number starting with 1.

Note: An NC block should not contain any unnecessary blank spaces or NC comments. Equally, "PROGRAM END" may not occur as it is language-dependent.

Example:

```
[Data]
Max_Index_Data=25
00001=N0000 G0 X0 Y0 Z0
...
00025=N0024 .Start
```

Explanation

This FI command merely compiles NC programs without triggering the subsequent download. Compiling of NC programs is done through an administration file and NC program files.

FI command

NC program compile.

BW_NCA3_(1) **(Single Write)**

(1) = Administration file with path details.

Note: Enclose file and path details in inverted commas.

Response Structure

The response to the NCA3 FI command consists of three lines, each with one column. The meaning of the elements is as follows:

- Line 1 = Job ID [01...20]
  (see Chapter "FI Commands for the MPCX Device Group", IFJ).
- Line 2 = FI command
  [String, in accordance with Chapter "Elements of the FI Command"]
- Line 3 = FI Job Error Code
  (see Chapter "Error Codes")

....

Example NVA

00_BW_NCA3_"D:\Program Files\Indramat\Mtgui\Temp\compile.ini"/3

FI command		**00_BW_NCA3_"D:\Program Files\Indramat\Mtgui\Temp\compile.ini"/3**
Line	**Column**	**Answer**
1	1	01
2	1	00_BW_NCA3_"D:\Program Files\Indramat\Mtgui\Temp\compile.ini"/3
3	1	0

5.98 NC Program Upload: NCA

MWCX device group

Designation **NCA** **NC**-Program **A**ccess

Explanation

NC programs are uploaded via all active processes; during upload, a basic file (upload file) and NC program files are created.

FI command

NC-Program upload.

BR_NCA1_(1)_(2) **(Single Read)**

(1) = NC memory [1 = NC memory A,
2 = NC memory B]

(2) = Upload file with path details

Note: Enclose file and path details in inverted commas.

In this command, the progress information is implemented in %. It can be interrogated via the command IFJ of the MPCX device group.

Response Structure

The response to the "NCA1" FI command consists of three lines, each with one column. The meaning of the elements is as follows:

- Line 1 = Job ID [01...20]
  (see Chapter "FI Commands for the MPCX Device Group", IFJ).
- Line 2 = FI command
  [String, in accordance with Chapter "Elements of the FI Command"]
- Line 3 = FI Job Error Code
  (see Chapter "Error Codes")

Example NCA

00_BR_NCA1_1_"D:\Program Files\Indramat\Mtgui\Temp\Upload.ini"/3

FI command		**00_BR_NCA1_"D:\Program Files\Indramat\Mtgui\ Temp\upload.ini"/3**
Line	**Column**	**Answer**
1	1	01
2	1	00_BR_NCA1_"D:\Program Files\Indramat\Mtgui\ Temp\upload.ini"/3
3	1	0

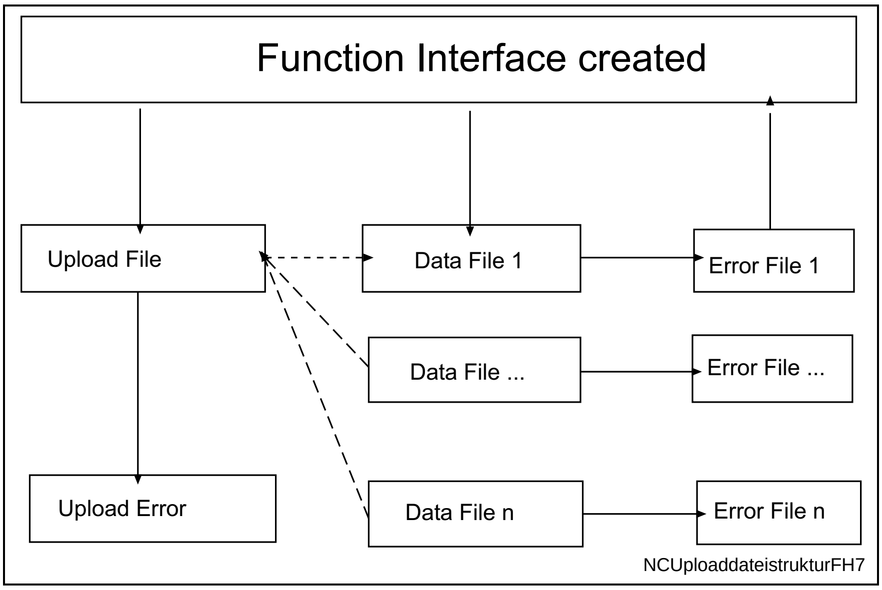


Fig. 5-7: File structure of the upload file

Structure of Upload File

The upload file is structured in the Windows – "Ini" format structure.

Note: Care must be taken in the use of upper and lower case letters.

Section [Common]
General information is stored in the COMMON section.

Key **UploadError**
Indicates whether or not an error has occurred during uploading. This value is only set in the event of an error.

Example:

```
[Common]
UploadError = YES       ; error
```

Section NC Program package information [NCPackage_Info]
Key **Memory**
Identifies the memory into which the NC program package is loaded.

Memory=1 ;Memory A
Memory=2 ;Memory B

Key **Program package information**
The package identification is compiled from several keys. The total length of all package identifications must not exceed **a maximum of 84** characters. The length of the individual identifications is described below:

Package number"**PackageNo**"	max. 2 characters
Package name "**PackageName**"	max. 32 characters
Package size: "**PackageSize**"	max. 8 characters left-justified
Package time: "**PackageTime**"	max. 8 characters
Package date: "**PackageDate**"	max. 8 characters
Package default:"**PackageDefault**"	max. 26 characters (optional)
--	--
Total:	max. 84 characters

Information on date and time is given in the format

Date :	dd.mm.yy
Time:	hh:mm:ss

Example:

```
[NCPackage_Info]
Memory=
PackageNo =         1
PackageName =       NC program package
PackageSize =       1234
PackageTime =       13:10:10
PackageDate =       24.12.00
```

Section list of NC programs [ListOfNCPrograms]
The list of the NC programs to be transferred is stored in the section "ListOfCycPrograms".

Key **Max_Index_Data**
Corresponds to the number of NC programs to be transmitted.

Key **consecutive index of the NC programs**
Four-digit number starting with 1, identifies with a value the full file name of the NC programs including the setup lists. The names of the NC programs and setup lists are structured as follows:

zzzzzzz	Data type (NC-PRG or SetupList)
xx	Process number
yyy	Program number of the NC program (with free NC programs, the index number)

The file extension can be freely selected. ".dat" has been used in the following example.

Examples:

NC-PRG-00-086	NC program for process 0 program 86
SETUPLIST-03-025	Setup list for process 3 program 25

Example:

```
[ListOfNCPrograms]
Max_Index_Data=50
```

```
0001=K:\Program Files\Indramat\Mtgui\Project_000\NC-PRG-00-001.dat
0002=K:\Program Files\Indramat\Mtgui\Project_000\NC-PRG-01-001.dat
...
0050=K:\Program Files\Indramat\Mtgui\Project_000\NC-PRG-06-099.dat
```

Data File Structure

Contains the actual data for the upload. Their structure corresponds to the Windows "Ini" structure.

Note: Care must be taken in the use of upper and lower case letters.

Data for the NC program is stored in the respective files as a section. It is composed of general data and the actual program.

Section [Common]

Program version:	**Version**	
Process:	**Process**	[0..6]
Program number:	**No**	[0.0.99]
Program name:	**Name**	max. 32 characters
Program size:	**Size**	
Program time:	**Time**	max. 8 characters
Program date:	**Date**	max. 8 characters
Program short identification:	**ShortID**	max. 8 characters
Program status:	**Status, (always 'N')**	

Information on date and time is given in the format

Date :	dd.mm.yy
Time:	hh:mm:ss

Status flag	**Description**
C	Compiled
E	Error
The marked section is then printed out.	Not compiled
No details	No compiler call

Fig. 5-8: Status flags

Section [Data]

Key **Max_Index_Data**
Corresponds to the number of NC blocks to be transmitted

Key **consecutive index of NC records**
Five-digit number starting with 1.

<u>Example:</u>

```
[Data]
Max_Index_Data=25
00001=N0000 G0 X0 Y0 Z0
...
00025=N0024 .Start
```

5.99 NC Messages: NCM

MWCX device group

Designation **NCM** **NC M**essages

Explanation Bosch Rexroth NC messages are output. These messages are assigned to a specific module and message type.

FI command Output all NC messages.

For reasons of optimization, not all data is called up. Accordingly, the Diagnosis server must be informed that the data is required (see ADW).

BR_NCM1_(1)_(2)	**(Single Read)**
BC_NCM1_(1)_(2)	**(Cyclic Read)**
(1) = Message type	[1 = error, 2 = messages]
(2) = Module number	[1...99]

Output of first NC message.

BR_NCM2_(1)_(2)	**(Single Read)**
BC_NCM2_(1)_(2)	**(Cyclic Read)**
(1) = Message type	[1 = error, 2 = messages]
(2) = Module number	[1...99]

Response Structure The following table shows the general structure of the FI commands "NCM1" and "NCM2". The number of lines depends on the number of messages pending.

If there are no messages, the number of lines is 0.

Line 1...n	Column 1	...	Column 8

Meaning of the Columns

1 = Message text	[ASCII characters]
2 = Message number	[ASCII characters]
3 = Time stamp day	[mm.dd.yyyy]
4 = Time stamp time	[hh:mm:ss]
5 = Mechanism number	[0..31]
6 = 2 byte additional information	[ASCII characters]
7 = NC note	[ASCII characters]
8 = Reference text exists	[YES, NO]
9 = Filename for additional information for message text	[e.g.HTML format]

Example NCM1 All NC errors from module 3 in control unit 0.

There are two messages.

FI command		00_BR_NCM1_1_3
Line	**Column**	**Answer**
1	1	24 volt supply absent
	2	12
	3	01.27.2000
	4	14:56:32
	5	0
	6	
	7	[Note 1]
	8	YES
	9	
2	1	Program stop
	2	152
	3	01.27.2000
	4	15:03:10
	5	1
	6	
	7	
	8	NO
	9	

Example NCM2 The first NC errors from module 3 in control unit 0.

There are two messages:

FI command		00_BR_NCM2_1_3
Line	**Column**	**Answer**
1	1	24 volt supply absent
	2	12
	3	01.27.2000
	4	14:56:32
	5	0
	6	
	7	[Note 1]
	8	YES
	9	

5.100 NC Events Download: NEA

MWCX device group

Designation **NEA** **N**C-**E**vent **A**ccess

Explanation NC events are downloaded by means of the download file via all processes.

FI command Download NC events.

BW_NEA1_(1) **(Single Write)**

(1) = Download file with path details.

Note: Enclose file and path details in inverted commas.

Response Structure The response to the "NEA1" FI command consists of three lines, each with one column. The meaning of the elements is as follows:

- Line 1 = Job ID
  [01...20] (see Chapter "FI Commands for the MPCX Device Group: IFJ").
- Line 2 = FI command
  [String, in accordance with Chapter "Elements of the FI Command"]
- Line 3 = FI Job Error Code
  (see Chapter "Error Codes")

Example NEA1 00_BW_NEA1_"D:\Program Files\Indramat\Mtgui\Temp\download.ini"/3

FI command		**00_BW_NEA1_"D:\Program Files\Indramat\Mtgui\Temp\download.ini"/3**
Line	**Column**	**Answer**
1	1	01
2	1	00_BW_NEA1_"D:\Program Files\Indramat\Mtgui\Temp\download.ini"/3
3	1	0

Structure of the download file The structure of the "download.ini" file used in this example corresponds to an Ini file in Windows.

Note: Care must be taken in the use of upper and lower case letters.

Section [Common]
This is currently only used for error processing, i.e., if an error is detected during a process, then the *DownloadError* key is written with "YES" within this section.

Example:

```
[Common]
DownloadError = YES   ; error
```

Section NC events information [NCEventsPackage_Info]
The package identification is compiled from several keys. The total length of all package identifications must not exceed **a maximum of 84** characters. The length of the individual identifications is described below:

Package number"**PackageNo**"	max. 2 characters
Package name "**PackageName**"	max. 32 characters
Package size: "**PackageSize**"	max. 8 characters left-justified
Package time: "**PackageTime**"	max. 8 characters
Package date: "**PackageDate**"	max. 8 characters
Package default:"**PackageDefault**"	max. 26 characters (optional)
--	------------------------------------
Total:	max. 84 characters

Information on date and time is given in the format
Date : dd.mm.yy
Time: hh:mm:ss

Example:

```
[NCEventsPackage_Info]
PackageNo =         1
PackageName =       NC events
PackageSize =       1234
PackageTime =       13:10:10
PackageDate =       24.12.00
```

Section NC events download [NCEvents_A]
A: corresponds to a process number [0..6]

For external events the section name is extended with X in place of the process number.

A section entry ([NCEvents_A]) is an optional entry, i.e., if a section for a process is absent it is not regarded as an error.

Key values correspond to the event numbers [0..31] and values are the write values of the NC events. Missing key values are not regarded as errors.

```
[NCEvents_0]
000=0
001=1
...
031=1

[NCEvents_1]
000=1
...
016=1

[NCEvents_6]
000=1
010=0
031=1

[NCEvents_X]
000=1
010=0
031=1
```

5.101 NC Events Upload: NEA

MWCX device group

Designation **NEA** NC-**E**vent **A**ccess

Explanation NC events are uploaded through all processes and external events.

FI command Upload NC events.

BR_NEA1_(1) **(Single Read)**

(1) = Upload file with path details

Note: Enclose file and path details in inverted commas.

In this command, the progress information is implemented in %. It can be interrogated via the command IFJ of the MPCX device group.

Response Structure The response to the NEA1 FI command consists of three lines, each with one column. The meaning of the elements is as follows:

- Line 1 = Job ID [01...20]
 (see Chapter "FI Commands for the MPCX Device Group", IFJ).
- Line 2 = FI command
 [String, in accordance with Chapter "Elements of the FI Command"]
- Line 3 = FI Job Error Code
 (see Chapter "Error Codes")

Example NVA 00_BR_NEA1_"D:\Program Files\Indramat\Mtgui\Temp\Upload.ini"/3

FI command		**00_BR_NEA1_"D:\Program Files\Indramat\Mtgui\Temp\upload.ini"/3**
Line	**Column**	**Answer**
1	1	01
2	1	00_BR_NEA1_"D:\Program Files\Indramat\Mtgui\Temp\upload.ini"/3
3	1	0

Structure of Upload File The upload file is structured in the Windows – "Ini" format structure.

Section [Common]
General information is stored in the COMMON section.

Key **UploadError**
Indicates whether or not an error has occurred during uploading. This value is only set in the event of an error.

Example:

[Common]
UploadError = YES ; error

Section NC Variables Information [NCEventsPackage_Info]

Key **Program package information**
The package identification is compiled from several keys. The total length of all package identifications must not exceed **a maximum of 84** characters. The length of the individual identifications is described below:

Package number"**PackageNo**"	max. 2 characters
Package name "**PackageName**"	max. 32 characters
Package size: "**PackageSize**"	max. 8 characters left-justified
Package time: "**PackageTime**"	max. 8 characters
Package date: "**PackageDate**"	max. 8 characters
Package default:"**PackageDefault**"	max. 26 characters (optional)
--	
Total:	max. 84 characters

Information on date and time is given in the format
Date : dd.mm.yy
Time: hh:mm:ss

Example:

[NCEventsPackage_Info]
PackageNo = 1
PackageName = NC events
PackageSize = 1234
PackageTime = 13:10:10
PackageDate = 24.12.00

Section NC variables download [NCEvents_A]

A: corresponds to a process number [0..6]

For external events the section name is extended with "X" in place of the process number.

Key values correspond to the variable numbers [0..31] and values are the NC events values.

[NCEvents_0]
000=0
001=1
...
031=0

[NCEvents_1]
000=1
...
031=0

[NCEvents_6]
000=1
...
031=1

[NCEvents_X]
000=1
...
031=1

5.102 Status of NC Events: NEV

MWCX device group

Designation **NEV** **N**C-**EV**ent

FI command Read the status of an NC event of the selected device of the MWCX device group.

CR_NEV_(1){_(2)}	**(Single Read)**
(1) = NC process number	[0...6]
(2) = Number of the NC event	[0...31] ! Optional !

Note: If the optional parameter is not specified then the status of all NC events is output.

Response Structure One line is output, whereby the number of columns depends on the number of event statuses requested. When the optional parameter has not been specified, the response consists of one line with 32 columns. If the optional parameter has been specified then the answer consists of one line with one column which contains the status [0] or [1] of the requested NC event.

Example NEV Read the status of the 17th NC event in NC process 0 of device address 00.

FI command		**00_CR_NEV_0_17**
Line	**Column**	**Answer**
1	1	0

FI command Write the status of an NC event of the selected device of the MWCX device group.

CW_NEV_(1)_(2)	**(Single Write)**
(1) = NC process number	[0...6]
(2) = Event number	[0...31]

Value to be written Status of NC Event 0 = delete NC event; 1 = set NC event

Response Structure One line is output with a column for acknowledgement of whether or not the FI command has been executed successfully.

(P_ACK) = **P**ositive **ACK**nowledge	NC event has been deleted or set.

Example NEV Set the 17th NC event in NC process 0 at device address 00.

FI command		**Value to write: 1** **00_CW_NEV_0_17**
Line	**Column**	**Answer**
1	1	(P_ACK)

Reference to Literature See chapter entitled "Literature" [7].

5.103 Selection of NC Memory: NMM

MWCX device group

Designation **NMM** **N**C-**M**e**M**ory

Explanation Used in selecting the NC memory for processing the NC program. The NC programs are managed on the NC in two NC memories. During the processing of an NC program, for instance in NC memory A, another NC program package can be transmitted into NC memory B. Both NC memories (A and B) are identically structured and completely equal; however, only one NC memory can ever be active at any given time.

FI command **CW_NMM** **(Single Write)**

Value to be written NC memory [1 = memory A; 2 = memory B]

Note: It is only possible to select an NC memory when the NC is ready for operation or is in the starting position. Otherwise, the request is acknowledged by an error message. The value to be written is passed to the "acValue" parameter in the "DataTransfer" routine.

Response Structure One line with one column is output to acknowledge the FI command issued. The meaning of the elements is as follows:

(P_ACK) = **P**ositive **ACK**nowledge	The selected NC memory has been selected.

Example NMM Select NC memory B at device 00 for processing the NC program.

FI command		**Value to write: 2** **00_CW_NMM**
Line	**Column**	**Answer**
1	1	(P_ACK)

Reference to Literature See chapter entitled "Literature" [37].

5.104 Reading NC Parameters: NPA

MWCX device group

Designation **NPA** N**C**-**PA**rameter

FI command Read a parameter line.

BR_NPA1_(1)_(2)	**(Single Read)**
(1) = Parameter record number	[1..99]
(2) = Parameter number	[A00.000..Cxx.120]

Response Structure The following table shows the general structure of the response to the FI command "NPA1". One line is output with 3 columns for the identifier, the value and the name respectively.

Line 1	Column 1	Column 2	Column 3

Value Range/Meaning of Columns

1 = Identifier	Parameter ID [max. 32 ASCII characters].
2 = Value	[ASCII text]
3 = Name	[unit, related to the value or empty]

Example NPA1 Return the parameter line from parameter record 10 with parameter number B00.007.

Assumption:
Parameter record 10 has been created and process 00 has been defined. In this place, the following information is located:
Max. path acceleration 75 mm/sec^2.

FI command		**00_BR_NPA1_10_B00.007**
Line	**Column**	**Answer**
1	1	B00.007
	2	75
	3	mm/sec^2

FI command Read out several parameter lines from a parameter record.

BR_NPA2_(1)_(2)_(3)	**(Single Read)**
(1) = Parameter record number	[1..99]
(2) = Parameter number [from]	[A00.000..Cxx.120]
(3) = Parameter number [to]	[A00.000..Cxx.120]

Response Structure The following table shows the general structure of the response to the FI command "NPA2". As many lines as are requested are output, each with three columns for the identifier, the value and the name respectively.

Line 1...n:	Column 1	...	Column 3

Value Range/Meaning of Columns

1 = Identifier	[max. 32 ASCII characters]
2 = Value	[ASCII text]
3 = Name	[unit, related to the value or empty]

Example NPA2 Return the parameter lines from parameter record 10 of parameter number A00.000 to parameter number A00.001.

Assumption:
Parameter record 10 has been created and contains the following information in this location:

FI command		**00_BR_NPA2_10_A00.000_A00.001**
Line	**Column**	**Answer**
1	1	A00.000
	2	Master
	3	
2	1	A00.001
	2	Process 1
	3	

FI command Read a particular element of a parameter line.

Note: Command NPA3 is not supported any more from FI Version 06!

BR_NPA3_(1)_(2)_(3)	**(Single Read)**
(1) = Parameter record number	[1.99]
(2) = Parameter number	[A00.000..Cxx.120]
(3) = Element number	[1..1000]

Response Structure The following table shows the general structure of the response to the FI command "NPA3". One line is output with one column for either the name or value or designated name.

Line 1	**Column 1**

Value Range/Meaning of Columns

1 = Name/value/designated name	[ASCII-Text]

Example NPA3 Return element 1 of the parameter line from parameter record 10 with parameter number C01.079.

Assumption:
The parameter record has been created and contains the following information in this location:

FI command		**00_BR_NPA3_10_C01.079_19**
Line	**Column**	**Answer**
1	1	Required value (here 19) from existing compensation table of axis 1.

FI command Read all elements from a parameter line (such as "NPA1").

Note: Command "NPA4" is not supported any more from FI Version 06!

BR_NPA4_(1)_(2)	**(Single Read)**
(1) = Parameter record number	[1..99]
(2) = Parameter number	[A00.000..Cxx.120]

Response Structure The following table shows the general structure of the response to the FI command "NPA4". One line is output with 3 columns for the identifier, the value and the name respectively.

Line 1	**Column 1**	**Column 2**	**Column 3**

Value Range/Meaning of Columns

1 = Identifier	[max. 32 ASCII characters]
2 = Value	[ASCII text]
3 = Name	[unit, related to the value]

Example NPA4

Return the parameter line from parameter record 10 with parameter number A00.000.

Assumption:
The parameter record has been created and contains the following information in this location: Master.

FI command		**00_BR_NPA4_10_A00.000**
Line	**Column**	**Answer**
1	1	A00.000
	2	Master
	3	

Note: The commands supported in this version are listed using the command "00_NPA1_?".

Explanation

It is possible to read a list with a maximum of 10 parameters of the same type (system parameters, process parameters or axis parameters).

FI command

Read NC parameters for a selected device.

BR_NPA5_(1)_(2)_{(3)..(12)}	**(Single Read)**
(1) = Parameter type	1 = System parameter 2 = Process parameter 3 = Axis parameter
(2) = Process number or axis number	If "system parameter" has been selected as the type of parameter, then this parameter is NOT evaluated – set to 0.
(3).....(12) = List of requested parameters	A maximum of 10 parameters of the same type may be listed here. Please take the parameter number from the general description of parameters for the control unit.

Response Structure

The following table shows the general structure of the response to the FI command "NPA5".

Line 1...n	**Column 1**	**...**	**Column 3**

Value Range/Meaning of Columns

1 =	Parameter number	Parameter number that has been requested.
2 =	Parameter value	Data setup – see general description of parameters.
3 =	Parameter unit	Data setup – see general description of parameters.

Example NPA5 NC parameter request for system parameters 0,52,53.

FI command		00_BR_NPA5_1_0_0_52_53
Line	Column	Answer
1	1	0
	2	Master
	3	--
2	1	52
	2	0
	3	--
3	1	53
	2	1
	3	--

Reference to Literature See chapter entitled "Literature" [38].

5.105 Activate NC Compiler: NPC

MWCX device group

Designation **NPC** **N**C-**P**ackage **C**ompiling

FI command Compiles the selected NC package.

BR_NPC1_(1)	**(Single Read)**
(1) = Number in NC package directory	[1...99]

Response Structure The following table shows the general construction of the answer of the FI command NPC1. A line with three columns for job ID, FI command and the FI job ErrorCode is output.

Line 1	Column 1	...	Column 3

Value Range/Meaning of Columns

1 = Job ID	[01...20] (refer to chapter entitled "FI Commands for the MPCX Device Group", IFJ).
2 = FI command	[string, in accordance to chapter entitled "Elements of the FI Command"]
3 = FI job error code	(see chapter entitled "Error Codes")

Example NPC Compile the 2nd NC package.

FI command	00_BR_NPC1_2		
Answer			
Line	Column 1	Column 2	Column 3
1	01	00_BR_NPC1_2	0

5.106 Activate NC Download: NPD

MWCX device group

Designation **NPD** **N**C-**P**ackage **D**ownload

FI command

Downloads the selected NC package into the identified device **without the setup lists**.

BW_NPD1_(1)_(2)	**(Single Write)**
(1) = NC memory	[1 = NC memory A, 2 = NC memory B]
(2) = Number in NC package directory	[1...99]

Value to be written Initialization 1 = Trigger NC download

Note: The value to be written is passed to the "acValue" parameter as an ASCII value in the "DataTransfer" routine.

Response Structure

The answer of the FI command NPD1 consists of three lines, each with one column. The meaning of the elements is as follows:

Line 1 = Job ID	[01...20] (refer to chapter entitled "FI Commands for the MPCX Device Group", IFJ).
Line 2 = FI command	[string, in accordance to chapter entitled "Elements of the FI Command"]
Line 3 = FI job error code	(see chapter entitled "Error Codes")

Example NPA1

Load the 2nd NC package (**without setup lists)** into the NC memory A of the device with device address 00.

FI command		**Value to be written: 1** **00_BW_NPD1_1_2**
Line	**Column**	**Answer**
1	1	02
2	1	00_BW_NPD1_1_2
3	1	0

Note: If an attempt is made to transfer once again an NC package which is already in the device, the "DataTransfer" routine will terminate the process with error code 1030 (see chapter entitled "Error Codes").

FI command

Downloads the selected NC package into the identified device **with the setup lists**.

BW_NPD2_(1)_(2)	**(Single Write)**
(1) = NC memory	[1= NC memory A, 2= NC memory B]
(2) = Number in NC package directory	[1...99]

Value to be written Initialization 1 = Trigger NC download

Note: The value to be written is passed to the "acValue" parameter as an ASCII value in the "DataTransfer" routine.

Response Structure

The answer of the FI command NPD2 consists of three lines, each with one column. The meaning of the elements is as follows:

Line 1 = Job ID	[01...20] (refer to chapter entitled "FI Commands for the MPCX Device Group", IFJ).
Line 2 = FI command	[string, in accordance to chapter entitled "Elements of the FI Command"]
Line 3 = FI job error code	(see chapter entitled "Error Codes")

Example NPD2

Load the 3rd NC package (**with setup lists)** into the NC memory B of the device with device address 00.

FI command		**Value to be written: 1** **00_BW_NPD2_2_3**
Line	**Column**	**Answer**
1	1	03
2	1	00_BW_NPD2_2_3
3	1	0

Note: If an attempt is made to transfer once again an NC package which is already in the device, the "DataTransfer" routine will terminate the process with error code 1030 (see chapter entitled "Error Codes").

Notes on NP3 and NP4

These FI command have been speed-optimized. They are suited especially well for the transmission of small NC programs (guide value: up to a maximum of 100 NC program lines). As the transmission of small NC programs takes less than two seconds, a status query does not make much sense. Therefore, the function interface job administration was left out with these FI commands (see chapter entitled "FI Command for the MPCX Device Group", IFJ).

Note: The "DataTransfer" routine remains for all the transmission period (remain period = transmission period). This is only valid for these FI commands.

FI command

Downloads the selected NC package into the identified device **without the setup lists**.

BW_NPD3_(1)_(2)	**(Single Write)**
(1) = NC memory	[1 = NC memory A, 2 = NC memory B]
(2) = Number in NC package directory	[1...99]

Value to be written

Initialization 1 = Trigger NC download

Note: The value to be written is passed to the "acValue" parameter as an ASCII value in the "DataTransfer" routine.

Rexroth
Bosch Group

Response Structure

The answer of the FI command NPD3 consists of three lines, each with one column. The meaning of the elements is as follows:

Line 1 = Job ID	[01...20] (refer to chapter entitled "FI Commands for the MPCX Device Group", IFJ).
Line 2 = FI command	[string, in accordance to chapter entitled "Elements of the FI Command"]
Line 3 = FI job error code	(see chapter entitled "Error Codes")

Example NPA3

Load the 2nd NC package (**without setup lists)** into the NC memory A of the device with device address 00.

FI command		**Value to be written: 1** **00_BW_NPD3_1_2**
Line	**Column**	**Answer**
1	1	03
2	1	00_BW_NPD3_1_2
3	1	0

Note: If an attempt is made to transfer once again an NC package which is already in the device, the "DataTransfer" routine will terminate the process with error code 1030 (see chapter entitled "Error Codes").

FI command

Downloads the selected NC package into the identified device **with the setup lists**.

BW_NPD4_(1)_(2)	**(Single Write)**
(1) = NC memory	[1 = NC memory A, 2 = NC memory B]
(2) = Number in NC package directory	[1...99]

Value to be written

Initialization 1 = Trigger NC download

Note: The value to be written is passed to the "acValue" parameter as an ASCII value in the "DataTransfer" routine.

Response Structure

The answer of the FI command NPD4 consists of three lines, each with one column. The meaning of the elements is as follows:

Line 1 = Job ID	[01...20] (refer to chapter entitled "FI Commands for the MPCX Device Group", IFJ).
Line 2 = FI command	[string, in accordance to chapter entitled "Elements of the FI Command"]
Line 3 = FI job error code	(see chapter entitled "Error Codes")

Example NPA4

Load the 3rd NC package (**with setup lists)** into the NC memory B of the device with device address 00.

FI command		**Value to be written: 1** **00_BW_NPD4_2_3**
Line	**Column**	**Answer**
1	1	03
2	1	00_BW_NPD4_2_3
3	1	0

Note:	If an attempt is made to transfer once again an NC package which is already in the device, the "DataTransfer" routine will terminate the process with error code 1030 (see chapter entitled "Error Codes").

5.107 Read NC Package Directory: NPI

MWCX device group

Designation **NPI** **N**C-**P**ackage D**I**rectory

Explanation Reads the entries of the NC package directories.

FI command **BR_NPI** **(Single Read)**

Response Structure The following table shows the general structure of the response to the FI command NPI. The response consists of up to a maximum of n=99 lines, each with 5 columns.

Line 1...n:	**Column 1**	**...**	**Column 5**

Value Range/Meaning of Columns

1 =	Number in NC package directory	[01...99]
2 =	Name of the NC package	[max. 32 ASCII characters]
3 =	Length of the NC package	[byte]
4 =	Date of creation/last change to NC package	[DD.MM.YY]
5 =	Time of creation/last change to NC package	[HH:MM:SS]

Example NPI Read the entries in the NC package directory at device address 00.

FI command		**00_BR_NPI**
Line	**Column**	**Answer**
1	1	01
	2	KEY1
	3	3579
	4	16.05.99
	5	10:41:08
2	1	10
	2	KEY2
	3	4589
	4	18.05.99
	5	10:12:10

Reference to Literature See chapter entitled "Literature" [31].

5.108 Selection of the NC Program in the Active NC Memory: NPS

MWCX device group

Designation **NPS** **N**C-**P**rogram **S**election

Explanation Used in selecting the NC program located for processing in the active NC memory. The NC programs are managed on the NC in two NC memories. During the processing of an NC program, for instance in NC memory A, another NC program package can be transmitted into NC memory B. Both NC memories (A and B) are identically structured and completely equal; however, only one NC memory can ever be active at any given time.

FI command **CW_NPS_(1)** **(Single Write)**

(1) = NC process number [0...6]

Value to be written Number in NC package directory [1...99]

Note: It is only possible to select an NC program when there is a valid NC program package in the active NC memory. Otherwise, the request is acknowledged by an error message. The value to be written is passed to the "acValue" parameter as an ASCII value in the "DataTransfer" routine.

Response Structure One line with one column is output to acknowledge the FI command issued. The meaning of the elements is as follows:

(P_ACK) = **P**ositive **ACK**nowledge	The selected NC program has been selected.

Example NPS Select NC process number 0 for processing NC program 01 in the active NC memory.

Assumption:
There is a valid NC program package in the active NC memory.

FI command		**Value to write: 1** **00_CW_NPS_0**
Line	**Column**	**Answer**
1	1	(P_ACK)

Reference to Literature See chapter entitled "Literature" [37].

5.109 Next Tool Number: NTN

MWCX device group

Designation **NTN** **N**ext **T**ool-**N**umber

Explanation Returns the next pre-selected tool number of the selected device of the MWCX device group.

FI command

CR_NTN_(1)	**(Single Read)**
CC_NTN_(1)	**(Cyclic Read)**
CB_NTN_(1)	**(Break Cyclic Read)**
(1) = NC process number	[0...6]

Response Structure One line with two columns is output for the identifier [T= Tool] and for the next tool number.

Example NTN Read the next tool number in NC process 0 of device address 00.

FI command		**00_CR_NTN_0**
Line	**Column**	**Answer**
1	1	T
	2	1

Reference to Literature See chapter entitled "Literature" [43].

5.110 NC Zero Point Download: NUA

MWCX device group

Designation **NUA** **N**C-Offset Data **A**ccess

Explanation Zero points are downloaded by means of the download file via all active processes.

FI command Download NC zero points.

BW_NUA1_(1) **(Single Write)**

(1) = Download file with path details.

Note: Enclose file and path details in inverted commas.

Response Structure The response to the "NUA1" FI command consists of three lines, each with one column. The meaning of the elements is as follows:

- Line 1 = Job ID [01...20]
  (see Chapter "FI Commands for the MPCX Device Group: IFJ").
- Line 2 = FI command
  [String, in accordance with Chapter "Elements of the FI Command"]
- Line 3 = FI Job Error Code
  (see Chapter "Error Codes")

Example NUA1 00_BW_NUA1_"D:\Program Files\Indramat\Mtgui\Temp\download.ini"/3

FI command		**00_BW_NUA1_"D:\Program Files\Indramat\Mtgui\Temp\download.ini"/3**
Line	**Column**	**Answer**
1	1	01
2	1	00_BW_NUA1_"D:\Program Files\Indramat\Mtgui\Temp\download.ini"/3
3	1	0

Structure of the download file

The structure of the "download.ini" file used in this example corresponds to an Ini file in Windows.

Note: Care must be taken in the use of upper and lower case letters.

Section [Common]
This is currently only used for error processing, i.e., if an error is detected during a process, then the *DownloadError* key is written with "YES" within this section.

Example:

```
[Common]
DownloadError = YES   ; error
```

Section [OffsetDataPackage_Info]
The package identification consists of several keys; the total length of all package identifications may not exceed a **maximum of 84** characters. The length of the individual identifications is described below:

Key **Memory**
Indicates the memory into which the NC package is loaded.
Memory=1 ;Memory A
Memory=2 ;Memory B

Package number	"**PackageNo**"	max. 2 characters
Package name	"**PackageName**"	max. 32 characters
Package size:	"**PackageSize**"	max. 8 characters left-justified
Package time:	"**PackageTime**"	max. 8 characters
Package date:	"**PackageDate**"	max. 8 characters
Package default:	"**PackageDefault**"	max. 26 characters (optional)
--		
Total:		max. 84 characters

Information on date and time is given in the format
Date : dd.mm.yy
Time: hh:mm:ss

Example:

```
[OffsetDataPackage_Info]
Memory=
PackageNo =        1
PackageName =      Offset Data
PackageSize =      1234
PackageTime =      13:10:10
PackageDate =      24.12.00
```

Section Zero-point data download

Consists of several pieces of information and is structured as follows:

[OffsetData_A\Number of zero-point bank\code of axis meaning]

A:	Process number [0..6]
Number of zero-point bank:	[0..9]
Code of axis meaning: [0..8]	
	[9] angle of rotation "PHI"

A section entry is an optional entry, i.e., if a section for a process is absent it is not regarded as an error.

Key values correspond to the types of offset [3..9] and values are the write values of the types of offset in the base unit. Missing key values are not regarded as errors.

Offset Type

Code	Meaning	Explanation
3	General	acts additive to all offset types
4	G54	adjustable zero offset
...		
9	G59	adjustable zero offset

Note: The axis meanings are contained in chapter "Data Tables".

```
[OffsetData_0\0\0]      ;Process 0, zero-point data bank 0, axis X
03=1.0000               ;Gen. offset
04=2.0000               ;G54
05=3.0000               ;G55
06=4.0000               ;G56
07=5.0000               ;G57
08=6.0000               ;G58
09=7.0000               ;G59

[OffsetData_0\2\3]      ;Process 0, zero-point data bank 3, axis X
03=1.0000               ;Gen. offset
04=2.0000               ;G54
05=3.0000               ;G55
06=4.0000               ;G56
07=5.0000               ;G57
08=6.0000               ;G58
09=7.0000               ;G59
```

5.111 NC Zero Point Upload: NUA

MWCX device group

Designation **NUA** **N**C-Offset Data **A**ccess

Explanation Zero-points are uploaded via all active processes.

FI command Zero-point upload.

BR_NUA1_(1)_(2) **(Single Read)**

(1) = memory [1 = memory A;
2 = memory B]

(2) = Upload file with path details

Note: Enclose file and path details in inverted commas.

In this command, the progress information is implemented in %. It can be interrogated via the command IFJ of the MPCX device group.

Response Structure The response to the NUA1 FI command consists of three lines, each with one column. The meaning of the elements is as follows:

- Line 1 = Job ID [01...20]
  (see Chapter "FI Commands for the MPCX Device Group", IFJ).
- Line 2 = FI command
  [String, in accordance with Chapter "Elements of the FI Command"]
- Line 3 = FI Job Error Code
  (see Chapter "Error Codes")

Example NUA 00_BR_NUA1_1_"D:\Program Files\Indramat\Mtgui\Temp\Upload.ini"/3

FI command		**00_BR_NUA1_1_"D:\Program Files\Indramat\Mtgui\Temp\upload.ini"/3**
Line	**Column**	**Answer**
1	1	01
2	1	00_BR_NUA1_1_"D:\Program Files\Indramat\Mtgui\Temp\upload.ini"/3
3	1	0

Structure of Upload File The upload file is structured in the Windows – "Ini" format structure.

Section [Common]
General information is stored in the COMMON section.

Key **UploadError**
Indicates whether or not an error has occurred during uploading. This value is only set in the event of an error.

Example:

[Common]
UploadError = YES ; error

Section NC variables information [OffsetDataPackage_Info]
Key **Memory**
Identifies the memory from which the NC package is loaded.
Memory=1 ;Memory A
Memory=2 ;Memory B

Key **Program package information**
The package identification is compiled from several keys. The total length of all package identifications must not exceed **a maximum of 84** characters. The length of the individual identifications is described below:

Package number"**PackageNo**"	max. 2 characters
Package name "**PackageName**"	max. 32 characters
Package size: "**PackageSize**"	max. 8 characters left-justified
Package time: "**PackageTime**"	max. 8 characters
Package date: "**PackageDate**"	max. 8 characters
Package default:"**PackageDefault**"	max. 26 characters (optional)
Total:	max. 84 characters

Information on date and time is given in the format
Date : dd.mm.yy
Time: hh:mm:ss

Example:

```
[OffsetDataPackage_Info]
Memory=
PackageNo =          1
PackageName =        Offset Data
PackageSize =        1234
PackageTime =        13:10:10
PackageDate =        24.12.00
```

Section Zero-point data download
Consists of several pieces of information and is structured as follows:

[OffsetData_A\Number of zero-point bank\code of axis meaning]
A: Process number [0..6]
Number of zero-point bank: [0..9]
Code of axis meaning: [0..8]
[9] angle of rotation "PHI"

Key values correspond to the types of offset [3..9] and values are the read values of the types of offset in the base unit.

Offset Type

Code	Meaning	Explanation
3	General	acts additive to all offset types
4	G54	adjustable zero offset
...		
9	G59	adjustable zero offset

Note: The axis meanings are contained in chapter entitled "Data Tables".

```
[OffsetData_0\0\0]    ;Process 0, zero-point data bank 0, axis X
03=1.0000             ;Gen. offset
04=2.0000             ;G54
05=3.0000             ;G55
06=4.0000             ;G56
07=5.0000             ;G57
08=6.0000             ;G58
09=7.0000             ;G59
```

```
[OffsetData_0\2\3]      ;Process 0, zero-point data bank 3, axis X
03=1.0000               ;Gen. offset
04=2.0000               ;G54
05=3.0000               ;G55
06=4.0000               ;G56
07=5.0000               ;G57
08=6.0000               ;G58
09=7.0000               ;G59
```

5.112 NC Variables Download: NVA

MWCX device group

Designation **NVA** **NC**-**V**ariable **A**ccess

Explanation NC variables are downloaded by means of the download file via all processes.

FI command Download NC variables.

BW_NVA1_(1) **(Single Write)**

(1) = Download file with path details.

Note: Enclose file and path details in inverted commas.

Response Structure The response to the "NVA1" FI command consists of three lines, each with one column. The meaning of the elements is as follows:

- Line 1 = Job ID [01...20]
  (see Chapter "FI Commands for the MPCX Device Group: IFJ").
- Line 2 = FI command
  [String, in accordance with Chapter "Elements of the FI Command"]
- Line 3 = FI Job Error Code
  (see Chapter "Error Codes")

Example NVA1 00_BW_NVA1_"D:\Program Files\Indramat\Mtgui\Temp\download.ini"/3

FI command		**00_BW_NVA1_"D:\Program Files\Indramat\Mtgui\Temp\download.ini"/3**
Line	**Column**	**Answer**
1	1	01
2	1	00_BW_NVA1_"D:\Program Files\Indramat\Mtgui\Temp\download.ini"/3
3	1	0

Structure of Download File The structure of the "download.ini" file used in this example corresponds to an Ini file in Windows.

Note: Care must be taken in the use of upper and lower case letters.

Section [Common]
This is currently only used for error processing, i.e., if an error is detected during a process, then the *DownloadError* key is written with "YES" within this section.

Example:

```
[Common]
DownloadError = YES   ; error
```

Section [NCVariablesPackage_Info]

The package identification is compiled from several keys. The total length of all package identifications must not exceed **a maximum of 84** characters. The length of the individual identifications is described below:

Package number"**PackageNo**"	max. 2 characters
Package name "**PackageName**"	max. 32 characters
Package size: "**PackageSize**"	max. 8 characters left-justified
Package time: "**PackageTime**"	max. 8 characters
Package date: "**PackageDate**"	max. 8 characters
Package default:"**PackageDefault**"	max. 26 characters (optional)
Total:	max. 84 characters

Information on date and time is given in the format

Date : dd.mm.yy

Time: hh:mm:ss

Example:

```
[NCVariablesPackage_Info]
PackageNo =          1
PackageName =        NC variables
PackageSize =        1234
PackageTime =        13:10:10
PackageDate =        24.12.00
```

Section NC variables download [NCVariables_A]

A: corresponds to a process number [0..6]

A section entry ([NCVariables_A]) is an optional entry, i.e., if a section for a process is absent it is not regarded as an error.

Key values correspond to the variable numbers [0..255] and values are the write values of the NC events. Missing key values are not regarded as errors.

```
[NCVariables_0]
000=1
001=3.14
...
255=255

[NCVariables_1]
000=1
...
100=255

[NCVariables_6]
000=1
010=3.14
255=255
```

5.113 NC Variables Upload: NVA

MWCX device group

Designation **NVA** **N**C-**V**ariable **A**ccess

Explanation NC variables are uploaded via all processes.

FI command NC variables upload.

BR_NVA1_(1) **(Single Read)**

(1) = Upload file with path details

Note: Enclose file and path details in inverted commas.

In this command, the progress information is implemented in %. It can be interrogated via the command IFJ of the MPCX device group.

Response Structure The response to the NVA1 FI command consists of three lines, each with one column. The meaning of the elements is as follows:

- Line 1 = Job ID [01...20]
  (see Chapter "FI Commands for the MPCX Device Group", IFJ).
- Line 2 = FI command
  [String, in accordance with Chapter "Elements of the FI Command"]
- Line 3 = FI Job Error Code
  (see Chapter "Error Codes")

Example NVA 00_BR_NVA1_"D:\Program Files\Indramat\Mtgui\Temp\Upload.ini"/3

FI command		**00_BR_NVA1_"D:\Program Files\Indramat\Mtgui\Temp\upload.ini"/3**
Line	**Column**	**Answer**
1	1	01
2	1	00_BR_NVA1_"D:\Program Files\Indramat\Mtgui\Temp\upload.ini"/3
3	1	0

Structure of Upload File The upload file is structured in the Windows – "Ini" format structure.

Section [Common]
General information is stored in the COMMON section.

Key **UploadError**
Indicates whether or not an error has occurred during uploading. This value is only set in the event of an error.

Example:

[Common]
UploadError = YES ; error

NC variables information section [NCVariablesPackage_Info]

Key **Program package information**
The package identification is compiled from several keys. The total length of all package identifications must not exceed **a maximum of 84** characters. The length of the individual identifications is described below:

Package number"**PackageNo**"	max. 2 characters
Package name "**PackageName**"	max. 32 characters
Package size: "**PackageSize**"	max. 8 characters left-justified
Package time: "**PackageTime**"	max. 8 characters
Package date: "**PackageDate**"	max. 8 characters
Package default:"**PackageDefault**"	max. 26 characters (optional)
Total:	max. 84 characters

Information on date and time is given in the format
Date : dd.mm.yy
Time: hh:mm:ss

Example:

```
[NCVariablesPackage_Info]
PackageNo =         1
PackageName =       NC variables
PackageSize =       1234
PackageTime =       13:10:10
PackageDate =       24.12.00
```

Section NC variables download [NCVariables_A]
A: corresponds to a process number [0..6]

Key values correspond to the variable numbers [0..255] and values are the NC variables values.

```
[NCVariables_0]
000=1
001=3.14
...
255=255

[NCVariables_1]
000=1
...
100=255

[NCVariables_6]
000=1
010=3.14
255=255
```

5.114 Reading and Writing NC Variables: NVS

MWCX device group

Designation **NVS** **N**C-**V**ariable **S**ingle

Explanation Reads the NC variables of the selected device of the MWCX device group.

FI command

CR_NVS_(1)_(2){_(3)}	**(Single Read)**
CC_NVS_(1)_(2){_(3)}	**(Cyclic Read)**
CB_NVS_(1)_(2){_(3)}	**(Break Cyclic Read)**
(1) = NC process number	[0...6]
(2) = NC variable number {from}	[0...255]
(3) = NC variable number {to}	[0...255] !Optional !

Note: If the optional parameter is specified then up to 20 NC variables are output.

Response Structure One line with a maximum of 20 columns containing the corresponding value of the requested NC variable is output.

Note: If the requested NC variable does not exist then [--] is entered in the corresponding column.

Example NVS without optional Parameter Read the value of the NC variable numbered 1 at device address 00 in NC process 0.

FI command		**00_CR_NVS_0_1**
Line	**Column**	**Answer**
1	1	1.111000

Example NVS with optional Parameter Read the value of the 1st NC variable to the 3rd NC variable at device address 00 in NC process 0.

Assumption:
The 2nd NC variable is not defined.

FI command	**00_CR_NVS_0_1_3**		
Answer			
Line	**Column 1**	**Column 2**	**Column 3**
1	1.111000	--	23.100000

Explanation Writes an NC variable of the selected device of the MWCX device group.

FI command

CW_NVS_(1)_(2)	**(Single Write)**
(1) = NC process number	[0...6]
(2) = NC variable number	[0...255]

Value to be written NC variable [Format, long, or doublereal]

Note: The of the NC variables is set to long or doublereal in accordance with the entered format. With doublereal, the decimal point '.' must be used by all means.

Note: Only defined NC variables can be written. The value to be written is passed to the "acValue" parameter as an ASCII value in the "DataTransfer" routine.

Response Structure

One line with one column is output to acknowledge the FI command issued. The meaning of the elements is as follows:

(P_ACK) = **P**ositive **ACK**nowledge	variable has been written.

Example NVS

Write the value 1.111000 in the 1st NC variable in NC process 0 at device address 00.

FI command		**Value to be written: 1.111000** **00_CW_NVS_0_1**
Line	**Column**	**Answer**
1	1	(P_ACK)

Reference to Literature

See chapter entitled "Literature" [39].

5.115 Optimum Position Distance from Axes: OPD

MWCX device group

Designation

OPD **O**ptimal **P**osition **D**istance

Explanation

The optimum position distance of a selected axis of the MWCX device group is read out. The FI command "OPD1" returns the position distance of an axis, related to the code of the axis meaning. On the other hand, the FI command "OPD2" returns the position distance of an axis, related to the physical axis number.

FI command

Output of the optimum position distance of the selected axis of the device specified, related to the code of the axis meaning.

Using the optional third parameter it is possible to pre-select conversion of the result into mm or inches. If, however, a spindle is selected as an axis, indicating a measurement system serves no purpose.

CR_OPD1_(1)_(2){_(3)}	**(Single Read)**
CC_OPD1_(1)_(2){_(3)}	**(Cyclic Read)**
CB_OPD1_(1)_(2){_(3)}	**(Break Cyclic Read)**
(1) = NC process number	[0...6]
(2) = Axis meaning	[0...11; 20] (see Chapter 6.2, "Data Tables")
(3) = Required measurement system (opt.)	[mm, inch]

FI command

Output the optimum position distance of the selected axis of the device specified, related to the physical axis number.

Using the optional second parameter it is possible to pre-select conversion of the result into mm or inches. If, however, a spindle is selected as an axis, indicating a measurement system serves no purpose.

CR_OPD2_(1){_(2)}	**(Single Read)**
CC_OPD2_(1){_(2)}	**(Cyclic Read)**
CB_OPD2_(1){_(2)}	**(Break Cyclic Read)**
(1) = Physical axis number	[1...32]

(2) = Required measurement system [mm, inch] (opt.)

Response Structure

The following table shows the general structure of the response to the FI commands "OPD1" and "OPD2". One line with four columns is output for the name of the axis, value of the optimum position distance, the unit and the opt. position distance limited to "indicated decimal places".

Line 1	Column 1	Column 2	Column 3	Column 4

Value Range/Meaning of the Columns

1 = Axis name	[Xi, Yi, Zi, Ui, Vi, Wi, Ai, Bi, Ci, Si] with i = [,1,2,3]
2 = Optimum position distance	[acc. to settings in the process parameters]
3 = Unit	[mm, inch]
4 = Optimum position distance	[as Column 2, but rounded up or down according to the parameter "Indicated decimal places"]

Note: If the specified axis is not defined in the selected NC process then the response in all columns is [--].

Example OPD1

Read the optimum position distance of the Z axis in NC process 0 of device address 00.

FI command	**00_CR_OPD1_0_2**			
Answer				
Line	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	Z	-5.9897	[mm]	-5.990

Example OPD1

Read the optimum position distance of the Z axis in NC process 0 of device address 00. Values are displayed in inches:

FI command	**00_CR_OPD1_0_2**			
Answer				
Line	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	Z	-0.2358	[inch]	-0.236

Example OPD2

Read the optimum position distance of the Z axis (physical axis number = 3) at device address 00.

FI command	**00_CR_OPD2_3**			
Answer				
Line	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	Z	-5.9897	[mm]	-5.990

5.116 Parameter Download: PAA

MWCX device group

Designation **PAA** **PA**rameter **A**ccess

Explanation Complete parameter records are downloaded by means of a download file.

FI command Parameter download command whereby two predefined functions are to be programmed by the user. These two functions concern:

1. Function for creating the download file itself:

```
LONG ParameterDownloadBegin(    Long lProjectNumber,
                                Long lDeviceNumber,
                                Long lIndexNumber,
                                Char* pcPDLFileName,
                                Long lMaxLengthFileNameBuffer,
                                Char* pcErrorText,
                                Long lMaxLengthErrorTextBuffer)
```

Pass parameters:

lProjectNumber:	Currently selected project number
lDeviceNumber:	Currently selected device address
lIndexNumber:	Currently selected parameter directory number [1..99]
pcPDLFileName:	Contains the complete file name for the created parameter download file.
lMaxLengthFileNameBuffer:	max. length of the buffer for the name of the parameter download file.
pcErrorText:	If necessary, user error text
lMaxLengthErrorTextBuffer:	Max. length of the buffer for the user error text.

2. Function called up at the end of the parameter download:

```
Long ParameterDownloadEnd(    Char* pcPDLFileName,
                              Long lResult)
```

Pass parameters:

- pcPDLFileName:
  Contains the complete file names for the created parameter download file.
- lResult:
  Contains the status message of the parameter download procedure
  Here, 0 = Parameter download procedure O.K.
  > 0 = Error has occurred

The two functions must be programmed in a DLL by the user and also exported from it.

BW_PAA1_(1)_{(2)} **(Single Write)**

(1) = Parameter directory number; the two functions to be implemented are located in INDIF410.DLL.

(2) = Complete DLL name, if required, in which the two functions to be

implemented are located.

Response Structure

The response to the "PAA1" FI command consists of three lines, each with one column.
The meaning of the elements is as follows:

- Line 1 = Job ID [01...20]
  (see Chapter "FI Commands for the MPCX Device Group: IFJ").
- Line 2 = FI command
  [String, in accordance with Chapter "Elements of the FI Command"]
- Line 3 = FI Job Error Code
  (see Chapter "Error Codes")

Note: File and path details must be enclosed in inverted commas.

Example PAA1

00_BW_PAA1_2_"D:\UserDir\USER.DLL"

FI command		**00_BW_PAA1_2_D:\UserDir\USER.DLL**
Line	**Column**	**Answer**
1	1	01
2	1	00_BW_PAA1_2_D:\UserDir\USER.DLL
3	1	0

FI command

Parameter download command whereby the parameter download file is directly indicated.

BW_PAA2_(1) **(Single Write)**

(1) = Complete parameter download file name

Response Structure

The response to the "PAA2" FI command consists of three lines, each with one column. The meaning of the elements is as follows:

- Line 1 = Job ID [01...20]
  (see Chapter "FI Commands for the MPCX Device Group", IFJ).
- Line 2 = FI command
  [String, in accordance with Chapter "Elements of the FI Command"]
- Line 3 = FI Job Error Code
  (see Chapter "Error Codes")

Note: File and path details must be enclosed in inverted commas.

Example PAA2

00_BW_PAA2_"D:\DOWNLOAD.DAT"

FI command		**00_BW_PAA2_"D:\DOWNLOAD.DAT"**
Line	**Column**	**Answer**
1	1	01
2	1	00_BW_PAA2_"D:\DOWNLOAD.DAT"
3	1	0

Structure of Download File

The structure of the download file corresponds to that of a Windows Ini file. Bosch Rexroth’s own description in V20_Param_08_Definitions_Parameter_Download_01.doc is recommended for a more detailed account of the structure of the download file.

Summary:

Section [ID_PARAMETER]
Information concerning parameter identification.

Section [ID_SYSTEM]
Information concerning system parameter identification.

Section [DATA_SYSTEM]
Listing of system parameter data.

Section [ID_PROCESSX]
Information concerning process parameter identification.

Section [DATA_PROCESSX]
Listing of process parameter data.

Section [ID_AXISX]
Information concerning axis parameter identification.

Section [DATA_AXISX]
Listing of axis parameter data.

5.117 Parameter Upload: PAA

MWCX device group

Designation **PAA** **PA**rameter **A**ccess

Explanation Uploads complete parameter records from a selected device. The data read is written into an upload file with an identical structure to that of a download file.

FI command Parameter upload command whereby two predefined functions are to be programmed by the user. These two functions concern:

1. The function supplies the complete name of the upload file:

```
LONG ParameterUploadBegin(   Long lProjectNumber,
                             Long lDeviceNumber,
                             Char* pcUploadFileName,
                             Long lMaxLengthFileNameBuffer,
                             Char* pcErrorText,
                             Long lMaxLengthErrorTextBuffer)
```

Pass parameters:

- lProjectNumber:
  Currently selected project number
- lDeviceNumber:
  Currently selected device address
- pcUploadFileName:
  Contains the complete file name for the parameter upload file to be created.
- lMaxLengthFileNameBuffer:
  Max. length of the buffer for the name of the parameter upload file.
- pcErrorText:
  If necessary, user error text
- lMaxLengthErrorTextBuffer:
  max. length of the buffer for the user error text

2. Function called up at the end of the parameter upload:

```
LONG ParameterUploadEnd(   Char* pcUploadFileName,
                           Long lResult)
```

Pass parameters:

- pcUploadFileName:
  Contains the complete file names for the created parameter upload file.
- lResult:
  Contains the status message of the parameter upload procedure
  Here: 0 = Parameter upload procedure O.K.
  > 0 = Error has occurred

The two functions must be programmed in a DLL by the user and also exported from it.

BR_PAA1_(1)_{(2)} **(Single Read)**

(1) = Parameter directory number; the two functions to be implemented are located in INDIF410.DLL.

(2) = Complete DLL name, if required, in which the two functions to be implemented are located.

Response Structure

The response to the "PAA1" FI command consists of three lines, each with one column. The meaning of the elements is as follows:

- Line 1 = Job ID [01...20]
  (see Chapter "FI Commands for the MPCX Device Group: IFJ").
- Line 2 = FI command
  [String, in accordance with Chapter "Elements of the FI Command"]
- Line 3 = FI Job Error Code (see Chapter "Error Codes")

Note: File and path details must be enclosed in inverted commas.

Example PAA1

00_BR_PAA1_2_"D:\UserDir\USER.DLL"

FI command		**00_BR_PAA1_2_D:\UserDir\USER.DLL**
Line	**Column**	**Answer**
1	1	01
2	1	00_BR_PAA1_2_D:\UserDir\USER.DLL
3	1	0

FI command

Parameter upload command whereby the parameter upload file is directly indicated.

BR_PAA2_(1) **(Single Read)**

(1) = complete name of the parameter upload file

Response Structure

The response to the "PAA2" FI command consists of three lines, each with one column. The meaning of the elements is as follows:

- Line 1 = Job ID [01...20]
  (see Chapter "FI Commands for the MPCX Device Group: IFJ").
- Line 2 = FI command
  [String, in accordance with Chapter "Elements of the FI Command"]
- Line 3 = FI Job Error Code
  (see Chapter "Error Codes")

Note: File and path details must be enclosed in inverted commas.

Example PAA2

00_BR_PAA2_"D:\UPLOAD.DAT"

FI command		**00_BR_PAA2_"D:\UPLOAD.DAT"**
Line	**Column**	**Answer**
1	1	01
2	1	00_BR_PAA2_"D:\UPLOAD.DAT"
3	1	0

Structure of Upload File

The structure of the upload file corresponds to that of a Windows Ini file. Bosch Rexroth's own description in V20_Param_08_Definitions_Parameter_Download_01.doc is recommended for a more detailed account of the structure of the upload file.

For a summary refer to the description under Parameter Download Command.

5.118 Process Axis Configuration Data: PAC

MWCX device group

Designation

PAC **P**rocess **A**xis **C**onfiguration Parameter

Explanation

The axis configuration data of a process is returned.

FI command

Output the axis configuration parameters of all NC processes.

BR_PAC1 **(Single Read)**

Response Structure

The following table shows the general structure of the response to the FI command "PAC1". The number of lines depends on the number of defined CN processes. Each line consists of five columns for the NC process number, the physical axis number, the main axis meaning, the main axis name and the axis type.

Line 1...n:	**Column 1**	**....**	**Column 5**

Value Range/Meaning of Columns

(1) = NC process number	[0...6]
2 = Physical axis number	[1...32]
3 = Main axis meaning	[see Chapter 6.2 "Data Tables"]
4 = Main axis name	[Xi, Yi, Zi, Ui, Vi, Wi, Ai, Bi ,Ci, Si, --] (i=[ ], [1...3])
5 = Axis type	[see Chapter 6.2 "Data Tables"]

Example PAC1

Read all process axis configuration data of device address 00.

FI command		**00_BR_PAC1**
Line	**Column**	**Answer**
1	1	0
	2	1
	3	0
	4	X1
	5	0x81
2	1	1
	2	2
	3	1
	4	Y1
	5	0x82
3	1	2
	2	3
	3	5
	4	--
	5	--

FI command

Output the axis configuration data of an NC process.

BR_PAC2_(1)	**(Single Read)**
(1) = NC process number	[0...6]

Response Structure

The following table shows the general structure of the response to the FI command "PAC2". One line is output with five columns for the NC process number, the physical axis number, the main axis meaning, the main axis name and the axis type.

Line 1	**Column 1**	**....**	**Column 5**

Value Range/Meaning of Columns

(1) = NC process number	[0...6]
2 = Physical axis number	[1...32]
3 = Main axis meaning	[see the chapter entitled "Data Tables"]
4 = Main axis name	[Xi, Yi, Zi, Ui, Vi, Wi, Ai, Bi ,Ci, Si, --] (i=[ ], [1...3])
5 = Axis type	[see Chapter 6.2 "Data Tables"]

Example PAC2

Read the axis configuration data of process 0 at device address 00.

FI command		**00_BR_PAC2_0**
Line	**Column**	**Answer**
1	1	0
	2	1
	3	0
	4	X1
	5	0x81

5.119 Deactivate Parameters for an Offline Device PAD

MWCX device group

Designation **PAD** **PA**rameter **D**eactivate

Explanation If a device is in offline mode (DeviceStatus=OFF), this FI command deactivates the parameter record in the offline device; then, NO valid parameter record is present.

FI command **BW_PAD1** **(Single Write)**

Response Structure The response to the "PAD1" FI command consists of one line with one column.

Line 1	**Column 1**

Value Range/Meaning of Columns 1 = Status message (P_ACK) (P_ACK)

Example PAS1 The parameter records are deactivated for the offline device 00, i.e., there is NO valid parameter record in the device 00.

FI command		**00_BW_PAD1**
Line	**Column**	**Answer**
1	1	(P_ACK)

5.120 Setting Parameters Active for an Offline Device: PAS

MWCX device group

Designation **PAS** **PA**rameter **S**et Active

Explanation If a device is in offline mode (DeviceStatus=OFF), this FI command sets a parameter record active.

FI command **BW_PAS1_(1)** **(Single Write)**

(1) = Complete parameter download file name

Response Structure The response to the "PAS1" FI command consists of three lines, each with one column.

The meaning of the elements is as follows:

- Line 1 = Job ID [01...20]
  (see Chapter "FI Commands for the MPCX Device Group: IFJ").
- Line 2 = FI command
  [String, in accordance with Chapter "Elements of the FI Command"]
- Line 3 = FI Job Error Code (see Chapter "Error Codes")

Note: File names must be enclosed in inverted commas. This command is an FI job command.

Example PAS1

For the offline device 00, the parameter data of the parameter download file D:\DOWNLOAD.DAT are set active.

FI command		**00_BW_PAS1_"D:\DOWNLOAD.DAT"**
Line	**Column**	**Answer**
1	1	01
2	1	00_BW_PAS1_"D:\DOWNLOAD.DAT"
3	1	0

5.121 ProVi Diagnosis Data: PDD

MWCX device group

Designation **PDD** Provi **D**iagnosis **D**ata

Explanation Data for ProVi criteria analysis is output.

FI command Output of files to indicate the detail in the editor.

BR_PDD1_(1)_(2){_(3)}	**(Single Read)**
(1) = Message ID	[ASCII characters]
(2) = Message type	[1 = errors, 2 = messages, 10 = warnings, 11 = start requirements, 12=setup diagnosis]
(3) = Module number	[1...99] ! only for message type 1 -2!

Response Structure The following table shows the general structure of the PDD1 FI command.

Line 1	**Column 1**	**...**	**Column 5**

Meaning of the Columns

1 = POU ID	[ASCII characters]
2 = Detail morpheme	[ASCII characters] (DWORD, decimal)
3 = Error ID	[ASCII characters] (DWORD, decimal)
4 = POU entity name	[ASCII characters]
5 = Nw ID (network ID)	[ASCII characters]

Example PDD1 Indication of data of a ProVi error with ID 43923028 from module 3 in control unit 0.

FI command		**00_BR_PDD1_43923028_1_1**
Line	**Column**	**Answer**
1	1	STATION_1_2
	2	98243823
	3	34985304
	4	Station2.Module3
	5	43493454

FI command

Output the I/O addresses to display a detail.

BR_PDD2_(1)_(2){_(3)}	**(Single Read)**
(1) = Message ID	[ASCII characters]
(2) = Message type	[1 = error, 2 = messages, 10 = warnings, 11 = start requirements, 12 = setup diagnosis]
(3) = Module number	[1...99] ! only for message type 1 -2!

Response Structure

The following table shows the general structure of the PDD2 FI command.

Line 1-n	Column 1	Column 2

Meaning of the Columns

1 = Variable morpheme	[ASCII characters] (DWORD, decimal)
2 = I/O address	[ASCII characters]

Example PDD2

Query of the I/O addresses of a ProVi error with ID 43923028 from module 3 in control unit 0.

Three variables have an I/O address.

FI command		**00_BR_PDD2_43923028_1_1**
Line	**Column**	**Answer**
1	1	98243823
	2	%I3.2.0
2	1	40923423
	2	%Q23.21.7
3	1	34985304
	2	%I100.3.5

FI command

Determine the multilingual comments for displaying a detail.

BR_PDD3_(1)_(2){_(3)}	**(Single Read)**
(1) = Message ID	[ASCII characters]
(2) = Message type	[1 = error, 2 = messages, 10 = warnings, 11 = start requirements, 12 = setup diagnosis]
(3) = Module number	[1...99] ! only for message type 1 -2!

Response Structure

The following table shows the general structure of the PDD3 FI command.

Line 1-n	Column 1	Column 2

Meaning of the Columns

1 = Comment morpheme	[ASCII characters] (DWORD, decimal)
2 = New comment	[ASCII characters]

Example PDD3

Query of the comments for indication of a ProVi error with ID 43923028 from module 3 in control unit 0.

Two comments are replaced by another text.

FI command		**00_BR_PDD3_43923028_1_1**
Line	**Column**	**Answer**
1	1	98243823
	2	Clamp open
2	1	40923423
	2	Clamp closed

FI command

Query of the status of a certain message

BR_PDD4_(1)_(2){_(3)}	**(Single Read)**
(1) = Message number	[ASCII characters]
(2) = Message type	[1 = error, 2 = messages, 10 = warnings, 11 = start requirements, 12 = setup diagnosis]
(3) = Module number	[1...99] ! only for message type 1 -2!

Response Structure

The following table shows the general structure of the PDD4 FI command.

Line 1-n	**Column 1**	**Column 2**

Meaning of the Columns

1 = Message is present	[YES, NO]
2 = Criteria analysis exists	[YES, NO]

Example PDD4

Query of the status of a ProVi error, number 1001 from module 3 in control 0.

This message is not present at the moment, and there is a criteria analysis.

FI command		**00_BR_PDD4_1001_1_1**
Line	**Column**	**Answer**
1	1	NO
	2	YES

FI command

Determination of the MessageID of a certain message

BR_PDD5!(1)!(2)!(3)!(4){!(5)}	**(Single Read)**
(1) = POU entity name	[ASCII characters]
(2) = Nw ID	[ASCII characters]
(3) = Message number	[ASCII characters]
(4) = Message type	[1 = error, 2 = messages, 10 = warnings, 11 = start requirements, 12 = setup diagnosis]
(5) = Module number	[1...99] ! only for message type 1 -2!

Note: The separator "!" is used in this command.

Response Structure

The following table shows the general structure of the PDD5 FI command.

Line 1-n	Column 1	...	Column 3

Meaning of the Columns

1 = Message ID	[ASCII characters] (DWORD, decimal)
2 = Message is present	[YES, NO]
3 = Criteria analysis exists	[YES, NO]

Example PDD5

Determination of the MessageID of a ProVi error, number 1001 from module 25.40 mm control 0.

Assumption:
This message is not present at the moment, and there is a criteria analysis.

FI command		**00_BR_PDD5!Station2.Modul3!43493454!1001!1!1**
Line	**Column**	**Answer**
1	1	240872342
	2	NO
	3	YES

5.122 Reading the Parameter Definition Table: PDT

MWCX device group

Designation **PDT** **P**arameter **D**efinition **T**able

Explanation The parameter definition table for the selected device can be read. Note: This command ONLY returns binary data, which means that knowledge of the structure of the parameter definition table is necessary in order to interpret this binary data!

FI command Read parameter definition table for the selected device.

BR_PDT **(Single Read)**

Response Structure The following table shows the general structure of the response to the FI command "PDT".

Line 1	Column 1

Value Range/Meaning of Columns

1 =	Parameter definition table in binary form	Binary encoding of the parameter definition table in accordance with conventional control

Example PDT Read the parameter definition table for device 0.

FI command		**00_BR_PDT1**
Line	**Column**	**Answer**
1	1	Binary data for the parameter definition table

Rexroth
Bosch Group

5.123 Programmed Feed Velocity: PFR

MWCX device group

Designation **PFR** **P**rogrammed **F**eed **R**ate

Explanation The value of the programmed feedrate of the selected device of the MWCX device group is read out.

FI command Output the current value of the programmed feedrate of an NC process.

Using the optional second parameter it is possible to pre-select conversion of the result into mm or inches.

FI command

CR_PFR_(1){_(2)}	**(Single Read)**
CC_PFR_(1){_(2)}	**(Cyclic Read)**
CB_PFR_(1){_(2)}	**(Break Cyclic Read)**
(1) = NC process number	[0...6]
(2) = Required measurement system (opt.)	[mm, inch]

Response Structure The following table shows the general structure of the response to the FI command "PFR". One line is output with three columns for the identifier, the current value of the programmed feedrate and the unit.

Line 1	Column 1	....	Column 3

Value Range/Meaning of Columns

1 = Identifier	[F = feedrate]
2 = Feedrate	[format according to settings of the parameters]
3 = Unit	[according to settings of the parameters]

Example PFR Read the programmed feedrate in NC process 0 of device address 00.

FI command	**00_CR_PFR_0**		
Answer			
Line	**Column 1**	**Column 2**	**Column 3**
1	F	30000.0	[mm/min]

Example PFR Read the programmed feedrate in NC process 0 of device address 00. The displayed value is to be converted into inch/min:

FI command	**00_CR_PFR_0**		
Answer			
Line	**Column 1**	**Column 2**	**Column 3**
1	F	1181.1	[inch/min]

5.124 Reading the Size of the PLC Memory: PMI

MWCX device group

Designation **PMI** **P**lc **M**emory **I**nformation

Explanation The current size of the PLC memory is read out.

FI command **CR_PMI** **(Single Read)**

Response Structure One line with two values in BYTE is output:

1. Total memory
2. Free memory available now.

Line 1	Column 1	Column 2

Example PMI Read the current size of the PLC memory at device address 00.

FI command		00_CR_PMI
Line	**Column**	**Answer**
1	1	123456
	2	3210

5.125 Active NC Program Information: PPA

MWCX device group

Designation **PPA** **P**art **P**rogram **A**ctive

Explanation Reads the active NC program with information about the NC memory and NC program number.

FI command

BR_PPA_(1) **(Single Read)**
BC_PPA_(1) **(Cyclic Read)**
(1) = Process number [0...6]

Response Structure The following table shows the general structure of the response to the FI command "PPA". One line is output with 3 columns for the NC memory, NC program number and NC program name.

Line 1	Column 1	Column 2	Column 3

Value Range/Meaning of Columns

1 = NC memory	[A = memory A, B = memory B]
2 = NC program number	[01...99]
3 = NC program name	[max. 32 ASCII characters]

Example PPA Read in NC process 0 at device address 00.

Assumption:
The NC program numbered 01 and the with the name "Block4" is located in NC memory A; the memory is currently active.

FI command		00_BR_PPA_0
Line	**Column**	**Answer**
1	1	A
	2	01
	3	Block4

Reference to Literature See chapter entitled "Literature" [37].

5.126 Read NC Program Directory: PPD

MWCX device group

Designation **PPD** Part-**P**rogram **D**irectory

FI command Reads the entries of the NC program directory.

BR_PPD_(1)_(2)	**(Single Read)**
(1) = Number in NC package directory	[1...99]
(2) = NC process number	[0...6]

Response Structure The following table shows the general structure of the response to the FI command "PPD". The response consists of up to a maximum of n=99 lines, each with 5 columns.

Line 1...n:	**Column 1**	**...**	**Column 5**

Value Range/Meaning of Columns

1 = NC program number	[00...99]
2 = Program designation	[max. 32 ASCII characters]
3 = Program length	[byte]
4 = Date of creation/last change of program	[DD.MM.YY]
5 = Time of creation/last change of program	[HH:MM:SS]

Example PPD Read the entries in the NC program directory of the NC package number 1 of the NC process 0 at device address 00.

FI command		**00_BR_PPD_1_0**
Line	**Column**	**Answer**
1	1	01
	2	TransAM
	3	3579
	4	16.05.99
	5	10:41:08
2	1	10
	2	BMW 3 series
	3	4589
	4	18.05.99
	5	10:12:10

5.127 Expert or Import NC Program: PPN

MWCX device group

Designation **PPN** Part-**P**rogram **N**C

FI command Transfers an NC program from the NC program directory into an ASCII file (export).

BR_PPN_(1)_(2)_(3)_(4)	**(Single Read)**
(1) = Number in NC package directory	[1...99]
(2) = NC process number	[0...6]
(3) = Number of the NC program	[1...99]
(4) = NC block numbering	[0 = without number; 1 = with numbers] !

Response Structure The response of the FI command PPN consists of one line and one column for information on the drive, the directory, and the file which contains the NC program.

Example PPN Without any NC block numbering, import the NC program with the NC program number 1 of the 2nd NC package of the NC process 0 at the device address 00 into a file.

FI command		**00_BR_PPN_2_0_1**
Line	**Column**	**Answer**
1	1	C:\MT-CNC\ANLAGE01\MT_TEMP\T1010001.TMP

Excerpt from the file "C:\MT-CNC\ANLAGE01\MT_TEMP\T1010001.TMP":

```
START

SPF 1 [select reference spindle]
T1 BSR .M6
G90 G96 G54 S1 2000 F5000 M03

G00 X60 Y-30
Z-6 [infeed motion]
G01 X60 Y0 F2000
X5 Y0
Z100
M05 [spindle stop]

T0 BSR .M6
BST .START
PROGRAM END
```

FI command Transfers an NC program from an ASCII file into the NC program directory (import).

BW_PPN_(1)_(2)_(3)_(4)_(5)_(6)		**(Single Write)**
(1) =	Number in NC package directory	[1...99]
(2) =	NC process number	[0...6]
(3) =	Number of the NC program	[1...99]
(4) =	NC block numbering	[0 = without number; 1 = with numbers] !
(5) =	Is the NC package directory entry empty?	[0 = without check (preset); 1 = with check] ! Optional !
(6) =	Complete information on the directory	[DRIVE:\..\X.Y]

Note: This FI command does not have any "Value to be written".

Response Structure One line with one column is output to acknowledge the FI command issued. The meaning of the elements is as follows:

(P_ACK) = **P**ositive **ACK**nowledge NC programs was exported.

Example PPN From the file "C:\Data\T1010001.TMP", export the NC program in NC program number 1 of the 2nd NC package of the NC process 0 at the device address 0.

FI command		**00_BW_PPN_2_0_1_0_1_C:\Data\T1010001.TMP**
Line	**Column**	**Answer**
1	1	(P_ACK)

5.128 Renaming of an NC Part Program: PPN

MWCX device group

Designation **PPN** Part **P**rogram **N**C

Explanation This FI command renames an NC part program.

FI command

BA_PPN_(1)_(2)_(3)	**(Single Write)**
(1) = Number in NC package directory	[1...99]
(2) = NC process number	[0...6]
(3) = Number of the NC program	[1...99]

Value to be written Name of the NC part program [max. 32 ASCII characters]

Note: The value to be written is passed to the "acValue" parameter as an ASCII value in the "DataTransfer" routine.

Response Structure One line with one column is output to acknowledge the FI command issued. The meaning of the elements is as follows:

(BOF_FCT_OK) = **BOF_F**un**CT**ion**_OK** program package has been renamed.

Example PPP The name of the NC part program numbered 1 in the NC package directory is to be renamed "PART1".

FI command		**Value to be written: PART1** **00_BA_PPN_2_0_1**
Line	**Column**	**Answer**
1	1	(BOF_FCT_OK)

Reference to Literature See chapter entitled "Literature" [37].

5.129 Renaming of an NC Program Package: PPP

MWCX device group

Designation **PPP** Part **P**rogram **P**ackage

Explanation The name of an NC program package of the selected device of the MWCX device group is changed.

FI command **BA_PPP_(1)** **(Single Alternate)**

(1) = NC program package [1...99]

Value to be written Name of the NC program package [max. 32 ASCII characters]

Note: The value to be written is passed to the "acValue" parameter as an ASCII value in the "DataTransfer" routine.

Response Structure One line with one column is output to acknowledge the FI command issued. The meaning of the elements is as follows:

(BOF_FCT_OK) = **BOF_F**un**CT**ion**_OK** program package has been renamed.

Example PPP The name of the NC program package numbered 1 in the NC package directory is to be renamed "FORM1".

FI command		**Value to be written: FORM1** **00_BA_PPP_1**
Line	**Column**	**Answer**
1	1	(BOF_FCT_OK)

Reference to Literature See chapter entitled "Literature" [37].

5.130 Reading an NC Record: PPS

MWCX device group

Designation **PPS** Part **P**rogram **S**equence

Explanation An NC record of an NC program from the selected device of the MWCX device group is read out.

FI command **CR_PPS_(1)_(2)_(3)_(4)** **(Single Read)**

(1) = NC memory [1=memory A, 2=memory B]

(2) = NC process number [0...6]

(3) = NC program number [0...99]

(4) = NC record number [0...9999]

Response Structure One line with one column containing the requested NC record is output.

Example PPS Read NC record number 2 from NC program memory A, NC process number 0 or NC program number 1.

FI command		00_CR_PPS_1_0_1_2
Line	**Column**	**Answer**
1	1	N0002 G01 X50.0000 Y50.0000 Z20.0000 F2500.0

Reference to Literature See chapter entitled "Literature" [4].

5.131 Issuing SYS Messages Specific to the PCL: PSM

MWCX device group

Designation **PSM** PCL **S**ys **M**essage

Explanation Issues the most important SYS messages regarding the PCL programming interface – required for remote programming.

Note:
The appropriate device address is passed as the write value.

It allows the following SYS messages to be initiated:

- Start of PCL download,
- end of PCL download,
- start of PLC online edit,
- end PLC online edit,
- start of PCL declaration change, and
- end of PCL declaration change.

FI command Issue the most important PCL SYS messages.

BW_PSM1_(1) **(Single Write)**

(1) = Requested SYS message [1= start of PCL download
2= end of PCL download
3= start of PCL online edit
4= end of PCL online edit
5= start of PCL declaration change
6= end of PCL declaration change]

Value to be written: Device address

Response Structure

The following table shows the general structure of the response to the FI command "PSM1".

Line 1	Column 1	...	Column 8

Value Range/Meaning of Columns

1 =	Status report	[READY=SYS message has been correctly acknowledged by the WIN32 applications] [ERROR=SYS message has NOT been acknowledged by a WIN32 application within the pre-set time]
2 =	Task name (LogInIf name)	[Task name that has triggered the SYS message]
3 =	SYS message number	[contains the issued SYS message number]
4 =	Acknowledgement time	[contains the pre-set acknowledgement time]
5 =	Reference information	[contains, where applicable, the reference information transferred as a write value]
6 =	Length of reference information	[0 where NO reference information has been transferred]
7 =	Where applicable, LOG channel of the FI that has NOT acknowledged	[-- = acknowledgements have been completed in time or the LOG channel number of the WIN32 application that has NOT acknowledged in time]
8 =	Where applicable, task name that has NOT acknowledged in time.	[-- = acknowledgements have been completed in time or the task name that has NOT acknowledged in time]

Example PSM1

Issue the SYS message Beginning PCL Download. The reference information, device address 00, is also transferred as a write value.

FI command		**value to be written: 00** **XX_BW_PSM1_1**
Line	**Column**	**Answer**
1	1	READY
	2	WINPCL.EXE
	3	14
	4	30000
	5	00
	6	2
	7	--
	8	--

5.132 Programmed Spindle Speed: PSS

MWCX device group

Designation **PSS** **P**rogrammed **S**pindle **S**peed

Explanation The value of the programmed spindle speed of the selected device of the MWCX device group is read out.

FI command

CR_PSS_(1)_(2)	**(Single Read)**
CC_PSS_(1)_(2)	**(Cyclic Read)**
CB_PSS_(1)_(2)	**(Break Cyclic Read)**
(1) = NC process number	[0...6]
(2) = Number of spindle	[1...3]

Response Structure The following table shows the general structure of the response to the FI command "PSS". One line with three columns is output for the axis name, the speed and the unit [1/min].

Line 1	Column 1	....	Column 3

Value Range/Meaning of Columns

1 = Identifier	[S = spindle]
2 = Speed	[format according to settings of the parameters]
3 = Unit	1/min

Example PSS Read the speed of the 1st spindle in NC process 0 of device address 00.

FI command	**00_CR_PSS_0_1**		
Answer			
Line	**Column 1**	**Column 2**	**Column 3**
1	S:	7500.0	1/min

Reference to Literature See chapter entitled "Literature" [4].

5.133 Process Tool Management Configuration: PTC

MWCX device group

Designation **PTC** **P**rocess **T**ool Management **C**onfiguration

Explanation Returns the most significant process parameter data of the tool magazine of the selected device of the MWCX device group.

FI command Read tool management data of all defined NC processes.

BR_PTC1	**(Single Read)**
BC_PTC1	**(Cyclic Read)**

Response Structure The following table shows the general structure of the response to the FI command "PTC1". The number of lines depends on the number of defined CN processes. Each line consists of 9 columns for the returned values.

Line 1...n:	Column 1	...	Column 9

Value Range/Meaning of the Columns

1 = NC process number	[0...6]
2 = Process name	
3 = Tool management	[YES, NO]
4 = Tool memory	[[MAGAZINE], [TURRET]]
5 = Endlessly turning tool memory	[YES, NO]
6 = Number of tool memory locations	[0...999]
7 = Number of tool spindles	[0...4]
8 = Number of tool grabbers	[0...4]
9 = Axis number of tool axis	[0...20]
10 = Number of tool groups	[0...99]

Note: If there is no tool management (Column 3: NO), then all partial results from Column 4 are marked as [--].

Example PTC1

Returns the process parameter data of the defined processes. This example assumes that there are two processes, On process with and another one without tool management.

FI command		**00_BR_PTC1**
Line	**Column**	**Answer**
1	1	0
	2	MILLING
	3	YES
	4	[MAGAZINE]
	5	YES
	6	8
	7	1
	8	2
	9	4
2	1	1
	2	TRANSFER
	3	NO
	4	--
	5	--
	6	--
	7	--
	8	--
	9	--
	10	0

FI command

Read tool management data of an NC process.

BR_PTC2_(1)	**(Single Read)**
BC_PTC2_(1)	**(Cyclic Read)**
(1) = NC process number	[0...6]

Response Structure

The following table shows the general structure of the response to the FI command "PTC2". One line with 9 columns is output for the returned values.

Line 1	Column 1	...	Column 9

Meaning of the Columns

1 = NC process number	[0...6]
2 = Process name	
3 = Tool management	[YES, NO]
4 = Tool memory	[[MAGAZINE], [TURRET]]
5 = Endlessly turning tool memory	[YES, NO]
6 = Number of tool memory locations	[0...999]
7 = Number of tool spindles	[0...4]
8 = Number of tool grabbers	[0...4]
9 = Axis number of tool axis	[0...20]
10 = Number of tool groups	[0...99]

Notes: If there is no tool management (Column 3: NO), then all partial results from Column 4 are marked as [--].

If the requested process does not exist then there is no results line.

Example PTC2

Returns the process parameter data of the process 0.

FI command		00_BR_PTC2_0
Line	**Column**	**Answer**
1	1	0
	2	MILLING
	3	YES
	4	MAGAZINE
	5	YES
	6	8
	7	1
	8	2
	9	4
	10	0

FI command

Returns the main process parameter data of the tool management of the selected device.

BR_PTC3	**(Single Read)**
BC_PTC3	**(Cyclic Read)**

Response Structure

The answer consists of n lines with 10 columns each, one line being supplied for each defined process.

Line n	Column 1	..	Column 10

Value Range/Meaning of Columns

1 =	Process number	[0..6]
2 =	Process designation	
3 =	Is a tool list available?	[YES,NO]
4 =	Name of the tool list	Max. 32 ASCII characters
5 =	Number of spindles according to process parameters	[S0..S4]
6 =	Number of grippers according to process parameters	[G0..G4]

7 =	Number of magazine locations according to process parameters	[M0..M999]
8 =	Number of edges according to system parameters	[E0..E9]
9 =	Type of tool magazine	[MAGAZIN,REVOLVER]
10 =	Endlessly turning tool magazine	[YES,NO]

Example PTC3 Returns the main process parameter data of the tool management for device 1. Here, the processes 0,1,2,3, and 4 are defined, and there are no tool lists in processes 3 and 4.

FI command		**01_BR_PTC3**
Line	**Column**	**Answer**
1	1	0
	2	Master
	3	YES
	4	Tool test list 1
	5	S1
	6	G1
	7	M99
	8	E9
	9	[MAGAZINE]
	10	YES
2	1	1
	2	Process 1
	3	YES
	4	Tool test list 2
	5	S4
	6	G2
	7	M88
	8	E9
	9	[MAGAZINE]
	10	YES
3	1	3
	2	Process 3
	3	NO
	4	--
	5	S4
	6	G4
	7	M66
	8	E9
	9	[MAGAZINE]
	10	YES

4	1	4
	2	Process 4
	3	NO
	4	--
	5	S4
	6	G3
	7	M55
	8	E9
	9	[MAGAZINE]
	10	YES

5.134 Edit PROVI Message Files: PVA

MWCX device group

Designation **PVA** **PRO**V**I**-Messages **A**ccess

Explanation This write command creates PROVI message files. With this write value, it is possible to decide whether the PROVI messages are to be generated according to the current PLC project, or selectively.

FI command **BW_PVA1** **(Single Write)**

Note: This command is an FI job command.

Value to be written

No write value exists	PROVI message files according to the current PLC project.
Write value exists	List of the requested PROVI message files (separated by a comma) according to the format: [PROVI-Diag-type: module number] Example: 01:01,01:02,02:02

Note: The value to be written is passed to the "acValue" parameter as an ASCII value in the "DataTransfer" routine.

Response Structure The response to the "BW_PVA1" FI command consists of three lines, each with one column. The meaning of the elements is as follows:

- Line 1 = Job ID [01...20]
  (see Chapter "FI Commands for the MPCX Device Group", IFJ).
- Line 2 = FI command
  [String, in accordance with Chapter "Elements of the FI Command"]
- Line 3 = FI Job Error Code (see Chapter "Error Codes")

....

Example PVA1

No write value is passed, i.e. the PROVI message files are generated according to the current PLC project.

FI command		**00_BW_PVA1**
Line	**Column**	**Answer**
1	1	01
2	1	00_BW_PVA1
3	1	0

Explanation

The read command returns the most significant information on the created PROVI message files.

FI command **BR_PVA1** **(Single Read)**

....

Response Structure

The following table shows the general construction of the answer of the FI command BR_PVA1. For each available PROVI message file, 1 line with 10 columns each is created.

Line 1...n	**Column 1**	**...**	**Column 10**

Value Range/Meaning of Columns

1 =	PROVI diagnosis type	[1..20]
2 =	PROVI diagnosis type designation	[The following designations can be returned: StartCondition, Error, Message, Warning, Setup]
3 =	Module number	[1..99]
4 =	PROVI diagnosis type and module number	[PROVI diagnosis type: module number, see write value for BW_PVA2]
5 =	Complete name of the PROVI message text file	[max. 200 ASCII characters]
6 =	Memory required for the PROVI messages in the control	[figure in ASCII format]
7 =	Complete name of the PROVI index file	[max. 200 ASCII characters]
8 =	Memory required for the PROVI index data in the control	[figure in ASCII format]
9 =	Total memory (for text + index) required in the control	[figure in ASCII format]
10 =	Total memory for ALL PROVI data (text + index) required in the control	[figure in ASCII format]

Example PVA1

The most significant information of 2 available PROVI message files are returned.

FI command		00_BR_PVA1_1
Line	**Column**	**Answer**
1	1	1
	2	Error
	3	1
	4	01:01
	5	D:\Programs\indramat\Mtgui\Project_000\ Programmdata\Device_000\Diag\De\ERROR 1.TXT
	6	1345
	7	D:\Programs\indramat\Mtgui\Project_000\ Programmdata\Device_000\Diag\De\ERROR 1.TXT
	8	234
	9	1579
	10	4491
2	1	2
	2	Message
	3	1
	4	02:01
	5	D:\Programs\indramat\Mtgui\Project_000\ Programmdata\Device_000\Diag\De\ERROR 1.TXT
	6	2456
	7	D:\Programs\indramat\Mtgui\Project_000\ Programmdata\Device_000\Diag\De\ERROR 1.TXT
	8	456
	9	2912
	10	4491

Explanation

This write command transmits PROVI message files into the selected device. Through the write value, it is possible to chose whether ALL or only the PROVI messages selected via the write value are to be transmitted.

FI command **BW_PVA2** **(Single Write)**

Note: This command is an FI job command.

Value to be written

No write value exists	All PROVI message files are transmitted into the selected device
Write value exists	List of the requested PROVI message files (separated by a comma) according to the format: [PROVI-Diag-type: module number] Example: 01:01,01:02,02:02

Note: The value to be written is passed to the "acValue" parameter as an ASCII value in the "DataTransfer" routine.

Response Structure

The response to the "BW_PVA2" FI command consists of three lines, each with one column. The meaning of the elements is as follows:

- Line 1 = Job ID [01...20]
  (see Chapter "FI Commands for the MPCX Device Group", IFJ).
- Line 2 = FI command
  [String, in accordance with Chapter "Elements of the FI Command"]
- Line 3 = FI Job Error Code (see Chapter "Error Codes")

....

Example PVA2

No write value is passed, i.e. all PROVI message files should be transmitted.

FI command		**00_BW_PVA2**
Line	**Column**	**Answer**
1	1	01
2	1	00_BW_PVA2
3	1	0

5.135 Formatted Input / Output of PLC Variables: PVF

MWCX device group

Designation **PVF** **P**LC **V**ariable **F**ormatted

Explanation

Formatted reading and writing of PLC variables, arrays and structures.

FI command

Read PLC variables.

CR_PVF_(1)	**(Single Read)**
CC_PVF_(1)	**(Cyclic Read)**
CB_PVF_(1)	**(Break Cyclic Read)**
(1) = Identifier of the PLC variable	[acc. to declaration part of the PLC]

Response Structure

One line with one column is output for single variables. For array and structure variables, one line per element is output, depending on the number of elements.

Line 1...n:	**Column 1**

n = number of elements.

Note: Only defined PLC variables can be read and written. Addressing a non-declared variable results in an error message. A PLC variable can only be read if its data length does not exceed 240 byte (refer also to chapter on "Programming" and "Guidelines").

Value Ranges ANSI / ASCII

The value range of the response depends on the data type of the variable read. The following table indicates the range in which the results string is to be expected when reading out a single variable and into which C-data type this string can be converted without loss of information:

Data Type	Value Range	Can be converted to C-data type
BOOL	[0;1]	unsigned char
SINT	[-128...127]	char
INT	[-32768...32767]	short
DINT	[2147483648...2147483647]	long
USINT	[0...255]	unsigned char
UINT	[0...65535]	unsigned short
UDINT	[0...4294967295]	unsigned long
BYTE	[0x00...0xFF]	unsigned char
WORD	[0x0000....0xFFFF]	unsigned short
DWORD;	[0x00000000...0xFFFFFFFF]	unsigned long
TIME	[0...4294967295]	unsigned long (msec)
CHAR	[$00...$20,!...~,$7F...$FF]	char
STRING	<String> whereby <String> string is a character string with a maximum of as many characters as are declared for the string in the PLC	Char[xx+1]] +1 i.e. room for the zero byte
REAL	[-3.402823567E+38...3.402823567E+38]	Float

Note: An empty string is identified by two single inverted commas: ' ' (do not confuse with the double inverted commas '')!

All single variables can be part of array and structure variables. The value ranges maintain their validity, even when within structured data types.

Binary Value Range

The value range of the response depends on the data type of the variable read. The following table indicates the value range in which to expect the binary value of a single variable and how many bytes are included in the binary byte sequence:

Data Type	Value Range	Length (bytes)
BOOL	[00_H...01_H]	1
SINT	[80_H...$7F_H$] i.e. –128...127	1
INT	[8000_H (-32768)...$7FFF_H$ (32767)]	2
DINT	[80000000_H (-2147483648)... $7FFFFFFF_H$ (2147483647)]	4
USINT	[00_H (0)...FF_H (255)]	1
UINT	[00_H (0)...$FFFF_H$ (65535)]	2
UDINT	[0...4294967295]	4
BYTE	[0x00...0xFF]	1
WORD	[0x0000....0xFFFF]	2
DWORD;	[0x00000000...0xFFFFFFFF]	4
TIME	[0...4294967295]	4
CHAR	[$00...$20,!...~,$7F...$FF]	1
STRING	<String> whereby <String> string is a character string with a maximum of as many characters as are declared for the string in the PLC	XX+1
REAL	[-3.402823567E+38...3.402823567E+38]	4

Note: Binary array and structure elements will be connected to without space between (1 Byte Alignment).

PLC - Example 1 PVF

Read the value of the PLC variable "STK_TXT" in ASCII format from device address 00.

Assumption:
The "STK_TXT" variable is declared as STRING in the PLC program.

FI command		**00_CR_PVF_STK_TXT/1**
Line	**Column**	**Answer**
1	1	Repeat counter

WinPCL - Example 1 PVF

Read the value of WinPCL program entity variable "STK_TXT" in ASCII format in WinPCL program "Prog" at device address 00.

Assumption:
The WinPCL variable "STK_TXT" is declared in WinPCL program "Prog" as STRING.

FI command		**00_CR_PVF_:Prog.STK_TXT/1**
Line	**Column**	**Answer**
1	1	Repeat counter

PLC - Example 2 PVF

Read the value of the PLC array "BEG_END" in ANSI format from device address 00.

Assumption:
The "BEG_END" variable is declared as BYTE with 2 elements in the PLC program.

FI command		**00_CR_PVF_BEG_END/3**
Line	**Column**	**Answer**
1	1	0x00
2	1	0x1F

WinPCL - Example 2 PVF

Read the value of WinPCL program entity array "BEG_END" in ASCII format in WinPCL program "Prog" at device address 00.

Assumption:
The WinPCL variable "BEG_END" is declared in WinPCL program "Prog" as BYTE with two elements.

FI command		**00_CR_PVF_:Prog.BEG_END/3**
Line	**Column**	**Answer**
1	1	0x00
2	1	0x1F

PLC - Example 3 PVF

Read the value of the PLC structure "MSTRCT" in ASCII format from device address 00.

Assumption:
The "MSTRCT" variable is declared as a structure in the PLC program as follows:

```
TYP STRUCT
        T1      BOOL
        T2      CHAR
        T3      STRING[16]
        T4      TIME
END
```

FI command		**00_CR_PVF_MSTRCT/1**
Line	**Column**	**Answer**
1	1	0
2	1	A
3	1	ROBOT AXIS X
4	1	2000

WinPCL - Example 3 PVF

Read the value of WinPCL program entity structure "MSTRCT" in ASCII format in WinPCL program "Prog" at device address 00.

Assumption:
The WinPCL variable "MSTRCT" is declared as a structure in WinPCL program "Prog" as follows:

```
TYP STRUCT
        T1      BOOL
        T2      CHAR
        T3      STRING[16]
        T4      TIME
END
```

FI command		**00_CR_PVF_:Prog.MSTRCT/1**
Line	**Column**	**Answer**
1	1	0
2	1	A
3	1	ROBOT AXIS X
4	1	2000

FI command

Write PLC variable.

CW_PVF_(1)	**(Single Write)**
(1) = Identifier of the PLC variable	[acc. to declaration part of the PLC]

Value to be written

Value of data element	[see value ranges]

Note: The value to be written is passed to the "acValue" parameter in the "DataTransfer" routine. The data code of the value is passed to the parameter "ValType".

Response Structure

One line is output with a column for acknowledgement of whether or not the FI command has been executed successfully.

(P_ACK) = **P**ositive **ACK**nowledge Data element has been set

Value Range of the value to be written in ANSI / ASCII Format

The value ranges agree for the most part with the ANSI / ASCII result-value ranges during read access. ANSI umlauts are thereby converted into ASCII umlauts. Only ASCII umlauts are stored in the control unit. For deviations to this, please refer to the following note:

Note: Strings are enclosed by two single inverted commas ' ' , e.g. 'drill'.

Special characters can be indicated in accordance with DIN-1131 by a $ sign.

The following are used:

- $' '
- $$ $
- $R \r (Carriage Return)
- $L \n (Linefeed)
- $P \f (Formfeed)
- $T \t (Tab)
- $xx xx refers to a character written as a hexadecimal value. e.g. $20 (space)

Array and structure elements are separated by a space.

Value Range of the Value to be written in Binary Format

The value ranges agree with the binary result-value range during read access. For deviations to this, please refer to the following note:

PLC - Example 4 PVF

Write into the PLC variable "STK_TXT" at device address 00. The value is passed in ANSI format.

Assumption:
The "STK_TXT" variable is declared as STRING in the PLC program.

FI command		**00_CW_PVF_STK_TXT/3**
Line	**Column**	**Answer**
1	1	(P_ACK)

Value to be written:

Value of data element	'item counter'
Data code	/3

WinPCL - Example 4 PVF

Write into the WinPCL program entity variable "STK_TXT" in WinPCL program "Prog" at device address 00. The value is passed in ANSI format.

Assumption:
The WinPCL variable "STK_TXT" is declared in WinPCL program "Prog" as STRING.

FI command		**00_CW_PVF_:Prog.STK_TXT/3**
Line	**Column**	**Answer**
1	1	(P_ACK)

Value to be written:

Value of data element	'item counter'
Data code	/3

PLC - Example 5 PVF

Write into the PLC byte array "BEG_END" at device address 00. The value is passed in ANSI format.

Assumption:
The "BEG_END" variable is declared as a BYTE array with 2 elements in the PLC program.

FI command		**00_CR_PVF_BEG_END/3**
Line	**Column**	**Answer**
1	1	(P_ACK)

Value to be written:

Value of data element	0x20 0x3f
Data code	/3

WinPCL - Example 5 PVF

Write into the WinPCL program entity byte array "BEG_END" in WinPCL program "Prog" at device address 00. The value is passed in ANSI format.

Assumption:
The WinPCL variable "BEG_END" is declared in WinPCL program "Prog" as BYTE with two elements.

FI command		**00_CW_PVF_:Prog.BEG_END/3**
Line	**Column**	**Answer**
1	1	(P_ACK)

Value to be written:

Value of data element	0x20 0x3f
Data code	/3

PLC - Example 6 PVF

Write the value of element T3 of the PLC structure "MSTRCT" at device address 00. The string "COUNTER" is transferred in binary format.

Assumption:
The "MSTRCT" variable is declared as a structure in the PLC program as follows:

```
TYP STRUCT
        T1      BOOL
        T2      CHAR
        T3      STRING[16]
        T4      TIME
END
```

FI command		**00_CW_PVF_MSTRCT.T3/2**
Line	**Column**	**Answer**
1	1	(P_ACK)

Value to be written:

Value of data element	Binary sequence: 43 4F 55 4E 54 45 52 00
Data code	/2

WinPCL - Example 6 PVF

Write the value of element T3 of the WinPCL program entity structure "MSTRCT" at device address 00. The string "COUNTER" is transferred in binary format.

Assumption:
The WinPCL variable "MSTRCT" is declared as a structure in WinPCL program "Prog" as follows:

```
TYP STRUCT
        T1      BOOL
        T2      CHAR
        T3      STRING[16]
        T4      TIME
END
```

FI command		**00_CW_PVF_:Prog.MSTRCT.T3/2**
Line	**Column**	**Answer**
1	1	(P_ACK)

Value to be written:

Value of data element	Binary sequence: 43 4F 55 4E 54 45 52 00
Data code	/2

PLC - Example 7 PVF

Write the value of the PLC structure "MSTRCT" from the structure "mstrct" previously stored in the C program at device address 00.

Assumption:
The "MSTRCT" variable is declared as a structure in the PLC program as follows:

```
TYP STRUCT
        T1      BOOL
        T2      CHAR
        T3      STRING[16]
        T4      TIME
END
```

To exchange binary data in a C program, the following "C" data type can be used:

```
#pragma pack(1)    //Write all elements
                   //without spaces next to each other.
typeder struct
{
      unsigned char T1;
      char         T2;
      char         T3[17]; //Space for zero byte
      unsigned long T4;
} Tymstrct;        // Declare structure
Tymstrct mstrct;   // Apply structure
```

FI command		**00_CW_PVF_MSTRCT/2**
Line	**Column**	**Answer**
1	1	(P_ACK)

Value to be written: address of the C structure.

Value of data element	&mstrct
Data code	/2

WinPCL - Example 7 PVF

Write the value of the WinPCL program entity structure "MSTRCT" from the structure "mstrct" previously stored in the C program at device address 00.

Assumption:
The WinPCL variable "MSTRCT" is declared as a structure in WinPCL program "Prog" as follows:

```
TYP STRUCT
        T1      BOOL
        T2      CHAR
        T3      STRING[16]
        T4      TIME
END
```

To exchange binary data in a C program, the following "C" data type can be used:

```
#pragma pack(1)   //Write all elements
                  //without spaces next to each other.
typeder struct
{
      unsigned char T1;
      char          T2;
      char          T3[17]; //Space for zero byte
      unsigned long T4;
} Tymstrct;        // Declare structure
Tymstrct mstrct;  // Apply structure
```

FI command		00_CW_PVF_:Prog.MSTRCT/2
Line	**Column**	**Answer**
1	1	(P_ACK)

Value to be written: address of the C structure.

Value of data element	&mstrct
Data code	/2

5.136 ProVi Messages: PVM

MWCX device group

Designation **PVM** **ProVi Messages**

Explanation

ProVi messages are output. These messages are assigned to a particular message type or module.

FI command

Output all ProVi messages.

For reasons of optimization, not all data is called up. Accordingly, the Diagnosis server must be informed that the data is required (see ADW).

BR_PVM1_(1){_(2)} **(Single Read)**

BC_PVM1_(1){_(2)} **(Cyclic Read)**

(1) = Message type	[1 = error, 2 = messages, 10 = warnings, 11 = start requirements, 12 = setup diagnosis]
(2) = Module number	[1...99] ! only for message type 1 -2!

Output first ProVi messages.

BR_PVM2_(1){_(2)}	**(Single Read)**
BC_PVM2_(1){_(2)}	**(Cyclic Read)**
(1) = Message type	[1 = error, 2 = messages, 10 = warnings, 11 = start requirements, 12 = setup diagnosis]
(2) = Module number	[1...99] ! only for message type 1 -2!

Response Structure

The following table shows the general structure of the FI commands "PVM1" and "PVM2". The number of lines depends on the number of messages pending.

If there are no messages, the number of lines is 0.

Line 1...n	Column 1	...	Column 8

Meaning of the Columns

1 = Message text	[ASCII characters]
2 = Message number	[ASCII characters]
3 = Time stamp day	[mm.dd.yyyy]
4 = Time stamp time	[hh:mm:ss]
5 = Message ID	[ASCII characters] (DWORD, decimal)
6 = Reference text available	[YES, NO]
7 = Criteria analysis exists	[YES, NO]
8 = Filename for additional information for message text	[e.g.HTML format]

Example PVM1

All ProVi errors from module 3 in control unit 0.

There are two messages.

FI command		00_BR_PVM1_1_3
Line	**Column**	**Answer**
1	1	Guard not closed
	2	34
	3	01.27.2000
	4	14:56:32
	5	43923028
	6	YES
	7	NO
	8	
2	1	Oil pressure too low
	2	124
	3	01.27.2000
	4	15:03:10
	5	98234039
	6	NO
	7	YES
	8	

Example PVM2

The first ProVi error from module 3 in control unit 0.

There are two messages:

FI command		**00_BR_PVM2_1_3**
Line	**Column**	**Answer**
1	1	Guard not closed
	2	34
	3	01.27.2000
	4	14:56:32
	5	43923028
	6	YES
	7	NO
	8	

FI command

Output the reference information of a ProVi message.

BR_PVM3_(1)_(2){_(3)} **(Single Read)**

(1) = Message ID [ASCII characters]

(2) = Message type [1 = error, 2 = messages,
10 = warnings,
11 = start requirements,
12 = setup diagnosis]

(3) = Module number [1...99] ! only for message type 1 -2!

Response Structure

The following table shows the general structure of the "PVM3" FI command.

Line 1	**Column 1**	**...**	**Column 16**

Meaning of the Columns

1 = Message text [ASCII characters]

2 = Message number [ASCII characters]

3 = Error category [ASCII characters] (empty no category)

4 = Time stamp day [mm.dd.yyyy]

5 = Time stamp hour [hh:mm:ss]

6 = Reference text available [YES, NO]

7 = Reference text [ASCII characters]

8 = Message ID [ASCII characters] (DWORD, decimal)

9 = Diagnosis source [ASCII characters] (PLC, CNC)

10 = POE name [ASCII characters]

11 = Detail name [ASCII characters] (leer Implementation)

12 = Detail type [1 = Action block,
3 = Transition,
4 = Implementation]

13 = Network number [ASCII characters]

14 = Variable name [ASCII characters]

15 = POU entity name [ASCII characters]

16 = POU type [2 = program,
3 = function block]

17 = Analysis of criteria available [YES, NO]

18 = File name for additional information for message text [e.g.HTML format]

19 = File name for additional information for reference text [e.g.HTML format]

Example PVM3

Reference text of a ProVi error with ID 43923028 from module 3 in control unit 0.

FI command		**00_BR_PVM3_43923028_1_3**
Line	**Column**	**Answer**
1	1	Guard not closed
	2	34
	3	1
	4	01.27.2000
	5	14:56:32
	6	YES
	7	Oil pressure too low Oil pipe leaking or insufficient oil.
	8	43923028
	9	PLC
	10	MODULE3
	11	
	12	4
	13	34
	14	EschutzT
	15	Station2.Module3
	16	3
	17	NO
	18	
	19	D:\Program Files\Indramat\MtGui\Project_000\ProgramData\HMTL\DE\Error34.html

FI command

One after the other, all active ProVi messages are output. In the result, one line each is returned. After expiry of the set time, the next message is returned. The clock frequency can be set via the last parameter. This value can only be set once for the PC. The value transmitted last is always the valid value. Default setting is 1 second.

BR_PVM4_(1){_(2)_(3)} **(Single Read)**

BC_PVM4_(1){_(2)_(3)} **(Cyclic Read)**

(1) = Message type [1 = error, 2 = messages,
10 = warnings,
11 = start requirements,
12 = setup diagnosis]

(2) = Module number [1...99] ! only for message type 1 -2!

(3) = Clock frequency [ASCII characters] Time in ms

Response Structure

The following table shows the general structure of the "PVM4" FI command.

If there are no messages, the number of lines is 0.

Line 1	**Column 1**	**...**	**Column 8**

Meaning of the Columns

1 = Message text	[ASCII characters]
2 = Message number	[ASCII characters]
3 = Time stamp day	[mm.dd.yyyy]
4 = Time stamp time	[hh:mm:ss]
5 = Message ID	[ASCII characters] (DWORD, decimal)
6 = Reference text available	[YES, NO]
7 = Criteria analysis exists	[YES, NO]
8 = Message index (1 = 1. message)	[ASCII characters]
9 = Filename for additional information for message text	[e.g.HTML format]

Example PVM1

ProVi errors from module 3 in control unit 0.

The 2nd message is being output. The clock frequency is to be 2 seconds.

FI command		**00_BR_PVM4_1_3_2000**
Line	**Column**	**Answer**
1	1	Oil pressure too low
	2	124
	3	01.27.2000
	4	15:03:10
	5	98234039
	6	NO
	7	YES
	8	2
	9	

5.137 Download of PLC Retain Variables: PVR

MWCX device group

Designation **PVR** **P**LC **V**ariable **R**etain Backup

Explanation Download of PLC retain variables.

FI command **BW_PVR1!(1)** **(Single Write)**

(1) = Download file with path details.

Note: File and path details must be enclosed in inverted commas.
The separator "!" is used in this command.

Response Structure The response to the "PVR1" FI command consists of three lines, each with one column. The meaning of the elements is as follows:

- Line 1 = Job ID [01...20]
(see Chapter "FI Commands for the MPCX Device Group: IFJ").
- Line 2 = FI command
[String, in accordance with Chapter "Elements of the FI Command"]
- Line 3 = FI Job Error Code
(see Chapter "Error Codes")

Example PVR1 00_BW_PVR1!"D:\Program Files\Indramat\Mtgui\Temp\download.ini"/3

FI command		**00_BW_PVR1!"D:\Program Files\Indramat\Mtgui\ Temp\download.ini" /3**
Line	**Column**	**Answer**
1	1	01
2	1	00_BW_PVR1!"D:\Program Files\Indramat\Mtgui\ Temp\download.ini" /3
3	1	0

Structure of Download File The structure of the "download.ini" file used in this example corresponds to an Ini file in Windows.

Note: Care must be taken in the use of upper and lower case letters.

5.138 Upload of PLC Retain Variables: PVR

MWCX device group

Designation **PVR** **P**LC **V**ariable **R**etain Backup

Explanation PLC retain variables are uploaded via all active processes.

FI command **BR_PVR1!(1)** **(Single Read)**

(1) = Upload file with path details

Note: Enclose file and path details in inverted commas.
The separator "!" is used in this command.

Response Structure The response to the "PVR1" FI command consists of three lines, each with one column. The meaning of the elements is as follows:

- Line 1 = Job ID [01...20]
  (see Chapter "FI Commands for the MPCX Device Group", IFJ).
- Line 2 = FI command
  [String, in accordance with Chapter "Elements of the FI Command"]
- Line 3 = FI Job Error Code (see Chapter "Error Codes")

Example PVR 00_BR_PVR1!"D:\Program Files\Indramat\Mtgui\Temp\Upload.ini"/3

FI command		**00_BR_PVR1!"D:\Program Files\Indramat\Mtgui\Temp\upload.ini" /3**
Line	**Column**	**Answer**
1	1	01
2	1	00_BR_PVR1!"D:\Program Files\Indramat\Mtgui\Temp\upload.ini" /3
3	1	0

Structure of Upload File The upload file is structured in the Windows – "Ini" format structure.

Note: Care must be taken in the use of upper and lower case letters.

5.139 Reading the PLC Variable Declaration: PVT

MWCX device group

Designation **PVT** **P**LC **V**ariable **T**ype

Explanation A PLC variable has a particular type. To evaluate complex variables such as structures and arrays, their components and types must be read out.

Refer also to PVF, Reading Structured PLC Variables.

FI command Read the PLC variable type.

BR_PVT_(1)	**(Single Read)**
(1) = Identifier of the PLC variable	[acc. to declaration part of the PLC]

Response Structure One line with 2 columns is output for each element of the variables.

Line 1...n:	Column 1	Column 2

n = number of elements.

Value Range/Meaning of Columns

(1) = Identifier of the PLC variable	[acc. to declaration part of the PLC]
2 = Type	[see value range PVF]

Examples:
PLC: Reading of a variable

Assumption:
The "TEST" variable is declared as WORD in the PLC program.

FI command	**00_BR_PVT_TEST**	
Answer		
Line	**Column 1 (Name)**	**Name**
1	TEST	WORD

WinPCL: Reading a Variable

Assumption:
The WinPCL variable "TEST" is declared as WORD in WinPCL program entity "Prog".

FI command	**00_BR_PVT_:Prog.TEST**	
Answer		
Line	**Column 1 (Name)**	**Name**
1	TEST	WORD

PLC: Reading a Structure

Assumption:
The "TEST1" variable is declared as STRUCT in the PLC program.

```
STRUCT
        E1      BOOL
        E2      INT
        E3      SINT
END
```

FI command	**00_BR_PVT_TEST1**	
Answer		
Line	**Column 1**	**Column 2**
1	TEST1.E1	BOOL
2	TEST1.E2	INT
3	TEST1.E3	SINT

WinPCL: Reading a Structure

Assumption:
The WinPCL variable "TEST1" is declared as STRUCT in WinPCL program entity "Prog".

```
STRUCT
        E1      BOOL
        E2      INT
        E3      SINT
END
```

FI command	**00_BR_PVT_:Prog.TEST1**	
Answer		
Line	**Column 1**	**Column 2**
1	TEST1.E1	BOOL
2	TEST1.E2	INT
3	TEST1.E3	SINT

PLC: Reading an Array

Assumption:
The "TEST2" variable is declared as ARRAY in the PLC program.

```
ARRAY [
        0 .. 3
] OF    BOOL
```

FI command	**00_BR_PVT_TEST2**	
Answer		
Line	**Column 1**	**Column 2**
1	TEST2[0]	BOOL
2	TEST2[1]	BOOL
3	TEST2[2]	BOOL
4	TEST2[3]	BOOL

WinPCL: Reading an Array

Assumption:
The WinPCL variable "TEST2" is declared as ARRAY in WinPCL program entity "Prog".

```
ARRAY [
        0 .. 3
] OF    BOOL
```

FI command	**00_BR_PVT_:Prog.TEST2**	
Answer		
Line	**Column 1**	**Column 2**
1	TEST2[0]	BOOL
2	TEST2[1]	BOOL
3	TEST2[2]	BOOL
4	TEST2[3]	BOOL

PLC: Reading an Array of a Structure

Assumption:
The "TEST3" variable is declared as ARRAY in the PLC program.

```
ARRAY [
        0 .. 1
] OF    STRUCT1,
```

where STRUCT1 is declared as follows:

```
STRUCT
        E1      BOOL
        E2      INT
        E3      SINT
END
```

FI command	00_BR_PVT_TEST3	
Answer		
Line	**Column 1**	**Column 2**
1	TEST3[0].E1	BOOL
2	TEST3[0].E2	INT
3	TEST3[0].E3	SINT
1	TEST3[1].E1	BOOL
2	TEST3[1].E2	INT
3	TEST3[1].E3	SINT

WinPCL: Reading an Array of a Structure

Assumption:
The WinPCL variable "TEST3" is declared as ARRAY in WinPCL program entity "Prog".

```
ARRAY [
        0 .. 1
] OF    STRUCT1,
```

where STRUCT1 is declared as follows:

```
STRUCT
        E1      BOOL
        E2      INT
        E3      SINT
END
```

FI command	00_BR_PVT_:Prog.TEST3	
Answer		
Line	**Column 1**	**Column 2**
1	TEST3[0].E1	BOOL
2	TEST3[0].E2	INT
3	TEST3[0].E3	SINT
1	TEST3[1].E1	BOOL
2	TEST3[1].E2	INT
3	TEST3[1].E3	SINT

Assumption:
The data types are output according to IEC1131.

See also command PVF.

5.140 Repositioning Data: REP

MWCX device group

Designation **REP** **REP**ositioning Data

Explanation The data for re-approaching to contour of the selected device of the MWCX device group is read.

At the start of repositioning, the fixed data (end/setpoint values) can be called up with a 1st read command (CR_REP1). After that, the variable data must be called up repeatedly through a 2nd read command (CR_REP2, or CC_REP2) to show the actual status. With both commands, the reference system (machine or work piece coordinates) can be selected via the 2nd parameter.

FI command **CR_REP1_(1)_(2)** **(Single Read)**

(1) = NC process number [0...6]

(2) = Reference system [1...2] Machine/ workpiece coordinates

Response Structure 16 lines with varying numbers of columns are output:

- Line 1 contains the adjusting bit mask (13 bit values for the axes X …. S3, tool) which informs on whether an axis has already been adjusted.
- Line 1 contains the repos status (5 bit values).
- Line 3 to 14 for each possible axis (X ... S3) axis name, repos end position with full resolution, unit and repos end position with limited resolution.
- Line 15 contains the setpoint M functions of the spindles S1... S3.
- Line 16 contains the setpoint magazine position.

Example REP1 Read the fixed repositioning data in the NC process 0 of device address 00. The values are to be indicated in machine coordinates.

FI command		**00_CR_REP1_0_1**	
Line	**Column**	**Answer**	
1	1	0,1	Adjustment bit: X axis
	2	0,1	Adjustment bit: Y axis
	3	0,1	Adjustment bit: Z axis
	4	0,1	Adjustment bit: U axis
	5	0,1	Adjustment bit: V axis
	6	0,1	Adjustment bit: W axis
	7	0,1	Adjustment bit: A axis
	8	0,1	Adjustment bit: B axis
	9	0,1	Adjustment bit: C axis
	10	0,1	Adjustment bit: S1 axis
	11	0,1	Adjustment bit: S2 axis
	12	0,1	Adjustment bit: S3 axis
	13	0,1	Adjustment bit: tool axis

2	1	0,1	Status bit: repos active
	2	0,1	Status bit: restart active
	3	0,1	Status bit: NPV data changed
	4	0,1	Status bit: tool corrections changed
	5	0,1	Status bit: repos/restart data prepared
3...14	1	X ... S3	Axis designation
	2	0.0000 ... +-999.9999	End position (full resolution)
	3	[mm], [inch], [deg]	Unit
	4	0.00 ... +- 999.99	End position (limited resol.)
15	1	M103 ... M119	Setpoint M function S1
	2	M203 ... M219	Setpoint M function S2
	3	M303 ... M319	Setpoint M function S3
16	1	1 ... 999	Setpoint magazine position

FI command

Reading variable repositioning data.

CR_REP2_(1)_(2)	**(Cyclic Read)**
CC_REP2_(1)_(2)	**(Cyclic Read)**
CB_REP2_(1)_(2)	**(Cyclic Break)**
(1) = NC process number	[0...6]
(2) = Reference system	[1...2] Machine/ workpiece coordinates

Response Structure

16 lines with varying numbers of columns are output:

- Line 1 contains the adjusting bit mask (13 bit values for the axes X …. S3, tool) which informs on whether an axis has already been adjusted.
- Line 2 contains the repos status (5 bit values).
- Line 3 to 14 for each possible axis (X ... S3) current setpoint value (full resolution), unit, current setpoint value (limited resolution), repos distance to go (full resolution), unit and repos distance to go (limited resolution).
- Line 15 contains the setpoint M functions of the spindles S1... S3.
- Line 16 contains the setpoint magazine position.

Example REP2

Read the variable repositioning data in the NC process 0 of device address 00. The values are to be indicated in machine coordinates.

FI command		**00_CR_REP2_0_1**	
Line	**Column**	**Answer**	
1	1	0,1	Adjustment bit: X axis
	2	0,1	Adjustment bit: Y axis
	3	0,1	Adjustment bit: Z axis
	4	0,1	Adjustment bit: U axis
	5	0,1	Adjustment bit: V axis
	6	0,1	Adjustment bit: W axis
	7	0,1	Adjustment bit: A axis
	8	0,1	Adjustment bit: B axis
	9	0,1	Adjustment bit: C axis
	10	0,1	Adjustment bit: S1 axis
	11	0,1	Adjustment bit: S2 axis
	12	0,1	Adjustment bit: S3 axis
	13	0,1	Adjustment bit: tool axis
2	1	0,1	Status bit: repos active
	2	0,1	Status bit: restart active
	3	0,1	Status bit: NPV data changed
	4	0,1	Status bit: tool corrections changed
	5	0,1	Status bit: repos/restart data prepared
3...14	1	0.0000 ... +-999.9999	Set path (full resolution)
	2	[mm], [inch], [deg]	Unit
	3	0.00 ... +- 999.99	Set path (limited resol.)
	4	0.00 ... +- 999.99	Distance to go (full resol.)
	5	[mm], [inch], [deg]	Unit
	6	0.00 ... +- 999.99	Distance to go (limited resol.)
15	1	M103 ... M119	Setpoint M function S1
	2	M203 ... M219	Setpoint M function S2
	3	M303 ... M319	Setpoint M function S3
16	1	1 ... 999	Setpoint magazine position

Reference to Literature

See chapter entitled "Literature" [11].

5.141 Relative Axis Position: RPO

MWCX device group

Designation **RPO** **R**elative Axis **PO**sition

Explanation In respect of a preset basic position value, the difference to the actual position is to be indicated.

With the optional 5th parameter, either mm or Inch can be selected for the conversion of the result. The basic position value will likewise be returned in this measuring unit. If the given axis is no linear axis but a rotary axis or spindle, an optional 5th parameter (mm or Inch) is ignored, and the measuring unit is according to the axis (units, deg, or 1/min).

FI command

CR_RPO_(1)_(2)_(3)_(4){_(5)}	**(Single Read)**
CC_RPO_(1)_(2)_(3)_(4){_(5)}	**(Cyclic Read)**
CB_RPO_(1)_(2)_(3)_(4){_(5)}	**(Cyclic Break)**

(1) = CNC process number	[0...6]
(2) = Axis meaning	[0...11]
(3) = System of coordinates	[1...2]
(4) = Basic position value	
(5) = Requested measuring unit (opt.)	[mm, inch]

Response Structure One line is output with 4 columns for the axis designation, difference position, unit and the difference position rounded off according to the parameter "indicated decimal places".

Line 1	Column 1	Column 2	Column 3	Column 4

Meaning of the Columns

1 = Axis name	[according to settings of axis parameters]
2 = Relative position	[with reference to the basic position value]
3 = Unit	[according to settings of process parameters and required measurement system]
4 = Relative position	[as Column 2, but rounded up or down according to the parameter "indicated decimal places"]

Note: If the given axis is not defined then the response in all columns is [--].

Example 1 RPO Indicate the relative position of the Y axis at the basic positions value 12.3456 Inch of the workpieces system of coordinates in process 0:

Assumption:
The Y axis stands at the 12,0000 mm position
Parameter "Indicated decimal places" = 3

FI command	**00_CR_RPO_0_1_2_12.3456**			
Answer				
Line	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	Y1	-0.3456	[mm]	-0.346

Example 2 RPO Indicate the relative position of the Y axis at the basic positions value 1.23456 Inch of the workpieces system of coordinates in process 0:

Assumption:
The Y axis stands at the 12,0000 mm position (=0.47244 inch)
Parameter "Indicated decimal places" = 3

FI command	**00_CR_RPO_0_1_2_1.23456_inch**			
Answer				
Line	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	Y1	-0.7621	[inch]	-0.762

Reference to Literature See chapter entitled "Literature" [11].

5.142 SFC Diagnosis Data: SDD

MWCX device group

Designation **SDD** **S**FC **D**iagnosis **D**ata

Explanation Data for step chain diagnosis is output. Depending on the FI command this data can concern disrupted steps, actions, transitions or a definite ID to display the action or transition.

FI command Output the disrupted step of a step chain.

BR_SDD1!(1)!(2)	**(Single Read)**
(1) = Module number	[1...99]
(2) = SFC entity name	[ASCII characters]

Note: The separator "!" is used in this command.

Response Structure The following table shows the general structure of the FI command "SDD1".

Line 1	**Column 1**	**...**	**Column 7**

Meaning of the Columns

1 = Step name	[ASCII characters]
2 = Detail type	[1 = action block, 2 = action network, 3 = transition]
3 = Detail name	[ASCII characters]
4 = POU ID	[ASCII characters]
5 = Detail morpheme	[ASCII characters] (DWORD, decimal)
6 = Error ID	[ASCII characters] (DWORD, decimal)
7 = POU entity name	[ASCII characters]

Example SDD1 Query disrupted step of the "clamp" chain in module 3 in control unit 0.

FI command		**00_BR_SDD1!3!Station03A.Clamp**
Line	**Column**	**Answer**
1	1	Open
	2	1
	3	Aopen
	4	SFC_1_2
	5	98243823
	6	34985304
	7	Station2.Module3

FI command

Output the faulty action, monitor error or transition of a disrupted step.

BR_SDD2!(1)!(2)!(3)	**(Single Read)**
(1) = Module number	[1...99]
(2) = SFC entity name	[ASCII characters]
(3) = Step name	[ASCII characters]

Note: The separator "!" is used in this command.

Response Structure

The following table shows the general structure of the FI command "SDD2".

Line 1	**Column 1**	**...**	**Column 6**

Meaning of the Columns

1 = Detail type	[1 = action block, 2 = action network, 3 = transition]
2 = Detail name	[ASCII characters]
3 = POU ID	[ASCII characters]
4 = Detail morpheme	[ASCII characters] (DWORD, decimal)
5 = Error ID	[ASCII characters] (DWORD, decimal)
6 = POU entity name	[ASCII characters]

Example SDD2

Query faulty action of the disrupted step "open" of the "clamp" chain in module 3 in control unit 0.

FI command		**00_BR_SDD1!3!Station03A.Clamp_Open**
Line	**Column**	**Answer**
1	1	1
	2	AOpen
	3	SFC_1_2
	4	98243823
	5	34985304
	6	Station2.Module3

FI command

Output the definite ID to display the action, monitor error or transition.

BR_SDD3!(1)!(2)!(3)!(4)	**(Single Read)**
(1) = Module number	[1...99]
(2) = SFC entity name	[ASCII characters]
(3) = Detail type	[1 = action block, 2 = action network, 3 = transition]
(4) = Detail name	[ASCII characters]

Note: The separator "!" is used in this command.

Response Structure

The following table shows the general structure of the FI command "SDD3".

Line 1	**Column 1**	**...**	**Column 4**

Meaning of the Columns

1 = POU ID	[ASCII characters]
2 = Detail morpheme	[ASCII characters] (DWORD, decimal)
3 = Error ID	[ASCII characters] (DWORD, decimal)
4 = POU entity name	[ASCII characters]

Example SDD3

Query ID to display the action "aOpen" of the "clamp" chain in module 3 in control unit 0.

FI command		00_BR_SDD3!3!Station03A.Clamp!1!aOpen
Line	**Column**	**Answer**
1	1	SFC_1_2
	2	98243823
	3	34985304
	4	Station2.Module3

FI command

Output the I/O addresses to display a detail.

BR_SDD4!(1)!(2)!(3)!(4)	**(Single Read)**
(1) = Module number	[1...99]
(2) = SFC entity name	[ASCII characters]
(3) = Detail type	[1 = action block, 2 = action network, 3 = transition]
(4) = Detail name	[ASCII characters]

Note: The separator "!" is used in this command.

Response Structure

The following table shows the general structure of the FI command "SDD4".

Line 1-n	Column 1	Column 2

Meaning of the Columns

1 = Variable morpheme	[ASCII characters] (DWORD, decimal)
2 = I/O address	[ASCII characters]

Example SDD4

Query I/O addresses to display the action "aOpen" of the "clamp" chain in module 3 in control unit 0.

Three variables have an I/O address.

FI command		00_BR_SDD4!3!Station03A.Clamp!1!aOpen
Line	**Column**	**Answer**
1	1	98243823
	2	%I3.2.0
2	1	40923423
	2	%Q23.21.7
3	1	34985304
	2	%I100.3.5

FI command

Determine the multilingual comments for displaying a detail.

BR_SDD5!(1)!(2)!(3)!(4)	**(Single Read)**
(1) = Module number	[1...99]
(2) = SFC entity name	[ASCII characters]
(3) = Detail type	[1 = action block, 2 = action network, 3 = transition]
(4) = Detail name	[ASCII characters]

Note: The separator "!" is used in this command.

Response Structure

The following table shows the general structure of the FI command "SDD5".

Line 1-n	Column 1	Column 2

Meaning of the Columns

1 = Comment morpheme	[ASCII characters] (DWORD, decimal)
2 = New comment	[ASCII characters]

Example SDD5

Query comments to display the action "aOpen" of the "clamp" chain in module 3 in control unit 0.

Two comments are replaced by another text.

FI command		**00_BR_SDD5!3!Station03A.Clamp!1!aOpen**
Line	**Column**	**Answer**
1	1	98243823
	2	Clamp open
2	1	40923423
	2	Clamp closed

FI command

Output the action that has not been performed, or the transition of a step calculated based on the online status.

BR_SDD6!(1)!(2)!(3)	**(Single Read)**
(1) = Module number	[1...99]
(2) = SFC entity name	[ASCII characters]
(3) = Step name	[ASCII characters]

Note: The separator "!" is used in this command.

Response Structure

The following table shows the general structure of the FI command "SDD6".

Line 1	Column 1	...	Column 6

Meaning of the Columns

1 = Detail type	[1 = action block, 3 = transition]
2 = Detail name	[ASCII characters]
3 = POU ID	[ASCII characters]
4 = Detail morpheme	[ASCII characters] (DWORD, decimal)
5 = Error ID	[ASCII characters] (DWORD, decimal)
6 = POE entity name	[ASCII characters]

Example SDD6

Query the action that has not been performed for the step "open" of the "clamp" chain in module 3 in control unit 0.

FI command		**00_BR_SDD6!3!Station03A.Clamp_Open**
Line	**Column**	**Answer**
1	1	1
	2	AOpen
	3	SFC_1_2
	4	98243823
	5	34985304
	6	Station2.Module3

FI command

Determine the module number of a step chain.

BR_SDD7!(1)	**(Single Read)**
(2) = SFC instances name	[ASCII characters]

Note: The separator "!" is used in this command.

Response Structure

The following table shows the general structure of the FI command "SDD7".

Line 1	Column 1

Meaning of the Columns

1 = Module number [1...99]

Example SDD7

Inquiry of the module of the chain "clamp".

FI command		**00_BR_EXD1!Station03A.Clamp**
Line	**Column**	**Answer**
1	1	3

5.143 Set the Device Status Information: SDS

MWCX device group

Designation

SDS Set **D**evice **S**tatus

Explanation

By this command, the device status information can be set; here, the configuration file IND_DEV.INI is adjusted as well.

Note: When this command is transmitted, the following system messages are generated:
MSG_DEVICEOFF or MSG_DEVICE_ON !

FI command

With this command, the device status information of **ALL** defined devices can be set.

BW_SDS1_(1)	**(Single Write)**
Device status information to be set	0 = Device status information OFF 1 = Device status information ON

Response Structure

The following table shows the general structure of the response to the "SDS1" FI command.

Line 1	Column 1

Value Range/Meaning of Columns

1 = Status report [(P_ACK)]

Example SDS1

Set device status information to OFF for **ALL** defined devices.

FI command		**00_BW_SDS1_0**
Line	**Column**	**Answer**
1	1	(P_ACK)

FI command

With this command, the device status information for a selected device can be set.

BW_SDS2_(1)	**(Single Write)**
(1) = Device status information to be set	0 = Device status information OFF 1 = Device status information ON

Response Structure

The following table shows the general structure of the response to the "SDS2" FI command.

Line 1	Column 1

Value Range/Meaning of Columns

1 = Status report [(P_ACK)]

Example: SDS2

Set device status information to OFF for the selected device 00.

FI command		**00_BW_SDS2_0**
Line	**Column**	**Answer**
1	1	(P_ACK)

5.144 Setting the FI Exclusive Mode: SEM

MWCX device group

Designation

SEM **S**et FI **E**xclusive **M**ode

Explanation

This command is used to activate FI Exclusive mode for the selected device address.

FI Exclusive mode: In this mode, **ALL** the processes logged in at the FI – **with the exception** of the process issuing the SEM command – are blocked from data communication with this device address. This mode is used for example for data communication which must NOT be interrupted by other data requests. However, it is **imperative** that this FI Exclusive mode is deleted once more through the DEM command.

FI command

BW_SEM1 **(Single Read)**

Response Structure

The following table shows the general structure of the response to the FI command "BW_SEM1". A line of 1 column is output.

Line 1	Column 1

Value Range/Meaning of Columns

1 = Status message (P_ACK) (P_ACK)

Example SEM1

Activate FI Exclusive mode for device address 0.

FI command		**00_BW_SEM1**
Line	**Column**	**Answer**
1	1	(P_ACK)

5.145 Sequencer Data: SFD

MWCX device group

Designation

SFD **S**FC **D**ata

Explanation

Data for a step chain is output. Depending on the FI command this can concern a step chain comment, POU name, step comment, maximum time, action / transition / monitor error name (comment), qualifier and time value.

FI command

Query the data for a step chain

BR_SFD1!(1)!(2) **(Single Read)**

(1) = Module number [1...99]

(2) = SFC entity name [ASCII characters]

Note: The separator "!" is used in this command.

Response Structure

The following table shows the general structure of the "SFD1" FI command.

Line 1	Column 1	Column 2

Meaning of the Columns

1 = Step chain comment	[ASCII characters]
2 = POU name	[ASCII characters]

Example SFD1

Query data of the "clamp" chain in module 3 in control unit 0.

FI command		**00_BR_SFD1!3!Station03A.Clamp**
Line	**Column**	**Answer**
1	1	Clamping device
	2	CLAMP

FI command

Query the data of a step.

BR_SFD2!(1)!(2)!(3)	**(Single Read)**
(1) = Module number	[1...99]
(2) = SFC entity name	[ASCII characters]
(3) = Step name	[ASCII characters]

Note: The separator "!" is used in this command.

Response Structure

The following table shows the general structure of the "SFD2" FI command. The number of lines depends on the number of actions and transitions.

If there are no details the line number is 1.

Line 1	Column 1	...	Column 3
Line 2...n:	**Column 1**	**...**	**Column 6**

Meaning of the Columns

Line 1

1 = Step comment	[ASCII characters]
2 = Maximum time	[ASCII characters]
3 = Minimum time	[ASCII characters]

Line 2...n:

1 = Detail type	[1 = action block, 3 = transition]
2 = Name	[ASCII characters]
3 = Comment	[ASCII characters]
4 = Boolean variable	[YES, NO]
5 = Qualifier	[ASCII characters]
6 = Time value	[ASCII characters]

Example SFD2

Data for the step "Open" in the "clamp" chain in module 3 on control unit 0.

FI command		**00_BR_SFD2!3!Station03A.Clamp!Open**
Line	**Column**	**Answer**
1	1	Open clamping device
	2	T#5s
	3	
2	1	1
	2	aOpen
	3	Clamp open
	4	NO
	5	D
	6	T#3s
3	1	3
	2	tOpen
	3	Clamping device is open
	4	NO
	5	
	6	

FI command

Output the data for a detail.

BR_SFD3!(1)!(2)!(3)!(4)	**(Single Read)**
(1) = Module number	[1...99]
(2) = SFC entity name	[ASCII characters]
(3) = Detail type	[1 = action block, 2 = action network, 3 = transition]
(4) = Detail name	[ASCII characters]

Note: The separator "!" is used in this command.

Response Structure

The following table shows the general structure of the "SFD3" FI command.

Line 1	**Column 1**	**Column 2**

Meaning of the Columns

1 = Comment	[ASCII characters]
2 = Boolean variable	[YES, NO]

Example SFD3

Data for the action "aOpen" in the "clamp" chain in module 3 on control unit 0.

FI command		**00_BR_SFD3!3!Station03A.Clamp!aOpen**
Line	**Column**	**Answer**
1	1	Clamp open
	2	NO

5.146 Sequencer Messages: SFE

MWCX device group

Designation **SFE** **SFC E**rror

Explanation The step chain messages of a module are output.

FI command Output all SFC messages.

For reasons of optimization, not all data is called up. Accordingly, the Diagnosis server must be informed that the data is required (see ADW).

BR_SFE1_(1)	**(Single Read)**
BC_SFE1_(1)	**(Cyclic Read)**
(1) = Module number	[1...99]

Output first SFC messages.

BR_SFE2_(1)	**(Single Read)**
BC_SFE2_(1)	**(Cyclic Read)**
(1) = Module number	[1...99]

Response Structure The following table shows the general structure of the FI commands "SFE1" and "SFE2". The number of lines depends on the number of messages pending.

If there are no messages, the number of lines is 0.

Line 1...n:	Column 1	...	Column 7

Meaning of the Columns

1 = Message text	[ASCII characters]
2 = SFC entity name	[ASCII characters]
3 = Step name	[ASCII characters]
4 = Time stamp day	[mm.dd.yyyy]
5 = Time stamp time	[hh:mm:ss]
6 = Type of error	[1 = time error, 2 = monitor error, 3 = monitor event]
7 = Is there condition analysis?	[YES, NO]

Example SFD1 All SFC messages from module 2 in control unit 0.

There are two messages.

FI command		**00_BR_SFE1_2**
Line	**Column**	**Answer**
1	1	TIME ERROR: Chain: chucking Step: up malfunction
	2	Station03A.Clamp
	3	Open
	4	01.27.2000
	5	11:56:32 AM
	6	1
	7	YES
2	1	ASSY ERROR: Chain: drilling Step: down malfunction
	2	Station02A.Drill
	3	Down
	4	01.27.200
	5	13:03:12
	6	2
	7	NO

Example SFE2 First SFC message from module 2 in control unit 0.

There are two messages.

FI command		**00_BR_SFE2_2**
Line	**Column**	**Answer**
1	1	TIME ERROR: Chain: chucking Step: up malfunction
	2	Station03A.Clamp
	3	Open
	4	01.27.2000
	5	14:56:32
	6	1
	7	YES

5.147 Sequencer Mode: SFM

MWCX device group

Designation **SFM** **SFC** **M**ode

Explanation Queries step chain mode.

FI command Query the mode of a step chain.

BR_SFM1!(1)!(2)	**(Single Read)**
BC_SFM1!(1)!(2)	**(Cyclic Read)**
(1) = Module number	[1...99]
(2) = SFC entity name	[ASCII characters]

Note: The separator "!" is used in this command.

Response Structure

The following table shows the general structure of the "SFM1" FI command.

Line 1	Column 1

Meaning of the Columns

1 = Mode [1 = time error, 2 = monitor error,
3 = monitor event, 10 = stop,
11 = auto, 12 = manual, 13 = jog]

Example SFM1

Query mode of the "clamp" chain in module 3 in control unit 0.

FI command		**00_BR_SFM1!3!Station03A.Clamp**
Line	**Column**	**Answer**
1	1	1

5.148 Software Installation Data: SID

MWCX device group

Designation **SID** **S**oftware **I**nstallation **D**ata

Explanation

Information is returned regarding installation. This information includes installation paths, the software version used, DLL mode, plus service pack and release information.

FI command

Read-in the installation data and/or the software version data

BR_SID1 **(Single Read)**

Response Structure

One line with 16 columns is output for the returned values.

Line 1	Column 1	...	Column 16

Meaning of the Columns

1 = Basic directory [EXE files of the BOF]

2 = FI installation directory [FI directory]

3 = Data directory [in accordance with BOF]

4 = GBO version [from INDRAMAT.ini]

5 = IF-DLL mode [from INDRAMAT.ini]

6 = IF version [from INDRAMAT.ini from DLL mode 400]

7 = Service pack info [from INDRAMAT.ini from DLL mode 420]

8 = Release info [from INDRAMAT.ini from DLL mode 420]

9 = IF-Build-Info [in accordance with Build process]

10 = Current context name [in accordance with the installation]

11 = Physical installation path [in accordance with the installation]

12 = Complete IF version indication string

13 = WinPCL build number [in accordance with WinPCL]

14 = Version number of the PLC compiler [in accordance with WinPCL]

15 = Version number of the PLC linker [in accordance with WinPCL]

16 = Version number of the PLC data basis [in accordance with WinPCL]

17 = Platform version

Example SID1 Return information on the current installation.

FI command		00_BR_SID1
Line	**Column**	**Answer**
1	1	--
	2	D:\Program Files\Indramat\MTGUI\Bin
	3	--
	4	005-22Vxx
	5	07.20
	6	07V00
	7	--
	8	--
	9	Build 3124 Mar 6 2003 08:53:55
	10	MTGUI_0-23T01 B3327
	11	D:\Program Files\Indramat\MTGUI\
	12	FI: 07V00 DLL-Mode: 07.20 Build 3124 Mar 6 2003 08:53:55
	13	347.15.4.11
	14	771
	15	515
	16	78
	17	Platform: 02V01 Build: 3214

Note: Refer to FI command "PHD" of the MPCX for working with absolute paths.

5.149 Servo Lag of an Axis: SLA

MWCX device group

Designation **SLA** **S**ervo **LA**g

Explanation The current servo lag of a selected axis of the MWCX device group is read out. The FI command "SLA1" returns the servo lag of an axis, related to the code of the axis meaning. The FI command "SLA2", on the other hand, returns the servo lag of an axis, related to the physical axis number.

FI command Output the servo lag of the selected axis of the device specified, related to the code of the axis meaning.

Using the optional third parameter it is possible to pre-select conversion of the result into mm or inches. If, however, a spindle is selected as an axis, indicating a measurement system serves no purpose.

CR_SLA1_(1)_(2){_(3)}	**(Single Read)**
CC_SLA1_(1)_(2){_(3)}	**(Cyclic Read)**
CB_SLA1_(1)_(2){_(3)}	**(Break Cyclic Read)**
(1) = NC process number	[0...6]
(2) = Axis meaning	[0...11; 20]; (see Chapter 6.2 "Data Tables")
(3) = Required measurement system (opt.)	[mm, inch]

FI command

Output the servo lag of the selected axis of the device specified, related to the physical axis number.

Using the optional second parameter it is possible to pre-select conversion of the result into mm or inches. If, however, a spindle is selected as an axis, indicating a measurement system serves no purpose.

CR_SLA2_(1){_(2)}	**(Single Read)**
CC_SLA2_(1){_(2)}	**(Cyclic Read)**
CB_SLA2_(1){_(2)}	**(Break Cyclic Read)**
(1) = Physical axis number	[1...32]
(2) = Required measurement system (opt.)	[mm, inch]

Response Structure

The following table shows the general structure of the response to the FI commands "SLA1" and "SLA2". One line is output with 4 columns for the axis designation, servo lag, unit and the servo lag limited to "indicated decimal places".

Line 1	Column 1	Column 2	Column 3	Column 4

Value Range/Meaning of Columns

1 = Axis name	[according to settings of axis parameters]
2 = Servo lag	[according to settings of process parameters]
3 = Unit	[according to settings of process parameters: [mm, inch]
4 = Servo lag	[as Column 2, but rounded up or down according to the parameter "indicated decimal places"]

Note: If the specified axis is not defined in the selected NC process then the response in all columns is [--].

Example SLA1

Read the servo lag of the Z axis in NC process 0 of device address 00.

FI command	**00_CR_SLA1_0_2**			
Answer				
Line	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	Z1	2.9124	[mm]	2.912

Example SLA1

Read the servo lag of the Z axis in NC process 0 of device address 00. Values are displayed in inches:

FI command	**00_CR_SLA1_0_2_inch**			
Answer				
Line	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	Z1	0.1147	[inch]	0.115

Example SLA2

Read the servo lag of the Z axis (e.g., physical axis number = 3) at device address 00.

FI command	**00_CR_SLA2_3**			
Answer				
Line	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	Z1	2.9124	[mm]	2.912

Reference to Literature

See chapter entitled "Literature" [40].

5.150 PLC Long Identification: SLI

MWCX device group

Designation **SLI** **P**LC **L**ong **I**dentification

Explanation Returns the unit data from the PLC long identification.

FI command Read PLC long identification.

BR_SLI **(Single Read)**

Response Structure One line with 15 columns is output for the returned values.

Line 1	Column 1	Column...	Column 15

Value Range/Meaning of the Columns

1 =	Device address	[00...63]
2 =	Program number	[01...99]
3 =	Project name	[max. 8 ASCII characters]
4 =	Program name	[max. 8 ASCII characters]
5 =	User name	[acc. to password entry]
6 =	Program length	[bytes]
7 =	Compilation time	[LONG] (coded in long value)
8 =	Compilation date	[8 ASCII characters]
9 =	Compilation time	[8 ASCII characters]
10 =	Download time	[LONG] (coded in long value)
11 =	Download date	[8 ASCII characters]1
12 =	Download time	[8 ASCII characters]
13 =	Version of PLC long identification	[LONG]
14 =	RUN flags	[HEX value]
15 =	Compiler info	[LONG]

Example SLI Read the unit data from the PLC long identification.

FI command		00_BR_SLI
Line	**Column**	**Answer**
1	1	02
	2	01
	3	--
	4	MOT12
	5	TEST
	6	17672
	7	630163960
	8	15.12.99
	9	17:15:48
	10	630163961
	11	15.12.99
	12	17:15:50
	13	2
	14	0x0000

FI command		00_BR_SLI
Line	Column	Answer
	15	13

Reference to Literature see chapter entitled "Literature" [30].

5.151 SERCOS Parameters: SPA

MWCX device group

Designation **SPA** **S**ERCOS **PA**rameter

Explanation A SERCOS drive parameter is output or written. Each parameter consists of 7 elements, whereby any combination of elements can be selected by element coding.

FI command

BR_SPA1_(1)_(2)_(3)	**(Single Read)**
BC_SPA1_(1)_(2)_(3)	**(Cyclic Read)**
BB_SPA1_(1)_(2)_(3)	**(Break Cyclic Read)**
BW_SPA1_(1)_(2)_(3)	**(Single Write)**
(1) = Drive address	[1...32]
(2) = Parameter No.	in the format: X-Y-ZZZZ
(3) = Element coding	[standard or advanced format]

Parameter No.

Format X-Y-ZZZZ	Value Range
X	S = standard data P = product data
Y	[0...7] = parameter record
Z	[0...4095] = data block no.

Element Coding Element coding in standard format allows individual elements, such as the operating date, to be requested. If several elements are to be read out in one request, then the element coding can be OR'd in advanced format, e.g. operating date (0x40) and unit (0x08) produces OR'd (0x48) → 48.

The advanced format 0x80 has priority over 0x40.

Element	Standard Format	Advanced Format	Format:	Example
Data status	S:	01H	Hexadecimal word	0x0000
Name	The marked section is then printed out.	02H	String	NC cycle time (TNcyc)
Attribute	A	04H	Hexadecimal double word	0x60110001
Unit	U	08H	String	µs
Min. input value	L	10H	Decimal word	2000
Max. input value	H	20H	Decimal word	20000
Operating date	D	40H	(see Displaying the Operating Date	
Operating date, when no list		80H		

Displaying the Operating Date The display of the operating date depends on the parameter number requested.

Decimal

Decimal values are given as floating points, e.g. 1.5. Leading spaces, zeros, plus and minus signs as well as trailing spaces are allowed.

Hexadecimal

Hexadecimal values are displayed by "0x...", e.g. 0x80. Up to a maximum of eight positions are allowed. Leading or trailing spaces are allowed. Leading additional zeros or plus and minus signs are not allowed.

Binary (max. 32 characters)

Leading or trailing spaces are allowed. The decimal point serves as separator:

e.g., 1111.0000.1010.1100.1111.0000.1010.1100

Note: Leading additional zeros or plus and minus signs are not allowed.

ID number

The following table shows the general way in which the ID number is displayed:

Format X-Y-ZZZZ	Value Range
X	S = standard data P = product data
Y	[0..0.7] = parameter record
Z	[0...4095] = data block no.

(see example SPA1/write).

Lists of Variable Length

Lists always begin with two decimal numbers for the actual length and maximum length of the list. The length specification refers to the length of the list in the drive and therefore designates the number of bytes for storage (storage bytes). The number of elements in the list can be calculated using the attribute. The list elements are displayed according to the attribute. All parts of the list are separated from each other by a line feed ("\n").

Example:

Parameter S-0-0017, IDN list of all parameters

"400\n400\nS-0-0001\nS-0-0002\n..."

ASCII List

ASCII lists are a special form of variable length lists. The individual string characters are not separated by a line feed. When displaying the lists, a distinction is made between standard format and advanced format. In standard format, only the character string is returned; in advanced format, the actual length and the maximum length of the list (string) are also transmitted.

Example:

Parameter S-0-0030, operation date
Standard format: "DKC2.1-SSE-01V09"
Advanced format: "16\n16\nDKC2.1-SSE-01V09"

Response Structure

The following table shows the general structure of the response to the FI command "SPA1". Line 1 is output both when reading and when writing. Additional lines are only output when reading depending on the element coding.

Note: If the element coding has been requested in standard format then the first line is not applicable.

Line 1 is a status line that either contains the Sercos error or displays the successful processing of the FI command. If the command has been processed successfully, then columns 1 and 3 contain the value [0x0000].

The number of the drive that reports the SERCOS error is output in the second column of the first line.

Line	Column 1	Column 2	Column 3	Column 4
1	<Sercos error>	<Drive no. SERCOS error>	0x0000	0x0000
2	Read: 1. Element corresponding to the element coding.			
...	...			
n	Reading: (n-1). Element corresponding to the element coding.			

Example SPA1/ read Read parameter S-0-0003 of the 3rd drive (element coding 0x48)

FI command	00_BR_SPA1_3_S-0-0003_48			
Answer				
Line	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	0x0000	0x0000	0x0000	0x0000
2	µs			
3	2000			

Example SPA1/ write Write the ID number P-0-0037 in parameter S-0-0305 of the 3rd drive (element coding 0x40).

Technical background:

- Realtime status bit 1 is to be assigned the trigger status word of the oscilloscope function of a DIAX04 drive.

FI command	Value to be written: : P-0-0037 00_BW_SPA1_3_S-0-0305_40			
Answer				
Line	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	0x0000	0x0003	0x0000	0x0000

Reference to Literature See chapter entitled "Literature" [41].

See chapter entitled "Literature" [46].

Explanation A Sercos parameter of a drive is read out of the transferred Sercos data file. Each Sercos parameter consists of 7 elements, whereby any combination of elements can be selected by element coding.

FI command **BR_SPA3_(1)_(2)_(3)_(4)** **(Single Read)**

(1) = Drive address	[1..32] with MTCNC systems [1..16] with MTA systems
(2) = Parameter No.	in the format: X-Y-ZZZZ
(3) = Element coding	[standard or advanced format]
(4) = Complete Sercos data file name	

Note: File and path details must be enclosed in inverted commas.

For a more detailed description of parameter no., element coding, displaying the operating date, see the description of the ‘SPA1’ command.

Response Structure The structure of the response data corresponds to the ‘SPA1’ command.

Example SPA3 Read the parameter S-0-0003 of the 3rd drive (element coding 0x48) of device 0 out of the Sercos data file D:\ SERCPAR .DAT.

FI command	**00_BR_SPA3_3_S-0-0003_48_"D:\SERCPAR.DAT"**			
Answer				
Line	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	0x0000	0x0000	0x0000	0x0000
2	µs			
3	2000			

Explanation A Sercos parameter of a drive is read out of the transferred Sercos data file and written into the addressed drive. Only the operating date can be written.

FI command **BR_SPA3_(1)_(2)_(3)_(4)** **(Single Read)**

(1) = Drive address [1..32] with MTCNC systems
[1..16] with MTA systems

(2) = Parameter No. in the format: X-Y-ZZZZ

(3) = Element coding [standard or advanced format]

(4) = Complete Sercos data file name

Note: File and path details must be enclosed in inverted commas.

For a more detailed description of parameter no., element coding, displaying the operating date, see the description of the ‘SPA1’ command.

Response Structure The structure of the response data corresponds to the ‘SPA1’ command.

Example SPA3 Write the operating date of the Sercos parameter S-0-0305, which is saved in the Sercos data file, into drive 3 of device 0. The name of the Sercos data file is D:\SERCPAR.DAT.

FI command	**00_BR_SPA3_3_S-0-0003_48_"D:\SERCPAR.DAT"**			
Answer				
Line	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	0x0000	0x0003	0x0000	0x0000

Explanation The "SPA4" command is identical with the "SPA1" command. They only differ in one feature: with the "SPA4" command, the system makes up to 100 attempts to compensate any SERCOS error 0x13EA (SERCOS busy flag set).

FI command **BR_SPA1_(1)_(2)_(3)** **(Single Read)**

(1) = Drive address [1...32]

(2) = Parameter No. in the format: X-Y-ZZZZ

(3) = Element coding [standard or advanced format]

For a more detailed description of parameter no., element coding, displaying the operating date, see the description of the ‘SPA1’ command.

Response Structure The structure of the response data corresponds to the ‘SPA1’ command.

FI command **BW_SPA4_(1)_(2)_(3)** **(Single Write)**

(1) = Drive address [1...32]

(2) = Parameter No. in the format: X-Y-ZZZZ

(3) = Element coding [standard or advanced format]

For a more detailed description of parameter no., element coding, displaying the operating date, see the description of the ‘SPA1’ command.

The write value is passed as with the "SPA1" command.

Response Structure The structure of the response data corresponds to the ‘SPA1’ command.

5.152 Active SERCOS Phase Switch-Over: SPH

MWCX device group

Designation **SPH** **S**ERCOS **PH**ase

Explanation All drives within a SERCOS ring are in the same communication phase. The phase status can be read-out or changed by this command.

As a result of the ".._SPH1" commands, the (stable) end phase of the drive is returned, while the ".._SPH2" commands also indicate the transitional statuses:

FI command Read SERCOS phase:

BR_SPH1_(1)	**(Single Read)**
BR_SPH2_(1)	**(Single Read)**
BC_SPH1_(1)	**(Cyclic Read)**
BC_SPH2_(1)	**(Cyclic Read)**
BB_SPH1_{(1)}	**(Cyclic Break)**
BB_SPH2_{(1)}	**(Cyclic Break)**
(1) = Drive address	[1...32]

FI command Change over SERCOS phase:

BW_SPH1_(1)	**(Single Write)**
BW_SPH2_(1)	**(Single Write)**
(1) = Drive address	[1...32]

Value to be written Requested phase [2, 4]

Response Structure One line is output with four columns existing for reason of compatibility with SYNAX, SERCANS and ECODRIVE. They always return the value 0x0000 and a 2nd line with the actual value. With .._SPH1, this value can be one of {0,1,2,3,4 }, and with .._SPH2 one of { 0,1,2,3,4,80,81,82,83,84 }. Here, the values { 0 ... 4 } stand for the stable phase, and { 80 … 84 } for the transitional statuses.

Line 1	**Column 1**	**Column 2**	**Column 3**	**Column 4**
Line 2	**Column 1**			

Note: The value to be written is passed to the "acValue" parameter in the "DataTransfer" routine.

Example BR_SPH1
Read SERCOS Phase

Read the active phase of the first axis at device address 00.

FI command		00_BR_SPH1_1
Line	**Column**	**Answer**
1	1	0x0000
	2	0x0000
	3	0x0000
	4	0x0000
2	1	2

Example SPH2
Read SERCOS Phase

Read the active phase of the first axis at device address 00.

Assumption: The drive is in the boot-up phase.

FI command		00_BR_SPH2_1
Line	**Column**	**Answer**
1	1	0x0000
	2	0x0000
	3	0x0000
	4	0x0000
2	1	82

Example BW_SPH2
Write SERCOS Phase

Switch-over the first axis (write) after phase 4; phase 2 is active.

FI command		Value to be written: 4 00_BW_SPH2_1
Line	**Column**	**Answer**
1	1	0x0000
	2	0x0000
	3	0x0000
	4	0x0000
2	1	82

Reference to Literature See chapter entitled "Literature" [42].

5.153 Selected NC Program: SPP

MWCX device group

Designation **SPP** **S**elected **P**art-**P**rogram Number

Explanation The selected NC program is read.

FI command

CR_SPP_(1)	**(Single Read)**
CC_SPP_(1)	**(Cyclic Read)**
CB_SPP_(1)	**(Break Cyclic Read)**
(1) = NC process number	[0...6]

Response Structure The response to the FI command "SPP" consists of one line with two columns for the identifier of the NC memory and the number of the selected NC program.

Line 1	Column 1	Column 2

Value Range/Meaning of Columns		
	1 = NC memory	[A = NC memory A; B = NC memory B]
	2 = Number of selected NC program	[according to settings of process parameters]

Example SPP Read the selected NC program in NC process 0 of device address 00.

FI command	**00_CR_SPP_0**	
Answer		
Line	**Column 1**	**Column 2**
1	B	55

5.154 Reading or Writing Tool Data Record: TDA

MWCX device group

Designation **TDA** Tool **DA**ta

Explanation A complete tool data record consisting of basic data and defined cutter data is read from or written into the control unit.

FI command Read the complete tool data record. For this FI command, the tool data record is addressed via the NC process number, the tool memory and the location number.

BR_TDA1_(1)_(2)_(3) **(Single Read)**

BC_TDA1_(1)_(2)_(3) **(Cyclic Read)**

(1) = NC process number [0...6]

(2) = Tool memory [M = magazine/turret, S = spindle, G = grabber]

(3) = Location number [1...999]

Response Structure The following table shows the general structure of the response to the FI command "BR_TDA1". The number of lines depends on the number of cutters. The first line contains the basic data. The cutter data is listed from line 2 onwards. The basic data consist of 28 basic data elements, and the tool edge data of 40 tool edge data elements.

Line 1	Column 1	...	Column 28	
Line 2	Column 1	Column 2	...	Column 40
...	...	...	...	...
Line n+1	Column 1	Column 2		Column 40

n = number of cutters

Example TDA1 Read the complete tool data record

FI command		**03_BR_TDA1_0_M_21**
Line	**Column**	**Answer**
1	01	10156
	02	Cutter head D80
	03	M
	04	21
	05	1
	06	1
	07	2

	08	1
	09	-p
	10	0
	11	M 21
	12	M
	13	--
	14	M
	15	--
	16	[cycl]
	17	[mm]
	18	0
	19	0
	20	0.000000
	21	0.000000
	22	0.000000
	23	0.000000
	24	0.000000
	25	0.000000
	26	0.000000
	27	0.000000
	28	0.000000
2	01	1
	02	_
	03	100.000000
	04	5.000000
	05	0.000000
	06	0.000000
	07	0.0000
	08	0.0000
	09	104.8000
	10	40.0000
	11	0.0000
	12	0.0000
	13	0.0000
	14	0.0000
	15	0.0000
	16	0.0000
	17	0.0000
	18	0.0000
	19	-999.0000
	20	999.0000
	21	-999.0000

	22	999.0000
	23	-999.0000
	24	999.0000
	25	-999.0000
	26	999.0000
	27	0.0000
	28	0.0000
	29	0.0000
	30	0.0000
	31	0.000000
	32	0.000000
	33	0.000000
	34	0.000000
	35	0.000000
	36	0.0000
	37	0.0000
	38	0.0000
	39	0.0000
	40	0.0000

FI command

Write the complete tool data record. For this FI command, the tool data record is addressed via the NC process number, the tool memory and the location number.

Note: To create a tool data record, the name (ID) must be transferred to the device (see example "TDA1", Write Tool Data).

BW_TDA1_(1)_(2)_(3)	**(Single Write)**
(1) = NC process number	[0...6]
(2) = Tool memory	[M = magazine/turret, S = spindle, G = grabber]
(3) = Location number	[1...999]

Values to be written

The values to be written are passed in a table. First, the 3-digit code of the data element must be passed and then the value to be written must be passed. The first position addresses the data record (0 = basic data record, 1 to 9 the corresponding cutter data record) and the second and third positions address the actual data element (also refer to "Basic Data" and "Tool Edge Data").

Data Element Code

1. Position	2. Position	3. Position
0 = basic data record or 1...9 = cutter data record	two-digit data element number	

Note: The character "|" (= 0x7D) is used as separator between the number of the data element and the value to be written. The individual lines of the table are also separated by a "|". <Element number n> <|> <Value n> <|> <Element number m> <|> <Value m> <|>

Example TDA1
Write Tool Data Record

Write the following data elements of the tool data record:

- Element number 002: Name (ID) "drill Z72"
- Element number 008: Number of tool edges "1" and
- Element number 107: Length L1 "100"

Assumption:

- NC process number: 0
- Tool magazine: M = magazine and
- location number: 2

FI command	**03_BW_TDA1_0_M_2**
Values to be written **002|Drill Z72|008|1|107|100**	

FI command

Read the complete tool data record. For this FI command, the tool data record is addressed via the NC process number, the tool number and the index number.

BR_TDA2_(1)_(2)_(3)	**(Single Read)**
BC_TDA2_(1)_(2)_(3)	**(Cyclic Read)**
(1) = NC process number	[0...6]
(2) = Tool number	[1...9999999]
(3) = Index number	[1...9999]

Response Structure

The following table shows the general structure of the response to the FI command "BR_TDA2". The number of lines depends on the number of cutters. The first line contains the basic data. The cutter data is listed from line 2 onwards. The basic data consist of 28 basic data elements, and the tool edge data of 40 tool edge data elements.

Line 1	Column 1	...	Column 28	
Line 2	Column 1	Column 2	...	Column 40
...	...	...	...	...
Line n+1	Column 1	Column 2	...	Column 40

n = number of cutters

Example TDA2

Read the complete tool data record

FI command		**03_BR_TDA2_0_1_1**
Line	**Column**	**Answer**
1	01	10156
	02	Cutter head D80
	03	M
	04	21
	05	1
	06	1
	07	2
	08	1
	09	-p
	10	0
	11	M 21
	12	M

	13	--
	14	M
	15	--
	16	[cycl]
	17	[mm]
	18	0
	19	0
	20	0.000000
	21	0.000000
	22	0.000000
	23	0.000000
	24	0.000000
	25	0.000000
	26	0.000000
	27	0.000000
	28	0.000000
2	01	1
	02	_
	03	100.000000
	04	5.000000
	05	0.000000
	06	0.000000
	07	0.0000
	08	0.0000
	09	104.8000
	10	40.0000
	11	0.0000
	12	0.0000
	13	0.0000
	14	0.0000
	15	0.0000
	16	0.0000
	17	0.0000
	18	0.0000
	19	-999.0000
	20	999.0000
	21	-999.0000
	22	999.0000
	23	-999.0000
	24	999.0000
	25	-999.0000
	26	999.0000

	27	0.0000
	28	0.0000
	29	0.0000
	30	0.0000
	31	0.000000
	32	0.000000
	33	0.000000
	34	0.000000
	35	0.000000
	36	0.0000
	37	0.0000
	38	0.0000
	39	0.0000
	40	0.0000

Reference to Literature See chapter entitled "Literature" [43].

5.155 Loading Tool Data into the Control Unit: TDD

MWCX device group

Designation **TDD** **T**ool **D**ata **D**ownload

Explanation Downloading of a tool data record. After the tool list download has been initiated with "CR_TDI", the entire data for a tool is transferred into the control unit for each position of the tool memory. The data consists of a data record for the basic data and a data record for the cutter data for each cutter of the tool.

FI command Write the basic data or cutter data of a tool data record.

CW_TDD_(1)_(2)_(3)_(4)	**(Single Read)**
(1) = NC process number	[0...6]
(2) = Tool memory	[M = magazine/turret, S =spindle, G = grabber, P = change position]
(3) = Tool memory location	In the magazine/turret: [1...999] In the spindle: [1...4] In the gripper: [1...4] In the change position: [1...4]
(4) = Cutter number	[0 = basic data, 1...9 = cutter data]

Value to be written Tool data record [basic and cutter data]

Note: The value to be written is passed to the "acValue" parameter as an ASCII string in the "DataTransfer" routine.

A tool data record consists of the individual writable tool data of the basic and cutter data, each separated from one another by a space (see Basic Data, Cutter Data).

The tool name (element No. 2 = 1st writable data of the basic data) can itself contain any characters (including spaces) and should therefore be character filled with exactly 28 characters with spaces.

Depending on the parameter setting it is possible that some of the basic or cutter data might not be relevant. Such data should nonetheless be included in the data record, e.g., with 0!

Response Structure

One line is output with a column for acknowledgement of whether or not the FI command has been executed successfully.

(P_ACK) = **P**ositive **ACK**nowledge	Data element has been set

Example TDD

In NC process 0 of device 00, write into the control unit the data record for the basic data of the tool in the magazine at location number 2.

FI command		**00_CW_TDD_0_M_2_0 <Data record>**
Line	**Column**	**Answer**
1	1	(P_ACK)

<Data record> =
"Tool 1 1234567 1234 4 3 0xFF301900 0 0 1 6 5 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0"

Example TDD

In NC process 0 of device 00, write into the control unit the data record for the 3rd cutter of the tool in the magazine at location number 2.

FI command		**00_CW_TDD_0_M_2_3 <Data record>**
Line	**Column**	**Answer**
1	1	(P_ACK)

<Data record> =
"1 0xF000 100.0 5 20000 1 2 3 4.0 0.1 0.2 0.3 0.4 0.01 0.02 0.03 0.04 0.001 0.002 0.003 0.004 1 2 3.1 4.1 5 11 0.222 -0.0333 9 10"

Status bits

The values for the status bits shown in the examples must be entered as a hexadecimal number (0x...), whereby the sequence should begin with the most significant bit 32 (tool status) or bit 16 (cutter status).

As only part of the status bit can be changed by the user, the form of these changeable bits is given here as an example:

W.Status: 0xFF301900 = 1111 1111 0011 0000 0001 1001 0000 0000

S.Status: 0xF000 = 1111 0000 0000 0000

Refer to Basic Data and Cutter Data for the meaning of the status bits.

Explanation

Download of a tool data record from version 23 (the number of basic data has been extended to 32). After the tool list download has been initiated with CR_TDI, the entire data for a tool is transferred into the control unit for each position of the tool memory. The data consists of a data record for the basic data and a data record for the cutter data for each cutter of the tool.

FI command

Write the basic data or cutter data of a tool data record.

CW_TDD1_(1)_(2)_(3)_(4) **(Single Read)**

(1) = CNC process number	[0...6]
(2) = Tool memory	[M = magazine/turret, S =spindle, G = grabber, P = change position]
(3) = Tool memory location	In the magazine/turret: [1...999] In the spindle: [1...4] In the gripper: [1...4] In the change position: [1...4]
(4) = Cutter number	[0 = basic data, 1...9 = cutter data]

Value to be written

Tool data record [basic and cutter data]

Note: The value to be written is passed to the "acValue" parameter as an ASCII string in the "DataTransfer" routine.

A tool data record consists of the individual writable tool data of the basic and cutter data, each separated from one another by a space (see Basic Data, Cutter Data).

The tool name (element No. 2 = 1st writable data of the basic data) can itself contain any characters (including spaces) and should therefore be character filled with exactly 28 characters with spaces.

Depending on the parameter setting it is possible that some of the basic or cutter data might not be relevant. Such data should nonetheless be included in the data record, e.g., with 0!

Response Structure

One line is output with a column for acknowledgement of whether or not the FI command has been executed successfully.

(P_ACK) = **P**ositive **ACK**nowledge Data element has been set

Example TDD1

In CNC process 0 of device 00, write into the control unit the data record for the basic data of the tool in the magazine at location number 2.

FI command		**00_CW_TDD1_0_M_2_0 <Data record>**
Line	**Column**	**Answer**
1	1	(P_ACK)

<Data record> =
"Tool 1 1234567 1234 4 3 0xFF301900 0 0 1 6 5 1.0 2.0 3.0 4.0
5.0 6.0 7.0 8.0 9.0 0 0 0 0x0000"

Example TDD1

In CNC process 0 of device 00, write into the control unit the data record for the 3rd cutter of the tool in the magazine at location number 2.

FI command		**00_CW_TDD1_0_M_2_3 <Data record>**
Line	**Column**	**Answer**
1	1	(P_ACK)

<Data record> =
"1 0xF000 100.0 5 20000 1 2 3 4.0 0.1 0.2 0.3 0.4 0.01 0.02 0.03 0.04
0.001 0.002 0.003 0.004 1 2 3.1 4.1 5 11 0.222 -0.0333 9 10"

Status bits

The values for the status bits shown in the examples must be entered as a hexadecimal number (0x...), whereby the sequence should begin with the most significant bit 32 (tool status) or bit 16 (cutter status / tool group status).

As only part of the status bit can be changed by the user, the form of these changeable bits is given here as an example:

T. status: 0xFF301900 = 1111 1111 0011 0000 0001 1001 0000 0000

C. status: 0xF000 = 1111 0000 0000 0000

Refer to Basic Data and Cutter Data for the meaning of the status bits.

5.156 Tool List Download Escape: TDE

MWCX device group

Designation **TDE** **T**ool List **D**ownload **E**scape

Explanation A tool list download can be interrupted, e.g. when an error occurs. This turns the data downloaded into the controller up to the abortion invalid.

FI command Abortion of a tool list download.

CR_TDE_(1) **(Single Read)**

(1) = Process number [0 ... 6]

Response Structure The controller transmits one line with a column for the acknowledgement of an abortion recognition.

(P_ACK) = **P**ositive **ACK**nowledge Download has been aborted

Example CR_TDE Aborting a tool list download:

FI command		**00_CR_TDE_0**
Line	**Column**	**Answer**
1	1	(P_ACK)

Other tool list download commands See **CR_TDI, BW_TDF, CW_TDD1.**

5.157 Tool List Download Finish: TDF

MWCX device group

Designation **TDF** **T**ool List **D**ownload **F**inish

Explanation The controller is informed about the end of a tool list download by transmission of the ident string of the tool list. This command can also be used to delete a tool list if **NO** write value is passed.

FI command Finish a tool list download.

BW_TDF_(1) **(Single Write)**

(1) = Process number [0 ... 6]

Value to be written Tool list ident string Directory entry (also see CR_DIS4)

The ident string consists of 5 parts, each separated by a space.

1 =	Number in tool list directory	[01...99]
2 =	Name of the tool list	[always 32 characters, fill up with spaces if applicable]
3 =	Length of the tool list	[No. of bytes]
4 =	Date of creation/last change to the tool list	[DD.MM.YY]

5 = Time of creation/last change to the tool list [HH:MM:SS]

Note: The value to be written is passed to the "acValue" parameter as an ASCII string in the "DataTransfer" routine. The tool list name must comprise exactly 32 characters – fill up with spaces if applicable. If there is **NO** write value, this is used to delete the tool list, and accordingly also to delete the tool list ident string.

Response Structure

The controller transmits one line with a column for the acknowledgement of a successfully finished transmission.

(P_ACK) = **P**ositive **ACK**nowledge Download has been completed

Example BW_TDF

Finish a tool list download in process 0:

FI command		**00_BW_TDF_0 <Tool list ident string>**
Line	**Column**	**Answer**
1	1	(P_ACK)

Other tool list download commands

See **CR_TDI, CR_TDE, CW_TDD1.**

5.158 Initialize Tool List Download: TDI

MWCX device group

Designation

TDI **T**ool List **D**ownload **I**nitialize

Explanation

To download a tool list into the controller, the controller must be informed of the start of the download process so that the necessary preconditions are checked (no active process, etc.).

FI command

Initialize a tool list download.

CW_TDI_(1) **(Single Read)**

(1) = Process number [0 ... 6]

Value to be written

Short ID/date/time

Format:

Short ID = always 8 ASCII characters (fill up with spaces if applicable)

Date = day.month.year

Time = hour:minute:second

The individual elements are separated by one space each.

Example:

TOOL_LIST 12.08.03 13:05:54

Response Structure

If the preconditions for a download are met, the controller sends:

(P_ACK) = **P**ositive **ACK**nowledge Download has been initialized

If the preconditions for a download are not met, the controller sends:

(N_ACK) = **N**egative **ACK**nowledge Download has been refused

Example CW_TDI

Initialize a tool list download in process 0:

FI command		**00_CW_TDI_0**
Line	**Column**	**Answer**
1	1	(P_ACK)

Or, if the process is still active:

Error 1014 = BOF_NEGATIVE_ACKNOWLEDGE (N_ACK):

Line	Column	Answer
1	**1**	1 **(=N_ACK)**
	2	**37** (=text number of N_ACK)
	3	**0x00000000** (=additional information for some texts)
	4	**Process still active** (=text of the N_ACK error)

Other tool download commands

See **BW_TDF, CW_TDD1, CR_TDE.**

5.159 Access to Tool Data Record: TDR

MWCX device group

Designation

TDR **T**ool **D**ata **R**ecord

Explanation

Returns a complete basic data record and/or cutter data record of a tool.

FI command

Read the basic data record or cutter data record of a tool in the tool memory.

CR_TDR1_(1)_(2)_(3)_(4) **(Single Read)**
CC_TDR1_(1)_(2)_(3)_(4) **(Cyclic Read)**
CB_TDR1_(1)_(2)_(3)_(4) **(Break Cyclic Read)**

(1) = NC process number [0...6]

(2) = Tool memory [M = magazine/turret, S = spindle, G = grabber, P = change position, X = index address]

(3) = Tool location
In the magazine/turret: [1...999]
In the spindle: [1...4]
In the gripper: [1...4]
In the change position: [1...4]
As an index address: [0...9999999]

(4) = Data record [0 = tool basic data, 1...9 = cutter data]

Note: The index address of a tool is set by the device. For this reason, during the first access, access can only be made via tool memories M, S, G and P. Thereafter, the tool can also be addressed via the received index address.

Response Structure

The following table shows the general structure of the response to the "CR_TDR1" FI command. One line is output with 28 (basic data) or 40 (cutter data) columns for the returned values.

Line 1	Column 1	...	Column 28/40

Value Range/Meaning of the Columns

1.0.28 = requested basic tool data [max. 28 data elements] (see basic value range data)

1..0.40 = requested tool cutter data [max. 40 data elements] (see value range of cutter data)

Data elements 20 to 28 of the basic data and data elements 31 to 40 of the cutter data are only available as options (depending on the system parameters).

Example TDR1

Read the basic tool data record of the 2nd tool in the magazine in NC process 0.

FI command		00_CR_TDR1_0_M_2_0
Line	**Column**	**Answer**
1	1	928
	2	Miller D20
	3	M
	4	2
	5	1234567
	6	1234
	7	2
	8	1
	9	..............+..p............
	10	0
	11	M1
	12	M
	13	--
	14	M
	15	--
	16	[cycl]
	17	[mm]
	18	4
	19	102
	20	0.000000
	21	0.000000
	22	0.000000
	23	0.000000
	24	0.000000
	25	0.000000
	26	0.000000
	27	0.000000
	28	0.000000

Designation

TDR2 **T**ool **D**ata **R**ecord

FI command

Read basic data record or cutter data record of a tool. Addressing is by means of the tool number and index number.

Attention: Before this command is executed, a tool identification run is required!

CR_TDR2_(1)_(2)_(3)_(4)	**(Single Read)**
CC_TDR2_(1)_(2)_(3)_(4)	**(Cyclic Read)**
CB_TDR2_(1)_(2)_(3)_(4)	**(Break Cyclic Read)**

(1) = NC process number	[0...6]
(2) = Tool number	[1...9999999]
(3) = Index number	[1...9999]

(4) = Data record [0 = tool basic data,
1...9 = cutter data]

Response Structure

The following table shows the general structure of the response to the "CR_TDR2" FI command. One line is output with 28 (basic data) or 40 (cutter data) columns for the returned values.

Line 1	Column 1	...	Column 28/40

Value Range/Meaning of the Columns

1.0.28 = requested basic tool data [max. 28 data elements]
(see basic value range data)

1..0.40 = requested tool cutter data [max. 40 data elements]
(see value range of cutter data)

Data elements 20 to 28 of the basic data and data elements 31 to 40 of the cutter data are only available as options (depending on the system parameters).

Example TDR2

Read the basic tool-data record of tool 2 / duplo number 1 in NC process 0.

FI command		**00_CR_TDR2_0_2_1_0**
Line	**Column**	**Answer**
1	1	928
	2	Miller D20
	3	M
	4	2
	5	2
	6	1
	7	2
	8	1
	9	-p
	10	0
	11	M0
	12	M
	13	0
	14	M
	15	0
	16	~~0~~ [cycl]
	17	~~0~~ [mm]
	18	4
	19	102
	20	0.000000
	21	0.000000
	22	0.000000
	23	0.000000
	24	0.000000
	25	0.000000
	26	0.000000
	27	0.000000
	28	0.000000

Explanation

Returns a complete basic data record and/or cutter data record of a tool.

Note: From Version V23, the basic data has been extended to 32. From Version V23, it is recommended only to use the TDR3 command as this also allows the basic data 29-32 to be read.

FI command

Read the basic data record or cutter data record of a tool in the tool memory.

CR_TDR3_(1)_(2)_(3)_(4) **(Single Read)**

CC_TDR3_(1)_(2)_(3)_(4) **(Cyclic Read)**

CB_TDR3_(1)_(2)_(3)_(4) **(Break Cyclic Read)**

(1) = CNC process number [0...6]

(2) = Tool memory [M = magazine/turret, S = spindle, G = gripper, P = change position, X = index address]

(3) = Tool location
In the magazine/turret: [1...999]
In the spindle: [1...4]
In the gripper: [1...4]
In the change position: [1...4]
As an index address: [0...9999999]

(4) = Data record [0 = tool basic data, 1...9 = cutter data]

Note: The index address of a tool is set by the device. For this reason, during the first access, access can only be made via tool magazines M, S, G and P. Thereafter, the tool can also be addressed via the received index address.

Response Structure

The following table shows the general structure of the response to the "CR_TDR3" FI command. One line is output with 32 (basic data) or 40 (cutter data) columns for the returned values.

Line 1	Column 1	...	Column 32/40

Value Range/Meaning of the Columns

1..0.32 = requested tool basic data [max. 32 data elements] (see basic value range data)

1..00.40 = requested tool cutter data [max. 40 data elements] (see value range of cutter data)

Data elements 20 to 28 of the basic data and data elements 31 to 40 of the cutter data are only available as options (depending on the system parameters).

Example TDR3

Read the basic tool data record of the 2nd tool in the magazine in CNC process 0.

FI command		**00_CR_TDR3_0_M_2_0**
Line	**Column**	**Answer**
1	1	928
	2	Miller D20
	3	M
	4	2
	5	1234567
	6	1234
	7	2
	8	1
	9	..............+..p............
	10	0
	11	M1
	12	M
	13	--
	14	M
	15	--
	16	[cycl]
	17	[mm]
	18	4
	19	102
	20	0.000000
	21	0.000000
	22	0.000000
	23	0.000000
	24	0.000000
	25	0.000000
	26	0.000000
	27	0.000000
	28	0.000000
	29	--
	30	0
	31	0
	32	.?.............

Reference to Literature

See chapter entitled "Literature" [43].

5.160 Tool Insert Finish: TIF

MWCX device group

Designation **TIF** **T**ool **I**nsert **F**inish

Explanation Complete the insertion of a tool. The reservation of the tool memory location is lifted.

Refer also to: **CR_TII** and **CW_TLD1**

FI command Complete insertion.

CR_TIF_(1)_(2)_(3)		**(Single Read)**
(1) =	NC process number	[0...6]
(2) =	Tool memory	[M = magazine/turret, S =spindle, G = grabber, P = change position]
(3) =	Location number in the tool storage	in the magazine/turret: [1...999] in the spindle: [1...4] in the gripper: [1...4] in the change position: [1...4]

Response Structure One line is output with a column for acknowledgement of whether or not the FI command has been executed successfully.

(P_ACK) = **P**ositive **ACK**nowledge Data element has been set

Example TIF Finish the insertion of a tool at location 5 in magazine in NC process 0 of device 00.

FI command		**00_CR_TIF_0_M_5**
Line	**Column**	**Answer**
1	1	(P_ACK)

Reference to Literature See chapter entitled "Literature" [43].

5.161 Tool Insert Initiate: TII

MWCX device group

Designation **TII** **T**ool **I**nsert **I**nitiate

Explanation Initiate the insertion of an individual tool. Reserves a location in the tool memory.

After this, the basic data and the cutter data are to be entered by repeated inputting of **"**CW_TLD1". After the tool has actually been inserted in the tool memory, the procedure is completed by "CR_TIF".

FI command Initiate insertion.

CR_TII_(1)_(2)_(3)		**(Single Read)**
(1) =	NC process number	[0...6]
(2) =	Tool memory	[M = magazine/turret, S =spindle, G = grabber, P = change position]
(3) =	Location number in the tool storage	in the magazine/turret: [1...999] in the spindle: [1...4] in the gripper: [1...4] in the change position: [1...4]

Response Structure

One line is output with a column for acknowledgement of whether or not the FI command has been executed successfully.

(P_ACK) = **P**ositive **ACK**nowledge   Data element has been set.

Example TII

Initiate the procedure for inserting tools in tool location at location number 5 in NC process 0 of device 00.

FI command		**00_CR_TII_0_M_5**
Line	**Column**	**Answer**
1	1	(P_ACK)

In the event of an error:

Error is returned by N_ACK error:

FI command		**00_CR_TII_0_M_5**
Line	**Column**	**Answer**
1	1	1 (= N_ACK error class)
	2	131 (= error number)
	3	0x00000000 (= additional information 0)
	4	Tool storage occupied (= error text)

Reference to Literature

See chapter entitled "Literature" [43].

5.162 Downloading/ Uploading of Tool Lists: TLA

MWCX device group

Designation

TLA **T**ool **L**ist **A**ccess

Explanation

A tool list is downloaded by means of a download file. This download command is an FI job.

Structure of Download File

The structure of the download file corresponds to that of a Windows Ini file.

Summary:

Section [COMMONDATA]

Contains general information on the generation of this file.

DeviceAddr=Device address for which the tool list has been collected.

DeviceName=Device name

DeviceType=Device type

MTGUIVersion=GUI version used to generate this download file.

MaxBasicUserData=Number of the greatest defined user date (see system parameters)

MaxEdgeNumber=max. defined number of edges (see system parameters)

MaxEdgeUserData=Number of the greatest defined edge user date (see system parameters)

ProcessList=List of the processes in this download file

SaveDate=Date when this file was generated (e.g. through an UPLOAD process).

SaveTime=Time of the day when this file was generated.

Section [PROCESSxx]

Contains the main process information in respect of tool management.

xx=process number

Grippers=Number of the defined grippers in this process (see process parameters)

MaxCurrentGripper=Number of the largest gripper assigned with a tool

MaxCurrentSpindle=Number of the largest spindle assigned with a tool

MaxCurrentToolPocket=Number of the largest magazine location assigned with a tool

Spindle=Number of the defined spindles in this process (see process parameters)

ToolListId=Full, completed tool list identification (used for unambiguous identification of a tool list)

ToolNumber=Number of tools in a tool list

ToolPockets=Number of the defined magazine locations in this process (see process parameters)

Section [PROCESSxx:Gyyy]

Contains the basic tool data and/or the tool edge data of the tools located in the grippers yyy.

xx=Process number
yyy=Number of the gripper

General structure of a tool data element:

Ez.Dww=Data value

z=designates the edge; the following applies:

0 = basic tool data (sequence index 1..32)

1..9 = Tool edge data (sequence index 1..40)

ww=Data element number; the following applies:

Basic tool data 1..32

Tool edge data 1..40

Section [PROCESSxx:Syyy]

Contains the basic tool data and/or the tool edge data of the tools located in the spindles yyy.

xx=Process number
yyy=Number of the spindle

See **Section [PROCESSxx:Gyyy]** for information on the general structure of a tool data element.

Section [PROCESSxx:Myyy]

Contains the basic tool data and/or the tool edge data of the tools located in the magazine locations yyy.

xx=Process number
yyy=Magazine location

See **Section [PROCESSxx:Gyyy]** for information on the general structure of a tool data element.

BW_TLA1_(1)_{(2)}	**(Single Write)**
(1) = Complete download file name	Download file according to the preset structure
(2) = If applicable, the process number into which the tool list is to be downloaded	[0..6]

Response Structure

The response to the "TLA1" FI command consists of three lines, each with one column. The meaning of the elements is as follows:

- Line 1 = Job ID [01...20]
  (see Chapter 7.1 "FI Commands for the MPCX Device Group", IFJ).
- Line 2 = FI command
  [String, in accordance with Chapter 6.1 "Elements of the FI Command"]
- Line 3 = FI Job Error Code
  (see Chapter 8 "Error Codes")

Note: File and path details must be enclosed in inverted commas.

Example TLA1

The tool lists stored in the download file D:\Tool.dat are to be transferred into the parameterized drives of device 0.

As there optional 2nd parameter does not exist, the data is transmitted according to the current process configuration.

FI command		**00_BW_TLA1_"D:\Tool.dat"**
Line	**Column**	**Answer**
1	1	01
2	1	00_BW_ TLA1_0_"D:\Tool.dat"
3	1	0

Explanation

If applicable, reads the tool lists from the selected device and saves them to the upload file. This upload command is an FI job.

Structure of upload file

The structure of the upload file corresponds to that of a Windows Ini file. The structure is identical with that of a download file.

BR_TLA1_(1)_{(2)} **(Single Read)**

(1) = Complete upload file name

(2) = Process no. if applicable, if the tool list is to be collected only from this selected process. [0..6]

Response Structure

The response to the "TLA1" FI command consists of three lines, each with one column. The meaning of the elements is as follows:

- Line 1 = Job ID [01...20]
  (see Chapter 7.1 "FI Commands for the MPCX Device Group", IFJ).
- Line 2 = FI command
  [String, in accordance with Chapter 6.1 "Elements of the FI Command"]
- Line 3 = FI Job Error Code
  (see Chapter 8 "Error Codes")

Note: File and path details must be enclosed in inverted commas.

Example TLA1

All possible tool lists of device 0, later to be saved to the upload file D:\Tool.dat, are to be saved.

FI command		**00_BR_TLA1_"D:\Tool.dat"**
Line	**Column**	**Answer**
1	1	01
2	1	00_BR_TLA1_"D:\Tool.dat"
3	1	0

Explanation

Reads the log file generated during download of the tool lists.

Note: File and path details must be enclosed in inverted commas.

BR_TLA2_(1) **(Single Read)**

(1) = Complete download file name

Response Structure

The response to the "TLA2" command consists of n lines, each with 5 columns. One line is provided for each process.

Line n	Column 1	...	Column 5

Value Range/Meaning of Columns

1 =	Process number	[0..6]
2 =	Download status	[READY] = Download completed successfully [WARNING] = NOT all tools of the lists have been transmitted [ERROR] = Download has NOT been possible
3 =	Error Text	[--] = No error text available; otherwise, the error text for the failed download process is indicated here
4 =	Number of the loaded tools	
5 =	Number of tools to be loaded according to the download file	

Example TLA2

There is a parameter set in device 0, the processes 0,1,2,3, and 4 being defined and process 4 having NO active tool management. For the processes 0, 1 and 2, tool lists have been transferred according to the download file.

FI command		**00_BR_TLA2_"D:\Tool.dat"**
Line	**Column**	**Answer**
1	1	0
	2	READY
	3	--
	4	46
	5	46
2	1	1
	2	READY
	3	--
	4	32
	5	32

Rexroth
Bosch Group

3	1	2
	2	READY
	3	--
	4	25
	5	25
4	1	3
	2	ERROR
	3	2:724 There is NO TOOL LIST DATA for the selected process in the download file
	4	0
	5	0
5	1	4
	2	ERROR
	3	2:714: The selected process has NO tool management
	4	0
	5	0

Explanation

Reads out the main information from the download file.

Note: File and path details must be enclosed in inverted commas.

FI command

BR_TLA3_(1) **(Single Read)**

(1) = Complete download file name

Response Structure

The response to the "TLA3" command consists of n lines, each with 14 columns. One line is provided for each process.

Line n	Column 1	..	Column 14

Value Range/Meaning of Columns

1 =	Process number	[0..6]
2 =	Tool list name	Max. 32 ASCII characters
3 =	Number of tool edges	[E0..E9]
4 =	Greatest defined edge user date	[EU0..EU10]
5 =	Greatest defined user date	[BU0..BU9]
6 =	Number of spindles	[S0..S4]
7 =	Number of grippers	[G0..G4]
8 =	Number of magazine locations	[M0..M999]
9 =	Number of the largest spindle assigned with a tool	[S0..S4]
10 =	Number of the largest gripper assigned with a tool	[G0..G4]
11 =	Number of the largest magazine location assigned with a tool	[M0..M999]
12 =	Number of tools	[T0..T999]
13 =	Date of generation	
14 =	Time of generation	

Example TLA3 Reads the main information from the download file D:\Tool.dat, tool lists for the processes 0 and 1 being defined in the download file.

FI command		**00_BR_TLA3_"D:\Tool.dat"**
Line	**Column**	**Answer**
1	1	0
	2	Tool test list 1
	3	E9
	4	EU6
	5	BU7
	6	S1
	7	G1
	8	M99
	9	S1
	10	G1
	11	M84
	12	T46
	13	01.09.2003
	14	13:42:45
2	1	1
	2	Tool test list 2
	3	E9
	4	EU6
	5	BU7
	6	S4
	7	G2
	8	M88
	9	S4
	10	G1
	11	M88
	12	T10
	13	01.09.2003
	14	13:42:45

5.163 Tool Basic Data List: TLB

MWCX device group

Designation **TLB** **T**oo**L** **B**asic Data List

Explanation

Returns the basic data of the tool list of the selected device of the MWCX device group.

FI command

Read selected basic data of the tool list.

BR_TLB1_(1)_(2)_(3)_(4)_(5)	**(Single Read)**
(1) = NC process number	[0...6]
(2) = Tool memory	[M = magazine/turret, S = spindle, G = grabber]
(3) = Location from	[1...999]
(4) = Location to	[1...999]
(5) = Data element	[1...28]

If more than one element is required as the 5th entry parameter then these are attached to the command with "_" and corresponding numbers.

Response Structure

The following table shows the general structure of the response to the FI command "BR_TLB1". The number of lines depends on the number of tools. One line with 2 columns is output per tool for the returned values. If more than one data element is requested then the number of columns increases accordingly.

Line 1...n:	Column 1	Column 2	...	Column 29

Value Range/Meaning of the Columns

1 =	Tool memory	[xxx = magazine/turret, SPx = spindle, GRx = gripper]
2...29 =	Requested base tool data	{ [max. 28 data elements] (see value range, page 5-309)

Example TLB1

Read data elements 2, 5, 6, 7.

Explanation of elements:

- Element number 002: Name (ID) [max. 28 ASCII characters]
- Element number 005: Tool number [1..9999999]
- Element number 006: Index number [1...9999] and
- Element number 007: Compensation type [1...5]

For additional elements, refer to basic data value range p. 5-309

Assumption:

• NC process number:	0
• Tool magazine:	M = magazine and
• location number from:	2
• Location number to:	4

FI command		**00_BR_TLB1_0_M_2_4_2_5_6_7**
Line	**Column**	**Answer**
1	1	002
	2	TAPPER M6
	3	0
	4	1
	5	2
2	1	003
	2	DRILL MILLER D12
	3	0
	4	1
	5	1
3	1	004
	2	TWIST DRILL D4.8
	3	0
	4	1
	5	2

FI command

Read all basic data of the tool list.

BR_TLB2_(1)_(2)	**(Single Read)**
(1) = NC process number	[0...6]
(2) = Data element	[1...28]

If more than one element is required as the 2nd entry parameter then these are attached to the command with "_" and corresponding numbers.

Response Structure

The following table shows the general structure of the response to the FI command "BR_TLB2". The number of lines depends on the number of tools. One line with 2 columns is output per tool for the returned values. If more than one data element is requested then the number of columns increases accordingly.

Line 1...n:	**Column 1**	**Column 2**	**...**	**Column 29**

Value Range/Meaning of the Columns

1 =	Tool memory	[xxx = magazine/turret, SPx = spindle, GRx = gripper]
2...29 =	Requested base tool data	[max. 28 data elements] (refer to basic data value range, p. 5-309)

Example TLB2

Read data elements 2, 5, 6, 7 in NC process 0.

Explanation of elements:

- Element number 002: Name (ID) [max. 28 ASCII characters]
- Element number 005: Tool number [1..9999999]
- Element number 006: Index number [1...9999] and
- Element number 007: Compensation type [1...5]

For more elements, refer to value range "Basic Data".

FI command		**00_BR_TLB2_0_2_5_6_7**
Line	**Column**	**Answer**
1	1	SP1
	2	--
	3	0
	4	0
	5	0
2	1	001
	2	END MILL D16
	3	0
	4	1
	5	2
3	1	002
	2	TAPPER M6
	3	0
	4	1
	5	2
4	1	003
	2	DRILL MILLER D12
	3	0
	4	1
	5	1
5	1	004
	2	TWIST DRILL D4.8
	3	0
	4	1
	5	2
6	1	005
	2	DRILL MILLER D8
	3	0
	4	1
	5	2
7	1	006
	2	SLAB MILLING CUTTER D60
	3	0
	4	1
	5	1
8	1	007
	2	--
	3	0
	4	0
	5	0

Reference to Literature See chapter entitled "Literature" [43].

5.164 Tool Data Record Elements: TLD

MWCX device group

Designation **TLD** **T**oo**L** **D**ata

Explanation Returns elements of the basic data or cutter data of a tool in the tool memory. Only basic data or data from one cutter can be returned in any one telegram. If data elements are to be combined from basic data and cutter data then the command "CR_TLD3" or "CR_TLD4" must be used.

For a complete data record of basic data or cutting data, please refer to "CR_TDR".

FI command Read element(s) of the basic data or cutter data of a tool.

CR_TLD1_(1)_(2)_(3)_(4)_(5) **(Single Read)**

CC_TLD1_(1)_(2)_(3)_(4)_(5) **(Cyclic Read)**

CB_TLD1_(1)_(2)_(3)_(4)_(5) **(Break Cyclic Read)**

(1) = NC process number [0...6]

(2) = Tool memory [M = magazine/turret, S = spindle, G = gripper, P = change position, X = index address]

(3) = Tool location
in the magazine/turret: [1...999]
in the spindle: [1...4]
in the gripper: [1...4]
in the change position: [1...4]
as Index address: [0...9999999]

(4) = Data record [0 = tool basic data, 1...9 = cutter data]

(5) = Data element
The basic data: [1...28]
of the tool edge data: [1...40]

Data elements 20 to 28 of the basic data and data elements 31 to 40 of the cutter data are only available as options (depending on the system parameters). The response to access to data elements that are not available is "N_ACK" (Negative Acknowledge).

If more than one element is required as the 5th entry parameter then these are attached to the command with "_" and corresponding numbers.

Note: The index address of a tool is set by the device. For this reason, during the first access, access can only be made via tool memories M, S, G and P. Thereafter, the tool can also be addressed via the received index address.

Response Structure The following table shows the general structure of the response to the FI command "CR_TLD1". One line with one column is output for the returned value. If more than one data element is requested then the number of columns increases correspondingly.

Line 1	Column 1	...	Up to column 28/40

Value Range/Meaning of the Columns

1.0.28 = requested basic tool data [max. 28 data elements] (see basic value range data)

1..0.40 = requested tool cutter data [max. 40 data elements] (see value range of cutter data)

Example TLD1

Read the name (basic data 2) of the 4th tool in the magazine in NC process 0.

FI command		**00_CR_TLD1_0_M_4_0_2**
Line	**Column**	**Answer**
1	1	MILLER D24

FI command

Write single element of basic data or cutter data of a tool in the tool memory.

CW_TLD1_(1)_(2)_(3)_(4)_(5) (Single Write)

(1) = NC process number	[0...6]
(2) = Tool memory	[M = magazine/turret, S = spindle, G = gripper, P = change position, X = index address]
(3) = Tool memory location	[in the magazine/turret: [1...999] in the spindle: [1...4] in the gripper: [1...4] in the change position: [1...4] as Index address: [0...9999999]
(4) = Data record	[0 = tool basic data, 1...9 = cutter data]
(5) = Data element	The basic data: [1...28] of the tool edge data: [1...40]

Data elements 20 to 28 of the basic data and data elements 31 to 40 of the cutter data are only available as options (depending on the system parameters). The response to access to data elements that are not available is N_ACK (Negative Acknowledge).

Note: The index address of a tool is set by the device. For this reason, during the first access, access can only be made via tool memories M, S, G and P. Thereafter, the tool can also be addressed via the received index address.

Value to be written

Value of data element	see value ranges for basic and cutter data

Note: The value to be written is passed to the "acValue" parameter as an ASCII value in the "DataTransfer" routine.

Response Structure

One line is output with a column for acknowledgement of whether or not the FI command has been executed successfully.

(P_ACK) = **P**ositive **ACK**nowledge Data element has been set.

Example TLD1

Write data element 4 (warning limit) in NC process 0 for the tool at the 3rd magazine position in cutter 1.

FI command		**Value to be written: 6.5** **00_CW_TLD1_0_M_3_1_4**
Line	**Column**	**Answer**
1	1	(P_ACK)

Designation

TLD2 **T**oo**L** **D**ata

Explanation

Read basic data or cutter data element(s) of a tool. Addressing is by means of the tool number and index number.

Attention: Before this command is executed, a tool identification run is required!

FI command

CR_TLD2_(1)_(2)_(3)_(4)_(5)	**(Single Read)**
CC_TLD2_(1)_(2)_(3)_(4)_(5)	**(Cyclic Read)**
CB_TLD2_(1)_(2)_(3)_(4)_(5)	**(Break Cyclic Read)**
(1) = NC-Process number	[0...6]
(2) = Tool number	[1...9999999]
(3) = Index number	[1...9999]
(4) = Data record	[0 = tool basic data, 1...9 = cutter data]
(5) = Data element	of the base data: [1...28] of the tool edge data: [1...40]

Data elements 20 to 28 of the basic data and data elements 31 to 40 of the cutter data are only available as options (depending on the system parameters). The response to access to data elements that are not available is N_ACK (Negative Acknowledge).

If more than one element is required as the 5th entry parameter then these are attached to the command with "_" and corresponding numbers.

Response Structure

The following table shows the general structure of the response to the FI command "CR_TLD2". One line with one column is output for the returned value. If more than one data element is requested then the number of columns increases correspondingly.

Line 1	Column 1	...	Column 28/40

Value Range/Meaning of the Columns

1.0.28 = requested basic tool data	[max. 28 data elements] (see basic value range data)
1..0.40 = requested tool cutter data	[max. 40 data elements] (see value range of cutter data)

Example TLD2

Read the name (basic data 2) of the 3th tool/index no. 1 in NC process 0.

FI command		**00_CR_TLD2_0_3_1_0_2**
Line	**Column**	**Answer**
1	1	TAPPER M5

FI command

Write single element of basic data or cutter data of a tool. Addressing via tool number + index number.

CW_TLD2_(1)_(2)_(3)_(4)_(5)	**(Single Write)**
(1) = NC-Process number	[0...6]
(2) = Tool number	[1...9999999]
(3) = Index number	[1...9999]
(4) = Data record	[0 = tool basic data, 1...9 = cutter data]
(5) = Data element	of the base data: [1...28] of the tool edge data: [1...40]

Data elements 20 to 28 of the basic data and data elements 31 to 40 of the cutter data are only available as options (depending on the system parameters). The response to access to data elements that are not available is N_ACK (Negative Acknowledge).

Value to be written

Value of data element	see value ranges for basic and cutter data

---

Note: The value to be written is passed to the "acValue" parameter as an ASCII value in the "DataTransfer" routine.

Response Structure

One line is output with a column for acknowledgement of whether or not the FI command has been executed successfully.

(P_ACK) = **P**ositive **ACK**nowledge	Data element has been set

Example TLD2

Write data element 4 (warning limit) in NC process 0 for tool number 3/index number 1 in cutter 1.

FI command		**Value to be written: 6.5** **00_CW_TLD2_0_3_1_1_4**
Line	**Column**	**Answer**
1	1	(P_ACK)

Designation

TLD3 **T**oo**L** **D**ata

Explanation

Returns any element of the basic data or cutter data of a tool in any order.

In contrast with the command "TLD1", for this FI command addressing an element is extended to three positions. The first position addresses the data record (0 = basic data record, 1-9= cutter data) and the second and third positions address the actual data element.

Addressing Examples

002	Basic data – tool name
103	Cutter 1 – remaining tool life
203	Cutter 2 – remaining tool life

Note: You should always make sure when requesting tool data that the maximum net data length of 240 bytes is not exceeded. If more than 240 bytes are requested then the control unit returns the error message (NACK) /FI (1014).

FI command

Reading of basic data and cutter data of a tool in the tool memory.

CR_TLD3_(1)_(2)_(3)_(4)	**(Single Read)**
CC_TLD3_(1)_(2)_(3)_(4)	**(Cyclic Read)**
CB_TLD3_(1)_(2)_(3)_(4)	**(Break Cyclic Read)**

(1) = NC-Process number	[0...6]	
(2) = Tool memory	[M = magazine/turret, S = spindle, G = gripper, P = change position, X = index address]	
(3) = Tool memory location	In the magazine/turret:	[1...999]
	In the spindle:	[1...4]
	In the gripper:	[1...4]
	In the change position:	[1...4]
	As an index address:	[0...9999999]
(4) = Data element	[001...940]	

Data elements 020 to 028 of the basic data and data elements x31 to x40 of the cutter data are only available as options (depending on the system parameters). The response to access to data elements that are not available is "N_ACK" (Negative Acknowledge).

If more than one element is required as the 4th entry parameter then these are attached to the command with "_" and corresponding numbers.

Note: The index address of a tool is set by the device. For this reason, during the first access, access can only be made via tool memories M, S, G and P. Thereafter, the tool can also be addressed via the received index address.

Response Structure

The following table shows the general structure of the response to the FI command "CR_TLD3". One line with one column is output for the returned value. If more than one data element is requested then the number of columns increases accordingly.

Line 1...n:	Column 1	...	Column xxx

Value Range/Meaning of the Columns

1...xxx = requested basic tool data and cutter data	see value ranges for basic and cutter data

Example TLD3

Read the name of the basic tool data of the 4th tool in the magazine and the remaining tool life of cutter 1 in NC process 0.

FI command		**00_CR_TLD3_0_M_4_002_103**
Line	**Column**	**Answer**
1	1	MILLER D24
	2	100.00

Designation

TLD4 **T**oo**L** **D**ata

Explanation

Returns any element of the basic data or cutter data of a tool in any order.

In contrast with the command "TLD2", for this FI command addressing an element is extended to three positions. The first position addresses the data record (0 = basic data record, 1-9= cutter data) and the second and third positions address the actual data element.

Addressing Examples

002	Basic data – tool name
103	Cutter 1 – remaining tool life
203	Cutter 2 – remaining tool life

Note: You should always make sure when requesting tool data that the maximum net data length of 240 bytes is not exceeded. If more than 240 bytes are requested then the control unit returns the error message (NACK) /FI (1014).

FI command

Read the basic and cutter data of a tool according to the tool number and index number.

CR_TLD4_(1)_(2)_(3)_(4)	**(Single Read)**
CC_TLD4_(1)_(2)_(3)_(4)	**(Cyclic Read)**
CB_TLD4_(1)_(2)_(3)_(4)	**(Break Cyclic Read)**
(1) = NC-Process number	[0...6]
(2) = Tool number	[1...9999999]
(3) = Index number	[1...9999]
(4) = Data element	[001...940]

Data elements 020 to 028 of the basic data and data elements x31 to x40 of the cutter data are only available as options (depending on the system parameters). The response to access to data elements that are not available is N_ACK (Negative Acknowledge).

If more than one element is required as the 4th entry parameter then these are attached to the command with "_" and corresponding numbers.

Response Structure

The following table shows the general structure of the response to the FI command "CR_TLD4". One line with one column is output for the returned value. If more than one data element is requested then the number of columns increases correspondingly.

Line 1...n:	Column 1	...	Column xxx

Value Range/Meaning of the Columns

1...xxx = requested basic tool data and cutter data	see value ranges for basic and cutter data

Example TLD4

Read the name of tool number 3/index number 1 and the remaining tool life of cutter 4 in NC process 0 of device 00.

FI command		00_CR_TLD4_0_3_1_002_403
Line	**Column**	**Answer**
1	1	TAPPER M5
	2	100.00

Reference to Literature

See chapter entitled "Literature" [43].

5.165 Tool Edge Data List: TLE

MWCX device group

Designation **TLE** **T**oo**L** **E**dge Data List

Explanation Returns the cutter data of the tool list.

FI command

Read selected cutter data of the tool list.

BR_TLE1_(1)_(2)_(3)_(4)_(5)_(6)	**(Single Read)**
(1) = NC process number	[0...6]
(2) = Tool edge	[1...9]
(3) = Tool memory	[M = magazine/turret, S = spindle, G = gripper]
(4) = Location from	[0...999]
(5) = Location to	[0...999]
(6) = Data element	[1...40]

If more than one element is required as the 6th entry parameter then these are attached to the command with "_" and corresponding numbers.

Response Structure

The following table shows the general structure of the response to the FI command "BR_TLE1". The number of lines depends on the number of tools. One line with 2 columns is output per tool for the returned values. If more than one data element is requested then the number of columns increases accordingly.

Line 1...n:	Column 1	Column 2	...	Column 41

Value Range/Meaning of the Columns

1 =	Tool memory	[xxx = magazine/turret, SPx = spindle, GRx = gripper]
2...41 =	Requested tool cutter data	[max. 40 data elements] (see value range "Tool Edge Data", p. 5-311)

Example TLE1

Element number 002: Tool edge status is requested.

Assumption:

- NC process number: 0
- Tool edge: 1
- Tool magazine: M = magazine and
- location number from: 1
- location number to: 3

Read data elements 2 and 3.

FI command		**00_BR_TLE1_0_1_M_1_3_2_3**
Line	**Column**	**Answer**
1	1	001
	2	d (tool worn out)
2	1	002
	2	_ (tool ok)
3	1	003
	2	w (fallen below warning limit)

FI command

Read all cutter data of the tool list.

BR_TLE2_(1)_(2)_(3)	**(Single Read)**
(1) = NC process number	[0...6]
(2) = Cutter position	[0...8]
(3) = Data element	[1...40]

If more than one element is required as the 3rd entry parameter then these are attached to the command with "_" and corresponding numbers.

Response Structure

The following table shows the general structure of the response to the FI command "BR_TLE2". The number of lines depends on the number of cutters. One line with 2 columns is output per cutter for the returned values. If more than one data element is requested then the number of columns increases accordingly.

Line 1...n:	**Column 1**	**Column 2**	**...**	**Column 41**

Value Range/Meaning of the Columns

1 =	Tool memory	[00 = magazine/turret, SP = spindle, GR = gripper]
2...41 =	Requested base tool data	[max. 40 data elements] (see value range "Tool Edge Data")

Example TLE2

- Element number 003: Residual tool life [0.0000...100.0000]
- Element number 004: Warning limit [0.1...100.00]
- Element number 005: Maximum period of use [0...9999999]
- Element number 009: Length L3 [-9999.9999...9999.9999]

Read in NC process 0 the data elements 3, 4, 5, 9 for all tools at cutter position 1.

FI command		**00_BR_TLE2_0_1_3_4_5_9**
Line	**Column**	**Answer**
1	1	SP1
	2	0.0000
	3	0.0000
	4	0.0000
	5	0.0000
2	1	001
	2	100.0000
	3	5.0000
	4	0.0000
	5	106.8500
3	1	002
	2	100.0000
	3	5.0000
	4	0.0000
	5	132.9600
4	1	003
	2	48.0000
	3	5.0000
	4	100.0000
	5	106.8000
5	1	004
	2	99,8617
	3	5.0000
	4	0.0000
	5	180.0900
6	1	005
	2	100.0000
	3	5.0000
	4	0.0000
	5	78.7000
7	1	006
	2	100.0000
	3	0.0000
	4	0.0000
	5	116.0000
8	1	007
	2	0.0000
	3	0.0000
	4	0.0000
	5	0.0000

Reference to Literature See chapter entitled "Literature" [43].

5.166 Tool Move: TMV

MWCX device group

Designation **TMV** **T**ool **M**o**V**e

Explanation A complete tool data record consisting of basic data and defined cutter data is moved. This corresponds to the Bosch Rexroth BOF function "Tool Move".

FI command Move the selected tool data record.

CR_TMV_(1)_(2)_(3)_(4)_(5)	**(Single Read)**
(1) = NC process number	[0...6]
(2) = Current tool memory	[M = magazine/turret, S = spindle, G = grabber]
(3) = Current location number	[1...999]
(4) = Target tool memory	[M = magazine/turret, S = spindle, G = grabber]
(5) = Target location number	[1...999]

Response Structure One line with one column is output to acknowledge the FI command issued. The meaning of the elements is as follows:

(P_ACK) = **P**ositive **ACK**nowledge	Data record has been moved

Example TMV Move the 24th tool data record in the magazine to the 25th tool data record in the magazine.

Assumption:
There is a valid tool in magazine location 24 in NC process 0 at device address 00.

FI command		**00_CR_TMV_0_M_24_M_25**
Line	**Column**	**Answer**
1	1	(P_ACK)

Reference to Literature See chapter entitled "Literature" [43].

5.167 Information on Grippers/Spindles/Tool Magazine Locations: TPI

MWCX device group

Designation **TPI** **T**ool **P**osition **I**nformation

FI command

This command is used to read the index addresses of the currently occupied tool storage locations. Through the parameters <VON location> ('f'om location') and <BIS location> ('to location'), the requested range can be determined. ' When these two parameters are NOT set, ALL occupied index addresses of the tool storage locations defined in the process parameters are returned.

BR_TPI1_(1)_{(2)_(3)}		**(Single Read)**
(1) =	Process number	[0..6]
(2) =	<VON location> - Start location index in the tool storage location administration (optional parameter)	[1..1007] 1..4 = Gripper 1..4 5..8 = Spindle 1..4 9..1007 = Magazine location 1..999
(3) =	<VON location> - End location index in the tool storage location administration (optional parameter)	[1..1007] 1..4 = Gripper 1..4 5..8 = Spindle 1..4 9..1007 = Magazine location 1..999

Response Structure

The following table shows the general structure of the response to the FI command "TPI1". N lines, each with 3 columns, are output. Each line corresponds to one occupied tool storage location.

Line 1...n	Column 1	Column 2	Column 3

Value Range/Meaning of Columns

1 =	Location name	[G1..G4 = Gripper 1..4 S1..S2 = Spindle 1..4 M = Tool storage location]
2 =	Location name	[1..999]
3 =	Index address of the tool as a LONG value	[LONG value]

Example TPI1

Reads the index addresses of ALL occupied tool storage locations of the process 0 of device 00.

FI command		**00_BR_TPI1_0**
Line	**Column**	**Answer**
1	1	G1
	2	1
	3	1834
2	1	S1
	2	1
	3	2345
3	1	M
	2	1
	3	1456
4	1	M
	2	3
	3	3456
5	1	M
	2	9
	3	1678

FI command

This command is used to read the location status bytes of the occupied or free tool storage locations. Through the parameters <VON location> ('f'om location') and <BIS location> ('to location'), the requested range can be determined. ' When these two parameters are NOT set, ALL location status bytes of the tool storage locations defined in the process parameters are returned.

BR_TPI2_(1)_{(2)_(3)}		**(Single Read)**
(1) =	Process number	[0..6]
(2) =	<VON location> - Start location index in the tool storage location administration (optional parameter)	[1..1007] 1..4 = Gripper 1..4 5..8 = Spindle 1..4 9..1007 = Magazine location 1..999
(3) =	<VON location> - End location index in the tool storage location administration (optional parameter)	[1..1007] 1..4 = Gripper 1..4 5..8 = Spindle 1..4 9..1007 = Magazine location 1..999

Response Structure

The following table shows the general structure of the response to the FI command "TPI2". A line of n columns is output. Here, the column index corresponds to the location index.

Line 1	**Column 1**	**...**	**Column n**

Value Range/Meaning of Columns

1 = Location status byte for location index	[0x00-0xFF]
2 = Location status byte for location index+1	[0x00-0xFF]
3 = Location status byte for location index+2	[0x00-0xFF]
....	[0x00-0xFF]
n = Location status byte for location index+n	[0x00-0xFF]

Example TPI2

Read the location status bytes of ALL tool storage locations of the process 0 of device 00. Here, 1 gripper and 1 spindle and 2 magazine locations are defined in the process parameters of the process 0. However, the location status bytes of the grippers 1…4 and the spindles 1…4 are ALWAYS returned.

FI command		**00_BR_TPI2_0**
Line	**Column**	**Answer**
1	1	0x80 (gripper 1)
	2	0x00 (gripper 2)
	3	0x00 (gripper 3)
	4	0x00 (gripper 4)
	5	0x80 (spindle 1)
	6	0x00 (spindle 2)
	7	0x00 (spindle 3)
	8	0x00 (spindle 4)
	9	0x80 (magazine location 1)
	10	0x80 (magazine location 2)

5.168 Torque: TQE

MWCX device group

Designation **TQE** **TorQuE**

Explanation

The torque at a selected axis of the MWCX device group is read. The FI command "TQE1" returns the torque of an axis, related to the code of the axis meaning. On the other hand, the FI command "TQE2" returns the torque of an axis, related to the physical axis number.

FI command

Output the torque of the selected device of the MWCX device group, related to the code of the axis meaning.

CR_TQE1_(1)_(2)	**(Single Read)**
CC_TQE1_(1)_(2)	**(Cyclic Read)**
CB_TQE1_(1)_(2)	**(Break Cyclic Read)**
(1) = NC process number	[0...6]
(2) = Axis meaning	[0...11; 20];

Response Structure

The following table shows the general structure of the response to the FI command "TQE1". One line with three columns is output for the name of the axis, the torque and the unit [%].

Line 1	Column 1	...	Column 3

Value Range/Meaning of Columns

1 = Axis name	[according to settings of axis parameters]
2 = Torque	[format acc. to settings of the process parameter]
3 = Unit	[%]

Note: If the specified axis is not defined in the selected NC process then the response in all columns is [--].

Example TQE1

Read the torque at the Z axis in NC process 0 of device address 00.

FI command	**00_CR_TQE1_0_2**		
Answer			
Line	**Column 1**	**Column 2**	**Column 3**
1	Z	-25.6	[%]

FI command

Output the torque at the selected axis of the device specified, related to the physical axis number.

CR_TQE2_(1)	**(Single Read)**
CC_TQE2_(1)	**(Cyclic Read)**
CB_TQE2_(1)	**(Break Cyclic Read)**
(1) = Physical axis number	[1...32]

Response Structure

The following table shows the general structure of the response to the FI command "TQE2". One line with three columns is output for the name of the axis, the torque and the unit [%].

Line 1	**Column 1**	**...**	**Column 3**

Value Range/Meaning of Columns

1 = Axis name	[according to settings of axis parameters]
2 = Torque	[format acc. to settings of the process parameter]
3 = Unit	[%]

Note: If the specified axis is not defined in the selected NC process then the response in all columns is [--].

Example TQE2

Read the torque at the Z axis (physical axis number = 3) at device address 00.

FI command	**00_CR_TQE2_3**		
Answer			
Line	**Column 1**	**Column 2**	**Column 3**
1	Z	-25.6	[%]

5.169 Removing Tool Data Record: TRM

MWCX device group

Designation **TRM** Tool **R**e**M**ove

Explanation

A complete tool data record consisting of basic data and defined cutter data is removed from the device. This corresponds to the Bosch Rexroth BOF function "Remove Tool from the Magazine List".

FI command

Remove the selected tool data record.

CR_TRM_(1)_(2)_(3)	**(Single Read)**
(1) = NC process number	[0...6]
(2) = Tool memory	[M = magazine/turret, S = spindle, G = gripper]
(3) = Location number	[1...999]

Response Structure

One line with one column is output to acknowledge the FI command issued. The meaning of the elements is as follows:

(P_ACK) = **P**ositive **ACK**nowledge Data record has been removed

Example TRM

Remove the 24th tool data record.

Assumption:
There is a valid tool in magazine location 24 in NC process 0 at device address 00.

FI command		**00_CR_TRM_0_M_24**
Line	**Column**	**Answer**
1	1	(P_ACK)

Reference to Literature

See chapter entitled "Literature" [43].

5.170 Resetting Remaining Tool Life of a Tool: TRS

MWCX device group

Designation **TRS** **T**ool **R**e**S**et

Explanation

Resets the percentage remaining tool life of a tool to 100%. The positive or negative execution of the function is reported back via the return value of the routine.

FI command

Reset remaining tool life of a tool:

CR_TRS_(1)_(2)_(3) **(Single Read)**

(1) = NC process number [0...6]

(2) = Tool memory [M = magazine/turret, S = spindle, G = gripper, P = change position, X = index address]

(3) = Tool location

in the magazine/turret:	[1...999]
in the spindle:	[1...4]
in the gripper:	[1...4]
in the change position:	[1...4]
as Index address:	[0...9999999]

Note: The index address of a tool is set by the device. For this reason, during the first access, access can only be made via tool memories M, S, G and P. Thereafter, the tool can also be addressed via the received index address.

Response Structure

One line is output with a column for acknowledgement of whether or not the FI command has been executed successfully.

(P_ACK) = **P**ositive **ACK**nowledge Tool has been reset

Example TRS

Reset the remaining tool life for the tool located in change position 1 in NC process 0 of device 00.

FI command		**00_CR_TRS_0_P_1**
Line	**Column**	**Answer**
1	1	(P_ACK)

Reference to Literature

See chapter entitled "Literature" [43].

5.171 Requesting Watch List Allocations: WLA

MWCX device group

Designation **WLA** **W**atch **L**ist **A**llocation

Explanation Requests free watch list allocations. A maximum of ten free watch list allocations can be requested with one FI command.

BR_WLA1_(1)		**(Single Read)**
(1) =	Number of the requested free watch list numbers	The required number of free watch list allocations is identified here. The allowed value range: 1..10

Response Structure The following table shows the general structure of the response to the FI command "WLA1".

Line 1	Column 1	...	Column n

Value Range/Meaning of Columns

1 =	1. free watch list allocation	Value range: 1..15
2 =	2. free watch list allocation	Value range: 1..15
3 =	3. free watch list allocation	Value range: 1..15
n =	nth free watch list allocation	Value range: 1..15

Example WLA1 Request four free watch list allocations.

Assumption:
Watch list allocations 3 and 5 are already assigned!

FI command		**00_BR_WLA1_4**
Line	**Column**	**Answer**
1	1	1
	2	2
	3	4
	4	6

5.172 Freeing Watch List Allocations: WLF

MWCX device group

Designation **WLF** **W**atch **L**ist **F**ree

Explanation Previously requested watch list allocations are freed again.

FI command Free ALL assigned watch list allocations for the selected device.

BR_WLF1 **(Single Read)**

Note: The FI command "WLF1" frees ALL assigned watch list allocations, including those of other WIN32 applications.

Response Structure The following table shows the general structure of the response to the FI command "WLF1".

Line 1	Column 1	...	Column n

Value Range/Meaning of Columns

1 =	1. freed watch list allocation	Value range: 1..15
2 =	2. freed watch list allocation	Value range: 1..15
3 =	3. freed watch list allocation	Value range: 1..15
n =	nth freed watch list allocation	Value range: 1..15

Example WLF1

Free ALL assigned watch list allocations.

Assumption:
The following watch list numbers have been allocated: 1, 2, 3, 4.

FI command		**00_BR_WLF1**
Line	**Column**	**Answer**
1	1	1
	2	2
	3	3
	4	4

FI command

Free the required watch list allocations for a selected device.

BR_WLF2_(1)_{(2)..(10)}	**(Single Read)**
(1)..(10) = List of watch list allocations to be released	A maximum of 10 watch list allocations can be transferred here to be freed again.

Response Structure

The following table shows the general structure of the response to the FI command "WLF2".

Line 1	**Column 1**	**...**	**Column n**

Value Range/Meaning of Columns

1 =	1. freed watch list allocation	Value range: 1..16
2 =	2. freed watch list allocation	Value range: 1..16
3 =	3. freed watch list allocation	Value range: 1..16
n =	nth freed watch list allocation	Value range: 1..16

Example WLF2

Free required watch list allocations:
Assumption: Watch list allocations 1,3,4, and 8 have first been requested using the FI command "WLA1".

FI command		**00_BR_WLF2_1_3_4_8**
Line	**Column**	**Answer**
1	1	1
	2	3
	3	4

5.173 Data of the Zero Offset Table: ZOD

MWCX device group

Designation **ZOD** Zero **O**ffset **D**ata

Explanation

The zero-offset table data can be read and written. The zero offsets allow the origin of a coordinate axis to be shifted (offset) by a set value, related to the original position of the machine. The location of the machine zero point remains securely stored in the NC controls and is not changed by the zero offset.

Offset Type

The following offset types are available in the CNC:

- programmable absolute zero offset G50,
- programmable incremental zero offset G51,
- programmable workpiece zero point G52,
- adjustable zero offsets G54 ... G59 as well as
- adjustable general offset in the zero (origin) table.

Using the zero offsets G50, G51 and G54 to G59 and the workpiece zero point (origin) G52, the coordinate zero point of every NC axis can be laid onto any coordinate position within or outside of the respective travelling range. It is thereby possible to process and identical NC program at different machine positions. The position of the machine zero point of every axis is entered in the drive parameters as a difference to the reference point, whereby the value entered in the drive parameters corresponds to the coordinate value of the reference point in the machine coordinate system.

Code of displacement types

Code	Meaning	Explanation
0	Total	Sum of all active offset values
1	G50/G51	Programmable absolute / incremental zero offset
3	General offset	acts additive to all offset types
4...9	G54 - G59	Selectable zero offsets

Zero point database

As memory for a record of zero offsets, 10 zero offset tables (O0 … O9) are provided.

FI command

Write a zero offset.

CW_ZOD_(1)_(2)_(3)_(4)_(5)	**(Single Write)**
(1) = NC memory	[1 = memory A; 2 = memory B]
(2) = NC process number	[0...6]
(3) = Offset table number	[0...9]
(4) = Offset type	[offset type code]
(5) = Code of the axis meanings	[0...8] axis meanings [9] offset angle "PHI"

Value to be written

Offset	[with axes: format acc. to the parameter settings] [offset angle PHI always in format Y.XXXX]

Note: The value to be written is passed to the "acValue" parameter in the "DataTransfer" routine.

Offset Type

Code	Meaning	Explanation
3	General offset	acts additive to all offset types
4	G54	Adjustable zero offset
....	....	....
9	G59	Adjustable zero offset

Note: The axis meanings are contained in chapter entitled "Data Tables".

Response Structure

One line with one column is output to acknowledge the FI command issued. The meaning of the elements is as follows:

(P_ACK) = **P**ositive **ACK**nowledge Value has been written

Example ZOD

Write into zero offset table O2 the value of the general offset of axis X in NC memory A of NC process number 0 at device address 00.

Assumption:

- There is a valid parameter record in the device and
- the axes X, Y, Z are defined.

FI command		**Value to be written: 0.111** **00_CW_ZOD_1_0_2_3_0**
Line	**Column**	**Answer**
1	1	(P_ACK)

FI command

The values of the zero offset of all defined axes are output for the selected offset (shift) type.

CR_ZOD1_(1)_(2)_(3)_(4){_(6)}	**(Single Read)**
CC_ZOD1_(1)_(2)_(3)_(4){_(6)}	**(Cyclic Read)**
CB_ZOD1_(1)_(2)_(3)_(4){_(6)}	**(Break Cyclic Read)**
(1) = NC memory	[1 = memory A; 2 = memory B]
(2) = NC-Process number	[0...6]
(3) = Offset table number	[0...9]
(4) = Offset type	[0...9 Code of offset type]
(5) = Measuring unit (optional)	[mm, inch]

If there is no optional information for the unit {mm, inch}, then the length values are given in the base programming unit of the process. If the entered unit is different from the basic coordinate system, the length values are converted into the requested unit.

Note: The axis meanings are contained in chapter entitled "Data Tables".

Offset Type

Code	Meaning	Explanation
0	Total	Sum of all active offset values
1	G50/G51	Programmable absolute / incremental zero offset
2	G52	Programmable work piece zero point
3	General offset	Acts additive to all offset types
4	G54	Adjustable zero offset
....	....	....
9	G59	Adjustable zero offset

Response Structure

The following table shows the general structure of the response to the FI command "ZOD1". The answer consists of one to a maximum of n=10 lines (1 per axis), each with three columns for the name of the axis, value of zero offset and the unit.

Line 1...n:	**Column 1**	**...**	**Column 3**

Value Range/Meaning of Columns

1 =	Axis designation	[acc. to settings of the axis parameters; PHI] [Xi, Yi, Zi, Ui, Vi, Wi, Ai, Bi ,Ci, Si, --] i=[1...3])
2 =	Value	format acc. to parameter settings] [offset angle PHI always in format Y.XXXX]
3 =	Unit	[mm, inch], [offset angle PHI: deg]

Example ZOD1

Read in the zero offset table O2 the values of the general offset of all defined axes in NC memory A of CNC process number 0 at device address 00. The values are to be output in the basic coordinate system.

Assumption:

- There is a valid parameter record in the device and
- the axes X, Y, Z (assigned at certain times) are defined.

FI command	**00_CR_ZOD1_1_0_2_3**		
Answer			
Line	**Column 1**	**Column 2**	**Column 3**
1	X	0.111	[mm]
2	Y	0.000	[mm]
3	*Z	0.000	[mm]
4	PHI	0.0000	[deg]

FI command

Output all zero offset values for the axes selected in a list.

CR_ZOD2_(1)_(2)_(3)_(4)_(5){_(6)} **(Single Read)**

CC_ZOD2_(1)_(2)_(3)_(4)_(5){_(6)} **(Cyclic Read)**

CB_ZOD2_(1)_(2)_(3)_(4)_(5){_(6)} **(Break Cyclic Read)**

(1) = NC memory	[1 = memory A; 2 = memory B]
(2) = NC process number	[0...6]
(3) = Offset table number	[0...9]
(4) = Offset type	[offset type code]
(5) = Selection list for a max. of 10 elements	[0...8] axis meanings [9] offset angle "PHI"
(6) = Measuring unit (optional)	[mm, inch]

If there is no optional information for the unit {mm, inch}, then the length values are given in the base programming unit of the process. If the entered unit is different from the basic coordinate system, the length values are converted into the requested unit.

Offset Type

Code	Meaning	Explanation
0	Total	Sum of all active offset values
1	G50/G51	Programmable absolute / incremental zero offset
2	G52	Programmable work piece zero point
3	General offset	Acts additive to all offset types
4	G54	Adjustable zero offset
....	....	....
9	G59	Adjustable zero offset

Note: The axis meanings are contained in chapter 6.2, "Data Tables".

Response Structure

The following table shows the general structure of the response to the FI command "ZOD2". The answer consists of one to a maximum of n=10 lines (1 per requested axis), each with three columns for the code of the axis meaning, value of zero offset and the unit. The number of lines depends on the number of list elements.

Line 1...n:	Column 1	...	Column 3

Value Range/Meaning of Columns

1 =	Axis designation	[acc. to settings of the axis parameters; PHI] [Xi, Yi, Zi, Ui, Vi, Wi, Ai, Bi ,Ci, Si, --] i=[1...3])
2 =	Value	format acc. to parameter settings] [offset angle PHI always in format Y.XXXX]
3 =	Unit	[mm, inch], [offset angle PHI: deg]

Note: If a requested axis is not defined then the value of columns 1 to 3 is [--]. If the axis name is preceeded by "*", e.g. "*Z", then this access is only assigned to the process at certain times (GAX/FAX).

Example ZOD2

Read in zero offset table O2 the values of the general offset of axes X, Y, Z and U as well as the offset angle "PHI" in NC memory A of CNC process number 0 at device address 00.

Assumption:

- There is a valid parameter record in the device and
- the axes X, Y, Z (assigned at certain times) are defined.

FI command	**00_CR_ZOD2_1_0_2_3_0_1_2_3_9**		
Answer			
Line	**Column 1**	**Column 2**	**Column 3**
1	X	0.111	[mm]
2	Y	0.000	[mm]
3	*Z	0.000	[mm]
4	--	--	--
5	PHI	0.0000	[deg]

Reference to Literature

See chapter entitled "Literature" [44].

5.174 Value Ranges

Basic Data

MWCX device group

Element No.	Name of the File Element		Writable?
1	Index address	0...9999999	No
2	Name (ID)	Max. 28 ASCII characters	Yes
3	Memory	M = magazine/turret, S = spindle, G = grabber	No
4	Location	0...999	No
5	Tool number	1...9999999	Yes
6	Duplo number	1...9999	Yes
7	Correction type	1...5	Yes
8	Number of tool edges	1...9	Yes
9	Tool status	32 status bits with 0/1 (see following table)	Yes
10	Unassigned half-location	0...4	Yes
11	Former tool location	Memory [M/S/G] location [0..999]	No
12	Memory of the next replacement tool	M = magazine/turret, S = spindle, G = grabber	No
13	Location of the next replacement tool	0...999	No
14	Memory of the previous replacement tool	M = magazine/turret, S = spindle, G = grabber	No
15	Location of the previous replacement tool	0..999	No
16	Time unit	0 = min, 1 = cycle	Yes
17	Unit of length	0 = mm, 1 = inch	Yes
18	Tool code	0...9	Yes
19	Display type	0...65535	Yes
20	User data 1	+/- $1.2*10^{38}$ .. +/- $3.4*10^{-38}$	Yes
21	User data 2	+/- $1.2*10^{38}$ .. +/- $3.4*10^{-38}$	Yes
22	User data 3	+/- $1.2*10^{38}$ .. +/- $3.4*10^{-38}$	Yes
23	User data 4	+/- $1.2*10^{38}$ .. +/- $3.4*10^{-38}$	Yes
24	User data 5	+/- $1.2*10^{38}$ .. +/- $3.4*10^{-38}$	Yes
25	User data 6	+/- $1.2*10^{38}$ .. +/- $3.4*10^{-38}$	Yes
26	User data 7	+/- $1.2*10^{38}$ .. +/- $3.4*10^{-38}$	Yes
27	User data 8	+/- $1.2*10^{38}$ .. +/- $3.4*10^{-38}$	Yes
28	User data 9	+/- $1.2*10^{38}$ .. +/- $3.4*10^{-38}$	Yes
29			Nein
30	Tool group	0..99	YES
31	Tool group duplo number	0..99	YES
32	Tool group status bit	16 Statusbits with 0/1 (see following table)	YES

Note: Box 19 applies from FI version 06 of the tool classification. It can no longer be edited by the user. Refer also to the documentation "Tool Management".

Tool Status Bits

Bit	Sym bol	Value	Group name	Group information	Change able	Comment
1	!	1 0	Presence	Tool not available Tool available	No	Tool is missing
2	?	1 0		Tool not required Tool required	No	Tool not required for machining
3	t	1 0	Error correction type	Correction type faulty	No	Correction type does not comply with requirements
4	e	1 0	Error number of cutters	Wrong number of cutters Correct number	No	Number of tool edges does not comply with requirements
5	f	1 0	Error tool edge	Cutter faulty Cutter not faulty	No	Tool edge data does not comply with requirements
6	$	1 0	Error tool code	Tool code faulty Tool code not faulty	No	
7	*				No	Reserved
8	*				No	Reserved
9	B	1 0	Location locking	Location blocked Location not blocked	Yes	Location is damaged, for example
10		1 0		Upper half-location blocked. Not blocked	No	Blocked for fpc tool located in grabber or spindle
11		1 0		Lower half-location blocked. Not blocked	No	Blocked for fpc tool located in grabber or spindle
12		1 0	Location reservation	Upper half-location reserved. Not reserved	Yes	For a tool that is to be inserted, for example
13		1 0		Lower half-location reserved. Not reserved	Yes	For a tool that is to be inserted, for example
14		1 0	Location reservation	Upper half-location covered Not covered	No	The upper half-location is covered by a tool
15		1 0		Lower half-location covered Not covered	No	The lower half-location is covered by a tool
16		1 0		Location assigned Not assigned	No	There is a tool at this location
17	d	1 0	Wear state	Tool is worn Tool is not worn	No	The tool can no longer be used (replace)
18	w	1 0		Warning limit reached Warning limit not reached	No	The remaining tool life is near its end (replace)
19	p	1 0	Alternate tool identification	Processing tool No processing tool	No	There is a processing tool for every sister tool group
20	s	1 0		Replacement tool No replacement tool	No	A replacement tool is a tool still to be used, not a processing tool
21	C	1 0	Fixed position coding	Fixed position coding, tool No fixed position coding, tool	Yes	The tool always remains at the same location in the magazine
22	L	1 0	Tool status	Tool blocked Tool not blocked	Yes	E.g., cutter is broken by user or application
23	*				No	Reserved
24	*				No	Reserved
25	1	1 0	ANW 1	User tool status bit 1	Yes	Any meaning

Bit	Sym bol	Value	Group name	Group information	Change able	Comment
26	2	1 0	ANW 2	User tool status bit 2	Yes	Any meaning
27	3	1 0	ANW 3	User tool status bit 3	Yes	Any meaning
28	4	1 0	ANW 4	User tool status bit 4	Yes	Any meaning
29	5	1 0	ANW 5	User tool status bit 5	Yes	Any meaning
30	6	1 0	ANW 6	User tool status bit 6	Yes	Any meaning
31	7	1 0	ANW 7	User tool status bit 7	Yes	Any meaning
32	8	1 0	ANW 8	User tool status bit 8	Yes	Any meaning

Tool Edge Data

Element Number	Name of the Data Element	Value Range	Writable?
1	Tool edge position	0...8	Yes
2	Tool edge status	16 status bits with 0/1 (see following table)	Yes
3	Remaining tool life	-99.99...100.00	Yes
4	Warning limit	0.1...100.00	Yes
5	Max. life time	0...9999999	Yes
6	Time used	0...9999.999	No
7	Length L1	-99999.9999...+99999.9999 or -9999.99999...+9999.99999	Yes
8	Length L2	-99999.9999...+99999.9999 or -9999.99999...+9999.99999	Yes
9	Length L2	-99999.9999...+99999.9999 or -9999.99999..+9999.99999	Yes
10	Radius R	-99999.9999...+99999.9999 or -9999.99999...+9999.99999	Yes
11	Wear L1	-99999.9999...+99999.9999 or -9999.99999..+9999.99999	Yes
12	Wear L2	-99999.9999...+99999.9999 or -9999.99999...+9999.99999	Yes
13	Wear L3	-99999.9999...+99999.9999 or -9999.99999...+9999.99999	Yes
14	Wear R	-99999.9999...+99999.9999 or -9999.99999...+9999.99999	Yes
15	Offset L1	-99999.9999...+99999.9999 or -9999.99999...+9999.99999	Yes
16	Offset L2	-99999.9999...+99999.9999 or -9999.99999...+9999.99999	Yes
17	Offset L3	-99999.9999...+99999.9999 or -9999.99999...+9999.99999	Yes
18	Offset R	-99999.9999...+99999.9999 or -9999.99999...+9999.99999	Yes
19	L1_min	-99999.9999...+99999.9999 or -9999.99999...+9999.99999	No
20	L1_max	-99999.9999...+99999.9999 or -9999.99999...+9999.99999	No
21	L2_min	-99999.9999...+99999.9999 or -9999.99999...+9999.99999	No
22	L2_max	-99999.9999...+99999.9999 or -9999.99999...+9999.99999	No
23	L3_min	-99999.9999...+99999.9999 or -9999.99999...+9999.99999	No
24	L3_max	-99999.9999...+99999.9999 or -9999.99999...+9999.99999	No
25	R_min	-99999.9999...+99999.9999 or -9999.99999...+9999.99999	No
26	R_max	-99999.9999...+99999.9999 or -9999.99999...+9999.99999	No
27	Wear factor L1	-99999.9999...+99999.9999 or -9999.99999...+9999.99999	Yes
28	Wear factor L2	-99999.9999...+99999.9999 or -9999.99999...+9999.99999	Yes

Element Number	Name of the Data Element	Value Range	Writable?
29	Wear factor L3	-99999.9999...+99999.9999 or -9999.99999...+9999.99999	Yes
30	Wear factor R	-99999.9999...+99999.9999 or -9999.99999...+9999.99999	Yes
31	User data 1	+/- $1.2*10^{38}$ ... +/- $3.4*10^{-38}$	Yes
32	User data 2	+/- $1.2*10^{38}$ ... +/- $3.4*10^{-38}$	Yes
33	User data 3	+/- $1.2*10^{38}$ ... +/- $3.4*10^{-38}$	Yes
34	User data 4	+/- $1.2*10^{38}$ ... +/- $3.4*10^{-38}$	Yes
35	User data 5	+/- $1.2*10^{38}$ ... +/- $3.4*10^{-38}$	Yes
36	User data 6	-99999.9999...+99999.9999 or -9999.99999...+9999.99999	Yes
37	User data 7	-99999.9999...+99999.9999 or -9999.99999...+9999.99999	Yes
38	User data 8	-99999.9999...+99999.9999 or -9999.99999...+9999.99999	Yes
39	User data 9	-99999.9999...+99999.9999 or -9999.99999...+9999.99999	Yes
40	User data 10	-99999.9999...+99999.9999 or -9999.99999...+9999.99999	Yes

Cutter Status Bits

Bit	Symbol	Value	Group name	Group information	Change able	Comment
1	e	1 0	Wrong cutting edge position	Wrong cutter position Correct position	No	
2	1	1 0	L1 incorrect	L1 faulty Not faulty	No	
3	2	1 0	L2 incorrect	L2 faulty Not faulty	No	
4	3	1 0	L3 incorrect	L3 faulty Not faulty	No	
5	r	1 0	R incorrect	R faulty Not faulty	No	
6	*				No	Reserved
7	*				No	Reserved
8	*				No	Reserved
9	d	1 0	Wear condition	Cutter worn Cutter not worn	No	The cutter can no longer be used (replace)
10	w	1 0		Warning limit reached Warning limit not reached	No	The remaining life time is going to expire (replace).
11	*				No	Reserved
12	*				No	Reserved
13	A	1 0	ANW 1	User cutter status bit 1	Yes	Any meaning
14	B	1 0	ANW 2	User cutter status bit 2	Yes	Any meaning
15	C	1 0	ANW 3	User cutter status bit 3	Yes	Any meaning
16	D	1 0	ANW 4	User cutter status bit 4	Yes	Any meaning

Tool Group Status Bits

Bit	Sym bol	Value	Meaning	Bit information	Chang eable	Comment
1	!	1 0	Presence	Group not available Group available	No	Tool not available in this group
2	?	1 0	Presence	Group not required Group required	No	No tool required in this group
3	L	1 0	Group status	Group blocked Group not blocked	Yes	User-programmable
4	d	1 0	Wear condition	Group worn Group not worn	No	At least one T sister chain of the group is worn
5	w	1 0	Wear condition	Warning limit reached Warning limit not reached	No	At least one T sister chain of the group has reached warning limit
6	p	1 0	Sister identification	Machining group No machining group	No	Group is machining group
7	s	1 0	Sister identification	Spare group No spare group	No	Group is no spare group
8		1 0				Reserved for extension
9	ass.	1 0	User group status 1	User group status bit 1	Yes	Any meaning
10	ass.	1 0	User group status 2	User group status bit 2	Yes	Any meaning
11	ass.	1 0	User group status 3	User group status bit 3	Yes	Any meaning
12	ass.	1 0	User group status 4	User group status bit 4	Yes	Any meaning
13	ass.	1 0	User group status 5	User group status bit 5	Yes	Any meaning
14	ass.	1 0	User group status 6	User group status bit 6	Yes	Any meaning
15	ass.	1 0	User group status 7	User group status bit 7	Yes	Any meaning
16	ass.	1 0	User group status 8	User group status bit 8	Yes	Any meaning

5.175 Flow Diagram for Command Groups

NC Download Commands: IPP, NPC, NPD, NPI, PPD, PPN

MWCX device group

The following figure shows the procedure that is necessary for a complete NC download.

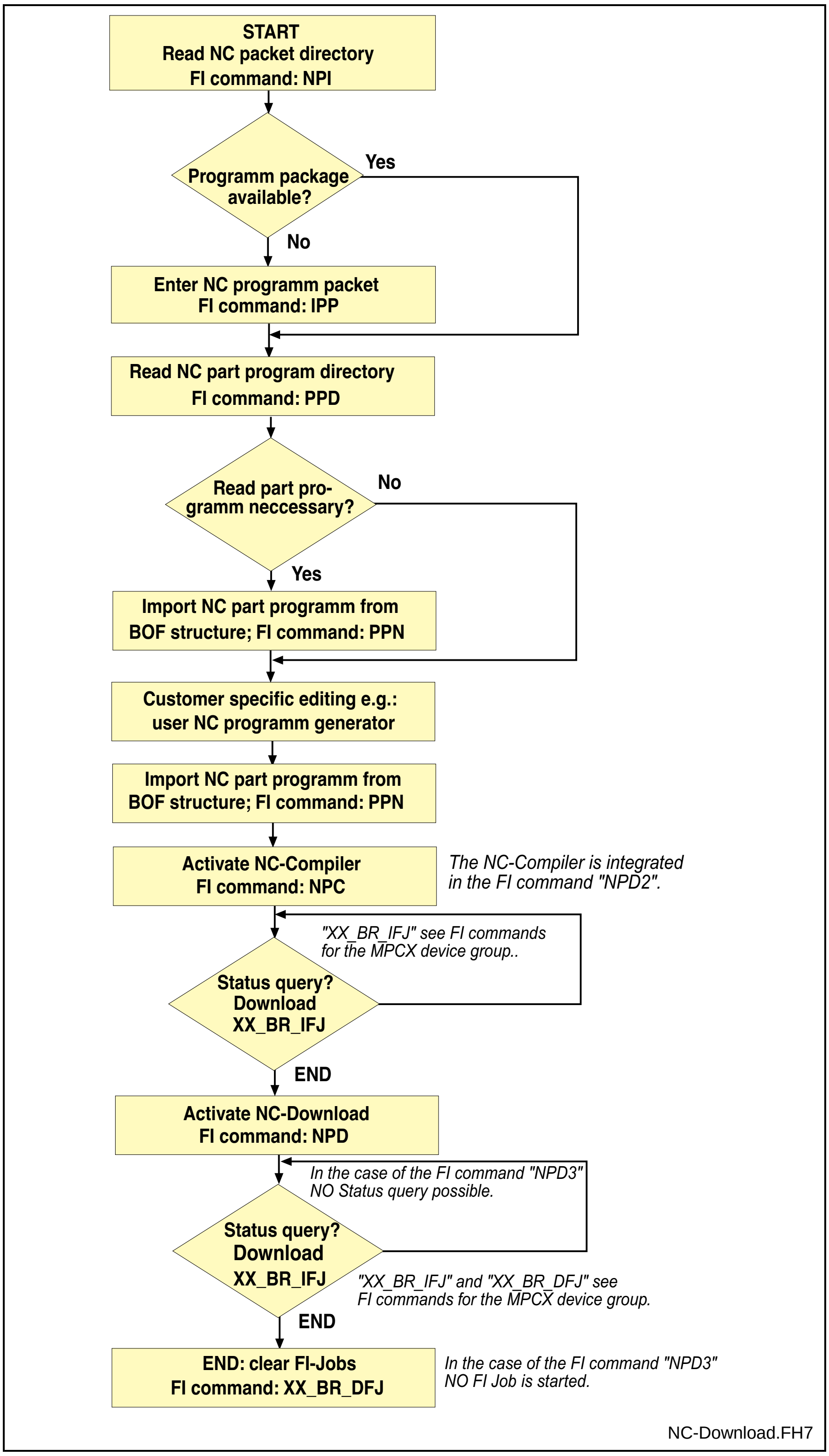


Fig. 5-9: Structure of NC download

Handling Tool Data Records: TDA, TRM

MWCX device group

The following diagram shows by way of an example the sequence (flow) required for editing complete tool data records.

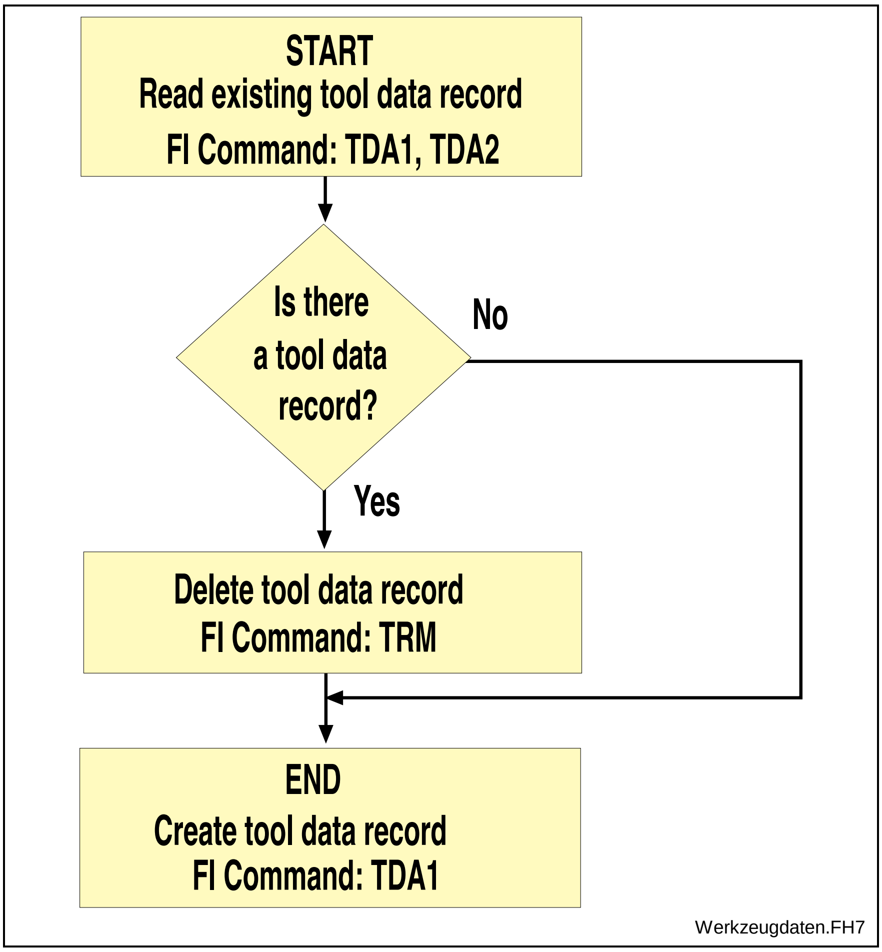


Fig. 5-10: Structure for handling tool data records

5.176 Using the Tool Command in Practice

In addition to exact knowledge of the individual commands, the multitude of tool commands also requires further information for practical usage.

This chapter therefore deals with this subject from the point of view of the user.

Fundamentals when Replacing a Tool

The control unit supports two different strategies:

i. The tool is transported to its previous location after use. The location remains reserved for the tool.

ii. The tool is transported to another, unassigned location after use. Only the control unit knows which tool is actually located where.

Point two is significant when a machine is equipped with a replacement grabber that fetches the tool from its tool location before it is actually used and then queues it. In some circumstances after use the old tool location may already be occupied by a tool that has previously been put down and therefore the next free location must be allocated.

Reading Tool Data

Note: Only the values from the tool database are read. No recognition is made of the tool that is physically inserted.

CR_TLD Returns data elements of a tool of the basic data or cutter data from the tool memory.

Note: No additional command required.

BR_TLB Returns one or more elements of the basic tool data of several tools from the tool memory.

Note: No additional command required.

BR_TLE Returns one or more elements of the tool cutter data of several tools from the tool memory.

Note: No additional command required.

CR_TDR Returns a complete basic data record or cutter data record of a tool in the tool memory.

Note: No additional command required.

BR_TDA Returns a complete tool data record consisting of the basic data and cutter data of a tool in the tool memory.

Note: No additional command required.

Block Tool Location

CR_TII The specified tool location is temporarily blocked from automatic assignment by the control unit.

Precondition: The tool location must be free (unassigned).

Note: No additional command required.

Release Tool Location

CR_TIF The indicated tool location is released after a temporary block.

Note: No additional command required.

Remove Tool

Note: Removing a tool means deleting the tool from the tool database. The actual tool itself must be removed previously by the user.

CR_TRM The tool data at this tool location is deleted from the database.

Note: No additional command required.

Modifying a Tool

Note: Only the tool data record in the tool memory is modified. The actual tool itself is not affected.

CW_TLD Writes a single element of the basic tool data or cutter data in the tool memory.

Note: No additional command required.

Replacing a Tool of the Same Type

Note: Inserting a tool should be understood as an updating of the tool database. The tool itself must have been previously inserted by the user at its location.

CW_TLD Writes a single element of the basic tool data or cutter data in the tool memory.

Note: This command possibly requires repeated calling up when a tool of the same type is to be replaced.

Replacing a Tool of a Different Type

Note: Inserting a tool should be understood as an updating of the tool database. The tool itself must have been previously inserted by the user at its location.

BW_TDA Writes a complete tool data record in the tool memory in a single access.

Note: This command must be carried out in the following order:

- CR_TRM Remove old tool.
- BW_TDA Write complete new tool data record.

Note: CR_TII and CR_TIF are already implemented in this command.

Moving a Tool

Note: Moving a tool should be understood as an updating of the tool database. The tool itself must have been previously inserted by the user at its new location.

CR_TMV A complete tool data record consisting of basic data and cutter data is moved.

Precondition: The target location must be free (unassigned).

Note: No additional command required.

Read Active Tool Number

CR_ATN The number of the active tool is read out.

Note: No additional command required.

Read Active Cutter Number

CR_AEN The number of the active cutter is read out.

Note: No additional command required.

Read Long Identification

CR_DIS4 The directory entry of the valid tool list is read out. It is updated after every download by CW_TDF.

Note: No additional command required.

Set Remaining Tool Life to 100%

CR_TRS The remaining tool life of a tool as a percentage is set to 100%.

Note: No additional command required.

Initiate Download

CW_TDI The control unit is prepared for the download of tool data.

Note: No additional command required.

Downloading Tool Data

CW_TDD The tool data for one or more tools is downloaded.

Note: This command must be carried out in the following order:

- CW_TDI Initiate download
- CW_TDD Write complete basic or cutting edge record data By means of repeated CW_TDD, all basic and cutting edge data of all tool of a tool magazine can be written (download).
- CW_TDF End download. the tool magazine is once more released

End Download

CW_TDF Download of tool data is completed.

Note: No additional command required.

6 FI Commands - MSCX Device Group (SERCANS)

The FI Commands described in this chapter are valid for the MSCX device group. The device types of this device group are listed in the following table:

Group	Device Type	Address
MSCX	SERCANS-A, SERCANS-P	[00]

Note: Please note that the device address must be set before the respective FI command, e.g., 00_BR_SPA1_3_S-0-0003_48 (refer also here to Chapter 2.1 "Elements of the FI Command").

6.1 Determining the Actual (Current) System Error: ASE

MSCX Device Group

Designation **ASE** Actual **S**ystem **E**rror

Explanation The current system error is read out, whereby the response 0x0000 indicates that the SERCANS card is functioning correctly.

FI command

CR_ASE **(Single Read)**

CC_ASE **(Cyclic Read)**

CB_ASE **(Break Cyclic Read)**

Response Structure The following table shows the general structure of the response to the FI command ASE. In line 1, column 4, the number of the drive is output that reports the current system error. Not all current system errors can be directly allocated to a drive. In this case, the single result "Drive No." is set to 0x0000.

Line 1	Column 1	...	Column 4

Value Range/Meaning of Columns

1 = 0x0000

2 = 0x0000

3 = Actual (current) system error

4 = Drive No.

Example ASE Reading the current system error returns LWL ring interrupted.

FI command		**00_CR_ASE**
Line	**Column**	**Answer**
1	1	0x0000
	2	0x0000
	3	0x8009
	4	0x0000

Reference to Literature See chapter entitled "Literature" [42].

6.2 Deleting the Actual (Current) System Error: CSE

MSCX Device Group

Designation **CSE** **C**lear **S**ystem **E**rror

Explanation An error reported by the SERCANS card is deleted.

FI command **CW_CSE** **(Single Write)**

Value to be written: The contents of the value parameter is not evaluated.

Response Structure The following table shows the general structure of the response to the FI command "CSE". In line 1, column 4, the number of the drive is output that reports the current system error. Not all current system errors can be directly allocated to a drive. In this case, the single result "Drive No." is set to 0x0000.

Line 1	Column 1	...	Column 4

Value Range/Meaning of Columns

1 = 0x0000

2 = 0x0000

3 = Actual (current) system error

4 = Drive No.

Example CSE Deleting the actual (current) system error:

FI command		**00_CW_CSE**
Line	**Column**	**Answer**
1	1	0x0000
	2	0x0000
	3	0x0000
	4	0x0000

Reference to Literature see chapter entitled "Literature" [45].

6.3 Setting the Communication Timeout Time DCT

MSCX Device Group

Designation **DCT** **D**evice **C**ommunication **T**imeout

Explanation By means of this command, the timeout time for the selected device is set dynamically (timeout time in ms).

FI command **BW_DCT1_(1)** **(Single Write)**

(1) = requested timeout time in ms

Response Structure The response to the "DCT1" FI command consists of one line with one column.

Line 1	**Column 1**

Value Range/Meaning of Columns

1 = Status message (P_ACK) (P_ACK)

Example DCT1 For the device 00, the timeout time is set 1500 ms.

FI command		**00_BW_DCT1_1500**
Line	**Column**	**Answer**
1	1	(P_ACK)

FI command With this command, the timeout time for the selected device can be reset to default value.

BW_DCT2 **(Single Write)**

Response Structure The response to the "DCT2" FI command consists of one line with one column.

Line 1	**Column 1**

Value Range/Meaning of Columns

1 = Status message (P_ACK) (P_ACK)

Example DCP2 For the device 00, the timeout time is reset to the default value.

FI command		**00_BW_DCT2**
Line	**Column**	**Answer**
1	1	(P_ACK)

6.4 Deleting the FI Exclusive Mode: DEM

MSCX Device Group

Designation **DEM** **D**elete FI **E**xclusive **M**ode

Explanation This command is used to deactivate FI Exclusive mode for the selected device address.

FI Exclusive mode: In this mode, **ALL** the processes logged in at the FI – **with the exception** of the process issuing the SEM command – are blocked from data communication with this device address. This mode is used for example for data communication which must NOT be interrupted by other data requests. However, it is **imperative** that this FI Exclusive mode is deleted once more through the DEM command.

FI command **BW_DEM1** **(Single Write)**

Response Structure The following table shows the general structure of the response to the FI command "BW_DEM1". A line of 1 column is output.

Line 1	Column 1

Value Range/Meaning of Columns 1 = Status message (P_ACK) (P_ACK)

Example DEM1 Deactivate the FI Exclusive mode for device address 0. The FI Exclusive mode for device address 0 has previously been activated by the SEM command.

FI command		**00_BW_DEM1**
Line	**Column**	**Answer**
1	1	(P_ACK)

6.5 Static Device Information: DIF

MSCX Device Group

Designation **DIF** **D**evice **I**n**F**ormation

Explanation Static device information and network information is read according to the "IND_DEV.INI" and "FAR_DEV.INI" files.

FI command Reading of the static device information and network information of a selected device.

BR_DIF1 **(Single Read)**

BC_DIF1 **(Cyclic Read)**

BB_DIF1 **(Break Cyclic Read)**

Response Structure The following table shows the general structure of the response to the "DIF1" FI command. The response consists of one line with 24 columns.

Line 1	Column 1	...	Column 24

Value Range/Meaning of Columns

1 =	Local/far device address	[00..63]
2 =	Device name	IND_DEV.INI entry: DeviceName=
3 =	Device type	IND_DEV.INI entry: DeviceType=
4 =	PLC support	IND_DEV.INI entry: PLC=
5 =	Device status	IND_DEV.INI entry: DeviceStatus=
6 =	Assignment of a simulation pair	IND_DEV.INI entry: DeviceAssign=
7 =	Device mode	IND_DEV.INI entry: MtvncMode=
8 =	Communication channel	IND_DEV.INI entry: [CommAddrX]
9 =	Description of the communication channel	IND_DEV.INI entry: CommStr=
10 =	Timeout value	IND_DEV.INI entry: Timeout=
11 =	Device group	(see Chapter 6.1 "Identifier")
12 =	PLC component type	IND_DEV.INI entry: Component type1=
13 =	CNC component type	IND_DEV.INI entry: Component type2=
14 =	Device log	IND_DEV.INI entry: DeviceProtocol=
15 =	Device simulation	IND_DEV.INI entry: DeviceSimulation=
16 =	Not yet assigned	[--]
17 =	Not yet assigned	[--]
18 =	Not yet assigned	[--]
19 =	Not yet assigned	[--]
20 =	Network ON/OFF	[ON = Network active OFF = No network active]
21 =	Network name	Max. 28 ASCII characters
22 =	PC number	[00..99,XX]
23 =	PC name	Max. 255 ASCII characters
24 =	Local device address	[00..63]

Example DIF1 Read the static device information and network information of device 1 while the network is active.

FI command		**01_BR_DIF1**
Line	**Column**	**Answer**
1	1	01
	2	DRIVE LEFT SIDE
	3	SERCANS-A
	4	NO
	5	ON
	6	NO
	7	OFF
	8	4
	9	V24,COM2,19200,EVEN,RS232,TCOFF
	10	3500
	11	MSCX
	12	NONE
	13	NONE
	14	CNC
	15	OFF
	16	--
	17	--
	18	--
	19	--
	20	ON
	21	PC network 1
	22	29
	23	BTV20-STATION-LEFT
	24	01

6.6 Reading the Device Status Information: DSI

MSCX Device Group

Designation **DSI** Device **S**tatus **I**nformation

Explanation This allows the most important device status information to be read out. The following information is returned:

Type of information	Status	Statement
System error information		Yes/No
Mechanism error information		Yes/No
Machine key information		4 Byte HEX
Machine key information	valid	Yes/No
Machine status information		4 Byte HEX
Sercans information		4 Byte HEX
Parameter download	running	Yes/No
PLC download	running	Yes/No
Firmware download	running	Yes/No
Offline/Online information	Communication?	Yes/No
Device simulation	switched on	Yes/No
Device status information		ON/ OFF
Communication channel defined		Yes/No
PLC components available		Yes/No
Monitor mode	Active	Yes/No

FI command Read out device status information for ALL defined devices.

BR_DSI1	**(Single Read)**
BC_DSI1	**(Cyclic Read)**
BB_DSI1	**(Break Cyclic Read)**

Note: The "DSI1" FI command refers to all devices within this device group. Therefore, any valid device address can be indicated in the command line (see example DSI1). The FI device polling mechanism **MUST** be switched on (see system configurator)!

Response Structure The following table shows the general structure of the response to the "DSI1" FI command.

Line 1...n	Column 1	...	Column 11

Value Range/Meaning of Columns

1 =	device address	[00...63]
2 =	System error information	[0 = there is no system error 1 = there is a system error]
3 =	Mechanism error information	[0 = there is no mechanism error 1 = there is a mechanism error
4 =	Machine key information	[4 byte in HEX coding]
5 =	Is machine key information valid?	[0 = not valid, 1=valid]
6 =	Machine status information	[4 byte in HEX coding]
7 =	Sercans information	[4 byte in HEX coding]
8 =	Is parameter download active?	[0 = parameter download not running 1 = parameter download running]
9 =	Is PLC download active?	[0 = PLC download not running 1 = PLC download running]
10 =	Is firmware download active?	[0 = PLC download not running 1 = PLC download running]
11 =	Offline/Online information	[0 = device connection interrupted 1 = device connection O.K.]
12 =	Device simulation switched on?	[0 = NO Simulation mode 1 = simulation mode]
13 =	Current device status information	[0 = Device status=OFF 1 = Device status=ON]
14 =	Communication channel defined?	[0 = NO communication channel 1 = Communication channel defined]
15 =	PLC components available ?	[0 = NO PLC component 1 = PLC component (DOS-Pcl) 2 = PLC component (WIN-Pcl)]
16 =	Monitor mode	[0 = NO monitor mode active 1 = Monitor mode active]

Example DSI1

Read the current device status information.
Assumption:
The following devices addresses are defined:

Device address 01 (SERCANS-A)

Device address 03 (SERCANS-P)

FI command		**01_BR_DSI1**
Line	**Column**	**Answer**
1	1	01
	2	0
	3	0
	4	00000000
	6	00000000
	5	0
	7	00000000
	8	0
	9	0
	10	0
	11	1
	12	0
	13	1
	14	1
	15	0
	16	0
2	1	03
	2	1
	3	0
	4	00000000
	5	0
	6	00000000
	7	00000000
	8	0
	9	0
	10	0
	11	1
	12	0
	13	1
	14	1
	15	0
	16	0

FI command

Read out device status information for a selected device.

BR_DSI2	**(Single Read)**
BC_DSI2	**(Cyclic Read)**
BB_DSI2	**(Break Cyclic Read)**

Response Structure

The following table shows the general structure of the response to the "DSI2" FI command.

Line 1...n	Column 1	...	Column 11

Value Range/Meaning of Columns

1 =	device address	[00...63]
2 =	System error information	[0 = there is no system error 1 = there is a system error]
3 =	Mechanism error information	[0 = there is no mechanism error 1 = there is a mechanism error]
4 =	Machine key information	[4 byte in HEX coding]
5 =	Machine key information valid?	[0 = not valid, 1=valid]
6 =	Machine status information	[4 byte in HEX coding]
7 =	Sercans information	[4 byte in HEX coding]
8 =	Is parameter download active?	[0 = parameter download not running 1 = parameter download running]
9 =	Is PLC download active?	[0 = PLC download not running 1 = PLC download running]
10 =	Is firmware download active?	[0 = PLC download not running 1 = PLC download running]
11 =	Offline/Online information	[0 = device connection interrupted 1 = device connection O.K.]
12 =	Device simulation switched on?	[0 = NO Simulation mode 1 = simulation mode]
13 =	Current device status information	[0 = Device status=OFF 1 = Device status=ON]
14 =	Communication channel defined?	[0 = NO communication channel 1 = Communication channel defined]
15 =	PLC components available ?	[0 = NO PLC component 1 = PLC component (DOS-Pcl) 2 = PLC component (WIN-Pcl)]
16 =	Monitor mode	[0 = NO monitor mode active 1 = Monitor mode active]

Example DSI2 Read the current device status information for the selected device.

FI command		00_BR_DSI2
Line	**Column**	**Answer**
1	1	00
	2	0
	3	0
	4	00000000
	5	0
	6	00000000
	7	00000000
	8	0
	9	0
	10	0
	11	1
	12	0
	13	1
	14	1
	15	0
	16	0

6.7 Device Type and Accompanying Components: DTY

MSCX Device Group

Designation **DTY** **D**evice **TY**pe

Explanation The device type and the accompanying components of the selected device address are output.

FI command **BR_DTY1** **(Single Read)**

Response Structure The following table shows the general structure of the response to the "DTY1" FI command. A line with three columns for the device type is output as well as the names of the first device component and the name of the second device component.

Line 1	Column 1	...	Column 3

Value Range/Meaning of Columns

1 =	Device Type	(see chapter entitled "Elements of the FI Command", and "Identifier")
2 =	Component type1	IND_DEV.INI entry: Component type1=
3 =	Component type 2	IND_DEV.INI entry: Component type2=

Example DTY1 Output the device type and the accompanying components of device address 00.

FI command	00_BR_DTY1		
Answer			
Line	**Column 1**	**Column 2**	**Column 3**
1	SERCANS-A	NONE	NONE

6.8 Read System Messages: MSG

MSCX Device Group

Designation **MSG** **M**e**S**sa**G**e

Explanation Reading of system messages

FI command Message

CC_MSG_(1) **(Cyclic Read)**

(1) = SYS-Message number

Note: Exists only as a cyclic command

Response Structure The response of the FI command 'MSG' consists of the system message data.

Example MSG 00_CC_MSG_64 (64 = MSG_SYSERRGEN)

FI command		**00_CC_MSG_64/3**
Line	**Column**	**Answer**
1	1	00

Restriction The following system messages:

SYS Message	SYS message numbers
MSG_PCLUPDBEG	52
MSG_PARUPDBEG	24
MSG_FWAUPDBEG	82

These commands cannot be used with the following programs:

- Bosch Rexroth OPC server
- Bosch Rexroth DDE server

6.9 Set the Device Status Information: SDS

MSCX Device Group

Designation **SDS** **S**et **D**evice **S**tatus

Explanation By this command, the device status information can be set; here, the configuration file IND_DEV.INI is adjusted as well.

Note: When this command is transmitted, the following system messages are generated:
MSG_DEVICEOFF or MSG_DEVICE_ON !

FI command With this command, the device status information of **ALL** defined devices can be set.

BW_SDS1_(1) **(Single Write)**

(1) =	Device status information to be set	0 = Device status information OFF 1 = Device status information ON

Response Structure The following table shows the general structure of the response to the "SDS1" FI command.

Line 1	Column 1

Value Range/Meaning of Columns

1 =	Status report	[(P_ACK)]

Example: SDS1 Set device status information to OFF for **ALL** defined devices.

FI command		**00_BW_SDS1_0**
Line	**Column**	**Answer**
1	1	(P_ACK)

FI command With this command, the device status information for a selected device can be set.

BW_SDS2_(1) **(Single Write)**

Device status information to be set	0 = Device status information OFF 1 = Device status information ON

Response Structure The following table shows the general structure of the response to the "SDS2" FI command.

Line 1	Column 1

Value Range/Meaning of Columns

1 =	Status report	[(P_ACK)]

Example: SDS2 Set device status information to OFF for the selected device 00.

FI command		**00_BW_SDS2_0**
Line	**Column**	**Answer**
1	1	(P_ACK)

6.10 Setting the FI Exclusive Mode: SEM

MSCX Device Group

Designation **SEM** **S**et FI **E**xclusive **M**ode

Explanation This command is used to activate FI Exclusive mode for the selected device address.

FI Exclusive mode: In this mode, **ALL** the processes logged in at the FI – **with the exception** of the process issuing the SEM command – are blocked from data communication with this device address. This mode is used for example for data communication which must NOT be interrupted by other data requests. However, it is **imperative** that this FI Exclusive mode is deleted once more through the DEM command.

FI command **BW_SEM1** **(Single Write)**

Response Structure The following table shows the general structure of the response to the FI command "BW_SEM1". A line of 1 column is output.

Line 1	Column 1

Value Range/Meaning of Columns 1 = Status message (P_ACK) (P_ACK)

Example SEM1 Activate FI Exclusive mode for device address 0.

FI command		**00_BW_SEM1**
Line	**Column**	**Answer**
1	1	(P_ACK)

6.11 Software Installation Data: SID

MSCX Device Group

Designation **SID** **S**oftware **I**nstallation **D**ata

Explanation Information is returned regarding installation. This information includes installation paths, the software version used, DLL mode, plus service pack and release information.

FI command Read-in the installation data.

BR_SID1 **(Single Read)**

Response Structure One line with 16 columns is output for the returned values.

Line 1	Column 1	...	Column 16

Meaning of the Columns

1 = Basic directory	[EXE files of the DOS-BOF]
2 = FI installation directory	[FI directory]
3 = Data directory	[in accordance with DOS-BOF]
4 = GBO version	[from INDRAMAT.ini]
5 = IF-DLL mode	[from INDRAMAT.ini]
6 = IF version	[from INDRAMAT.ini from DLL mode 400]
7 = Service pack info	[from INDRAMAT.ini from DLL mode 420]
8 = Release info	[from INDRAMAT.ini from DLL mode 420]
9 = IF-Build-Info	[in accordance with Build process]
10 = Current context name	[in accordance with the installation]
11 = Physical installation path	[in accordance with the installation]
12 = Complete IF version indication string	
13 = WinPCL build number	[in accordance with WinPCL]
14 = Version number of the PLC compiler	[in accordance with WinPCL]
15 = Version number of the PLC linker	[in accordance with WinPCL]
16 = Version number of the PLC data basis	[in accordance with WinPCL]
17 = Platform version	

Example SID1 Return information on the current installation.

FI command		**00_BR_SID1**
Line	**Column**	**Answer**
1	1	--
	2	D:\Program Files\Indramat\MTGUI\Bin
	3	--
	4	005-22Vxx
	5	07.20
	6	07V00
	7	--
	8	--
	9	Build 3124 Mar 6 2003 08:53:55
	10	MTGUI_0-23T01 B3327
	11	D:\Program Files\Indramat\MTGUI\
	12	FI: 07V00 DLL-Mode: 07.20 Build 3124 Mar 6 2003 08:53:55
	13	347.15.4.11
	14	771
	15	515
	16	78
	17	Platform: 02V01 Build: 3214

Note: Refer to FI command "PHD" for working with absolute paths of the MPCX group.

6.12 SERCOS Parameters: SPA

MSCX Device Group

Designation **SPA** **S**ERCOS **PA**rameter

Explanation One SERCOS parameter of a drive or a SERCANS parameter is read out or is written. Each parameter consists of 7 elements, whereby any combination of elements can be selected by element coding.

FI command

BR_SPA1_(1)_(2)_(3)	**(Single Read)**
BC_SPA1_(1)_(2)_(3)	**(Cyclic Read)**
BB_SPA1_(1)_(2)_(3)	**(Break Cyclic Read)**
BW_SPA1_(1)_(2)_(3)	**(Single Write)**
(1) = Drive address	[0...99]
(2) = Parameter No.	in the format: X-Y-ZZZZ
(3) = Element coding	[standard or advanced format]

Parameter No.

Format X-Y-ZZZZ	Value Range
X	S = standard data P = product data Y = SERCANS parameter
Y	[0..7] = parameter record
Z	[0...4095] = data block no.

Element Coding

Element coding in standard format allows individual elements, such as the operating date, to be requested. If several elements are to be read out in one request, then the element coding can be OR'd in advanced format, e.g. operating date (0x40) and unit (0x08) produces OR'd (0x48) → 48

The advanced format 0x80 has priority over 0x40.

Element	Standard Format	Advanced Format	Format:	Example
Data status	S:	01H	Hexadecimal word	0x0000
Name	The marked section is then printed out.	02H	String	NC cycle time (TNcyc)
Attribute	A	04H	Hexadecimal double word	0x60110001
Unit	U	08H	String	µs
Min. input value	L	10H	Decimal word	2000
Max. input value	H	20H	Decimal word	20000
Operating date	D	40H	(see Displaying the Operating Date	
Operating date, when no list		80H		

Displaying the Operating Date

The display of the operating date depends on the parameter number requested.

Decimal

Decimal values are given as floating points, e.g. 1.5. Leading spaces, zeros, plus and minus signs as well as trailing spaces are allowed.

Hexadecimal

Hexadecimal values are displayed by "0x...", e.g. 0x80. Up to a maximum of eight positions are allowed. Leading or trailing spaces are allowed. Leading additional zeros or plus and minus signs are not allowed.

Binary (max. 32 characters)

Leading or trailing spaces are allowed. The decimal point serves as separator:

e.g., 1111.0000.1010.1100.1111.0000.1010.1100

Note: Leading additional zeros or plus and minus signs are not allowed.

ID number

The following table shows the general way in which the ID number is displayed:

Format X-Y-ZZZZ	Value Range
X	S = standard data P = product data
Y	[0..7] = parameter record
Z	[0...4095] = data block no.

(see example SPA1/write).

Lists of Variable Length

Lists always begin with two decimal numbers for the actual length and maximum length of the list. The length specification refers to the length of the list in the drive and therefore designates the number of bytes for storage (storage bytes). The number of elements in the list can be calculated using the attribute. The list elements are displayed according to the attribute. All parts of the list are separated from each other by a line feed ("\n").

Example:

Parameter S-0-0017, IDN list of all parameters

"400\n400\nS-0-0001\nS-0-0002\n..."

ASCII List

ASCII lists are a special form of variable length lists. The individual string characters are not separated by a line feed. When displaying the lists, a distinction is made between standard format and advanced format. In standard format, only the character string is returned, whereas in advanced format the actual length and the maximum length of the list (string) is also transmitted.

Example:

Parameter S-0-0030, operation date
Standard format: "DKC2.1-SSE-01V09"
Advanced format: "16\n16\nDKC2.1-SSE-01V09"

Note: When requesting SERCANS parameters the drive address can be anywhere within the range [0..254].

Response Structure

The following table shows the general structure of the response to the FI command "SPA1". Line 1 is output both when reading and when writing. Additional lines are only output when reading depending on the element coding.

Note: If the element coding has been requested in standard format then the first line is not applicable.

Note: Line 1 is a status line that either contains SERCOS / SERCANS errors or displays the successful processing of the FI command. If the command has been processed successfully, then columns 1 and 3 contain the value [0x0000].

In the first line, column 2 or column 4, the number of the drive is output that reports the SERCOS error or the global SERCANS error. Not all global SERCANS errors can be directly assigned to a drive. In this case, the single result "Drive No." is set to 0x0000.

Line	Column 1	Column 2	Column 3	Column 4
1	<SERCOS error>	<Drive no. SERCOS error>	<Global SERCANS error>	<Drive No. Global SERCANS error>
2	Read: Element corresponding to the element coding.			
...	...			
n	Read: (n-1). Element corresponding to the element coding.			

Example SPA1 / read Read parameter S-0-0003 of the 3rd drive (element coding 0x48)

FI command	00_BR_SPA1_3_S-0-0003_48			
Answer				
Line	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	0x0000	0x0000	0x0000	0x0000
2	µs			
3	2000			

Example SPA1 / write Write the ID number P-0-0037 in parameter S-0-0305 of the 3rd drive (element coding 0x40).

Technical background:

- Realtime status bit 1 is to be assigned the trigger status word of the oscilloscope function of a DIAX04 drive.

FI command	Value to be written: : P-0-0037 00_BW_SPA1_3_S-0-0305_40			
Answer				
Line	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	0x0000	0x0003	0x0000	0x0000

Reference to Literature See chapter entitled "Literature" [41].

See chapter entitled "Literature" [46].

6.13 Active SERCOS Phase Switch-Over: SPH

MSCX Device Group

Designation **SPH** **S**ERCOS **PH**ase

Explanation All drives within a SERCOS ring are in the same communication phase. The phase status can be read-out or changed by this command.

FI command

BR_SPH	**(Single Read)**
BC_SPH	**(Cyclic Read)**
BB_SPH	**(Break Cyclic Read)**
BW_SPH	**(Single Write)**

Value to be written/ Result The phase conditions allowed are shown by the numbers [0...4].

Response Structure The following table shows the general structure of the response to the FI command "SPH". In the first line, column 2 or column 4, the number of the drive is output that reports the SERCOS error or the global SERCANS error. Not all global SERCANS errors can be directly assigned to a drive. In this case, the single result "Drive No." is set to 0x0000.

Line	Column 1	Column 2	Column 3	Column 4
1	<SERCOS error>	<Drive No. SERCOS error>	<Global SERCANS error>	<Drive no. that has caused the global SERCANS error>
2	Read: current phase Write: previously phase			

Example BR_SPH
Read SERCOS phase

Read the active phase of SERCAN sontrol on device address 00.

<table>
<tr><th colspan="2">FI command</th><th colspan="3">00_BR_SPH</th></tr>
<tr><th colspan="5">Response</th></tr>
<tr><th>Line</th><th>Column 1</th><th>Column 2</th><th>Column 3</th><th>Column 4</th></tr>
<tr><td>1</td><td>0x0000</td><td>0x0000</td><td>0x0000</td><td>0x0000</td></tr>
<tr><td>2</td><td>2</td><td></td><td></td><td></td></tr>
</table>

Example BW_SPH
Write SERCOS phase

Switching (writing) of SERCANS control to phase 4; phase 2 is active.

<table>
<tr><th colspan="2">FI command</th><th colspan="3">Value to be written: 4
00_BW_SPH</th></tr>
<tr><th colspan="5">Response</th></tr>
<tr><th>Line</th><th>Column 1</th><th>Column 2</th><th>Column 3</th><th>Column 4</th></tr>
<tr><td>1</td><td>0x0000</td><td>0x0000</td><td>0x0000</td><td>0x0000</td></tr>
<tr><td>2</td><td>2</td><td></td><td></td><td></td></tr>
</table>

Reference to Literature

See chapter entitled "Literature" [42].

7 FI Commands - MVMX Device Group (VM)

The FI Commands described in this chapter are valid for the MVMX device group. In this device group, the following types are listed as well as possible device addresses:

Group	Accompanying Types	Address
MVMX	VM-P, VM-R	[00...63]

Note: The Visual Motion component has been realized under SCP (Scalable Communication Platform).

Please note that the device address must be set before the respective FI command, e.g. 00_BR_ASM1 (refer also here to chapter 2.1 "Elements of the FI Command").

7.1 Active System Error Messages: ASM

MVMX device group

Designation **ASM** Active **S**ystem **M**essages

Explanation The active device information is output (system errors, device statuses) that affect the functioning of the entire electrical device. Depending on the FI command, the device address, device name, message number, type of message, short text and additional text are all output.

FI command Output the currently pending device information (system errors, device statuses) of the selected device from the MWAX device group.

BR_ASM2 **(Single Read)**

BC_ASM2 **(Cyclic Read)**

BB_ASM2 **(Break Cyclic Read)**

Response Structure The following table shows the general structure of the response to the FI command "ASM2". The answer consists of a line of 7 columns for the device address, device name, message number, message status, short text and indication of whether there is an additional text for this device information (system errors, device statuses).

Line 1...n	Column 1	...	Column 7

Value Range/Meaning of Columns

1 =	Device address	[00...15]
2 =	Device name	[max. 32 ASCII characters]
3 =	Message number	[0...150]
4 =	Type of message	[F = fault/error, D = diagnosis]
5 =	Short text	[max. 54 ASCII characters]
6 =	Reference text	[x= exists, -- = does not exist]
7 =	2 bytes of additional information for the message number	is required to resolve the information "@" (see ASM5)

Example ASM2

Read the current device information (system errors, device statuses) of device address 01.

Assumption:
the following three devices are defined:

- Device address 01
- Device address 07 and
- Device address 10

FI command		01_BR_ASM2
Line	**Column**	**Answer**
1	1	01
	2	Drill center
	3	71
	4	F
	5	PLC battery voltage too low.
	6	X
	7	0

FI command

Output the additional text for the currently pending device information (system errors, device statuses) related to the device and the message number.

BR_ASM5_(1)_(2) (Single Read)

(1) = Message number [0...150]

(2) = 2 bytes of additional information for the message number

Response Structure

The following table shows the general structure of the response to the FI command "ASM5". The answer consists of a line with 5 columns for the device address, device name, message number and additional text.

Line 1...n	Column 1	...	Column 5

Value Range/Meaning of Columns

1 =	Device address	[00...15]
2 =	Device name	[max. 32 ASCII characters]
3 =	Message number	[0...150]
4 =	Type of message	[F = fault/error, D = diagnosis]
6 =	Reference text	[max. 14 lines with a max. 78 characters/line]

Example ASM5

Read the additional text relating to the device information (system errors, device statuses) with message number 74 of device address 01.

FI command		01_BR_ASM5_74_0
Line	**Column**	**Answer**
1	1	01
	2	Drill center
	3	74
	4	F
	5	Replace the SLM module on the PLC card (slot 3).

Reference to Literature

See chapter entitled "Literature" [13].

7.2 Trigger Control Reset: CRT

MVMX device group

Designation **CRT** **C**ontrol **R**ese**T**

Explanation The control reset allows the selected device to be reset after a system error. If there is no system error at the selected device then the job is ignored.

CAUTION

Carrying out a reset completely re-initializes the device.
During initialization, communication is temporarily interrupted (inherent to design).

FI command **BW_CRT** **(Single Write)**

Value to be written Trigger reset 0

Note: The value to be written is passed to the "acValue" parameter in the "DataTransfer" routine.

Response Structure The return value of the "DataTransfer" routine is [0] if the write procedure has been successfully completed. In the event of an error, more information in the form of a general error result line can requested by the routine "ReadGroupItem" (refer here to Function Interface – Application Manual, chapter "Error Codes", "General Error Result Line").

Example CRT Trigger a control reset on the selected device.

FI command		**Value to be written: 0** **00_BW_CRT**
Line	**Column**	**Answer**
1	1	(P_ACK)

Reference to Literature See chapter entitled "Literature" [26].

7.3 Setting the Communication Timeout Time: DCT

MVMX device group

Designation **DCT** Device Communication Timeout

Explanation By means of this command, the timeout time for the selected device is set dynamically (timeout time in ms).

FI command **BW_DCT1_(1)** **(Single Write)**

(1) = desired timeout time in ms

Response Structure The response to the "DCT1" FI command consists of one line with one column.

Line 1	Column 1

Value Range/Meaning of Columns 1 = Status message (P_ACK) (P_ACK)

Example DCT1 For the device 00, the timeout time is set 1500 ms.

FI command		00_BW_DCT1_1500
Line	**Column**	**Answer**
1	1	(P_ACK)

FI command With this command, the timeout time for the selected device can be reset to default value.

BW_DCT2 **(Single Write)**

Response Structure The response to the "DCT2" FI command consists of one line with one column.

Line 1	Column 1

Value Range/Meaning of Columns 1 = Status message (P_ACK) (P_ACK)

Example DCP2 For the device 00, the timeout time is reset to the default value.

FI command		00_BW_DCT2
Line	**Column**	**Answer**
1	1	(P_ACK)

7.4 Deleting the FI Exclusive Mode: DEM

MVMX device group

Designation **DEM** **D**elete FI **E**xclusive **M**ode

Explanation This command is used to deactivate FI Exclusive mode for the selected device address.

FI Exclusive mode:

In this mode, **ALL** the processes logged in at the FI – **with the exception** of the process issuing the SEM command – are blocked from data communication with this device address. This mode is used for example for data communication which must NOT be interrupted by other data requests. However, it is **imperative** that this FI Exclusive mode is deleted once more through the DEM command.

FI command **BW_DEM1** **(Single Write)**

Response Structure The following table shows the general structure of the response to the FI command "BW_DEM1". A line of 1 column is output.

Line 1	Column 1

Value Range/Meaning of Columns 1 = Status message (P_ACK) (P_ACK)

Example DEM1 Deactivate the FI Exclusive mode for device address 0. The FI Exclusive mode for device address 0 has previously been activated by the SEM command.

FI command		**00_BW_DEM1**
Line	**Column**	**Answer**
1	1	(P_ACK)

7.5 Static Device Information: DIF

MVMX device group

Designation **DIF** **D**evice **I**n**F**ormation

Explanation Static device information and network information is read according to the "IND_DEV.INI" and "FAR_DEV.INI" files.

FI command Reading of the static device information and network information of a selected device.

BR_DIF1	**(Single Read)**
BC_DIF1	**(Cyclic Read)**
BB_DIF1	**(Break Cyclic Read)**

Response Structure The following table shows the general structure of the response to the "DIF1" FI command. The response consists of one line with 24 columns.

Line 1	Column 1	...	Column 24

Value Range/Meaning of Columns

1 =	Local/far device address	[00..63]
2 =	Device name	IND_DEV.INI entry: DeviceName=
3 =	Device type	IND_DEV.INI entry: DeviceType=
4 =	PLC support	IND_DEV.INI entry: PLC=
5 =	Device status	IND_DEV.INI entry: DeviceStatus=
6 =	Assignment of a simulation pair	IND_DEV.INI entry: DeviceAssign=
7 =	Device mode	IND_DEV.INI entry: MtvncMode=
8 =	Communication channel	IND_DEV.INI entry: [CommAddrX]
9 =	Description of the communication channel	IND_DEV.INI entry: CommStr=
10 =	Timeout value	IND_DEV.INI entry: Timeout=
11 =	Device group	(see Chapter 6.1 "Identifier")
12 =	PLC component type	IND_DEV.INI entry: Component type1=
13 =	CNC component type	IND_DEV.INI entry: Component type2=
14 =	Device log	IND_DEV.INI entry: DeviceProtocol=
15 =	Device simulation	IND_DEV.INI entry: DeviceSimulation=
16 =	Not yet assigned	[--]
17 =	Not yet assigned	[--]
18 =	Not yet assigned	[--]
19 =	Not yet assigned	[--]
20 =	Network ON/OFF	[ON = Network active OFF = No network active]
21 =	Network name	Max. 28 ASCII characters
22 =	PC number	[00..99,XX]
23 =	PC name	Max. 255 ASCII characters
24 =	Local device address	[00..63]

Example DIF1 Read the static device information and network information of device 1 while the network is active.

FI command		01_BR_DIF1
Line	**Column**	**Answer**
1	1	01
	2	Station right side
	3	VM-R
	4	NO
	5	ON
	6	NO
	7	OFF
	8	4
	9	V24,COM2,19200,EVEN,RS232,TCOFF
	10	3500
	11	MVMX
	12	NONE
	13	NONE
	14	SIS
	15	OFF
	16	--
	17	--
	18	--
	19	--
	20	ON
	21	PC network 1
	22	29
	23	BTV20-STATION-LEFT
	24	01

7.6 Reading the Device Status Information: DSI

MVMX device group

Designation **DSI** **D**evice **S**tatus **I**nformation

Explanation This enables the most important device status information to be read. The following information is returned:

Type of information	status	Statement
System error information		Yes/No
Mechanism error information		Yes/No
Machine key information		4 Byte HEX
Machine key information	valid	Yes/No
Machine status information		4 Byte HEX
Sercans information		4 Byte HEX
Parameter download	running	Yes/No
PLC download	running	Yes/No
Firmware download	running	Yes/No
Offline/Online information		Yes/No
Device simulation	switched on	Yes/No
Device status information		ON/ OFF
Communication channel defined		Yes/No
PLC components available		Yes/No
Monitor mode	active	Yes/No

FI command Read out device status information for ALL defined devices.

BR_DSI1	**(Single Read)**
BC_DSI1	**(Cyclic Read)**
BB_DSI1	**(Break Cyclic Read)**

Note: The "DSI1" FI command refers to all devices within this device group. Therefore, any valid device address can be indicated in the command line (see example DSI1). The FI device polling mechanism **MUST** be switched on (see system configurator)!

Response Structure The following table shows the general structure of the response to the "DSI1" FI command.

Line 1...n	Column 1	...	Column 11

Value Range/Meaning of Columns

1 =	Device address	[00...63]
2 =	System error information	[0 = there is no system error 1 = there is a system error]
3 =	Mechanism error information	[0 = there is no mechanism error 1 = there is a mechanism error]
4 =	Machine key information	[4 byte in HEX coding]
5 =	Is machine key information valid?	[0 = not valid, 1=valid]
6 =	Machine status information	[4 byte in HEX coding]
7 =	Sercans information	[4 byte in HEX coding]

8 =	Is parameter download active?	[0 = parameter download not running 1 = parameter download running]
9 =	Is PLC download active?	[0 = PLC download not running 1 = PLC download running]
10 =	Is firmware download active?	[0 = PLC download not running 1 = PLC download running]
11 =	Offline/Online information	[0 = device connection interrupted 1 = device connection O.K.]
12 =	Device simulation switched on?	[0 = NO Simulation mode 1 = simulation mode]
13 =	Current device status information	[0 = Device status=OFF 1 = Device status=ON]
14 =	Communication channel defined	[0 = NO communication channel 1 = Communication channel defined]
15 =	PLC components available ?	[0 = NO PLC component 1 = PLC component (DOS-Pcl) 2 = PLC component (WIN-Pcl)]
16 =	Monitor mode	[0 = NO monitor mode active 1 = Monitor mode active]

Example DSI1 Read the current device status information.

Assumption:
The following device addresses have been defined:

Device address 01 (VM-P-G2)

Device address 03 (VM-R-G2)

FI command		**01_BR_DSI1**
Line	**Column**	**Answer**
1	1	01
	2	0
	3	0
	4	00000000
	5	0
	6	00000000
	7	00000000
	8	0
	9	0
	10	0
	11	1
	12	0
	13	1
	14	1
	15	0
	16	0

2	1	03
	2	1
	3	0
	4	00000000
	5	0
	6	00000000
	7	00000000
	8	0
	9	0
	10	0
	11	1
	12	0
	13	1
	14	1
	15	0
	16	0

FI command

Read out device status information for a selected device.

BR_DSI2 **(Single Read)**

BC_DSI2 **(Cyclic Read)**

BB_DSI2 **(Break Cyclic Read)**

Response Structure

The following table shows the general structure of the response to the "DSI2" FI command.

Line 1	Column 1	...	Column 11

Value Range/Meaning of Columns

1 =	Device address	[00...63]
2 =	System error information	[0 = there is no system error 1 = there is a system error]
3 =	Mechanism error information	[0 = there is no mechanism error 1 = there is a mechanism error]
4 =	Machine key information	[4 byte in HEX coding]
5 =	Is machine key information valid?	[0 = not valid, 1=valid]
6 =	Machine status information	[4 byte in HEX coding]
7 =	Sercans information	[4 byte in HEX coding]
8 =	Is parameter download active?	[0 = parameter download not running 1 = parameter download running]
9 =	Is PLC download active?	[0 = PLC download not running 1 = PLC download running]
10 =	Is firmware download active?	[0 = PLC download not running 1 = PLC download running]
11 =	Offline/Online information	[0 = device connection interrupted 1 = device connection O.K.]
12 =	Device simulation switched on?	[0 = NO Simulation mode 1 = simulation mode]

13 = Current device status information
[0 = Device status=OFF
1 = Device status=ON]

14 = Communication channel defined
[0 = NO communication channel
1 = Communication channel defined]

15 = PLC components available ?
[0 = NO PLC component
1 = PLC component (DOS-Pcl)
2 = PLC component (WIN-Pcl)]

16 = Monitor mode
[0 = NO monitor mode active
1 = Monitor mode active]

Example DSI2 Read the current device status information for the selected device.

FI command		**00_BR_DSI2**
Line	**Column**	**Answer**
1	1	00
	2	0
	3	0
	4	00000000
	5	0
	6	00000000
	7	00000000
	8	0
	9	0
	10	0
	11	1
	12	0
	13	1
	14	1
	15	0
	16	0

7.7 Device Type and Accompanying Components: DTY

MVMX device group

Designation **DTY** **D**evice **TY**pe

Explanation The device type and the accompanying components of the selected device address are output.

FI command
BR_DTY1 **(Single Read)**
BC_DTY1 **(Cyclic Read)**

Response Structure The following table shows the general structure of the response to the "DTY1" FI command. A line with three columns is output for the device type, as well as the names of the first device component and the name of the second device component.

Line 1	Column 1	...	Column 3

Value Range/Meaning of Columns

1 =	Device type	(see Chapter 6.1 "Elements of the FI Command" and "Identifier")
2 =	Component type1	IND_DEV.INI entry: Component type1=
3 =	Component type 2	IND_DEV.INI entry: Component type2=

Example DTY1 Output the device type and the accompanying components of device address 00.

FI command	**00_BR_DTY1**		
Answer			
Line	**Column 1**	**Column 2**	**Column 3**
1	VMISP200-P-G2	MTS-P	NONE

7.8 Component Information for a System Error: ECI

MVMX device group

Designation **ECI** **E**rror **C**omponent **I**nformation

Explanation When a system error is present, this command is used to define which controller component is causing the error. Here, PLC components are differentiated from general controller components (e.g. CNC, Synax, MTA, …).

FI command **BR_ECI1** **(Single Read)**

Response Structure The response to the "ECI1" FI command consists of one line with two columns.

Line 1	Column 1	Column 2

Value Range/Meaning of Columns

1 =	PLC component information	[0 = There is NO system error at the PLC) 1 = There is a system error at the PLC]
2 =	General information on controller components	[0 = There is NO system error at the general control component [0 = There is a system error at the general control component; the following applies: 2 = CNC component 3 = SYNAX component 4 = VISUAL-MOTION component 5 = MTA component 6 = TRANS 200 component]

Example ECI1

There is a system error present in device 0 (VM-P) which is caused by the PLC component.

FI command		**00_BR_ECI1**
Line	**Column**	**Answer**
1	1	1
1	2	0

7.9 Device Data of the Module Configuration: MCD

MVMX device group

Designation

MCD **M**odule **C**onfiguration: **D**evice Information

Explanation

All device data for module configuration is read out from the "Moduldef.ini" file which is stored in the "[LW]:\Program Files\Indramat\MTGUI\CustomData\Resource" directory on standard installation. The device data is in the sections [DeviceAddrX], whereby "X" stands for the configured device addresses.

FI command

Read-out device data within the module configuration of the MVMX device group.

BR_MCD1	**(Single Read)**
BC_MCD1	**(Cyclic Read)**
BB_MCD1	**(Break Cyclic Read)**

Note: The "MCD1" FI command refers to all devices within the MVMX device group. Therefore, any valid device address can be indicated in the command line (see example MCD1).

Response Structure

The following table shows the general structure of the response to the "MCD1" FI command. The number of lines depends on the number of configured devices. Each line consists of four columns for the device address as well as PLC-FB names for providing setup diagnostics, warning messages and start requirements.

Value Range of the Columns

1 = Device address	[0...15]
2 = PLC-FB name for the setup diagnostics	[max. 9 ASCII characters]
3 = PLC-FB name for the warning messages	[max. 9 ASCII characters]
4 = PLC-FB name for the start requirements	[max. 9 ASCII characters]

Example MCD1

Read all device data of the module configuration

Assumption:
The following devices in the MVMX device group have been defined:

- Device address 01 (VM200-P-G2)
- Device address 03 (VM200-R-G2)

FI command	**03_BR_MCD1**			
Answer				
Line	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	01	PVSetup_1	PVWarn_1	PVStart_1
2	03	PVSetup_3	PVWarn_3	PVStart_3

Reference to Literature

See chapter entitled "Literature" [36].

7.10 Module Data of the Module Configuration: MCM

MVMX device group

Designation

MCM **M**odule **C**onfiguration: **M**odule Information

Explanation

All module data of a particular device is read out from the "Moduldef.ini"' file. This file is located in the directory "[LW]:\Program Files\Indramat\MTGUI\CustomData\Resource" and contains the data for all module configurations. The module data is located in sections [DeviceAddrX\ModulY], whereby "X" stands for the device addressed and "Y" for the configured module numbers.

FI command

Read module data from the module configuration with respect to a device from the MVMX device group.

BR_MCM1 **(Single Read)**

BC_MCM1 **(Cyclic Read)**

BB_MCM1 **(Break Cyclic Read)**

Response Structure

The following table shows the general structure of the response to the "MCM1" FI command. The number of lines depends on the number of configured modules of a device. Each line consists of four columns for the module number, module name and PLC-FB names for general module errors and module messages.

Line 1	**Column 1**	**...**	**Column 4**

Value Range of the Columns

1 = Module number	[0...99]
2 = Module name	[max. 28 ASCII characters]
3 = PLC-FB name for general module errors	[max. 9 ASCII characters]
4 = PLC-FB name for module messages	[max. 9 ASCII characters]

Example MCM1

Read the module data of device 03 from the module configuration:

Assumption:
The following modules have been defined:

- Module number 5
- Module number 7

FI command	**03_BR_MCM1**			
Answer				
Line	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	5	Module 5 - Milling	PVError_5	PVMsg_5
2	7	Module 7 - Drilling	PVError_7	PVMsg_7

Reference to Literature See chapter entitled "Literature" [36].

7.11 Setting the Device Status Information: SDS

MVMX device group

Designation **SDS** **S**et **D**evice **S**tatus

Explanation By this command, the device status information can be set; here, the configuration file IND_DEV.INI is adjusted as well.

Note: When this command is issued, the following system messages are generated as well: MSG_DEVICEOFF or MSG_DEVICE_ON !

FI command With this command, the device status information of **ALL** defined devices can be set.

BW_SDS1_(1)	**(Single Write)**
(1) = Device status information to be set	[0 = Device status information OFF 1 = Device status information ON]

Response Structure The following table shows the general structure of the response to the "SDS1" FI command.

Line 1	**Column 1**

Value Range/Meaning of Columns

1 = Status report [(P_ACK)]

Example: SDS1 Set device status information to OFF for **ALL** defined devices.

FI command		**00_BW_SDS1_0**
Line	**Column**	**Answer**
1	1	(P_ACK)

FI command With this command, the device status information for a selected device can be set.

BW_SDS2_(1)	**(Single Write)**
(1) = Device status information to be set	[0 = Device status information OFF 1 = Device status information ON]

Response Structure The following table shows the general structure of the response to the "SDS2" FI command.

Line 1	**Column 1**

Value Range/Meaning of Columns

1 = Status report [(P_ACK)]

Example: SDS2 Set device status information to OFF for the selected device 00.

FI command		**00_BW_SDS2_0**
Line	**Column**	**Answer**
1	1	(P_ACK)

7.12 Read Device Date and Time: SDT

MVMX device group

Designation **SDT** Acces**S** **D**ate **T**ime

FI command This FI command reads or writes date and time of the selected device.

BR_SDT **(Single Read)**

Response Structure The following table shows the general structure of the response to the FI command "BR_SDT1". A line of 1 column is output.

Line 1	Column 1	...	Column 3

Value Range/Meaning of Columns

1 = Coded LONG value	Date and time coded in 32 bit MS Windows time format (as a HEX figure)
2 = Date	[Day.month.year]
3 = Time	[hour:minute:second]

Example SDT Read the device time and the device date of device 0.

FI command		**00_BR_SDT**
Line	**Column**	**Answer**
1	1	0x2CB52F78
1	2	21.05.02
1	3	05:59:48

7.13 Setting the FI Exclusive Mode: SEM

MVMX device group

Designation **SEM** **S**et FI **E**xclusive **M**ode

Explanation This command is used to activate FI Exclusive mode for the selected device address.

FI Exclusive mode: In this mode, **ALL** the processes logged in at the FI – **with the exception** of the process issuing the SEM command – are blocked from data communication with this device address. This mode is used for example for data communication which must NOT be interrupted by other data requests. However, it is **imperative** that this FI Exclusive mode is deleted once more through the DEM command.

FI command **BW_SEM1** **(Single Write)**

Response Structure The following table shows the general structure of the response to the FI command "BW_SEM1". A line of 1 column is output.

Line 1	Column 1

Value Range/Meaning of Columns 1 = Status message (P_ACK) (P_ACK)

Example SEM1 Activate FI Exclusive mode for device address 0.

FI command		**00_BW_SEM1**
Line	**Column**	**Answer**
1	1	(P_ACK)

7.14 Software Installation Data: SID

MVMX device group

Designation **SID** **S**oftware **I**nstallation **D**ata

Explanation Information is returned regarding installation. This information includes installation paths, the software versions used, DLL mode, context information, plus service pack and release information.

FI command The installation data and/or software version data is read in.

BR_SID1 **(Single Read)**

Response Structure One line with 16 columns is output for the returned values.

Line 1	Column 1	...	Column 16

Meaning of the Columns

1 = Basic directory	[EXE files of the BOF]
2 = FI installation directory	[FI directory]
3 = Data directory	[in accordance with BOF]
4 = GBO version	[from INDRAMAT.ini]
5 = IF-DLL mode	[from INDRAMAT.ini]
6 = IF version	[from INDRAMAT.ini from DLL mode 400]
7 = Service pack info	[from INDRAMAT.ini from DLL mode 420]
8 = Release info	[from INDRAMAT.ini from DLL mode 420]
9 = IF-Build-Info	[in accordance with Build process]
10 = Current context name	[in accordance with the installation]
11 = Physical installation path	[in accordance with the installation]
12 = Complete IF version indication string	
13 = WinPCL build number	[in accordance with WinPCL]
14 = Version number of the PLC compiler	[in accordance with WinPCL]
15 = Version number of the PLC linker	[in accordance with WinPCL]
16 = Version number of the PLC data basis	[in accordance with WinPCL]
17 = Platform version	

Example SID1 Return information on the current installation.

FI command		00_BR_SID1
Line	**Column**	**Answer**
1	1	--
	2	D:\Program Files\Indramat\MTGUI\Bin
	3	--
	4	005-22Vxx
	5	07.20
	6	07V00

	7	--
	8	--
	9	Build 3124 Mar 6 2003 08:53:55
	10	MTGUI_0-23T01 B3327
	11	D:\Program Files\Indramat\MTGUI\
	12	FI: 07V00 DLL-Mode: 07.20 Build 3124 Mar 6 2003 08:53:55
	13	347.15.4.11
	14	771
	15	515
	16	78
	17	Platform: 02V01 Build: 3214

Note: Refer to FI command "PHD" of the MPCX for working with absolute paths.

7.15 Writing Visual Motion Data: VMD

MVMX device group

Designation **VMD** Visual **M**otion **D**ata

Explanation This command is used to write Visual Motion data. To identify the Visual Motion data to be written, the ASCII communication description is applicable.

FI command **BW_VMD1_(1)** **(Single Read)**

(1) = ASCII request string according to the ASCII communication description and the value to be written

Response Structure The following table shows the general structure of the response to the FI command "BW_VMD1". A line of 1 column is output.

Line 1	**Column 1**

Value Range/Meaning of Columns 1 = Status message (P_ACK) (P_ACK)

Example VMD1 Set the KV parameter for drive 1. The ASCII request string and the value to be written can also be written in inverted commas (00_BW_VMD1_"DP 1.104 1.00").

FI command		**00_BW_VMD1_DP 1.104 1.00**
Line	**Column**	**Answer**
1	1	(P_ACK)

8 FI Commands - MWMX Device Group (VMISP200)

The FI Commands described in this chapter are valid for the MWMX. In this device group, the following types are listed as well as possible device addresses:

Group	Accompanying Types	Address
MWMX	VMISP200-P-G2, VMISP200-R-G2	[00...63]

Note: The Visual Motion component has been realized under SCP (Scalable Communication Platform).

Please note that the device address must be set before the respective FI command, e.g. 00_BR_ASM1 (refer also here to the chapter 2.1 "Elements of the FI Command").

8.1 Active Diagnosis Window ADW

MWMX device group

Designation **ADW** Active **D**iagnosis **W**indow

Explanation Indicates the window types for which data is required.

For improved performance and with some diagnosis windows, not all the data is called up each time a diagnosis is performed, but only when the data is actually required.

Through this FI command, the diagnosis server can be informed that the data of the respective window type is required.

This command has to be issued at least every 3 seconds while the data is required. If this command is not issued any more, calling-up of all data will stop.

FI command Indicates the window types for which data is required.

BW_ADW1_(1){_(2)}	**(Single Write)**
(1) = Type of diagnosis window	[1 = CNC error, 2 = sequence errors, 3 = general errors, 4 = messages, 10 = start requirements, 11 = warnings, 12 = setup diagnosis]
(2) = Module number	[1...99] ! only for window type 1 -4!

Example ADW1 Call up data for CNC error in controller 0 module 1.

FI command	**00_BW_ADW1_1_1**

8.2 Active System Error Messages: ASM

MWMX device groups

Designation **ASM** Active **S**ystem **M**essages

Explanation

The active device information is output (system errors, device statuses) that affect the functioning of the entire electrical device. Depending on the FI command, the device address, device name, message number, type of message, short text and additional text are all output.

FI command

Output the currently pending device information (system errors, device statuses) of the selected device from the MWMX device group.

BR_ASM2	**(Single Read)**
BC_ASM2	**(Cyclic Read)**
BB_ASM2	**(Break Cyclic Read)**

Response Structure

The following table shows the general structure of the response to the FI command "ASM2". The answer consists of a line of 7 columns for the device address, device name, message number, message status, short text and indication of whether there is an additional text for this device information (system errors, device statuses).

Line 1...n	Column 1	...	Column 7

Value Range/Meaning of Columns

1 =	device address	[00...15]
2 =	Device name	[max. 32 ASCII characters]
3 =	Message number	[0...150]
4 =	Type of message	[F = fault/error, D = diagnosis]
5 =	Short text	[max. 54 ASCII characters]
6 =	Reference text	[x= exists, -- = does not exist]
7 =	2 bytes of additional information for the message number	is required to resolve the information "@" (see ASM5)

Example ASM2

Read the current device information (system errors, device statuses) of device address 01.

Assumption:
The following three devices are defined:

- Device address 01
- Device address 07 and
- Device address 10

FI command		**01_BR_ASM2**
Line	**Column**	**Answer**
1	1	01
	2	Drill center
	3	71
	4	F
	5	PLC battery voltage too low.
	6	X
	7	0

FI command

Output the additional text for the currently pending device information (system errors, device statuses) related to the device and the message number.

BR_ASM5_(1)_(2) **(Single Read)**

(1) = Message number [0...150]

(2) = 2 bytes of additional information for the message number

Response Structure

The following table shows the general structure of the response to the FI command "ASM5". The answer consists of a line with 5 columns for the device address, device name, message number and reference text.

Line 1...n	Column 1	...	Column 5

Value Range/Meaning of Columns

1 =	device address	[00...15]
2 =	Device name	[max. 32 ASCII characters]
3 =	Message number	[0...150]
4 =	Type of message	[F = fault/error, D = diagnosis]
6 =	Reference text	[max. 14 lines with a max. 78 characters/line]

Example ASM5

Read the reference text relating to the system error with message number 74 of device address 01.

FI command		**01_BR_ASM5_74_0**
Line	**Column**	**Answer**
1	1	01
	2	Drill center
	3	74
	4	F
	5	Replace the SLM module on the PLC card (slot 3).

Reference to Literature

See chapter entitled "Literature" [13].

8.3 Create MI Import Data: CMD

MWMX device group

Designation

CMD Create **M**I Import **D**ata

Explanation

Creation of the data the Message Integrator requires for data import.

FI command

Creation of the data of all ProVi messages.

BR_CMD1 **(Single Read)**

Response Structure

The command does not return any answer. If no error is signaled, the respective files have been generated.

FI command

Creation of the data of a certain ProVi message type.

BR_CMD2_(1){_(2)} **(Single Read)**

(1) = Message type [1 = error, 2 = messages, 10 = warnings, 11 = start requirements, 12 = setup diagnosis]

(2) = Module number [1...99] ! only for message type 1 -2!

Response Structure

The command does not return any answer. If no error is signaled, the respective files have been generated.

FI command Creation of the data of all step chain messages.

BR_CMD3 **(Single Read)**

Response Structure The command does not return any answer. If no error is signaled, the respective files have been generated.

FI command Creation of the data of a certain step chain register.

BR_CMD4!(1) **(Single Read)**

(1) = Register name [ASCII characters]

Note: The separator "!" is used in this command.

Response Structure The command does not return any answer. If no error is signaled, the respective files have been generated.

8.4 Trigger Control Reset: CRT

MWMX device groups

Designation **CRT** **Control-Reset**

Explanation The control reset allows the selected device to be reset after a system error. If there is no system error at the selected device then the job is ignored.

CAUTION

Carrying out a reset completely re-initializes the device.
During initialization, communication is temporarily interrupted (inherent to design).

FI command **CW_CRT** **(Single Write)**

Value to be written Trigger reset 0

Note: The value to be written is passed to the "acValue" parameter in the "DataTransfer" routine.

Response Structure The return value of the "DataTransfer" routine is [0] if the write procedure has been successfully completed. In the event of an error, more information in the form of a general error result line can be requested by the routine "ReadGroupItem" (refer here to chapter 8, "Error Codes" and "General Error Result Line").

Example CRT Trigger a control reset on the selected device.

<table>
<tr><th colspan="2">FI command</th><th>Value to be written: 0
00_CW_CRT</th></tr>
<tr><th>Line</th><th>Column</th><th>Answer</th></tr>
<tr><td>1</td><td>1</td><td>(P_ACK)</td></tr>
</table>

Reference to Literature See chapter entitled "Literature" [26].

8.5 Read Device Component Information: DCI

MWMX device group

Designation **DCI** **D**evice **C**omponent **I**nformation

Explanation The current device component information is read out of the device. From the device component information, the user is provided with information on the components the addressed device is equipped with, and the firmware each component contains. The command will not file if no access to firmware is possible (e.g. while the device is in monitor mode). Instead, the failed access is reported through the firmware access status.

FI command Read the device component information.

BR_DCI1 **(Single Read)**

Response Structure The following table shows the general structure of the response to the FI command "BR_DCI1". For each device component available in the device, one line is returned. Each line consists of 11 columns.

FI command		**00_BR_DCI1**
Line	**Column**	**Answer**
1	1	PCB type
	2	Configured component type
	3	Detected component type
	4	Firmware access status; i.e. has an error occurred accessing the firmware, Yes/No? Valid range of values [YES/NO] In case of an error, the error cause can be defined from one the two following columns.
	5	Error class on accessing firmware identification: (see Error Class Definition under General Error Result Line)
	6	Error code on accessing firmware identification: (see Error Code Definition under Error Codes)
	7	Firmware identification
	8	Firmware version
	9	Firmware release
	10	Is the component address in column 11 a sub-address, Yes/No? Valid range of values [YES/NO]
	11	Component address
2	1	PCB type
	..	..
	11	Component address

Example DCI1 At device address 00, read out the current device component information.

FI command		**00_BR_DCI1**
Line	**Column**	**Answer**
1	1	CPU
	2	MTC-P
	3	MTC-P
	4	NO
	5	0
	6	0
	7	CPU06/0006-23V10
	8	23
	9	10
	10	NO
	11	1
2	1	PLC
	2	MTS-P01.2
	3	MTS-P01.2
	4	NO
	5	0
	6	0
	7	PLC06S-M05-06V05
	8	06
	9	05
	10	NO
	11	3
3	1	APR
	2	--
	3	APR-P
	4	NO
	5	0
	6	0
	7	APR06/0003-23T06
	8	23
	9	06
	10	NO
	11	4

Example DCI1 while booting is blocked

While booting is blocked (i.e. while the device is in monitor mode), read out the current device component information at device address 00.

FI command		00_BR_DCI1
Line	**Column**	**Answer**
1	1	CPU
	2	MTC-P
	3	MTC-P
	4	YES
	5	1
	6	2082
	7	--
	8	--
	9	--
	10	NO
	11	1
2	1	PLC
	2	MTS-P01.2
	3	MTS-P01.2
	4	YES
	5	1
	6	2082
	7	--
	8	--
	9	--
	10	NO
	11	3
3	1	APR
	2	--
	3	APR-P
	4	YES
	5	1
	6	2082
	7	--
	8	--
	9	--
	10	NO
	11	4

8.6 Setting the Communication Timeout Time DCT

MWMX device groups

Designation **DCT** **D**evice **C**ommunication **T**imeout

Explanation By means of this command, the timeout time for the selected device is set dynamically (timeout time in ms).

FI command **BW_DCT1_(1)** **(Single Write)**

(1) = requested timeout time in ms

Response Structure The response to the "DCT1" FI command consists of one line with one column.

Line 1	Column 1

Value Range/Meaning of Columns

1 = Status message (P_ACK) (P_ACK)

Example DCT1 For the device 00, the timeout time is set 1500 ms.

FI command		**00_BW_DCT1_1500**
Line	**Column**	**Answer**
1	1	(P_ACK)

FI command With this command, the timeout time for the selected device can be reset to default value.

BW_DCT2 **(Single Write)**

Response Structure The response to the "DCT2" FI command consists of one line with one column.

Line 1	Column 1

Value Range/Meaning of Columns

1 = Status message (P_ACK) (P_ACK)

Example DCP2 For the device 00, the timeout time is reset to the default value.

FI command		**00_BW_DCT2**
Line	**Column**	**Answer**
1	1	(P_ACK)

8.7 Deleting the FI Exclusive Mode: DEM

MWMX device group

Designation **DEM** **D**elete FI **E**xclusive **M**ode

Explanation This command is used to deactivate FI Exclusive mode for the selected device address.

FI Exclusive mode: In this mode, **ALL** the processes logged in at the FI – **with the exception** of the process issuing the SEM command – are blocked from data communication with this device address. This mode is used for example for data communication which must NOT be interrupted by other data requests. However, it is **imperative** that this FI Exclusive mode is deleted once more through the DEM command.

FI command **BW_DEM1** **(Single Write)**

Response Structure The following table shows the general structure of the response to the FI command "BW_DEM1". A line of 1 column is output.

Line 1	Column 1

Value Range/Meaning of Columns 1 = Status message (P_ACK) (P_ACK)

Example DEM1 Deactivate the FI Exclusive mode for device address 0. The FI Exclusive mode for device address 0 has previously been activated by the SEM command.

FI command		**00_BW_DEM1**
Line	**Column**	**Answer**
1	1	(P_ACK)

8.8 Static Device Information: DIF

MWMX device group

Designation **DIF** **D**evice **I**n**F**ormation

Explanation Static device information and network information is read according to the "IND_DEV.INI" and "FAR_DEV.INI" files.

FI command Reading of the static device information and network information of a selected device.

BR_DIF1 **(Single Read)**

BC_DIF1 **(Cyclic Read)**

BB_DIF1 **(Break Cyclic Read)**

Response Structure The following table shows the general structure of the response to the "DIF1" FI command. The response consists of one line with 24 columns.

Line 1	Column 1	...	Column 24

Value Range/Meaning of Columns

1 =	Local/far device address	[00..63]
2 =	Device name	IND_DEV.INI entry: DeviceName=
3 =	Device type	IND_DEV.INI entry: DeviceType=
4 =	PLC support	IND_DEV.INI entry: PLC=
5 =	Device status	IND_DEV.INI entry: DeviceStatus=
6 =	Assignment of a simulation pair	IND_DEV.INI entry: DeviceAssign=
7 =	Device mode	IND_DEV.INI entry: MtvncMode=
8 =	Communication channel	IND_DEV.INI entry: [CommAddrX]
9 =	Description of the communication channel	IND_DEV.INI entry: CommStr=
10 =	Timeout value	IND_DEV.INI entry: Timeout=
11 =	Device group	(see Chapter 6.1 "Identifier")
12 =	PLC component type	IND_DEV.INI entry: Component type1=
13 =	CNC component type	IND_DEV.INI entry: Component type2=
14 =	Device log	IND_DEV.INI entry: DeviceProtocol=
15 =	Device simulation	IND_DEV.INI entry: DeviceSimulation=
16 =	Not yet assigned	[--]
17 =	Not yet assigned	[--]
18 =	Not yet assigned	[--]
19 =	Not yet assigned	[--]
20 =	Network ON/OFF	[ON = Network active OFF = No network active]
21 =	Network name	Max. 28 ASCII characters
22 =	PC number	[00..99,XX]
23 =	PC name	Max. 255 ASCII characters
24 =	Local device address	[00..63]

Example DIF1

Read the static device information and network information of device 1 while the network is active.

FI command		**01_BR_DIF1**
Line	**Column**	**Answer**
1	1	01
	2	Station right side
	3	VMISP200-R
	4	YES
	5	ON
	6	NO
	7	OFF
	8	4
	9	V24,COM2,19200,EVEN,RS232,TCOFF
	10	3500
	11	MWMX
	12	MTS-R
	13	NONE
	14	SIS
	15	OFF
	16	--
	17	--
	18	--
	19	--
	20	ON
	21	PC network 1
	22	29
	23	BTV20-STATION-LEFT
	24	01

8.9 Long ID of PLC Data Block: DIS

MWMX device groups

Designation **DIS** Data **I**dentification **S**tring

Explanation Reads the long ID (directory entries) of the PLC program. Included in the directory entries are the number of the entry in the directory, the name, length and date and time of creation and/or details of the last time the respective data record was changed.

FI command

BR_DIS2 **(Single Read)**

BC_DIS2 **(Cyclic Read)**

BB_DIS2 **(Break Cyclic Read)**

Response Structure The following table shows the general structure of the response to the "DIS2" FI command. The response consists of a line with six columns.

Line 1	Column 1	...	Column 6

Value Range/Meaning of Columns

1 =	Number in PLC directory	[01...99]
2 =	Name of the PLC program	[max. 8 ASCII characters]
3 =	Length of the PLC program	[byte]
4 =	Date of creation/last change to PLC program	[DD.MM.YY]
5 =	Time of creation/last change to the PLC program	[HH:MM:SS]
6 =	Date of creation/last change to PLC program	[DD.MM.YYYY]

Note: If there is no valid NC package in the selected NC memory then all columns contain [--] .

Example DIS2 Read the directory entries of the PLC program at address 00.
Assumption:
There is a valid PLC program in the selected device.

FI command		**00_BR_DIS2**
Line	**Column**	**Answer**
1	1	01
	2	KEY1
	3	20018
	4	10.05.99
	5	12:42:00
	6	10.05.1999

Reference to Literature See chapter entitled "Literature" [30].

8.10 Reading the Device Status Information: DSI

MWMX device groups

Designation **DSI** Device **S**tatus **I**nformation

Explanation This allows the most important device status information to be read out. The following information is returned:

Type of information	Status	Statement
System error information		Yes/No
Mechanism error information		Yes/No
Machine key information		4 Byte HEX
Machine key information	valid	Yes/No
Machine status information		4 Byte HEX
Sercans information		4 Byte HEX
Parameter download	running	Yes/No
PLC download	running	Yes/No
Firmware download	running	Yes/No
Offline/Online information		
Device simulation	switched on	Yes/No
Device status information		ON/OFF
Communication channel defined		Yes/No
PLC components available		Yes/No
Monitor mode	active	Yes/No

FI command Read out device status information for ALL defined devices.

BR_DSI1 **(Single Read)**
BC_DSI1 **(Cyclic Read)**
BB_DSI1 **(Break Cyclic Read)**

Note: The "DSI1" FI command refers to all devices within this device group. Therefore, any valid device address can be indicated in the command line (see example DSI1). The FI device polling mechanism **MUST** be switched on (see system configurator)!

Response Structure The following table shows the general structure of the response to the "DSI1" FI command.

Line 1...n	Column 1	...	Column 11

Value Range/Meaning of Columns

1 =	device address	[00...63]
2 =	System error information	[0 = there is no system error 1 = there is a system error]
3 =	Mechanism error information	[0 = there is no mechanism error 1 = there is a mechanism error
4 =	Machine key information	[4 byte in HEX coding]
5 =	Machine key information valid?	[0 = not valid, 1=valid]
6 =	Machine status information	[4 byte in HEX coding]
7 =	Sercans information	[4 byte in HEX coding]

8 =	Is parameter download active?	[0 = parameter download not running 1 = parameter download running]
9 =	Is PLC download active?	[0 = PLC download not running 1 = PLC download running]
10 =	Is firmware download active?	[0 = PLC download not running 1 = PLC download running]
11 =	Offline/Online information	[0 = device connection interrupted 1 = device connection O.K.]
12 =	Device simulation switched on?	[0 = NO Simulation mode 1 = simulation mode]
13 =	Current device status information	[0 = Device-Status=OFF 1 = Device-Status=ON]
14 =	Communication channel defined?	[0 = NO communication channel 1 = Communication channel defined]
15 =	PLC components available ?	[0 = NO PLC component 1 = PLC component (DOS-Pcl) 2 = PLC component (WIN-Pcl)]
16 =	Monitor mode	[0 = NO monitor mode active 1 = Monitor mode active]

Example DSI1 Read the current device status information.

Assumption:
The following devices addresses are defined:

- Device address 01 (VMISP200-P-G2)
- Device address 03 (VMISP200-R-G2)

FI command		**01_BR_DSI1**
Line	**Column**	**Answer**
1	1	01
	2	0
	3	0
	4	00000000
	5	0
	6	00000000
	7	00000000
	8	0
	9	0
	10	0
	11	1
	12	0
	13	1
	14	1
	15	2
	16	0

2	1	03
	2	1
	3	0
	4	00000000
	5	0
	6	00000000
	7	00000000
	8	0
	9	0
	10	0
	11	1
	12	0
	13	1
	14	1
	15	2
	16	0

FI command

Read out device status information for a selected device.

BR_DSI2	**(Single Read)**
BC_DSI2	**(Cyclic Read)**
BB_DSI2	**(Break Cyclic Read)**

Response Structure

The following table shows the general structure of the response to the "DSI2" FI command.

Line 1...n	Column 1	...	Column 11

Value Range/Meaning of Columns

1 =	device address	[00...63]
2 =	System error information	[0 = there is no system error 1 = there is a system error]
3 =	Mechanism error information	[0 = there is no mechanism error 1 = there is a mechanism error]
4 =	Machine key information	[4 byte in HEX coding]
5 =	Is machine key information valid?	[0 = not valid, 1=valid]
6 =	Machine status information	[4 byte in HEX coding]
7 =	Sercans information	[4 byte in HEX coding]
8 =	Is parameter download active?	[0 = parameter download not running 1 = parameter download running]
9 =	Is PLC download active?	[0 = PLC download not running 1 = PLC download running]
10 =	Is firmware download active?	[0 = PLC download not running 1 = PLC download running]
11 =	Offline/Online information	[0 = device connection interrupted 1 = device connection O.K.]
12 =	Device simulation switched on?	[0 = NO simulation mode 1 = Simulation mode]
13 =	Current device status	[0 = Device status=OFF

Rexroth
Bosch Group

	information	1 = Device status=ON]
14 =	Communication channel defined?	[0 = NO communication channel 1 = Communication channel defined]
15 =	PLC components available ?	[0 = NO PLC component 1 = PLC component (DOS-Pcl) 2 = PLC component (WIN-Pcl)]

Example DSI2 Read the current device status information for the selected device.

FI command		**00_BR_DSI2**
Line	**Column**	**Answer**
1	1	00
	2	0
	3	0
	4	00000000
	5	0
	6	00000000
	7	00000000
	8	0
	9	0
	10	0
	11	1
	12	0
	13	1
	14	1
	15	2
	16	0

8.11 Device Type and Accompanying Components: DTY

MWMX device groups

Designation **DTY** **D**evice **TY**pe

Explanation The device type and the accompanying components of the selected device address are output.

FI command

BR_DTY1 **(Single Read)**

BC_DTY1 **(Cyclic Read)**

Response Structure The following table shows the general structure of the response to the "DTY1" FI command. A line with three columns is output for the device type, as well as the names of the first device component and the name of the second device component.

Line 1	Column 1	...	Column 3

Value Range/Meaning of Columns

1 =	Device Type	(see Chapter 6.1 "Elements of the FI Command" and "Identifier")
2 =	Component type1	IND_DEV.INI-Entry: Componenttype1=
3 =	Component type 2	IND_DEV.INI-Entry: Componenttype2=

Example DTY1 Output the device type and the accompanying components of device address 00.

FI command	**00_BR_DTY1**		
Answer			
Line	**Column 1**	**Column 2**	**Column 3**
1	ISP200-P	MTS-P	NONE

8.12 Diagnosis Window Data: DWD

MWMX device groups

Designation **DWD** **D**iagnosis **W**indow **D**ata

Explanation Diagnostic messages are output. The data is edited in such a way that they can be output directly in the diagnosis overview, i.e., where applicable, different types of diagnosis, such as ProVi and a process report, are returned simultaneously.

FI command Output all diagnostic messages.

For reasons of optimization, not all data is called up. Accordingly, the Diagnosis server must be informed that the data is required (see ADW).

BR_DWD1_(1){_(2)}	**(Single Read)**
BC_DWD1_(1){_(2)}	**(Cyclic Read)**
(1) = Type of diagnosis window	[1 = CNC error, 2 = sequence errors, 3 = general errors, 4 = messages, 10 = start preconditions, 11 = warnings, 12 = setup diagnosis]
(2) = Module number	[1...99] ! only for window type 1 -4 !

Output first diagnostic messages.

BR_DWD2_(1){_(2)}	**(Single Read)**
BC_DWD2_(1){_(2)}	**(Cyclic Read)**
(1) = Type of diagnosis window	[1 = CNC error, 2 = sequence errors, 3 = general errors, 4 = messages, 10 = start preconditions, 11 = warnings, 12 = setup diagnosis]
(2) = Module number	[1...99] ! only for window type 1 -4 !

Response Structure

The following table shows the general structure of the "DWD1" and "DWD2" FI commands. The number of lines depends on the number of messages pending. Different columns are valid according to the type of diagnosis.

If there are no messages, the number of lines is 0.

Line 1...n	Column 1	...	Column 12

Meaning of the Columns

1 =	Message text	[ASCII characters]
2 =	Time stamp day	[mm.dd.yyyy]
3 =	Time stamp hour	[hh:mm:ss]
4 =	Reference text available	[YES, NO]
5 =	Type of diagnosis	[1 = ProVi, 2 = SFC, 3 = MTC-NC, 4 = MTA-NC]
6 =	Message number	[ASCII characters]
7 =	Message ID	[ASCII characters] (DWORD, decimal) (ProVi)
8 =	Mechanism number	[0..31] (MTC-NC) [0] (MTA-NC)
9 =	2 byte additional information	[ASCII characters] (MTC NC)
10 =	Message group	[1...9999] (MTA-NC)
11 =	SFC entity name	[ASCII characters]
12 =	NC note	[ASCII characters] (MTC NC)
13 =	Analysis of criteria available	[YES, NO] (ProVi, SFC)
14 =	Message HTML file	[ASCII characters] (ProVi, MTC-NC)

Example DWD1 All diagnostic messages from module 3 in control unit 0.

There are two messages.

FI command		00_BR_DWD1_4_3
Line	**Column**	**Answer**
1	1	Guard not closed
	2	01.27.2000
	3	14:56:32
	4	YES
	5	1
	6	34
	7	43923028
	8	
	9	
	10	
	11	
	12	
	13	YES
	14	
2	1	Station waiting until tool-change command has ended.
	2	01.27.2000
	3	15:03:10
	4	YES
	5	3
	6	79
	7	
	8	1
	9	0
	10	
	11	
	12	
	13	NO
	14	

Example DWD2 First diagnostic message from module 3 in control unit 0.

There are two messages.

FI command		**00_BR_DWD2_4_3**
Line	**Column**	**Answer**
1	1	Guard not closed
	2	01.27.2000
	3	14:56:32
	4	YES
	5	1
	6	34
	7	43923028
	8	
	9	
	10	
	11	
	12	
	13	YES
	14	

Reference to Literature See chapter entitled "Literature" [13].

8.13 Component Information for a System Error: ECI

MWMX device group

Designation **ECI** **E**rror **C**omponent **I**nformation

Explanation When a system error is present, this command is used to define which controller component is causing the error. Here, PLC components are differentiated from general controller components (e.g. CNC, Synax, MTA, …).

FI command **BR_ECI1** **(Single Read)**

Response Structure The response to the "ECI1" FI command consists of one line with two columns.

Line 1	**Column 1**	**Column 2**

Value Range/Meaning of Columns

1 =	PLC component information	[0 = There is NO system error at the PLC) 1 = There is a system error at the PLC]
2 =	General information on controller components	[0 = There is NO system error at the general control component [0 = There is a system error at the general control component; the following applies: 2 = CNC component 3 = SYNAX component 4 = VISUAL-MOTION component 5 = MTA component 6 = TRANS 200 component]

Example ECI1

There is a system error present in device 0 (VMISP200-P-G2)) which is caused by the PLC component.

FI command		**00_BR_ECI1**
Line	**Column**	**Answer**
1	1	1
1	2	0

8.14 Existing Errors: EDE

MWMX device groups

Designation

EDE **E**xisting **D**iagnosis **E**rror

Explanation

Whether or not errors exist in a control unit or in a module is queried. These can be step chain errors, NC errors, MTA 200 errors or ProVi errors.

FI command

Query whether there are errors in this control unit.

BR_EDE1 **(Single Read)**

BC_EDE1 **(Cyclic Read)**

Response Structure

The following table shows the general structure of the "EDE1" FI command.

Line 1	**Column 1**

Meaning of the Columns

1 = Error exists [YES, NO]

Example EDE1

Do errors exist in control unit 0?

FI command		**00_BR_EDE1**
Line	**Column**	**Answer**
1	1	YES

FI command

Query whether or not errors exist in a specific module.

BR_EDE2_(1) **(Single Read)**

BC_EDE2_(1) **(Cyclic Read)**

(1) = Module number [1...99]

Response Structure

The following table shows the general structure of the "EDE2" FI command.

Line 1	**Column 1**

Meaning of the Columns

1 = Error exists [YES, NO]

Example EDE2

Do errors exist in module 1 on control unit 0?

FI command		**00_BR_EDE2_2**
Line	**Column**	**Answer**
1	1	NO

8.15 Existing Diagnosis Window: EDW

MWMX device groups

Designation **EDW** **E**xisting **D**iagnosis **W**indow

Explanation Which types of diagnosis window exist is queried.

FI command Output all types of diagnosis window.

BR_EDW1 **(Single Read)**

Response Structure The following table shows the general structure of the "EDW1" FI command. The number of lines depends on the number of types of window existing.

Line 0...n	Column 1	Column 2

Meaning of the Columns

1 = Type of diagnosis window	[1 = CNC error, 2 = sequence errors, 3 = general errors, 4 = messages, 10 = start requirements, 11 = warnings, 12 = setup diagnosis]
2 = Module number	[ASCII characters] 0 = Diagnosis window type does not belong to any module

Example EDW1 All types of diagnosis window in control unit 0.

There are three diagnosis windows.

FI command		**00_BR_EDW1**
Line	**Column**	**Answer**
1	1	10
	2	0
2	1	1
	2	3
3	1	2
	2	3

FI command Output all diagnosis window types for a module.

BR_EDW2_(1) **(Single Read)**

(1) = Module number [1...99]

Response Structure The following table shows the general structure of the "EDW2" FI command. The number of lines depends on the number of types of window existing.

Line 0...n	Column 1	Column 2

Meaning of the Columns

1 = Type of diagnosis window	[1 = CNC error, 2 = sequence errors, 3 = general errors, 4 = messages]
2 = Module number	[ASCII characters] 0 = Diagnosis window type does not belong to any module

Example EDW2 All types of diagnosis window in Module 3, Control unit 0.

There are two diagnosis windows.

FI command		**00_BR_EDW2_3**
Line	**Column**	**Answer**
1	1	1
	2	3
2	1	2
	2	3

FI command Query a specific type of diagnosis window.

BR_EDW3_(1){_(2)} **(Single Read)**

(1) = Type of diagnosis window	[1 = CNC error, 2 = sequence errors, 3 = general errors, 4 = messages, 10 = start requirements, 11 = warnings, 12 = setup diagnosis]
(2) = Module number	[1...99] ! only for window type 1 -4 !

Response Structure The following table shows the general structure of the "EDW3" FI command.

Line 1	**Column 1**

Meaning of the Columns

1 = Type of diagnosis window exists	[YES, NO]

Example EDW3 Query whether or not an NC error window exists in module 3, control unit 0.

FI command		**00_BR_EDW3_1_3**
Line	**Column**	**Answer**
1	1	YES

Reference to Literature See chapter literature [13].

8.16 Existing PLC Diagnoses: EPD

MWMX device groups

Designation **EPD** **E**xisting **P**LC **D**iagnosis

Explanation Which PLC diagnostic types exist is queried. Depending on the FI command, specific types are queried or else the diagnostic types for a device or a module are output together.

FI command Query which PLC diagnostic types are available on a control unit.

BR_EPD1 **(Single Read)**

Response Structure The following table shows the general structure of the "EPD1" FI command.

Line 1	**Column 1-3**

Meaning of the Columns

1 = Start requirement exists	[YES, NO]
2 = Warning exists	[YES, NO]
3 = Setup diagnosis exists	[YES, NO]

Example EPD1 Query PLC diagnostic types in control unit 0.

FI command		00_BR_EPD1
Line	**Column**	**Answer**
1	1	YES
	2	NO
	3	YES

FI command

Query which PLC diagnostic types are available in a module.

BR_EPD2_(1) **(Single Read)**

(1) = Module number [1...99]

Response Structure

The following table shows the general structure of the "EPD2" FI command.

Line 1	Column 1-3

Meaning of the Columns

1 = Messages exist	[YES, NO]
2 = Errors exist	[YES, NO]
3 = Step chains exist	[YES, NO]

Example EPD2

Query the PLC diagnostic types in Module 2 on Control unit 0.

FI command		00_BR_EPD2_2
Line	**Column**	**Answer**
1	1	NO
	2	YES
	3	YES

FI command

Query a specific PLC diagnostic type.

BR_EPD3_(1){_(2)} **(Single Read)**

(1) = Message type [1 = error, 2 = messages, 3 = SFC, 10 = warnings, 11 = start requirements, 12 = setup diagnosis]

(2) = Module number [1...99] ! only for message type 1 -3!

Response Structure

The following table shows the general structure of the "EPD3" FI command.

Line 1	Column 1

Meaning of the Columns

1 = Diagnosis type exists [YES, NO]

Example EPD3

Are there any messages in module 4 in control unit 0?

FI command		00_BR_EPD3_2_4
Line	**Column**	**Answer**
1	1	YES

8.17 Existing ProVi Types: EPT

MWMX device groups

Designation **EPT** **E**xisting **P**roVi **T**ypes

Explanation Which ProVi types are programmed in the current PLC program is queried. The data is returned in a suitable form for the message texts of the small control panels. There is no need to define modules in Moduldef.ini.

FI command Output all ProVi types.

BR_EPT1 **(Single Read)**

Response Structure The following table shows the general structure of the "EPT1" FI command. The number of lines depends on the number of ProVi types existing.

Line 0...n	Column 1	Column 2

Meaning of the Columns

1 = Type [11 = error, 12 = messages,
20 = start requirements,
21 = warnings, 22 = setup diagnosis]

2 = Index [ASCII characters]

Example EPT1 All ProVi types in control unit 0.

There are three diagnosis windows.

FI command		**00_BR_EPT1**
Line	**Column**	**Answer**
1	1	20
	2	0
2	1	11
	2	3
3	1	12
	1	3

8.18 Error Status: EST

MWMX device groups

Designation **EST** **E**rror **ST**ate

Explanation Queries the error state of a variable.

FI command Query the frozen error state of a variable.

BR_EST1!(1)!(2) **(Single Read)**

BC_EST1!(1)!(2) **(Cyclic Read)**

(1) = Error ID [ASCII characters] (DWORD, decimal)

(2) = Variable name [ASCII characters]

Note: The separator "!" is used in this command.

Response Structure

The following table shows the general structure of the "EXD1" FI command.

Line 1	Column 1

Meaning of the Columns

1 = Error state

WinPCL - Example EST

Read the value of WinPCL variable "IB_EXT24" in WinPCL program "Prog", at device address 00.

Exception:
The WinPCL variable "IB_EXT24" is declared in the WinPCL Program "Prog" as BOOL.

FI command		**00_BR_EST1!5892855!:Prog.IB_EXT24**
Line	**Column**	**Answer**
1	1	1

8.19 Execution Display: EXD

MWMX device groups

Designation

EXD **EX**ecution **D**isplay

Explanation

Information for displaying the execution of a movement is output.

FI command

Query the execution of a step or of an action.

BR_EXD1!(1)!(2)!(3)	**(Single Read)**
BC_EXD1!(1)!(2)!(3)	**(Cyclic Read)**
(1) = SFC entity name	[ASCII characters]
(2) = Step or action name	[ASCII - characters]
(3) = Behaviour of mode	[1 = all modes, 2 = manual mode]

Note: The separator "!" is used in this command.

Response Structure

The following table shows the general structure of the "EXD1" FI command.

Line 1	Column 1

Meaning of the Columns

1 = Execution	[1 = can be executed, 0 = cannot be executed]

Example EXD1

Query the execution of the step "open" for the chain "clamp" in control unit 0 for all modes.

FI command		**00_BR_EXD1!Station03A.Clamp!Open!1**
Line	**Column**	**Answer**
1	1	1

FI command

Query whether the condition analysis (control image) of a step chain is enabled.

BR_EXD2!(1)	**(Single Read)**
(1) = SFC entity name	[ASCII characters]

Note: The separator "!" is used in this command.

Response Structure

The following table shows the general structure of the "EXD2" FI command.

Line 1	Column 1

Meaning of the Columns

1 = Enabled [1 = enabled, 0 = not enabled]

Example EXD2

Query whether the condition analysis of the "clamp" chain has been enabled.

FI command		**00_BR_EXD2!Station03A.Clamp**
Line	**Column**	**Answer**
1	1	1

8.20 Read Reference Name of a PLC Variable: MAR

MWMX device groups

Designation

MAR **M**ap **A**bsolute PCL-**R**eference

PLC Explanation

The absolute reference name of a symbolic PLC variable is read out.

FI command

Read the absolute reference name of a PLC variable.

BR_MAR_(1) **(Single Read)**

BC_MAR_(1) **(Cyclic Read)**

(1) = Identifier of the PLC variable

PLC – Example MAR

Read the absolute reference name of the PLC variable with the identifier "abref" at device address 00.

Assumption:
The PLC variable with the identifier "abref" is of the type "INTEGER".

FI command		**00_BR_MAR_abref**
Line	**Column**	**Answer**
1	1	%M100.0

WinPlc Explanation

The absolute reference name of a symbolic WinPlc PLC variable with program entity is read out.

FI command

Read the absolute reference name of a WinPlc PLC variable.

BR_MAR_(1) **(Single Read)**

BC_MAR_(1) **(Cyclic Read)**

(1) = Identifier of the PLC variable

Win PLC - Example MAR

Read the absolute reference name of the Win PLC variable with the identifier "Prog.abref" at device address 00.

Assumption:
The Win PLC variable with the identifier "Prog.abref" is of the type "INTEGER" and is present in Win PLC program entity "Prog".

FI command		**00_BR_MAR_:Prog.abref**
Line	**Column**	**Answer**
1	1	%M100.0

Reference to Literature

See chapter entitled "Literature" [30].

8.21 Device Data of the Module Configuration: MCD

MWMX device groups

Designation **MCD** **M**odule **C**onfiguration: **D**evice Information

Explanation All device data for module configuration is read out from the "Moduldef.ini" file which is stored in the "[LW]:\Program Files\Indramat\MTGUI\CustomData\Resource" directory on standard installation. The device data is in the sections [DeviceAddrX], whereby "X" stands for the configured device addresses.

FI command Read out device data within the module configuration of the MWSX device group.

BR_MCD1	**(Single Read)**
BC_MCD1	**(Cyclic Read)**
BB_MCD1	**(Break Cyclic Read)**

Note: The "MCD1" FI command refers to all devices within the MWSX device group. Therefore, any valid device address can be indicated in the command line (see example MCD1).

Response Structure The following table shows the general structure of the response to the "MCD1" FI command. The number of lines depends on the number of configured devices. Each line consists of four columns for the device address as well as PLC-FB names for providing setup diagnostics, warning messages and start requirements.

Value Range of the Columns

1 = Device address	[0...15]
2 = PLC-FB name for the setup diagnostics	[max. 9 ASCII characters]
3 = PLC-FB name for the warning messages	[max. 9 ASCII characters]
4 = PLC-FB name for the start requirements	[max. 9 ASCII characters]

Example MCD1 Read all device data of the module configuration

Assumption:
The following devices have been configured in the MWSX device group:

- Device address 01 (ISP200-P)
- Device address 03 (ISP200-R)

FI command	**03_BR_MCD1**			
Answer				
Line	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	01	PVSetup_1	PVWarn_1	PVStart_1
2	03	PVSetup_3	PVWarn_3	PVStart_3

Reference to Literature See chapter entitled "Literature" [36].

8.22 Module Data of the Module Configuration: MCM

MWMX device groups

Designation **MCM** **M**odule **C**onfiguration: **M**odule Information

Explanation All module data of a particular device is read out from the "Moduldef.ini"' file. This file is located in the directory "[LW]:\Program Files\Indramat\MTGUI\CustomData\Resource" and contains the data for all module configurations. The module data is located in sections [DeviceAddrX\ModulY], whereby "X" stands for the device addressed and "Y" for the configured module numbers.

FI command Read-out module data from the module configuration with respect to a device from the MWSX device group.

BR_MCM1	**(Single Read)**
BC_MCM1	**(Cyclic Read)**
BB_MCM1	**(Break Cyclic Read)**

Response Structure The following table shows the general structure of the response to the "MCM1" FI command. The number of lines depends on the number of configured modules of a device. Each line consists of four columns for the module number, module name and PLC-FB names for general module errors and module messages.

Line 1	Column 1	...	Column 4

Value Range of the Columns

1 = Module number	[0...99]
2 = Module name	[max. 28 ASCII characters]
3 = PLC-FB name for general module errors	[max. 9 ASCII characters]
4 = PLC-FB name for module messages	[max. 9 ASCII characters]

Example MCM1 Read the module data of device 03 from the module configuration:

Assumption:
The following modules have been defined:

- Module number 5
- Module number 7

FI command	**03_BR_MCM1**			
Answer				
Line	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	5	Module 5 - Milling	PVError_5	PVMsg_5
2	7	Module 7 - Drilling	PVError_7	PVMsg_7

Reference to Literature See chapter entitled "Literature" [36].

8.23 SFC Data of the Module Configuration: MCS

MWMX device groups

Designation **MCS** **M**odule **C**onfiguration: **S**FC Information

Explanation All SFC data of a particular module is read out from the "Moduldef.ini" file. This file is located in the directory "[LW]:\Program Files\Indramat\MTGUI\CustomData\Resource" and contains the data for all module configurations. The SFC data is located in sections [DeviceAddrX\ModulY\Sfc], whereby "X" stands for the device addressed and "Y" for the selected module number.

FI command Read out the SFC data with respect to the module of a device from the module configuration of the MWSX device group.

BR_MCS1_(1)	**(Single Read)**
BC_MCS1_(1)	**(Cyclic Read)**
BB_MCS1_(1)	**(Break Cyclic Read)**
(1) = Module number	[0...99]

Response Structure The number of lines depends on the number of configured Indrastep step chains for a device. Each line contains a column for the name of the Indrastep step chains.

Value Range of the Column 1 = Name of the Indrastep step chain [format W.X.Y.Z]

Format W.X.Y.Z	Value Range
W	Max. 9 ASCII characters
X	Max. 9 ASCII characters ! OPTIONAL !
Y	Max. 9 ASCII characters ! OPTIONAL !
Z	Max. 9 ASCII characters ! OPTIONAL !

Example MCS1 Read the name of the Indrastep step chain of module 5 from device 03 of the module configuration.

Assumption:
The following Indrastep step chains have been defined:

- ISFB_1
- FB_US.ISFB_3
- FB_US.ISFB_3.SW1
- FB_US.ISFB_3.SW1.ABBA

FI command		**03_BR_MCS1_5**
Line	**Column**	**Answer**
1	1	ISFB_1
2	1	FB_US.ISFB_3
3	1	FB_US.ISFB_3.SW1
4	1	FB_US.ISFB_3.SW1.ABBA

8.24 Downloading Message Texts: MFD

MWMX device groups

Designation **MFD** **M**essage **F**iles **D**ownload

FI command This is used to load the message texts into the device indicated. These message texts are required for small devices. The following message texts are transmitted, depending on the type of device:

- system error messages
- transmission error messages, and/or
- mechanism messages.

Note: This FI command is an FI job!

BW_MFD1 **(Single Write)**

Response Structure The response to the "MFD1" FI command consists of three lines, each with one column. The meaning of the elements is as follows:

Line 1 = Job ID	[01...20] (refer to chapter entitled "FI Commands for the MPCX Device Group", IFJ).
Line 2 = FI command	[string, in accordance to chapter entitled "Elements of the FI Command"]
Line 3 = FI job error code	(see chapter entitled "Error Codes")

Example MFD1 Load message texts into the device with device address 00.

FI command		**00_BW_MFD1**
Line	**Column**	**Answer**
1	1	01
2	1	00_BW_MFD1
3	1	0

8.25 Reading Machine Key Information : MKS

MWMX device groups

Designation **MKS** **M**achine **K**ey **S**tatus

Explanation Current machine key information can be read for the selected device.

FI command Read machine key information for selected device.

BR_MKS	**(Single Read)**
BC_MKS	**(Cyclic Read)**
BB_MKS	**(Break Cyclic Read)**

Response Structure The following table shows the general structure of the response to the FI command "MKS".

Line 1	Column 1	Column 2

Value Range/Meaning of Columns

1 =	Information of machine key	[4 byte in HEX coding]
2 =	Information valid?	[0 = not valid, 1=valid]

Example MKS Read the current machine key information for device 0.

FI command		00_BR_MKS
Line	**Column**	**Answer**
1	1	00000000
	2	0

8.26 Writing the GUI-SK Block: MKT

MWMX device groups

Designation **MKT** **M**achine **K**ey **T**able

Explanation Writes the GUI-SK16 block in the PLC.

FI command Write GUI-SK16 block.

BW_MKT1_(1) **(Single Write)**

(1) = List of the 48 PLC variables for writing the GUI-SK16 block.

A distinction is made between the following cases:

1. Clear GUI-SK16 block.
2. Write the GUI-SK16 block with the 48 PLC variables, filling gaps with $SPACE.

Response Structure (P_ACK) is returned following successful transmission.

Line 1	Column 1

Value Range/Meaning of the Columns

1 = Successfully completed (P_ACK)

1. Example MKT1

Clear GUI-SK16 block:

FI command		**Value to be written: $EMPTY** **00_BW_MKT1**
Line	**Column**	**Answer**
1	1	(P_ACK)

2. Example MKT1

Write GUI-SK16 block:

FI command		**Value to be written: $EMPTY** **SPSVAR1,SPSVAR2,$SPACE,...** **00_BW_MKT1**
Line	**Column**	**Answer**
1	1	(P_ACK)

FI command

Write the GUI-SK16 block, writing only those PLC variables which are defined in the current PLC program. All undefined PLC variables are automatically replaced by $SPACE and returned as a partial result (column 2).

BW_MKT2_(1)	**(Single Write)**
(1) = List of the 48 PLC variables for writing the GUI-SK16 block.	A distinction is made between the following cases: 1. Clear GUI-SK16 block: BW_MKT2 $EMPTY 2. Write the GUI-SK16 block with the 48 PLC variables, filling gaps with $SPACE: BW_MKT1 SPSVAR1,SPSVAR2, $SPACE,$SPACE,....

Response Structure

After successful transmission, one line with two columns is returned.

Line 1	**Column 1**	**Column 2**

Value Range/Meaning of Columns

1 =	Status report	[0 = at least 1 PLC variable in the current PLC program is NOT defined 1 = ALL PLC variables could be written]
2 =	List of the NON-defined PLC variables in the current PLC program	[-- = ALL PLC variables could be written, or else list of the PLC variables that could not be written.] The individual PLC variables are separated by a comma.

Example MKT1

Write GUI-SK16 component with 48 PLC variables, while the PLC variables SPSVAR11 and SPSVAR12 are NOT defined in the current PLC program.

FI command		**Value to be written:** **SPSVAR1,SPSVAR2,...SPSVAR48** **00_BW_MKT1**
Line	**Column**	**Answer**
1	1	(P_ACK)

Extended information

The variables are divided into 3 groups of 16 variables each and have the following meaning:

1. Variables 1 - 16: Machine function keys
2. Variables 17 - 32: Status pressed
3. Variables 33 - 48: Status shining

Note: When, for example, only the first 8 M keys are used, the telegram will contain only these 8 PLC variables. The other 40 variables need not be defined in the transmission parameter.

When certain areas, e.g. of M keys, are left unused, they must be filled up with '$SPACE' up to the next variable.

8.27 Read System Messages: MSG

MWMX device groups

Designation **MSG** **M**e**S**sa**G**e

Explanation Reading of system messages

FI command Message

CC_MSG_(1) **(Cyclic Read)**

(1) = SYS-Message number

Note: Exists only as a cyclic command

Response Structure The response of the FI command 'MSG' consists of the system message data.

Example MSG 00_CC_MSG_64 (64 = MSG_SYSERRGEN)

FI command		**00_CC_MSG_64/3**
Line	**Column**	**Answer**
1	1	00

Restriction The following system messages:

SYS Message	SYS Message number
MSG_PCLUPDBEG	52
MSG_PARUPDBEG	24
MSG_FWAUPDBEG	82

These commands cannot be used with the following programs:

- Bosch Rexroth OPC server
- Bosch Rexroth DDE server

8.28 Reading the Firmware Identification: MTC

MWMX device groups

Designation **MTC** **MT**-**C**NC Slot Software Version

FI command This command is used to read the firmware identification from the various control components (slot numbers).

Note: For the time this FI command is executed, the internal FI communication interlocks (fast timeout monitoring, offline operation, etc.) are switched off.

FI command

BR_MTC_(1)	**(Single Read)**
BC_MTC_(1)	**(Cyclic Read)**
(1) = Slot number	[1=CNC, 2=SIO, 3=PLC, 4=APR1 5=APR2, 6=APR3, 7=APR4]

Response Structure The following table shows the general structure of the response to the FI command "MTC". A line of 1 column is output.

Line 1	**Column 1**

Value Range/Meaning of Columns 1 = Firmware identification string [max. 16 ASCII characters]

Example MTC Read the firmware identification of slot number 1 (CPU) of device 00.

FI command		**00_BR_MTC_1**
Line	**Column**	**Answer**
1	1	CPU01/0004-20V00

8.29 ProVi Diagnosis Data: PDD

MWMX device groups

Designation **PDD** **P**rovi **D**iagnosis **D**ata

Explanation Data for ProVi criteria analysis is output.

FI command Output of files to indicate the detail in the editor.

BR_PDD1_(1)_(2){_(3)}	**(Single Read)**
(1) = Message ID	[ASCII characters]
(2) = Message type	[1 = error, 2 = messages, 10 = warnings, 11 = start requirements, 12 = setup diagnosis]
(3) = Module number	[1...99] ! only for message type 1 -2!

Response Structure The following table shows the general structure of the PDD1 FI command.

Line 1	**Column 1**	**...**	**Column 5**

Meaning of the Columns

1 = POU ID	[ASCII characters]
2 = Detail morpheme	[ASCII characters] (DWORD, decimal)
3 = Error ID	[ASCII characters] (DWORD, decimal)
4 = POE entity name	[ASCII characters]
5 = Nw ID (network ID)	[ASCII characters]

Example PDD1

Indication of data of a ProVi error with ID 43923028 from module 3 in control unit 0.

FI command		**00_BR_PDD1_43923028_1_1**
Line	**Column**	**Answer**
1	1	STATION_1_2
	2	98243823
	3	34985304
	4	Station2.Module3
	5	43493454

FI command

Output the I/O addresses to display a detail.

BR_PDD2_(1)_(2){_(3)}	**(Single Read)**
(1) = Message ID	[ASCII characters]
(2) = Message type	[1 = error, 2 = messages, 10 = warnings, 11 = start requirements, 12 = setup diagnosis]
(3) = Module number	[1...99] ! only for message type 1 -2!

Response Structure

The following table shows the general structure of the PDD2 FI command.

Line 1-n	**Column 1**	**Column 2**

Meaning of the Columns

1 = Variable morpheme	[ASCII characters] (DWORD, decimal)
2 = I/O address	[ASCII characters]

Example PDD2

Query of the I/O addresses of a ProVi error with ID 43923028 from module 3 in control unit 0.

Three variables have an I/O address.

FI command		**00_BR_PDD2_43923028_1_1**
Line	**Column**	**Answer**
1	1	98243823
	2	%I3.2.0
2	1	40923423
	2	%Q23.21.7
3	1	34985304
	2	%I100.3.5

FI command

Determine the multilingual comments for displaying a detail.

BR_PDD3_(1)_(2){_(3)}	**(Single Read)**
(1) = Message ID	[ASCII characters]
(2) = Message type	[1 = error, 2 = messages, 10 = warnings, 11 = start requirements, 12 = setup diagnosis]
(3) = Module number	[1...99] ! only for message type 1 -2!

Response Structure

The following table shows the general structure of the PDD3 FI command.

Line 1-n	Column 1	Column 2

Meaning of the Columns

1 = Comment morpheme	[ASCII characters] (DWORD, decimal)
2 = New comment	[ASCII characters]

Example PDD3

Query of the comments for indication of a ProVi error with ID 43923028 from module 3 in control unit 0.

Two comments are replaced by another text.

FI command		**00_BR_PDD3_43923028_1_1**
Line	**Column**	**Answer**
1	1	98243823
	2	Clamp open
2	1	40923423
	2	Clamp closed

FI command

Query of the status of a certain message

BR_PDD4_(1)_(2){_(3)}	**(Single Read)**
(1) = Message number	[ASCII characters]
(2) = Message type	[1 = error, 2 = messages, 10 = warnings, 11 = start requirements, 12 = setup diagnosis]
(3) = Module number	[1...99] ! only for message type 1 -2!

Response Structure

The following table shows the general structure of the PDD4 FI command.

Line 1-n	Column 1	Column 2

Meaning of the Columns

1 = Message is present	[YES, NO]
2 = Criteria analysis exists	[YES, NO]

Example PDD4

Query of the status of a ProVi error, number 1001 from module 3 in control 0.

This message is not present at the moment, and there is a criteria analysis.

FI command		**00_BR_PDD4_1001_1_1**
Line	**Column**	**Answer**
1	1	NO
	2	YES

FI command

Determination of the MessageID of a certain message

BR_PDD5!(1)!(2)!(3)!(4){!(5)}	**(Single Read)**
(1) = POU entity name	[ASCII characters]
(2) = Nw ID	[ASCII characters]
(3) = Message number	[ASCII characters]
(4) = Message type	[1 = error, 2 = messages, 10 = warnings, 11 = start requirements, 12 = setup diagnosis]
(5) = Module number	[1...99] ! only for message type 1 -2!

Note: The separator "!" is used in this command.

Response Structure

The following table shows the general structure of the PDD5 FI command.

Line 1-n	Column 1	...	Column 3

Meaning of the Columns

1 = Message ID	[ASCII characters] (DWORD, decimal)
2 = Message is present	[YES, NO]
3 = Criteria analysis exists	[YES, NO]

Example PDD5

Determination of the MessageID of a ProVi error, number 1001 from module 25.40 mm control 0.

Assumption:
This message is not present at the moment, and there is a criteria analysis.

FI command		**00_BR_PDD5!Station2.Modul3!43493454!1001!1!1**
Line	**Column**	**Answer**
1	1	240872342
	2	NO
	3	YES

8.30 Reading the Size of the PLC Memory: PMI

MWMX device group

Designation **PMI** **P**LC **M**emory **I**nformation

Explanation The current size of the PLC memory is read out.

FI command **CR_PMI** **(Single Read)**

Response Structure One line with two values in BYTE is output:

1. Total memory
2. Free memory available now.

Line 1	Column 1	Column 2

Example PMI Read the current size of the PLC memory at device address 00.

FI command		**00_CR_PMI**
Line	**Column**	**Answer**
1	1	123456
	2	3210

8.31 Issuing SYS Messages Specific to the PCL: PSM

MWMX device groups

Designation **PSM** **P**CL **S**ys **M**essage

Explanation Issues the most important SYS messages regarding the PCL programming interface – required for remote programming.

Note:
The appropriate device address is passed as the write value.

It allows the following SYS messages to be initiated:

- Start of PCL download,
- end of PCL download,
- start of PLC online edit,
- end PLC online edit,
- start of PCL declaration change, and
- end of PCL declaration change.

FI command Issue the most important PCL SYS messages.

BW_PSM1_(1) **(Single Write)**

(1) = Requested SYS message [1= start of PCL download
2= end of PCL download
3= start of PCL online edit
4= end of PCL online edit
5= start of PCL declaration change
6= end of PCL declaration change]

Value to be written
device address

Response Structure

The following table shows the general structure of the response to the FI command "PSM1".

Line 1	Column 1	...	Column 8

Value Range/Meaning of Columns

1 =	Status report	[READY=SYS message has been correctly acknowledged by the WIN32 applications] [ERROR=SYS message has NOT been acknowledged by a WIN32 application within the pre-set time]
2 =	Task name (LogInIf name)	[Task name that has triggered the SYS message]
3 =	SYS message number	[contains the issued SYS message number]
4 =	Acknowledgement time	[contains the pre-set acknowledgement time]
5 =	Reference information	[contains, where applicable, the additional information transferred as a write value]
6 =	Length of reference information	[0 where NO reference information has been transferred]
7 =	Where applicable, LOG channel of the FI that has NOT acknowledged	[-- = acknowledgements have been completed in time or the LOG channel number of the WIN32 application that has NOT acknowledged in time]
8 =	Where applicable, task name that has NOT acknowledged in time	[-- = acknowledgements have been completed in time or the task name that has NOT acknowledged in time]

Example PSM1

Issue the SYS message Beginning PCL Download. The reference information, device address 00, is also transferred as a write value.

FI command		**XX_BW_PSM1_1**
Line	**Column**	**Answer**
1	1	READY
	2	WINPCL.EXE
	3	14
	4	30000
	5	00
	6	2
	7	--
	8	--

8.32 Edit PROVI Message Files: PVA

MWMX device groups

Designation **PVA** **PROV**I-Messages **A**ccess

Explanation This write command creates PROVI message files. With this write value, it is possible to decide whether the PROVI messages are to be generated according to the current PLC project, or selectively.

FI command **BW_PVA1** **(Single Write)**

Note: This command is an FI job command.

Value to be written

No write value exists	PROVI message files according to the current PLC project.
Write value exists	List of the requested PROVI message files (separated by a comma) according to the format: [PROVI-Diag-type: module number] Example: 01:01,01:02,02:02

Note: The value to be written is passed to the "acValue" parameter as an ASCII value in the "DataTransfer" routine.

Response Structure The response to the "BW_PVA1" FI command consists of three lines, each with one column. The meaning of the elements is as follows:

- Line 1 = Job ID [01...20]
  (see Chapter "FI Commands for the MPCX Device Group", IFJ).
- Line 2 = FI command
  [String, in accordance with Chapter "Elements of the FI Command"]
- Line 3 = FI Job Error Code (see Chapter "Error Codes")

...

Example PVA1 No write value is passed, i.e. the PROVI message files are generated according to the current PLC project.

FI command		**00_BW_PVA1**
Line	**Column**	**Answer**
1	1	01
2	1	00_BW_PVA1
3	1	0

Explanation The read command returns the most significant information on the created PROVI message files.

FI command **BR_PVA1** **(Single Read)**

....

Response Structure The following table shows the general construction of the answer of the FI command BR_PVA1. For each available PROVI message file, 1 line with 10 columns each is created.

Line 1...n	**Column 1**	**...**	**Column 10**

Value Range/Meaning of Columns

1 =	PROVI diagnosis type	[1..20]
2 =	PROVI diagnosis type designation	[The following designations can be returned: StartCondition, Error, Message, Warning, Setup]
3 =	Module number	[1..99]
4 =	PROVI diagnosis type and module number	[PROVI diagnosis type: module number, see write value for BW_PVA2]
5 =	Complete name of the PROVI message text file	[max. 200 ASCII characters]
6 =	Memory required for PROVI messages in the control	[figure in ASCII format]
7 =	Complete name of the PROVI index file	[max. 200 ASCII characters]
8 =	Memory required for PROVI index files in the control	[figure in ASCII format]
9 =	Total memory (text+index) required in the control	[figure in ASCII format]
10 =	Total memory for ALL PROVI files (text+index) required in the control	[figure in ASCII format]

Example PVA1

The most significant information of 2 available PROVI message files are returned.

FI command		**00_BR_PVA1_1**
Line	**Column**	**Answer**
1	1	1
	2	Error
	3	1
	4	01:01
	5	D:\Programs\indramat\Mtgui\Project_000\ Programmdata\Device_000\Diag\De\ERROR 1.TXT
	6	1345
	7	D:\Program Files\indramat\Mtgui\Project_000\ Programmdata\Device_000\Diag\De\ERROR 1.IDX
	8	234
	9	1579
	10	4491
2	1	2
	2	Message
	3	1
	4	02:01
	5	D:\Programs\indramat\Mtgui\Project_000\ Programmdata\Device_000\Diag\De\ERROR 1.TXT
	6	2456
	7	D:\Programs\indramat\Mtgui\Project_000\ Programmdata\Device_000\Diag\De\ERROR 1.TXT
	8	456

	9	2912
	10	4491

Explanation

This write command transmits PROVI message files into the selected device. Through the write value, it is possible to chose whether ALL or only the PROVI messages selected via the write value are to be transmitted.

FI command

BW_PVA2 **(Single Write)**

Note: This command is an FI job command.

Value to be written

No write value exists	All PROVI message files are transmitted into the selected device
Write value exists	List of the requested PROVI message files (separated by a comma) according to the format: [PROVI-Diag-type: module number] Example: 01:01,01:02,02:02

Note: The value to be written is passed to the "acValue" parameter as an ASCII value in the "DataTransfer" routine.

Response Structure

The response to the "BW_PVA2" FI command consists of three lines, each with one column. The meaning of the elements is as follows:

- Line 1 = Job ID [01...20]
  (see Chapter "FI Commands for the MPCX Device Group", IFJ).
- Line 2 = FI command
  [String, in accordance with Chapter "Elements of the FI Command"]
- Line 3 = FI Job Error Code (see Chapter "Error Codes")

....

Example PVA1

No write value is passed, i.e. all PROVI message files should be transmitted.

FI command		**00_BW_PVA2**
Line	**Column**	**Answer**
1	1	01
2	1	00_BW_PVA2
3	1	0

8.33 Formatted Input / Output of PLC Variables: PVF

MWMX device groups

Designation **PVF** **P**LC **V**ariable **F**ormatted

Explanation Formatted reading and writing of PLC variables, arrays and structures.

FI command Read PLC variables.

CR_PVF_(1)	**(Single Read)**
CC_PVF_(1)	**(Cyclic Read)**
CB_PVF_(1)	**(Break Cyclic Read)**
(1) = Identifier of the PLC variable	[acc. to declaration part of the PLC]

Response Structure One line with one column is output for single variables. For array and structure variables, one line per element is output, depending on the number of elements.

Line 1...n:	Column 1

n = number of elements.

Note: Only defined PLC variables can be read and written. Addressing a non-declared variable results in an error message. A PLC variable can only be read if its data length does not exceed 240 byte. (Refer also to chapter on "Programming" and "Guidelines").

Value Ranges ANSI / ASCII The value range of the response depends on the data type of the variable read. The following table indicates the range in which the results string is to be expected when reading out a single variable and into which C-data type this string can be converted without loss of information:

Data Type	Value Range	Can be converted to C-data type
BOOL	[0;1]	unsigned char
SINT	[-128...127]	char
INT	[-32768...32767]	short
DINT	[2147483648...2147483647]	long
USINT	[0...255]	unsigned char
UINT	[0...65535]	unsigned short
UDINT	[0...4294967295]	unsigned long
BYTE	[0x00...0xFF]	unsigned char
WORD	[0x0000....0xFFFF]	unsigned short
DWORD	[0x00000000...0xFFFFFFFF]	unsigned long
TIME	[0...4294967295]	unsigned long (msec)
CHAR	[$00...$20,!...~,$7F...$FF]	char
STRING	<String> whereby <String> string is a character string with a maximum of as many characters as are declared for the string in the PLC	Char[xx+1]] +1 i.e. room for the zero byte
REAL	[-3.402823567E+38...3.402823567E+38]	Float

Note: An empty string is identified by two single inverted commas: ' ' (do not confuse with the double inverted commas ")!

All single variables can be part of array and structure variables. The value ranges maintain their validity, even when within structured data types.

Binary Value Range

The value range of the response depends on the data type of the variable read. The following table indicates the value range in which to expect the binary value of a single variable and how many bytes are included in the binary byte sequence:

Data Type	Value Range	Length (bytes)
BOOL	[00$_H$...01$_H$]	1
SINT	[80 $_H$...7F $_H$] i.e. –128...127	1
INT	[8000 $_H$ (-32768)...7FFF $_H$ (32767)]	2
DINT	[80000000 $_H$ (-2147483648)... 7FFFFFFF $_H$ (2147483647)]	4
USINT	[00 $_H$ (0)...FF $_H$ (255)]	1
UINT	[00 $_H$ (0)...FFFF $_H$ (65535)]	2
UDINT	[0...4294967295]	4
BYTE	[0x00...0xFF]	1
WORD	[0x0000....0xFFFF]	2
DWORD	[0x00000000...0xFFFFFFFF]	4
TIME	[0...4294967295]	4
CHAR	[$00...$20,!...~,$7F...$FF]	1
STRING	<String> whereby <String> string is a character string with a maximum of as many characters as are declared for the string in the PLC	XX+1
REAL	[-3.402823567E+38...3.402823567E+38]	4

Note: Binary array and structure elements are joined together without any spaces between (1-byte alignment).

PLC - Example 1 PVF

Read the value of the PLC variable "STK_TXT" in ASCII format from device address 00.

Assumption:
The "STK_TXT" variable is declared as STRING in the PLC program.

FI command		**00_CR_PVF_STK_TXT/1**
Line	**Column**	**Answer**
1	1	Repeat counter

WinPCL - Example 1 PVF

Read the value of WinPCL variable "STK_TXT" in ASCII format in WinPCL program "Prog" at device address 00.

Assumption:
The WinPCL variable "STK_TXT" is declared in WinPCL program entity "Prog" as STRING.

FI command		**00_CR_PVF_:Prog.STK_TXT/1**
Line	**Column**	**Answer**
1	1	Repeat counter

PLC - Example 2 PVF

Read the value of the PLC array "BEG_END" in ANSI format from device address 00.

Assumption:
The "BEG_END" variable is declared as BYTE with 2 elements in the PLC program.

FI command		00_CR_PVF_BEG_END/3
Line	**Column**	**Answer**
1	1	0x00
2	1	0x1F

WinPCL - Example 2 PVF

Read the value of WinPCL array "BEG_END" in ANSI format in WinPCL program "Prog" at device address 00.

Assumption:
The WinPCL variable "BEG_END" is declared in WinPCL program entity "Prog" as BYTE with two elements.

FI command		00_CR_PVF_:Prog.BEG_END/3
Line	**Column**	**Answer**
1	1	0x00
2	1	0x1F

PLC - Example 3 PVF

Read the value of the PLC structure "MSTRCT" in ASCII format from device address 00.

Assumption:
The "MSTRCT" variable is declared as a structure in the PLC program as follows:

```
TYP STRUCT
        T1      BOOL
        T2      CHAR
        T3      STRING[16]
        T4      TIME
END
```

FI command		00_CR_PVF_MSTRCT/1
Line	**Column**	**Answer**
1	1	0
2	1	A
3	1	ROBOT AXIS X
4	1	2000

WinPCL - Example 3 PVF

Read the value of WinPCL structure "MSTRCT" in ASCII format in WinPCL program "Prog" at device address 00.

Assumption:
The WinPCL variable "MSTRCT" is declared as a structure in WinPCL program entity "Prog" as follows:

```
TYP STRUCT
        T1      BOOL
        T2      CHAR
        T3      STRING[16]
        T4      TIME
END
```

FI command		**00_CR_PVF_:Prog.MSTRCT/1**
Line	**Column**	**Answer**
1	1	0
2	1	A
3	1	ROBOT AXIS X
4	1	2000

FI command

Write PLC variable.

CW_PVF_(1)	**(Single Write)**
(1) = Identifier of the PLC variable	[acc. to declaration part of the PLC]

Value to be written

Value of data element [see value ranges]

Note: The value to be written is passed to the "acValue" parameter in the "DataTransfer" routine. The data code of the value is passed to the parameter "ValType".

Response Structure

One line is output with a column for acknowledgement of whether or not the FI command has been executed successfully.

(P_ACK) = **P**ositive **ACK**nowledge Data element has been set

Value Range of the value to be written in ANSI / ASCII Format

The value ranges agree for the most part with the ANSI / ASCII result-value ranges during read access. ANSI umlauts are thereby converted into ASCII umlauts. Only ASCII umlauts are stored in the control unit. For deviations to this, please refer to the following note:

Note: Strings are enclosed by two single inverted commas ' ' , e.g. 'drill'.

Special characters can be indicated in accordance with DIN-1131 by a $ sign.

The following are used:

- $' '
- $$ $
- $R \r (Carriage Return)
- $L \n (Linefeed)
- $P \f (Form feed)
- $T \t (Tab)
- $xx xx refers to a character written as a hexadecimal value, e.g. $20 (space)

Array and structure elements are separated by a space.

Value Range of the Value to be written in Binary Format

The value ranges agree with the binary result-value range during read access. For deviations to this, please refer to the following note:

PLC - Example 4 PVF

Write into the PLC variable "STK_TXT" at device address 00. The value is passed in ANSI format.

Assumption:
The "STK_TXT" variable is declared as STRING in the PLC program.

FI command		**00_CW_PVF_STK_TXT/3**
Line	**Column**	**Answer**
1	1	(P_ACK)

Value to be written:

Value of data element	'item counter'
Data code	/3

WinPCL - Example 4 PVF

Write into the WinPCL variable "STK_TXT" in WinPCL program "Prog" at device address 00. The value is passed in ANSI format.

Assumption:
The WinPCL variable "STK_TXT" is declared in WinPCL program entity "Prog" as STRING.

FI command		**00_CW_PVF_:Prog.STK_TXT/3**
Line	**Column**	**Answer**
1	1	(P_ACK)

Value to be written:

Value of data element	'item counter'
Data code	/3

PLC - Example 5 PVF

Write into the PLC byte array "BEG_END" at device address 00. The value is passed in ANSI format.

Assumption:
The "BEG_END" variable is declared as a BYTE array with 2 elements in the PLC program.

FI command		**00_CR_PVF_BEG_END/3**
Line	**Column**	**Answer**
1	1	(P_ACK)

Value to be written:

Value of data element	0x20 0x3f
Data code	/3

WinPCL - Example 5 PVF

Write into the WinPCL byte array "BEG_END" in WinPCL program "Prog" at device address 00. The value is passed in ANSI format.

Assumption:
The WinPCL variable "BEG_END" is declared in WinPCL program entity "Prog" as BYTE with two elements.

FI command		**00_CW_PVF_:Prog.BEG_END/3**
Line	**Column**	**Answer**
1	1	(P_ACK)

Value to be written:

Value of data element	0x20 0x3f
Data code	/3

PLC - Example 6 PVF

Write the value of element T3 of the PLC structure "MSTRCT" at device address 00. The string "COUNTER" is transferred in binary format.

Assumption:
The "MSTRCT" variable is declared as a structure in the PLC program as follows:

```
TYP STRUCT
        T1      BOOL
        T2      CHAR
        T3      STRING[16]
        T4      TIME
END
```

FI command		00_CW_PVF_MSTRCT.T3/2
Line	**Column**	**Answer**
1	1	(P_ACK)

Value to be written:

Value of data element	Binary sequence: 43 4F 55 4E 54 45 52 00
Data code	/2

WinPCL - Example 6 PVF

Write the value of element T3 of the WinPCL structure "MSTRCT" at device address 00. The string "COUNTER" is transferred in binary format.

Assumption:
The WinPCL variable "MSTRCT" is declared as a structure in WinPCL program entity "Prog" as follows:

```
TYP STRUCT
        T1      BOOL
        T2      CHAR
        T3      STRING[16]
        T4      TIME
END
```

FI command		00_CW_PVF_:Prog.MSTRCT.T3/2
Line	**Column**	**Answer**
1	1	(P_ACK)

Value to be written:

Value of data element	Binary sequence: 43 4F 55 4E 54 45 52 00
Data code	/2

PLC - Example 7 PVF

Write the value of the PLC structure "MSTRCT" from the structure "mstrct" previously stored in the C program at device address 00.

Assumption:
The "MSTRCT" variable is declared as a structure in the PLC program as follows:

```
TYP STRUCT
        T1      BOOL
        T2      CHAR
        T3      STRING[16]
        T4      TIME
END
```

To exchange binary data in a C program, the following "C" data type can be used:

```
#pragma pack(1)    //Write all elements
                   //without spaces next to each other.
typeder struct
{
      unsigned char T1;
      char          T2;
      char          T3[17]; //Space for zero byte
      unsigned long T4;
} Tymstrct;        // Declare structure
Tymstrct mstrct;   // Apply structure
```

FI command		**00_CW_PVF_MSTRCT/2**
Line	**Column**	**Answer**
1	1	(P_ACK)

Value to be written: address of the C structure.

Value of data element	&mstrct
Data code	/2

WinPCL - Example 7 PVF

Write the value of the WinPCL structure "MSTRCT" from the structure "mstrct" previously stored in the C program at device address 00.

Assumption:
The WinPCL variable "MSTRCT" is declared as a structure in WinPCL program entity "Prog" as follows:

```
TYP STRUCT
        T1      BOOL
        T2      CHAR
        T3      STRING[16]
        T4      TIME
END
```

To exchange binary data in a C program, the following "C" data type can be used:

```
#pragma pack(1)    //Write all elements
                   //without spaces next to each other.
typeder struct
{
      unsigned char T1;
      char          T2;
      char          T3[17]; //Space for zero byte
      unsigned long T4;
} Tymstrct;        // Declare structure
Tymstrct mstrct;   // Apply structure
```

FI command		**00_CW_PVF_:Prog.MSTRCT/2**
Line	**Column**	**Answer**
1	1	(P_ACK)

Value to be written: address of the C structure.

Value of data element	&mstrct
Data code	/2

8.34 ProVi Messages: PVM

MWMX device groups

Designation **PVM** **Pro**Vi **M**essages

Explanation ProVi messages are output. These messages are assigned to a particular message type or module.

FI command Output all ProVi messages.

For reasons of optimization, not all data is called up. Accordingly, the Diagnosis server must be informed that the data is required (see ADW).

BR_PVM1_(1){_(2)} **(Single Read)**

BC_PVM1_(1){_(2)} **(Cyclic Read)**

(1) = Message type [1 = error, 2 = messages, 10 = warnings,
11 = start requirements,
12 = setup diagnosis]

(2) = Module number [1...99] ! only for message type 1 -2!

Output first ProVi messages.

BR_PVM2_(1){_(2)} **(Single Read)**

BC_PVM2_(1){_(2)} **(Cyclic Read)**

(1) = Message type [1 = error, 2 = messages, 10 = warnings,
11 = start requirements,
12 = setup diagnosis]

(2) = Module number [1...99] ! only for message type 1 -2!

Response Structure The following table shows the general structure of the FI commands "PVM1" and "PVM2". The number of lines depends on the number of messages pending.

If there are no messages, the number of lines is 0.

Line 1...n	**Column 1**	**...**	**Column 6**

Meaning of the Columns

1 = Message text	[ASCII characters]
2 = Message number	[ASCII characters]
3 = Time stamp day	[mm.dd.yyyy]
4 = Time stamp time	[hh:mm:ss]
5 = Message ID	[ASCII characters] (DWORD, decimal)
6 = Reference text exists	[YES, NO]
7 = Criteria analysis exists	[YES, NO]
Message HTML file	[ASCII characters]

Example PVM1 All ProVi errors from module 3 in control unit 0. There are two messages:

FI command		**00_BR_PVM1_1_3**
Line	**Column**	**Answer**
1	1	Guard not closed
	2	34
	3	01.27.2000
	4	14:56:32
	5	43923028
	6	YES
	7	NO
	8	
2	1	Oil pressure too low
	2	124
	3	01.27.2000
	4	15:03:10
	5	98234039
	6	NO
	7	YES
	8	

Example PVM2 The first ProVi error from module 3 in control unit 0.

There are two messages:

FI command		**00_BR_PVM2_1_3**
Line	**Column**	**Answer**
1	1	Guard not closed
	2	34
	3	01.27.2000
	4	14:56:32
	5	43923028
	6	YES
	7	NO
	8	

FI command Output the reference information of a ProVi message.

BR_PVM3_(1)_(2){_(3)}	**(Single Read)**
(1) = Message ID	[ASCII characters]
(2) = Message type	[1 = error, 2 = messages, 10 = warnings, 11 = start requirements, 12 = setup diagnosis]
(3) = Module number	[1...99] ! only for message type 1 -2!

Response Structure The following table shows the general structure of the "PVM3" FI command.

Line 1	**Column 1**	**...**	**Column 16**

Meaning of the Columns

1 =	Message text	[ASCII characters]
2 =	Message number	[ASCII characters]
3 =	Error category	[ASCII characters] (empty no category)
4 =	Time stamp day	[mm.dd.yyyy]
5 =	Time stamp hour	[hh:mm:ss]
6 =	Reference text available	[YES, NO]
7 =	Reference text	[ASCII characters]
8 =	Message ID	[ASCII characters] (DWORD, decimal)
9 =	Diagnosis source	[ASCII characters] (PLC, CNC)
10 =	POE name	[ASCII characters]
11 =	Detail name	[ASCII characters] (empty implementation)
12 =	Detail type	[1 = action block, 3 = transition, 4 = implementation]
13 =	Network number	[ASCII characters]
14 =	Variable name	[ASCII characters]
15 =	POU entity name	[ASCII characters]
16 =	POU type	[2 = program, 3 = function block]
17 =	Analysis of criteria available	[YES, NO]
18 =	Message HTML file	[ASCII characters]
19 =	Reference info HTML file	[ASCII characters]

Example PVM3

Reference text of a ProVi error with ID 43923028 from module 3 in control unit 0.

FI command		**00_BR_PVM3_43923028_1_3**
Line	**Column**	**Answer**
1	1	Guard not closed
	2	34
	3	1
	4	01.27.2000
	5	14:56:32
	6	YES
	7	Oil pressure too low Oil pipe leaking or insufficient oil.
	8	43923028
	9	PLC
	10	MODULE3
	11	
	12	4
	13	34
	14	EschutzT
	15	Station2.Module3
	16	3

	17	NO
	18	
	19	D:\Program Files\Indramat\MtGui\Project_000\ProgramData\HMTL\DE\Error34.html

FI command

One after the other, all active ProVi messages are output. In the result, one line each is returned. After expiry of the set time, the next message is returned. The clock frequency can be set via the last parameter. This value can only be set once for the PC. The value transmitted last is always the valid value. Default setting is one second.

BR_PVM4_(1){_(2)_(3)}	**(Single Read)**
BC_PVM4_(1){_(2)_(3)}	**(Cyclic Read)**
(1) = Message type	[1 = error, 2 = messages, 10 = warnings, 11 = start requirements, 12 = setup diagnosis]
(2) = Module number	[1...99] ! only for message type 1 -2!
(3) = Clock frequency	[ASCII characters] Time in ms

Response Structure

The following table shows the general structure of the "PVM4" FI command.

If there are no messages, the number of lines is 0.

Line 1	Column 1	...	Column 8

Meaning of the Columns

1 = Message text	[ASCII characters]
2 = Message number	[ASCII characters]
3 = Time stamp day	[mm.dd.yyyy]
4 = Time stamp time	[hh:mm:ss]
5 = Message ID	[ASCII characters] (DWORD, decimal)
6 = Reference text available	[YES, NO]
7 = Criteria analysis exists	[YES, NO]
8 = Message index (1 = 1. message)	[ASCII characters]
9 = Message HTML file	[ASCII characters]

Example PVM1

ProVi errors from module 3 in control unit 0.

The 2nd message is being output. The clock frequency is to be 2 seconds.

FI command		**00_BR_PVM4_1_3_2000**
Line	**Column**	**Answer**
1	1	Oil pressure too low
	2	124
	3	01.27.2000
	4	15:03:10
	5	98234039
	6	NO
	7	YES
	8	2
	9	

8.35 Download of PLC Retain Variables: PVR

MWMX device groups

Designation **PVR** **P**LC **V**ariable **R**etain Backup

Explanation Download of PLC retain variables.

FI command **BW_PVR1!(1)** **(Single Write)**

(1) = Download file with path details.

Note: File and path details must be enclosed in inverted commas.
The separator "!" is used in this command.

Response Structure The response to the "PVR1" FI command consists of three lines, each with one column. The meaning of the elements is as follows:

- Line 1 = Job ID [01...20]
  (see Chapter "FI Commands for the MPCX Device Group: IFJ").
- Line 2 = FI command
  [String, in accordance with Chapter "Elements of the FI Command"]
- Line 3 = FI Job Error Code
  (see Chapter "Error Codes")

Example PVR1 00_BW_PVR1!"D:\Program Files\Indramat\Mtgui\Temp\download.ini"/3

FI command		**00_BW_PVR1!"D:\Program Files\Indramat\Mtgui\Temp\download.ini"/3**
Line	**Column**	**Answer**
1	1	01
2	1	00_BW_PVR1!"D:\Program Files\Indramat\Mtgui\Temp\download.ini" /3
3	1	0

Structure of Download File The structure of the "download.ini" file used in this example corresponds to an Ini file in Windows.

Note: Care must be taken in the use of upper and lower case letters.

8.36 Upload of PLC Retain Variables: PVR

MWMX device groups

Designation **PVR** **P**LC **V**ariable **R**etain Backup

Explanation PLC retain variables are uploaded via all active processes.

FI command **BR_PVR1!(1)** **(Single Read)**

(1) = Upload file with path details

Note: Enclose file and path details in inverted commas.
The separator "!" is used in this command.

Response Structure

The response to the "PVR1" FI command consists of three lines, each with one column. The meaning of the elements is as follows:

- Line 1 = Job ID [01...20]
  (see Chapter "FI Commands for the MPCX Device Group", IFJ).
- Line 2 = FI command
  [String, in accordance with Chapter "Elements of the FI Command"]
- Line 3 = FI Job Error Code (see Chapter "Error Codes")

Example PVR

00_BR_PVR1_"D:\Program Files\Indramat\Mtgui\Temp\Upload.ini"/3

FI command		**00_BR_PVR1!"D:\Program Files\Indramat\Mtgui\ Temp\upload.ini"/3**
Line	**Column**	**Answer**
1	1	01
2	1	00_BR_PVR1!"D:\Program Files\Indramat\Mtgui\ Temp\upload.ini" /3
3	1	0

Structure of Upload File

The upload file is structured in the Windows – "Ini" format structure.

Note: Care must be taken in the use of upper and lower case letters.

8.37 Reading the PLC Variable Declaration: PVT

MWMX device groups

Designation

PVT **P**LC **V**ariable **T**ype

Explanation

A PLC variable has a particular type. To evaluate complex variables such as structures and arrays, their components and types must be read out.

Refer also to PVF, Reading Structured PLC Variables.

FI command

Read the PLC variable type.

BR_PVT_(1) **(Single Read)**

(1) = Identifier of the PLC variable [acc. to declaration part of the PLC]

Response Structure

One line with 2 columns is output for each element of the variables.

Line 1...n:	**Column 1**	**Column 2**

n = number of elements.

Value Range/Meaning of Columns

1 = Identifier of the PLC variable [acc. to declaration part of the PLC]

2 = Type [see value range PVF]

Examples:
PLC: Reading of a variable

Assumption:
The "TEST" variable is declared as WORD in the PLC program.

FI command	**00_BR_PVT_TEST**	
Answer		
Line	**Column 1 (Name)**	**Name**
1	TEST	WORD

WinPCL: Reading a Variable

Assumption:
The WinPCL variable "TEST" is declared as WORD in WinPCL program entity "Prog".

FI command	**00_BR_PVT_:Prog.TEST**	
Answer		
Line	**Column 1 (Name)**	**Name**
1	TEST	WORD

PLC: Reading a Structure

Assumption:
The "TEST1" variable is declared as STRUCT in the PLC program.

```
STRUCT
        E1      BOOL
        E2      INT
        E3      SINT
END
```

FI command	**00_BR_PVT_TEST1**	
Answer		
Line	**Column 1**	**Column 2**
1	TEST1.E1	BOOL
2	TEST1.E2	INT
3	TEST1.E3	SINT

WinPCL: Reading a Structure

Assumption:

The WinPCL variable "TEST1" is declared as STRUCT in WinPCL program entity "Prog".

```
STRUCT
        E1      BOOL
        E2      INT
        E3      SINT
END
```

FI command	**00_BR_PVT_:Prog.TEST1**	
Answer		
Line	**Column 1**	**Column 2**
1	TEST1.E1	BOOL
2	TEST1.E2	INT
3	TEST1.E3	SINT

PLC: Reading an Array

Assumption:
The "TEST2" variable is declared as ARRAY in the PLC program.

ARRAY [
    0 .. 3
] OF   BOOL

FI command	00_BR_PVT_TEST2	
Answer		
Line	**Column 1**	**Column 2**
1	TEST2[0]	BOOL
2	TEST2[1]	BOOL
3	TEST2[2]	BOOL
4	TEST2[3]	BOOL

WinPCL: Reading an Array

Assumption:
The WinPCL variable "TEST2" is declared as ARRAY in WinPCL program entity "Prog".

ARRAY [
    0 .. 3
] OF   BOOL

FI command	00_BR_PVT_:Prog.TEST2	
Answer		
Line	**Column 1**	**Column 2**
1	TEST2[0]	BOOL
2	TEST2[1]	BOOL
3	TEST2[2]	BOOL
4	TEST2[3]	BOOL

PLC: Reading an Array of a Structure

Assumption:
The "TEST3" variable is declared as ARRAY in the PLC program.

ARRAY [
    0 .. 1
] OF   STRUCT1,

where STRUCT1 is declared as follows:

STRUCT
    E1   BOOL
    E2   INT
    E3   SINT
END

END

FI command	00_BR_PVT_TEST3	
Answer		
Line	**Column 1**	**Column 2**
1	TEST3[0].E1	BOOL
2	TEST3[0].E2	INT
3	TEST3[0].E3	SINT
1	TEST3[1].E1	BOOL
2	TEST3[1].E2	INT
3	TEST3[1].E3	SINT

WinPCL: Reading an Array of a Structure

Assumption:
The WinPCL variable "TEST3" is declared as ARRAY in WinPCL program entity "Prog".

```
ARRAY [
        0 .. 1
] OF    STRUCT1,
```

where STRUCT1 is declared as follows:

```
STRUCT
        E1      BOOL
        E2      INT
        E3      SINT
END
```

FI command	00_BR_PVT_:Prog.TEST3	
Answer		
Line	**Column 1**	**Column 2**
1	TEST3[0].E1	BOOL
2	TEST3[0].E2	INT
3	TEST3[0].E3	SINT
1	TEST3[1].E1	BOOL
2	TEST3[1].E2	INT
3	TEST3[1].E3	SINT

Assumption:
The data types are output according to IEC1131.

See also command PVF.

8.38 SFC Diagnosis Data: SDD

MWMX device group

Designation **SDD** **S**FC **D**iagnosis **D**ata

Explanation Data for step chain diagnosis is output. Depending on the FI command this data can concern disrupted steps, actions, transitions or a definite ID to display the action or transition.

FI command Output the disrupted step of a step chain.

BR_SDD1!(1)!(2) **(Single Read)**

(1) = Module number	[1...99]
(2) = SFC entity name	[ASCII characters]

Note: The separator "!" is used in this command.

Response Structure The following table shows the general structure of the FI command "SDD1".

Line 1	Column 1	...	Column 7

Meaning of the Columns

1 = Step name	[ASCII characters]
2 = Detail type	[1 = action block, 2 =action network, 3 = transition]
3 = Detail name	[ASCII characters]
4 = POU ID	[ASCII characters]

5 = Detail morpheme	[ASCII characters] (DWORD, decimal)
6 = Error ID	[ASCII characters] (DWORD, decimal)
7 = POU entity name	[ASCII characters]

Example SDD1

Query disrupted step of the "clamp" chain in module 3 in control unit 0.

FI command		**00_BR_SDD1!3!Station03A.Clamp**
Line	**Column**	**Answer**
1	1	Open
	2	1
	3	Aopen
	4	SFC_1_2
	5	98243823
	6	34985304
	7	Station2.Module3

FI command

Output the faulty action, monitor error or transition of a disrupted step.

BR_SDD2!(1)!(2)!(3)	**(Single Read)**
(1) = Module number	[1...99]
(2) = SFC entity name	[ASCII characters]
(3) = Step name	[ASCII characters]

Note: The separator "!" is used in this command.

Response Structure

The following table shows the general structure of the FI command "SDD2".

Line 1	**Column 1**	**...**	**Column 6**

Meaning of the Columns

1 = Detail type	[1 = action block, 2 =action network, 3 = transition]
2 = Detail name	[ASCII characters]
3 = POU ID	[ASCII characters]
4 = Detail morpheme	[ASCII characters] (DWORD, decimal)
5 = Error ID	[ASCII characters] (DWORD, decimal)
6 = POU entity name	[ASCII characters]

Example SDD2

Query faulty action of the disrupted step "open" of the "clamp" chain in module 3 in control unit 0.

FI command		**00_BR_SDD1!3!Station03A.Clamp_Open**
Line	**Column**	**Answer**
1	1	1
	2	AOpen
	3	SFC_1_2
	4	98243823
	5	34985304
	6	Station2.Module3

FI command

Output the definite ID to display the action, monitor error or transition.

BR_SDD3!(1)!(2)!(3)!(4)	**(Single Read)**
(1) = Module number	[1...99]

(2) = SFC entity name [ASCII characters]

(3) = Detail type [1 = action block, 2 = action network, 3 = transition]

(4) = Detail name [ASCII characters]

Note: The separator "!" is used in this command.

Response Structure

The following table shows the general structure of the FI command "SDD3".

Line 1	Column 1	...	Column 4

Meaning of the Columns

1 = POU ID	[ASCII characters]
2 = Detail morpheme	[ASCII characters] (DWORD, decimal)
3 = Error ID	[ASCII characters] (DWORD, decimal)
4 = POU entity name	[ASCII characters]

Example SDD3

Query ID to display the action "aOpen" of the "clamp" chain in module 3 in control unit 0.

FI command		**00_BR_SDD3!3!Station03A.Clamp!1!aOpen**
Line	**Column**	**Answer**
1	1	SFC_1_2
	2	98243823
	3	34985304
	4	Station2.Module3

FI command

Output the I/O addresses to display a detail.

BR_SDD4!(1)!(2)!(3)!(4) **(Single Read)**

(1) = Module number [1...99]

(2) = SFC entity name [ASCII characters]

(3) = Detail type [1 = action block, 2 = action network, 3 = transition]

(4) = Detail name [ASCII characters]

Note: The separator "!" is used in this command.

Response Structure

The following table shows the general structure of the FI command "SDD4".

Line 1-n	Column 1	Column 2

Meaning of the Columns

1 = Variable morpheme	[ASCII characters] (DWORD, decimal)
2 = I/O address	[ASCII characters]

Example SDD4

Query I/O addresses to display the action "aOpen" of the "clamp" chain in module 3 in control unit 0.

Three variables have an I/O address.

FI command		**00_BR_SDD4!3!Station03A.Clamp!1!aOpen**
Line	**Column**	**Answer**
1	1	98243823
	2	%I3.2.0
2	1	40923423
	2	%Q23.21.7
3	1	34985304
	2	%I100.3.5

FI command

Determine the multilingual comments for displaying a detail.

BR_SDD5!(1)!(2)!(3)!(4) (Single Read)

(1) = Module number	[1...99]
(2) = SFC entity name	[ASCII characters]
(3) = Detail type	[1 = action block, 2 =action network, 3 = transition]
(4) = Detail name	[ASCII characters]

Note: The separator "!" is used in this command.

Response Structure

The following table shows the general structure of the FI command "SDD5".

Line 1-n	**Column 1**	**Column 2**

Meaning of the Columns

1 = Comment morpheme	[ASCII characters] (DWORD, decimal)
2 = New comment	[ASCII characters]

Example SDD5

Query comments to display the action "aOpen" of the "clamp" chain in module 3 in control unit 0.

Two comments are replaced by another text.

FI command		**00_BR_SDD5!3!Station03A.Clamp!1!aOpen**
Line	**Column**	**Answer**
1	1	98243823
	2	Clamp open
2	1	40923423
	2	Clamp closed

FI command

Output the action that has not been performed, or the transition of a step calculated based on the online status.

BR_SDD6!(1)!(2)!(3) (Single Read)

(1) = Module number	[1...99]
(2) = SFC entity name	[ASCII characters]
(3) = Step name	[ASCII characters]

Note: The separator "!" is used in this command.

Response Structure

The following table shows the general structure of the FI command "SDD6".

Line 1	Column 1	...	Column 6

Meaning of the Columns

1 = Detail type	[1 = action block, 3 = transition]
2 = Detail name	[ASCII characters]
3 = POU ID	[ASCII characters]
4 = Detail morpheme	[ASCII characters] (DWORD, decimal)
5 = Error ID	[ASCII characters] (DWORD, decimal)
6 = POU entity name	[ASCII characters]

Example SDD6

Query the action that has not been performed for the step "open" of the "clamp" chain in module 3 in control unit 0.

FI command		**00_BR_SDD6!3!Station03A.Clamp_Open**
Line	**Column**	**Answer**
1	1	1
	2	AOpen
	3	SFC_1_2
	4	98243823
	5	34985304
	6	Station2.Module3

FI command

Determine the module number of a step chain.

BR_SDD7!(1)	**(Single Read)**
(2) = SFC instances name	[ASCII characters]

Note: The separator "!" is used in this command.

Response Structure

The following table shows the general structure of the FI command "SDD7".

Line 1	Column 1

Meaning of the Columns

1 = Module number [1...99]

Example SDD7

Inquiry of the module of the chain "clamp".

FI command		**00_BR_EXD1!Station03A.Clamp**
Line	**Column**	**Answer**
1	1	3

Rexroth
Bosch Group

8.39 Set the Device Status Information: SDS

MWMX device groups

Designation **SDS** Set **D**evice **S**tatus

Explanation By this command, the device status information can be set; here, the configuration file IND_DEV.INI is adjusted as well.

Note: When this command is transmitted, the following system messages are generated:
MSG_DEVICEOFF or MSG_DEVICE_ON !

FI command With this command, the device status information of **ALL** defined devices can be set.

BW_SDS1_(1)	**(Single Write)**
(1) = Device status information to be set	0 = Device status information OFF 1 = Device status information ON

Response Structure The following table shows the general structure of the response to the "SDS1" FI command.

Line 1	Column 1

Value Range/Meaning of Columns

1 = Status report [(P_ACK)]

Example SDS1 Set device status information to OFF for **ALL** defined devices.

FI command		**00_BW_SDS1_0**
Line	**Column**	**Answer**
1	1	(P_ACK)

FI command With this command, the device status information for a selected device can be set.

BW_SDS2_(1)	**(Single Write)**
(1) = Device status information to be set	0 = Device status information OFF 1 = Device status information ON

Response Structure The following table shows the general structure of the response to the "SDS2" FI command.

Line 1	Column 1

Value Range/Meaning of Columns

1 = Status report [(P_ACK)]

Example: SDS2 Set device status information to OFF for the selected device 00.

FI command		**00_BW_SDS2_0**
Line	**Column**	**Answer**
1	1	(P_ACK)

8.40 Setting the FI Exclusive Mode: SEM

MWMX device group

Designation **SEM** **S**et FI **E**xclusive **M**ode

Explanation This command is used to activate FI Exclusive mode for the selected device address.

FI Exclusive mode: In this mode, **ALL** the processes logged in at the FI – **with the exception** of the process issuing the SEM command – are blocked from data communication with this device address. This mode is used for example for data communication which must NOT be interrupted by other data requests. However, it is **imperative** that this FI Exclusive mode is deleted once more through the DEM command.

FI command **BW_SEM1** **(Single Write)**

Response Structure The following table shows the general structure of the response to the FI command "BW_SEM1". A line of 1 column is output.

Line 1	Column 1

Value Range/Meaning of Columns 1 = Status message (P_ACK) (P_ACK)

Example SEM1 Activate FI Exclusive mode for device address 0.

FI command		**00_BW_SEM1**
Line	**Column**	**Answer**
1	1	(P_ACK)

8.41 Sequencer Data: SFD

MWMX device groups

Designation **SFD** **SF**C **D**ata

Explanation Data for a step chain is outputted. Depending on the FI command this can concern a step chain comment, POE name, step comment, maximum time, action / transition / monitor error name (comment), qualifier and time value.

FI command Query the data for a step chain.

BR_SFD1!(1)!(2)	**(Single Read)**
(1) = Module number	[1...99]
(2) = SFC entity name	[ASCII characters]

Note: The separator "!" is used in this command.

Response Structure The following table shows the general structure of the "SFD1" FI command.

Line 1	Column 1	Column 2

Meaning of the Columns

1 = Step chain comment	[ASCII characters]
2 = POE name	[ASCII characters]

Example SFD1

Query data of the "clamp" chain in module 3 in control unit 0.

FI command		**00_BR_SFD1!3!Station03A.Clamp**
Line	**Column**	**Answer**
1	1	Clamping device
	2	CLAMP

FI command

Query the data of a step.

BR_SFD2!(1)!(2)!(3)	**(Single Read)**
(1) = Module number	[1...99]
(2) = SFC entity name	[ASCII characters]
(3) = Step name	[ASCII characters]

Note: The separator "!" is used in this command.

Response Structure

The following table shows the general structure of the "SFD2" FI command. The number of lines depends on the number of actions and transitions.

If there are no details the line number is 1.

Line 1	**Column 1**	**...**	**Column 3**
Line 2...n:	**Column 1**	**...**	**Column 6**

Meaning of the Columns

Line 1

1 = Step comment	[ASCII characters]
2 = Maximum time	[ASCII characters]
3 = Minimum time	[ASCII characters]

Line 2...n:

1 = Detail type	[1 = action block, 3 = transition]
2 = Name	[ASCII characters]
3 = Comment	[ASCII characters]
4 = Boolean variable	[YES, NO]
5 = Qualifier	[ASCII characters]
6 = Time value	[ASCII characters]

Example SFD2

Data for the step "Open" in the "clamp" chain in module 3 on control unit 0.

FI command		**00_BR_SFD2!3!Station03A.Clamp!Open**
Line	**Column**	**Answer**
1	1	Open clamping device
	2	T#5s
	3	
2	1	1
	2	aOpen
	3	Clamp open
	4	NO
	5	D
	6	T#3s

FI command		**00_BR_SFD2!3!Station03A.Clamp!Open**
Line	**Column**	**Answer**
3	1	3
	2	tOpen
	3	Clamping device is open
	4	NO
	5	
	6	

FI command

Output the data for a detail.

BR_SFD3!(1)!(2)!(3)!(4)	**(Single Read)**
(1) = Module number	[1...99]
(2) = SFC entity name	[ASCII characters]
(3) = Detail type	[1 = action block, 2 = action network, 3 = transition]
(4) = Detail name	[ASCII characters]

Note: The separator "!" is used in this command.

Response Structure

The following table shows the general structure of the "SFD3" FI command.

Line 1	**Column 1**	**Column 2**

Meaning of the Columns

1 = Comment	[ASCII characters]
2 = Boolean variable	[YES, NO]

Example SFD3

Data for the action "aOpen" in the "clamp" chain in module 3 in control unit 0.

FI command		**00_BR_SFD3!3!Station03A.Clamp!aOpen**
Line	**Column**	**Answer**
1	1	Clamp open
	2	NO

8.42 Sequencer Messages: SFE

MWMX device groups

Designation **SFE** **S**FC **E**rror

Explanation The sequencer messages of a module are output.

FI command

Output all SFC messages.

For reasons of optimization, not all data is called up. Accordingly, the Diagnosis server must be informed that the data is required (see ADW).

BR_SFE1_(1)	**(Single Read)**
BC_SFE1_(1)	**(Cyclic Read)**
(1) = Module number	[1...99]

Output first SFC messages.

BR_SFE2_(1)	**(Single Read)**
BC_SFE2_(1)	**(Cyclic Read)**
(1) = Module number	[1...99]

Response Structure

The following table shows the general structure of the FI commands "SFE1" and "SFE2". The number of lines depends on the number of messages pending.

If there are no messages, the number of lines is 0.

Line 1...n:	Column 1	...	Column 7

Meaning of the Columns

1 = Message text	[ASCII characters]
2 = SFC entity name	[ASCII characters]
3 = Step name	[ASCII characters]
4 = Time stamp day	[mm.dd.yyyy]
5 = Time stamp time	[hh:mm:ss]
6 = Type of error	[1 = time error, 2 = monitor error, 3 = monitor event]
7 = Is there condition analysis?	[YES, NO]

Example SFD1

All SFC messages from module 2 in control unit 0.

There are two messages:

FI command		**00_BR_SFE1_2**
Line	**Column**	**Answer**
1	1	TIME ERROR: Chain: chucking Step: up malfunction
	2	Station03A.Clamp
	3	Open
	4	01.27.2000
	5	11:56:32 AM
	6	1
	7	YES
2	1	ASSY ERROR: Chain: drilling Step: down malfunction
	2	Station02A.Drill
	3	Down
	4	01.27.200
	5	13:03:12
	6	2
	7	NO

Example SFE2

First SFC message from module 2 in control unit 0.

There are two messages.

FI command		**00_BR_SFE2_2**
Line	**Column**	**Answer**
1	1	TIME ERROR: Chain: chucking Step: up malfunction
	2	Station03A.Clamp
	3	Open
	4	01.27.2000
	5	14:56:32
	6	1
	7	YES

8.43 Sequencer Mode: SFM

MWMX device groups

Designation

SFM **SF**C **M**ode

Explanation

Queries step chain mode.

FI command

Query the mode of a step chain.

BR_SFM1!(1)!(2)	**(Single Read)**
BC_SFM1!(1)!(2)	**(Cyclic Read)**
(1) = Module number	[1...99]
(2) = SFC entity name	[ASCII characters]

Note: The separator "!" is used in this command.

Response Structure

The following table shows the general structure of the "SFM1" FI command.

Line 1	**Column 1**

Meaning of the Columns

1 = Mode	[1 = time error, 2 = monitor error, 3 = monitor event, 10 = stop, 11 = auto, 12 = manual, 13 = jog]

Example SFM1

Query mode of the "clamp" chain in module 3 in control unit 0.

FI command		**00_BR_SFM1!3!Station03A.Clamp**
Line	**Column**	**Answer**
1	1	1

8.44 Software Installation Data: SID

MWMX device groups

Designation **SID** **S**oftware **I**nstallation **D**ata

Explanation Information is returned regarding installation. This information includes installation paths, the software version used, DLL mode, plus service pack and release information.

FI command Read-in the installation data and/or the software version data

BR_SID1 **(Single Read)**

Response Structure One line with 8 columns is output for the returned values.

Line 1	**Column 1**	**...**	**Column 16**

Meaning of the Columns

1 = Basic directory	[EXE files of the BOF]
2 = FI installation directory	[FI directory]
3 = Data directory	[in accordance with BOF]
4 = GBO version	[from INDRAMAT.ini]
5 = IF-DLL mode	[from INDRAMAT.ini]
6 = IF version	[from INDRAMAT.ini - from DLL mode 400]
7 = Service package info	[from INDRAMAT.ini - from DLL mode 420]
8 = Release info	[from INDRAMAT.ini - from DLL mode 420]
9 = IF-Build-Info	[in accordance with Build process]
10 = Current context name	[in accordance with the installation]
11 = Physical installation path	[in accordance with the installation]
12 = Complete IF version indication string	
13 = WinPCL build number	[in accordance with WinPCL]
14 = Version number of the PLC compiler	[in accordance with WinPCL]
15 = Version number of the PLC linker	[in accordance with WinPCL]
16 = Version number of the PLC data basis	[in accordance with WinPCL]
17 = Platform version	

Example SID1 Return information on the current installation.

FI command		**00_BR_SID1**
Line	**Column**	**Answer**
1	1	--
	2	D:\Program Files\Indramat\MTGUI\Bin
	3	--
	4	005-22Vxx
	5	07.00
	6	07V00
	7	--
	8	--
	9	Build 3124 Mar 6 2003 08:53:55
	10	MTGUI_0-23T01 B3327
	11	D:\Program Files\Indramat\MTGUI\
	12	FI: 07V00 DLL-Mode: 07.20 Build 3124 Mar 6 2003 08:53:55
	13	347.15.4.11
	14	771
	15	515
	16	78
	17	Platform: 02V01 Build: 3214

Note: Refer to FI command "PHD" of the MPCX for working with absolute paths.

8.45 PLC Long Identification: SLI

MWMX device groups

Designation **SLI** PLC (**S**PS) **L**ong **I**dentification

Explanation Returns the unit data from the PLC long identification.

FI command Read PLC long identification.

BR_SLI **(Single Read)**

Response Structure One line with 15 columns is output for the returned values.

Line 1	**Column 1**	**Column...**	**Column 15**

Value Range/Meaning of the Columns

1 =	device address	[00...15]
2 =	program number	[01...99]
3 =	Project name	[max. 8 ASCII characters]
4 =	Program name	[max. 8 ASCII characters]
5 =	User name	[acc. to password entry]
6 =	Program length	[bytes]
7 =	Compilation time	[LONG] (coded in long value)
8 =	Compilation date	[8 ASCII characters]
9 =	Compilation time	[8 ASCII characters]
10 =	Download time	[LONG] (coded in long value)
11 =	Download date	[8 ASCII characters]1
12 =	Download time	[8 ASCII characters]
13 =	Version of PLC long identification	[LONG]
14 =	RUN flags	[HEX value]
15 =	Compiler info	[LONG]

Example SLI Read the unit data from the PLC long identification.

FI command		**00_BR_SLI**
Line	**Column**	**Answer**
1	1	02
	2	01
	3	--
	4	MOT12
	5	TEST
	6	17672
	7	630163960
	8	15.12.99
	9	17:15:48
	10	630163961
	11	15.12.99
	12	17:15:50
	13	2
	14	0x0000
	15	13

Reference to Literature see chapter entitled "Literature" [30].

8.46 Reading and Writing Visual Motion Data: VMD

MWMX device group

Designation **VMD** **V**isual **M**otion **D**ata

Explanation This FI command is used to read out and to write all kind of Visual Motion data. To identify the Visual Motion data to be written or to describe the Visual Motion data requested, the ASCII communication description is applicable.

FI command **BR_VMD1_(1)** **(Single Read)**

(1) = ASCII request string according to
the ASCII communication description

Response Structure The following table shows the general structure of the response to the FI command "BR_VMD1". A line of 1 column is output.

Line 1	**Column 1**

Value Range/Meaning of Columns 1 = Read Visual Motion data string Is supplied as an ASCII string

Example VMD1 Read the drive status message. The ASCII request string can also be written in inverted commas (00_BR_VMD1_"DP 1.95").

FI command		**00_BR_VMD1_DP 1.95**
Line	**Column**	**Answer**
1	1	302 Position Mode Encoder 1

FI command	**BW_VMD1_(1)** (1) = ASCII request string according to the ASCII communication description and the value to be written	**(Single Read)**

Response Structure The following table shows the general structure of the response to the FI command "BW_VMD1". A line of 1 column is output.

Line 1	Column 1

Value Range/Meaning of Columns 1 = Status message (P_ACK) (P_ACK)

Example VMD1 Set the KV parameter for drive 1. The ASCII request string and the value to be written can also be written in inverted commas (00_BW_VMD1_"DP 1.104 1.00").

FI command		00_BW_VMD1_DP 1.104 1.00
Line	**Column**	**Answer**
1	1	(P_ACK)

8.47 Requesting Watch List Allocations: WLA

MWMX device groups

Designation **WLA** **W**atch **L**ist **A**llocation

Explanation Requests free watch list allocations. A maximum of ten free watch list allocations can be requested with one FI command.

BR_WLA1_(1)	**(Single Read)**
(1) =Number of requested free watch list allocations	The required number of free watch list allocations is identified here. The allowed value range: 1..10.

Response Structure The following table shows the general structure of the response to the FI command "WLA1".

Line 1	Column 1	...	Column n

Value Range/Meaning of Columns

1 =	1. free watch list allocation	Value range: 1..15
2 =	2. free watch list allocation	Value range: 1..15
3 =	3. free watch list allocation	Value range: 1..15
n =	nth free watch list allocation	Value range: 1..15

Example WLA1

Request four free watch list allocations.

Assumption:
Watch list allocations 3 and 5 are already assigned!

FI command		**00_BR_WLA1_4**
Line	**Column**	**Answer**
1	1	0
	2	1
	3	2
	4	4

8.48 Freeing Watch List Allocations: WLF

MWMX device groups

Designation

WLF **W**atch **L**ist **F**ree

Explanation

Previously requested watch list allocations are freed again.

FI command

Free ALL assigned watch list allocations for the selected device.

BR_WLF1 **(Single Read)**

Note: The FI command "WLF1" frees ALL assigned watch list allocations, including those of other WIN32 applications.

Response Structure

The following table shows the general structure of the response to the FI command "WLF1".

Line 1	**Column 1**	**...**	**Column n**

Value Range/Meaning of Columns

1 =	1. freed watch list allocation	Value range: 1..15
2 =	2. freed watch list allocation	Value range: 1..15
3 =	3. freed watch list allocation	Value range: 1..15
n =	nth freed watch list allocation	Value range: 1..15

Example WLF1

Free ALL assigned watch list allocations.

Assumption:
The following watch list numbers have been allocated: 0,1, 2, 3.

FI command		**00_BR_WLF1**
Line	**Column**	**Answer**
1	1	0
	2	1
	3	2
	4	3

FI command

Free the required watch list allocations for a selected device.

BR_WLF2_(1)_{(2)..(10)}	**(Single Read)**
(1)..(10) = List of watch list allocations to be released	A maximum of 10 watch list allocations can be transferred here to be freed again.

Response Structure

The following table shows the general structure of the response to the FI command "WLF2".

Line 1	Column 1	...	Column n

Value Range/Meaning of Columns

1 =	1. freed watch list allocation	Value range: 1..15
2 =	2. freed watch list allocation	Value range: 1..15
3 =	3. freed watch list allocation	Value range: 1..15
n =	nth freed watch list allocation	Value range: 1..15

Example WLF2

Free required watch list allocations:
Assumption: Watch list allocations 0,3,4 and 8 have first been requested using the FI command "WLA1".

FI command		**00_BR_WLF2_0_3_4_8**
Line	**Column**	**Answer**
1	1	0
	2	3
	3	4
	4	8

9 FI Commands - MWSX Device Group (ISP 200)

The FI Commands described in this chapter are valid for the MWSX device group. In this device group, the following types are listed as well as possible device addresses:

Group	Accompanying Types	Address
MWSX	ISP200-P-G2, ISP200-R-G2, ISP200-I-G2	[00...63]

Note: Please note that the device address must be set before the respective FI command, e.g. 00_BR_ASM1 (refer also here to the chapter 2.1 "Elements of the FI Command").

9.1 Active Diagnosis Window ADW

MWSX device group

Designation **ADW** Active **D**iagnosis **W**indow

Explanation Indicates the window types for which data is required.

For improved performance and with some diagnosis windows, not all the data is called up each time a diagnosis is performed, but only when the data is actually required.

Through this FI command, the diagnosis server can be informed that the data of the respective window type is required.

This command has to be issued at least every 3 seconds while the data is required. If this command is not issued any more, calling-up of all data will stop.

FI command Indicates the window types for which data is required.

BW_ADW1_(1){_(2)}	**(Single Write)**
(1) = Type of diagnosis window	[1 = CNC error, 2 = sequence errors, 3 = general errors, 4 = messages, 10 = start requirements, 11 = warnings, 12 = setup diagnosis]
(2) = Module number	[1...99] ! only for window type 1 -4!

Example ADW1 Call up data for CNC error in controller 0 module 1.

FI command	**00_BW_ADW1_1_1**

9.2 Active System Error Messages: ASM

MWSX device groups

Designation **ASM** Active **S**ystem **M**essages

Explanation The active device information is output (system errors, device statuses) that affect the functioning of the entire electrical device. Depending on the FI command, the device address, device name, message number, type of message, short text and reference text are all output.

FI command Output of the currently pending device information (system errors, device statuses) of all active devices from the MWSX device group.

BR_ASM1 **(Single Read)**

BC_ASM1 **(Cyclic Read)**

BB_ASM1 **(Break Cyclic Read)**

Note: The "ASM1" FI command refers to all devices within this device group. Therefore, any valid device address can be indicated in the command line (see example "ASM1").

Response Structure The following table shows the general structure of the response to the FI command "ASM1". The number of lines (1 .. n=15) depends on the number of defined devices. Each line consists of 7 columns for the device address, device name, message number, message status, short text and indication of whether there is an reference text for this device information (system errors, device statuses).

Line 1...n	Column 1	...	Column 7

Value Range/Meaning of Columns

1 =	device address	[00...15]
2 =	Device name	[max. 32 ASCII characters]
3 =	Message number	[0...150]
4 =	Type of message	[F = fault/error, D = diagnosis]
5 =	Short text	[max. 54 ASCII characters]
6 =	Reference text	[x= exists, -- = does not exist]
7 =	2 bytes of additional information for the message number	is required to resolve the information "@" (see ASM5)
8 =	File name for additional information for notification text	e.g. in HTML format

Example ASM1 Read the current system error messages of all defined devices of the MWSX device group.

Assumption: The following three devices are defined:

- Device address 01,
- Device address 07 and
- Device address 10.

FI command		07_BR_ASM1
Line	**Column**	**Answer**
1	1	01
	2	Drill center
	3	71
	4	F
	5	PLC battery voltage too low.
	6	X
	7	0
	8	0
2	1	07
	2	Drill station 1
	3	74
	4	F
	5	SLM time monitoring
	6	X
	7	0
	8	
3	1	10
	2	Drill station 2
	3	1
	4	D
	5	Error has been corrected.
	6	X
	7	0
	8	

FI command Output the currently pending system error message of the selected device from the MWSX device group.

BR_ASM2 **(Single Read)**

BC_ASM2 **(Cyclic Read)**

BB_ASM2 **(Break Cyclic Read)**

Response Structure The following table shows the general structure of the response to the FI command "ASM2". The answer consists of a line of 7 columns for the device address, device name, message number, message status, short text and indication of whether there is an reference text for this device information (system errors, device statuses).

Line 1...n	Column 1	...	Column 8

Value Range/Meaning of Columns

1 =	device address	[00...15]
2 =	Device name	[max. 32 ASCII characters]
3 =	Message number	[0...150]
4 =	Type of message	[F = fault/error, D = diagnosis]
5 =	Short text	[max. 54 ASCII characters]
6 =	Reference text	[x= exists, -- = does not exist]
7 =	2 bytes of additional	is required to resolve the

	information for the message number	information "@" (see ASM5)
8 =	File name for additional information for notification text	e.g. in HTML format

Example ASM2

Read the current device information (system errors, device statuses) of device address 01.

Assumption:
The following three devices are defined:

- Device address 01
- Device address 07 and
- Device address 10

FI command		**01_BR_ASM2**
Line	**Column**	**Answer**
1	1	01
	2	Drill center
	3	71
	4	F
	5	PLC battery voltage too low.
	6	X
	7	0
	8	

FI command

Output the current device information (system errors, device statuses) of the device listed from the MWSX device group.

BR_ASM3_(1) **(Single Read)**

BC_ASM3_(1) **(Cyclic Read)**

BB_ASM3_(1) **(Break Cyclic Read)**

(1) = Selection list for a max. of 10 MWSX devices [00_01_02_ ... _15]

Response Structure

The following table shows the general structure of the response to the FI command "ASM3". The number of lines (1 .. n=15) depends on the number of listed MWSX devices. Each line consists of 7 columns for the device address, device name, message number, message status, short text and indication of whether there is an reference text for this error message.

Line 1...n	Column 1	...	Column 8

Value Range/Meaning of Columns

1 =	device address	[00...15]
2 =	Device name	[max. 32 ASCII characters]
3 =	Message number	[0...150]
4 =	Type of message	[F = fault/error, D = diagnosis]
5 =	Short text	[max. 54 ASCII characters]
6 =	Reference text	[x= exists, -- = does not exist]
7 =	2 bytes of additional information for the message number	is required to resolve the information "@" (see ASM5)
8 =	File name for additional information for notification	e.g. in HTML format

text

Example ASM3

Read the current device information (system errors, device statuses) of the selected MWSX devices.

Assumption:
The following devices addresses are defined:

- Device address 01,
- Device address 07 and
- Device address 10.

FI command		**01_BR_ASM3_01_10**
Line	**Column**	**Answer**
1	1	01
	2	Drill center
	3	71
	4	F
	5	PLC - battery voltage too low
	6	X
	7	0
	8	0
2	1	10
	2	Drill center 2
	3	1
	4	D
	5	Error has been corrected.
	6	X
	7	0
	8	

FI command

Output the current device information (system errors, device statuses) of all defined devices (in accordance with the system configuration) from the MWSX device group.

BR_ASM4_(1)	**(Single Read)**
BC_ASM4_(1)	**(Cyclic Read)**
BB_ASM4_(1)	**(Break Cyclic Read)**
(1) = Device group	[MWSX]

Response Structure

The following table shows the general structure of the response to the FI command "ASM4". The number of lines (1 .. n=15) depends on the number of defined MWSX devices. Each line consists of 7 columns for the device address, device name, message number, message status, short text and indication of whether there is an reference text for this device information (system errors, device statuses).

Line 1...n	**Column 1**	**...**	**Column 8**

Value Range/Meaning of Columns

1 =	device address	[00...15]
2 =	Device name	[max. 32 ASCII characters]
3 =	Message number	[0...150]
4 =	Type of message	[F = fault/error, D = diagnosis]
5 =	Short text	[max. 54 ASCII characters]
6 =	Reference text	[x= exists, -- = does not exist]
7 =	2 byte additional information for the message number	is required to resolve the information "@" (see ASM5)
8 =	File name for additional information for notification text	e.g. in HTML format

Example ASM4

Read the current device information (system errors, device statuses) of all defined devices of the MWSX device group.

Assumption:
The following devices are defined:

- Device address 01 and
- Device address 10.

FI command		**01_BR_ASM4_MWSX**
Line	**Column**	**Answer**
1	1	01
	2	Drill center
	3	71
	4	F
	5	PLC battery voltage too low.
	6	X
	7	0
	8	
2	1	10
	2	Drill center 2
	3	1
	4	D
	5	Error has been corrected.
	6	X
	7	0
	8	

FI command

Output the additional text for the currently pending device information (system errors, device statuses) related to the device and the message number.

BR_ASM5_(1)_(2) **(Single Read)**

(1) = Message number [0...150]

(2) = 2 bytes of additional information for the message number

Response Structure

The following table shows the general structure of the response to the FI command "ASM5". The answer consists of a line with 5 columns for the device address, device name, message number and reference text.

Line 1...n	Column 1	...	Column 6

Value Range/Meaning of Columns

1 =	device address	[00...15]
2 =	Device name	[max. 32 ASCII characters]
3 =	Message number	[0...150]
4 =	Type of message	[F = fault/error, D = diagnosis]
5 =	Reference text	[max. 14 lines with a max. 78 characters/line]
6 =	File name for additional information for reference text	e.g. in HTML format

Example ASM5

Read the reference text relating to the system error with message number 74 of device address 01.

FI command		**01_BR_ASM5_74_0**
Line	**Column**	**Answer**
1	1	01
	2	Drill center
	3	74
	4	F
	5	Replace the SLM module on the PLC card (slot 3).
	6	

Reference to Literature

See chapter entitled "Literature" [13].

9.3 Creating MI Import Data: CMD

MWSX device group

Designation **CMD** Create **MI** Import **Data**

Explanation Creation of the data the Message Integrator requires for data import.

FI command Creation of the data of all ProVi messages.

BR_CMD1 **(Single Read)**

Response Structure The command does not return any answer. If no error is signaled, the respective files have been generated.

FI command Creation of the data of a certain ProVi message type.

BR_CMD2_(1){_(2)} **(Single Read)**

(1) = Message type	[1 = error, 2 = messages, 10 = warnings, 11 = start requirements, 12 = setup diagnosis]
(2) = Module number	[1...99] ! only for message type 1 -2!

Response Structure The command does not return any answer. If no error is signaled, the respective files have been generated.

FI command Creation of the data of all step chain messages.

BR_CMD3 **(Single Read)**

Response Structure The command does not return any answer. If no error is signaled, the respective files have been generated.

FI command Creation of the data of a certain step chain register.

BR_CMD4!(1)	**(Single Read)**
(1) = Register name	[ASCII characters]

Note: The separator "!" is used in this command.

Response Structure The command does not return any answer. If no error is signaled, the respective files have been generated.

9.4 Trigger Control Reset: CRT

MWSX device groups

Designation **CRT** **Control-Reset**

Explanation The control reset allows the selected device to be reset after a system error. If there is no system error at the selected device then the job is ignored.

CAUTION

Resetting causes a complete reinitialization of device.
A temporary communication error occurs during initialization.

FI command **BW_CRT** **(Single Write)**

Value to be written Trigger reset 0

Note: The value to be written is passed to the "acValue" parameter in the "DataTransfer" routine.

Response Structure The return value of the "DataTransfer" routine is [0] if the write procedure has been successfully completed. In the event of an error, more information in the form of a general error result line can be requested by the routine "ReadGroupItem" (refer here to chapter 8, "Error Codes" and "General Error Result Line").

Example CRT Trigger a control reset on the selected device.

FI command	**Value to be written: 0** **00_BW_CRT**
Value to be written	0

Reference to Literature See chapter entitled "Literature" [26].

9.5 Reading Device Component Information: DCI

MWSX device group

Designation **DCI** **D**evice **C**omponent **I**nformation

Explanation The current device component information is read out of the device. From the device component information, the user is provided with information on the components the addressed device is equipped with, and the firmware each component contains. The command will not file if no access to firmware is possible (e.g. while the device is in monitor mode). Instead, the failed access is reported through the firmware access status.

FI command Read the device component information.

BR_DCI1 **(Single Read)**

Response Structure The following table shows the general structure of the response to the FI command "BR_DCI1". For each device component available in the device, one line is returned. Each line consists of 11 columns.

FI command		**00_BR_DCI1**
Line	**Column**	**Answer**
1	1	PCB type
	2	Configured component type
	3	Detected component type
	4	Firmware access status; i.e. has an error occurred accessing the firmware, Yes/No? Valid range of values [YES/NO] In case of an error, the error cause can be defined from one the two following columns.
	5	Error class on accessing firmware identification: (see Error Class Definition under General Error Result Line)
	6	Error code on accessing firmware identification: (see Error Code Definition under Error Codes)
	7	Firmware identification
	8	Firmware version
	9	Firmware release
	10	Is the component address in column 11 a sub-address, Yes/No? Valid range of values [YES/NO]
	11	Component address
2	1	PCB type
	..	..
	11	Component address

Example DCI1 At device address 00, read out the current device component information.

FI command		**00_BR_DCI1**
Line	**Column**	**Answer**
1	1	CPU
	2	MTC-P
	3	MTC-P
	4	NO
	5	0
	6	0
	7	CPU06/0006-23V10
	8	23
	9	10
	10	NO
	11	1
2	1	PLC
	2	MTS-P01.2
	3	MTS-P01.2
	4	NO
	5	0
	6	0
	7	PLC06S-M05-06V05
	8	06
	9	05
	10	NO
	11	3
3	1	APR
	2	--
	3	APR-P
	4	NO
	5	0
	6	0
	7	APR06/0003-23T06
	8	23
	9	06
	10	NO
	11	4

Example DCI1 while booting is blocked

While booting is blocked (i.e. while the device is in monitor mode), read out the current device component information at device address 00.

FI command		**00_BR_DCI1**
Line	**Column**	**Answer**
1	1	CPU
	2	MTC-P
	3	MTC-P
	4	YES
	5	1
	6	2082
	7	--
	8	--
	9	--
	10	NO
	11	1
2	1	PLC
	2	MTS-P01.2
	3	MTS-P01.2
	4	YES
	5	1
	6	2082
	7	--
	8	--
	9	--
	10	NO
	11	3
3	1	APR
	2	--
	3	APR-P
	4	YES
	5	1
	6	2082
	7	--
	8	--
	9	--
	10	NO
	11	4

9.6 Setting the Communication Timeout Time DCT

MWSX device groups

Designation **DCT** **D**evice **C**ommunication **T**imeout

Explanation By means of this command, the timeout time for the selected device is set dynamically (timeout time in ms).

FI command **BW_DCT1_(1)** **(Single Write)**

(1) = requested timeout time in ms

Response Structure The response to the "DCT1" FI command consists of one line with one column.

Line 1	**Column 1**

Value Range/Meaning of Columns

1 = Status message (P_ACK) (P_ACK)

Example DCT1 For the device 00, the timeout time is set 1500 ms.

FI command		**00_BW_DCT1_1500**
Line	**Column**	**Answer**
1	1	(P_ACK)

FI command With this command, the timeout time for the selected device can be reset to default value.

BW_DCT2 **(Single Write)**

Response Structure The response to the "DCT2" FI command consists of one line with one column.

Line 1	**Column 1**

Value Range/Meaning of Columns

1 = Status message (P_ACK) (P_ACK)

Example DCP2 For the device 00, the timeout time is reset to the default value.

FI command		**00_BW_DCT2**
Line	**Column**	**Answer**
1	1	(P_ACK)

9.7 Deleting the FI Exclusive Mode: DEM

MWSX device group

Designation **DEM** **D**elete FI **E**xclusive **M**ode

Explanation This command is used to deactivate FI Exclusive mode for the selected device address.

FI Exclusive mode: In this mode, **ALL** the processes logged in at the FI – **with the exception** of the process issuing the SEM command – are blocked from data communication with this device address. This mode is used for example for data communication which must NOT be interrupted by other data requests. However, it is **imperative** that this FI Exclusive mode is deleted once more through the DEM command.

FI command **BW_DEM1** **(Single Write)**

Response Structure The following table shows the general structure of the response to the FI command "BW_DEM1". A line of 1 column is output.

Line 1	Column 1

Value Range/Meaning of Columns 1 = Status message (P_ACK) (P_ACK)

Example DEM1 Deactivate the FI Exclusive mode for device address 0. The FI Exclusive mode for device address 0 has previously been activated by the SEM command.

FI command		00_BW_DEM1
Line	**Column**	**Answer**
1	1	(P_ACK)

9.8 Static/Dynamic Device Information: DIF

MWSX device group

Designation **DIF** **D**evice **I**n**F**ormation

Explanation Static device information and network information is read according to the "IND_DEV.INI" and "FAR_DEV.INI" files.

FI command Reading of the static device information and network information of a selected device.

BR_DIF1 **(Single Read)**

BC_DIF1 **(Cyclic Read)**

BB_DIF1 **(Break Cyclic Read)**

Response Structure The following table shows the general structure of the response to the "DIF1" FI command. The response consists of one line with 24 columns.

Line 1	Column 1	...	Column 24

Value Range/Meaning of Columns

1 =	Local/far device address	[00..63]
2 =	Device name	IND_DEV.INI entry: DeviceName=
3 =	Device type	IND_DEV.INI entry: DeviceType=
4 =	PLC support	IND_DEV.INI entry: PLC=
5 =	Device status	IND_DEV.INI entry: DeviceStatus=
6 =	Assignment of a simulation pair	IND_DEV.INI entry: DeviceAssign=
7 =	Device mode	IND_DEV.INI entry: MtvncMode=
8 =	Communication channel	IND_DEV.INI entry: [CommAddrX]
9 =	Description of the communication channel	IND_DEV.INI entry: CommStr=
10 =	Timeout value	IND_DEV.INI entry: Timeout=
11 =	Device group	(see Chapter 6.1 "Identifier")
12 =	PLC component type	IND_DEV.INI entry: Component type1=
13 =	CNC component type	IND_DEV.INI entry: Component type2=
14 =	Device log	IND_DEV.INI entry: DeviceProtocol=
15 =	Device simulation	IND_DEV.INI entry: DeviceSimulation=
16 =	Not yet assigned	[--]
17 =	Not yet assigned	[--]
18 =	Not yet assigned	[--]
19 =	Not yet assigned	[--]
20 =	Network ON/OFF	[ON = Network active OFF = No network active]
21 =	Network name	Max. 28 ASCII characters
22 =	PC number	[00..99,XX]
23 =	PC name	Max. 255 ASCII characters
24 =	Local device address	[00..63]

Example DIF1 Read the static device information and network information of device 1 while the network is active.

FI command		01_BR_DIF1
Line	**Column**	**Answer**
1	1	01
	2	Handling station right side
	3	ISP200-R-G2
	4	YES
	5	ON
	6	NO
	7	OFF
	8	4
	9	V24,COM2,19200,EVEN,RS232,TCOFF
	10	3500
	11	MWMX
	12	MTS-R
	13	NONE
	14	CNC
	15	OFF
	16	--
	17	--
	18	--
	19	--
	20	ON
	21	PC network 1
	22	29
	23	BTV20-STATION-LEFT
	24	01

Explanation The dynamic device information and network information is read. The current data identifications are made available from the selected controller.

FI command Reading of the dynamic device information and network information of a selected device.

BR_DIF2 **(Single Read)**

BC_DIF2 **(Cyclic Read)**

BB_DIF2 **(Break Cyclic Read)**

Response Structure

The answer consists of 23 lines, each line having a specific meaning.

Line 1	Static device information
Line 2	Firmware information
Line 3	Current parameter set
Line 4	Current PLC program
Line 5	Current machine data set
Line 6	Current NC package for memory A
Line 7	Current NC package for memory B
Line 8	Current cycle package
Line 9	Current NC program name for process 0
Line 10	Current NC program name for process 1
Line 11	Current NC program name for process 2
Line 12	Current NC program name for process 3
Line 13	Current NC program name for process 4
Line 14	Current NC program name for process 5
Line 15	Current NC program name for process 6
Line 16	Current tool list for process 0
Line 17	Current tool list for process 1
Line 18	Current tool list for process 2
Line 19	Current tool list for process 3
Line 20	Current tool list for process 4
Line 21	Current tool list for process 5
Line 22	Current tool list for process 6
Line 23	Current I/O configuration table

Meaning of line 1

Line 1 returns the most significant static device information and network information and consists of 18 columns.

Line 1	Column 1	...	Column 18

Value Range/Meaning of Columns

1 =	Line number	[1]
2 =	Status information	Contains the information whether or not the subsequent data in this line is valid; the following applies: [0 = Data is invalid – further column results [--] 1 = Data is valid]
3 =	Local/far device address	[00..63]
4 =	Device name	According to device configuration
5 =	Device type	According to device configuration
6 =	PLC Components	According to device configuration
7 =	CNC components	According to device configuration
8 =	Device group	(see Chapter 6.1 "Identifier")
9 =	Device status	According to device configuration ON = DeviceStatus ON OFF = DeviceStatus OFF
10 =	Current device status	ON = Device ONLINE OFF = Device OFFLINE
11 =	Not yet assigned	[--]
12 =	Not yet assigned	[--]

13 =	Not yet assigned	[--]
14 =	Network ON/OFF	[ON = Network active OFF = No network active]
15 =	Network name	Max. 28 ASCII characters
16 =	PC number	[00..99,XX]
17 =	PC name	Max. 255 ASCII characters
18 =	Local device address	[00..63]

Meaning of line 2

Returns the firmware versions of the existing controller components. Each line consists of 8 columns.

Line 2	Column 1	...	Column 8

Value Range/Meaning of Columns

1 =	Line number	[2]
2 =	Status information	Contains the information whether or not the subsequent data in this line is valid; the following applies: [0 = Data is invalid – further column results [--] 1 = Data is valid]
3 =	Firmware version of the CNC component	Designation according to convention
4 =	Firmware version of the PLC component	Designation according to convention
5 =	Firmware version of the 1. APR component	Designation according to convention
6 =	Firmware version of the 2. APR component	Designation according to convention
7 =	Firmware version of the 3. APR component	Designation according to convention
8 =	Firmware version of the 4. APR component	Designation according to convention

Meaning of line 3

Returns the identification of the current parameter set and consists of 6 columns.

Line 3	Column 1	...	Column 6

Value Range/Meaning of Columns

1 =	Line number	[3]
2 =	Status information	Contains the information whether or not the subsequent data in this line is valid; the following applies: [0 = Data is invalid – further column results [--] 1 = Data is valid]
3 =	Index of the parameter set	[01..99]
4 =	Designation of the parameter set	Max. 32 ASCII characters
5 =	Date string	Date of generation/modification
6 =	Time string	Time of generation/modification

Meaning of line 4

Returns the identification of the current PLC program and consists of 6 columns.

Line 4	Column 1	...	Column 6

Value Range/Meaning of Columns

1 =	Line number	[4]
2 =	Status information	Contains the information whether or not the subsequent data in this line is valid; the following applies: [0 = Data is invalid – further column results [--] 1 = Data is valid]
3 =	Index of the PLC program	Always [00]
4 =	PLC resource name = PLC program name	Max. 32 ASCII characters
5 =	Date string	Date of generation/modification
6 =	Time string	Time of generation/modification

Meaning of line 5

Returns the identification of the current machine data set and consists of 6 columns.

Line 5	Column 1	...	Column 6

Value Range/Meaning of Columns

1 =	Line number	[5]
2 =	Status information	Contains the information whether or not the subsequent data in this line is valid; the following applies: [0 = Data is invalid – further column results [--] 1 = Data is valid]
3 =	Index of the machine data set	[01..99]
4 =	Designation of the machine data set	Max. 32 ASCII characters
5 =	Date string	Date of generation/modification
6 =	Time string	Time of generation/modification

Meaning of line 6

Returns the identification of the current NC package in memory A and consists of 6 columns.

Line 6	Column 1	...	Column 6

Value Range/Meaning of Columns

1 =	Line number	[6]
2 =	Status information	Contains the information whether or not the subsequent data in this line is valid; the following applies: [0 = Data is invalid – further column results [--] 1 = Data is valid]
3 =	Index of the NC package in memory A	[01..99]
4 =	Designation of the NC package in memory A	Max. 32 ASCII characters
5 =	Date string	Date of generation/modification
6 =	Time string	Time of generation/modification

Meaning of line 7

Returns the identification of the current NC package in memory B and consists of 6 columns.

Line 7	Column 1	...	Column 6

Value Range/Meaning of Columns

1 =	Line number	[7]
2 =	Status information	Contains the information whether or not the subsequent data in this line is valid; the following applies: [0 = Data is invalid – further column results [--] 1 = Data is valid]
3 =	Index of the NC package in memory B	[01..99]
4 =	Designation of the NC package in memory B	Max. 32 ASCII characters
5 =	Date string	Date of generation/modification
6 =	Time string	Time of generation/modification

Meaning of line 8

Returns the identification of the current cycle package and consists of 6 columns.

Line 8	Column 1	...	Column 6

Value Range/Meaning of Columns

1 =	Line number	[8]
2 =	Status information	Contains the information whether or not the subsequent data in this line is valid; the following applies: [0 = Data is invalid – further column results [--] 1 = Data is valid]
3 =	Index of the cycle package	[01..99]
4 =	Designation of the cycle package	Max. 32 ASCII characters
5 =	Date string	Date of generation/modification
6 =	Time string	Time of generation/modification

Meaning of the lines 9 - 15

These lines return information on the current NC program for the processes 0..6 and consist of 8 columns each.

Line 9..15	Column 1	...	Column 8

Value Range/Meaning of Columns

1 =	Line number	[9..15]
2 =	Status information	Contains the information whether or not the subsequent data in this line is valid; the following applies: [0 = Data is invalid – further column results [--] 1 = Data is valid]
3 =	Process number	[00..06]
4 =	Process name	Max. 40 ASCII characters
5 =	Current NC memory	[A,B]
6 =	Current NC program number	[01..99]
7 =	Current NC program name	Max. 32 ASCII characters
8 =	Current NC block	

Meaning of the lines 16 -22

These lines return information on the current tool lists for the processes 0..6 and consist of 12 columns each.

Line 16..22	Column 1	...	Column 12

Value Range/Meaning of Columns

1 =	Line number	[16..22]
2 =	Status information	Contains the information whether or not the subsequent data in this line is valid; the following applies: [0 = Data is invalid – further column results [--] 1 = Data is valid]
3 =	Process number	[00..06]
4 =	Process name	Max. 40 ASCII characters
5 =	Tool list index	Always [00]
6 =	Name of the tool list	Max. 32 ASCII characters
7 =	Date string	Date of generation/modification
8 =	Time string	Time of generation/modification
9 =	Tool magazine type	[MAGAZINE] [TURRET]
10 =	Number of spindles	[0..4]
11 =	Number of grippers	[0..4]
12 =	Number of magazine locations	[0..999]

Meaning of line 23

Returns the identification of the current I/O configuration list and consists of 6 columns.

Line 23	Column 1	...	Column 6

Value Range/Meaning of Columns

1 =	Line number	[23]
2 =	Status information	Contains the information whether or not the subsequent data in this line is valid; the following applies: [0 = Data is invalid – further column results [--] 1 = Data is valid]
3 =	Index of the I/O configuration list	[01..99]
4 =	Designation of the I/O configuration list	Max. 32 ASCII characters
5 =	Date string	Date of generation/modification
6 =	Time string	Time of generation/modification

Example DIF2

Read the dynamic device information of device 1. It is an ISP200-P-G2, and it is active in a network.

FI command		**01_BR_DIF2**
Line	**Column**	**Answer**
1	1	1
	2	1
	3	01
	4	Loading station right side
	5	ISP200-P-G2
	6	MTS-P
	7	NONE
	8	MWSX
	9	ON
	10	ON
	11	--
	12	--
	13	--
	14	ON
	15	PC network 1
	16	29
	17	BTV20-RIGHT
	18	01
2	1	2
	2	1
	3	--
	4	PLC06S-M05-06T03
	5	--
	6	--
	7	--
	8	--
3	1	3
	2	0
	3	--
	4	--
	5	--
	6	--
4	1	4
	2	1
	3	00
	4	Prg_3_Process
	5	30.04.03
	6	09:03:45

5	1	5
	2	0
	3	--
	4	--
	5	--
	6	--
6	1	6
	2	0
	3	--
	4	--
	5	--
	6	--
7	1	7
	2	0
	3	--
	4	--
	5	--
	6	--
8	1	8
	2	0
	3	--
	4	--
	5	--
	6	--
9	1	9
	2	0
	3	--
	4	--
	5	--
	6	--
	7	--
	8	--
10	1	10
	2	0
	3	--
	4	--
	5	--
	6	--
	7	--
	8	--

11	1	11
	2	0
	3	--
	4	--
	5	--
	6	--
	7	--
	8	--
12	1	12
	2	0
	3	--
	4	--
	5	--
	6	--
	7	--
	8	--
13	1	13
	2	0
	3	--
	4	--
	5	--
	6	--
	7	--
	8	--
14	1	14
	2	0
	3	--
	4	--
	5	--
	6	--
	7	--
	8	--
15	1	15
	2	0
	3	--
	4	--
	5	--
	6	--
	7	--
	8	--

16	1	16
	2	0
	3	--
	4	--
	5	--
	6	--
	7	--
	8	--
	9	--
	10	--
	11	--
	12	--
17	1	17
	2	0
	3	--
	4	--
	5	--
	6	--
	7	--
	8	--
	9	--
	10	--
	11	--
	12	--
18	1	18
	2	0
	3	--
	4	--
	5	--
	6	--
	7	--
	8	--
	9	--
	10	--
	11	--
	12	--
19	1	19
	2	0
	3	--
	4	--
	5	--
	6	--

	7	--
	8	--
	9	--
	10	--
	11	--
	12	--
20	1	20
	2	0
	3	--
	4	--
	5	--
	6	--
	7	--
	8	--
	9	--
	10	--
	11	--
	12	--
21	1	21
	2	0
	3	--
	4	--
	5	--
	6	--
	7	--
	8	--
	9	--
	10	--
	11	--
	12	--
22	1	22
	2	0
	3	--
	4	--
	5	--
	6	--
	7	--
	8	--
	9	--
	10	--
	11	--
	12	--

23	1	23
	2	0
	3	--
	4	--
	5	--
	6	--

FI command

Returns information on which controller data is currently available in the selected device.

BR_DIF3	**(Single Read)**
BC_DIF3	**(Cyclic Read)**

Response Structure

The answer consists of 1 line with 14 columns, each column having a specific meaning.

1 =	Active parameter set available	Yes/No
2 =	NC package memory A available	Yes/No
3 =	NC package memory B available	Yes/No
4 =	NC zero points memory A available	Yes/No
5 =	NC zero points memory B available	Yes/No
6 =	NC events available	Yes/No
7 =	NC variables available	Yes/No
8 =	NC D-corrections available	Yes/No
9 =	NC cycles available	Yes/No
10 =	Active machine data record available	Yes/No
11 =	PLC retain variables available	Yes/No
12 =	Tool lists available	Yes/No
13 =	Drive parameters available	Yes/No
14 =	I/O configuration list available	Yes/No

Line 1	Column 1	...	Column 14

Value Range/Meaning of Columns

1 =	Controller data available Yes/NO	[YES,NO]

Example DIF3 Return information on which controller data is currently available in the selected device.

FI command		01_BR_DIF3
Line	**Column**	**Answer**
1	1	NO
2	1	NO
3	1	NO
4	1	NO
5	1	NO
6	1	NO
7	1	NO
8	1	NO
9	1	NO
10	1	NO
11	1	YES
12	1	NO
13	1	NO
14	1	NO

9.9 Long ID of PLC Data Block: DIS

MWSX device groups

Designation **DIS** **D**ata **I**dentification **S**tring

Explanation Reads the long ID (directory entries) of the PLC program. Included in the directory entries are the number of the entry in the directory, the name, length and date and time of creation and/or details of the last time the respective data record was changed.

FI command

BR_DIS2 **(Single Read)**
BC_DIS2 **(Cyclic Read)**
BB_DIS2 **(Break Cyclic Read)**

Response Structure The following table shows the general structure of the response to the "DIS2" FI command. The response consists of a line with six columns.

Line 1	Column 1	...	Column 6

Value Range/Meaning of Columns

1 =	Number in PLC directory	[01...99]
2 =	Name of the PLC program	[max. 8 ASCII characters]
3 =	Length of the PLC program	[byte]
4 =	Date of creation/last change to PLC program	[DD.MM.YY]
5 =	Time of creation/last change to the PLC program	[HH:MM:SS]
6 =	Date of creation/last change to PLC program	[DD.MM.YYYY]

Note: If there is no valid NC package in the selected NC memory then all columns contain [--] .

Example DIS2

Read the directory entries of the PLC program at address 00.
Assumption:
There is a valid PLC program in the selected device.

FI command		**00_BR_DIS2**
Line	**Column**	**Answer**
1	1	01
	2	KEY1
	3	20018
	4	10.05.99
	5	12:42:00
	6	10.05.1999

Reference to Literature

See chapter entitled "Literature" [30].

9.10 Reading the Device Status Information: DSI

MWSX device groups

Designation **DSI** Device **S**tatus **I**nformation

Explanation

This allows the most important device status information to be read out. The following information is returned:

Type of information	**Status**	**Statement**
System error information		Yes/No
Mechanism error information		Yes/No
Machine key information		4 Byte HEX
Machine key information	valid	Yes/No
Machine status information		4 Byte HEX
Sercans information		4 Byte HEX
Parameter download	running	Yes/No
PLC download	running	Yes/No
Firmware download	running	Yes/No
Offline/Online information		
Device simulation	switched on	Yes/No
Device status information		ON/OFF
Communication channel defined		Yes/No
PLC components available		Yes/No
Monitor mode	Active	Yes/No

FI command

Read out device status information for ALL defined devices.

BR_DSI1 **(Single Read)**
BC_DSI1 **(Cyclic Read)**
BB_DSI1 **(Break Cyclic Read)**

Note: The "DSI1" FI command refers to all devices within this device group. Therefore, any valid device address can be indicated in the command line (see example DSI1). The FI device polling mechanism **MUST** be switched on (see system configurator)!

Response Structure

The following table shows the general structure of the response to the "DSI1" FI command.

Line 1...n	Column 1	...	Column 11

Value Range/Meaning of Columns

1 =	device address	[00...63]
2 =	System error information	[0 = there is no system error 1 = there is a system error]
3 =	Mechanism error information	[0 = there is no mechanism error 0 = there is a mechanism error
4 =	Machine key information	[4 byte in HEX coding]
5 =	Machine key information valid?	[0 = not valid, 1=valid]
6 =	Machine status information	[4 byte in HEX coding]
7 =	Sercans information	[4 byte in HEX coding]
8 =	Is parameter download active?	[0 = parameter download not running 1 = parameter download running]
9 =	Is PLC download active?	[0 = PLC download not running 1 = PLC download running]
10 =	Is firmware download active?	[0 = PLC download not running 1 = PLC download running]
11 =	Offline/Online information	[0 = device connection interrupted 1 = device connection O.K.]
12 =	Device simulation switched on?	[0 = NO Simulation mode 1 = simulation mode]
13 =	Current device status information	[0 = Device-Status=OFF 1 = Device-Status=ON]
14 =	Communication channel defined?	[0 = NO communication channel 1 = Communication channel defined]
15 =	PLC components available ?	[0 = NO PLC component 1 = PLC component (DOS-Pcl) 2 = PLC component (WIN-Pcl)]
16 =	Monitor mode	[0 = NO monitor mode active 1 = Monitor mode active]

Example DSI1

Read the current device status information.

Assumption:
The following devices addresses are defined:

- Device address 01 (ISP200-P-G2)
- Device address 03 (ISP200-R-G2)

FI command		**01_BR_DSI1**
Line	**Column**	**Answer**
1	1	01
	2	0
	3	0
	4	00000000
	5	0
	6	00000000
	7	00000000
	8	0
	9	0
	10	0
	11	1
	12	0
	13	1
	14	1
	15	2
	16	0
2	1	03
	2	1
	3	0
	4	00000000
	5	0
	6	00000000
	7	00000000
	8	0
	9	0
	10	0
	11	1
	12	0
	13	1
	14	1
	15	2
	16	0

FI command

Read out device status information for a selected device.

BR_DSI2	**(Single Read)**
BC_DSI2	**(Cyclic Read)**
BB_DSI2	**(Break Cyclic Read)**

Response Structure

The following table shows the general structure of the response to the "DSI2" FI command.

Line 1...n	Column 1	...	Column 11

Value Range/Meaning of Columns

1 =	device address	[00...63]
2 =	System error information	[0 = there is no system error 1 = there is a system error]
3 =	Mechanism error information	[0 = there is no mechanism error 1 = there is a mechanism error]
4 =	Machine key information	[4 byte in HEX coding]
5 =	Is machine key information valid?	[0 = not valid, 1=valid]
6 =	Machine status information	[4 byte in HEX coding]
7 =	Sercans information	[4 byte in HEX coding]
8 =	Is parameter download active?	[0 = parameter download not running 1 = parameter download running]
9 =	Is PLC download active?	[0 = PLC download not running 1 = PLC download running]
10 =	Is firmware download active?	[0 = PLC download not running 1 = PLC download running]
11 =	Offline/Online information	[0 = device connection interrupted 1 = device connection O.K.]
12 =	Device simulation switched on?	[0 = NO Simulation mode 1 = simulation mode]
13 =	Current device status information	[0 = Device status=OFF 1 = Device status=ON]
14 =	Communication channel defined?	[0 = NO communication channel 1 = Communication channel defined]
15 =	PLC components available ?	[0 = NO PLC component 1 = PLC component (DOS-Pcl) 2 = PLC component (WIN-Pcl)]
16 =	Monitor mode	[0 = NO monitor mode active 1 = Monitor mode active]

Example DSI2 Read the current device status information for the selected device.

FI command		00_BR_DSI2
Line	**Column**	**Answer**
1	1	00
	2	0
	3	0
	4	00000000
	5	0
	6	00000000
	7	00000000
	8	0
	9	0
	10	0
	11	1
	12	0
	13	1
	14	1
	15	2
	16	0

9.11 Device Type and Accompanying Components: DTY

MWSX device groups

Designation **DTY** **D**evice **TY**pe

Explanation The device type and the accompanying components of the selected device address are output.

FI command

BR_DTY1 **(Single Read)**

BC_DTY1 **(Cyclic Read)**

Response Structure The following table shows the general structure of the response to the "DTY1" FI command. A line with three columns is output for the device type, as well as the names of the first device component and the name of the second device component.

Line 1	Column 1	...	Column 3

Value Range/Meaning of Columns

1 =	Device Type	(see Chapter 6.1 "Elements of the FI Command" and "Identifier")
2 =	Component type1	IND_DEV.INI-Entry: Componenttype1=
3 =	Component type 2	IND_DEV.INI-Entry: Componenttype2=

Example DTY1

Output the device type and the accompanying components of device address 00.

FI command	**00_BR_DTY1**		
Answer			
Line	**Column 1**	**Column 2**	**Column 3**
1	ISP200-P-G2	MTS-P	NONE
1	ISP200-P	MTS-P	NONE

9.12 Diagnosis Window Data: DWD

MWSX device groups

Designation **DWD** **D**iagnosis **W**indow **D**ata

Explanation

Diagnostic messages are output. The data is edited in such a way that they can be output directly in the diagnosis overview, i.e., where applicable, different types of diagnosis, such as ProVi and a process report, are returned simultaneously.

FI command

Output all diagnostic messages.

For reasons of optimization, not all data is called up. Accordingly, the Diagnosis server must be informed that the data is required (see ADW).

BR_DWD1_(1){_(2)}	**(Single Read)**
BC_DWD1_(1){_(2)}	**(Cyclic Read)**
(1) = Type of diagnosis window	[1 = CNC error, 2 = sequence errors, 3 = general errors, 4 = messages, 10 = start preconditions, 11 = warnings, 12 = setup diagnosis]
(2) = Module number	[1...99] ! only for window type 1 -4 !

Output first diagnostic messages.

BR_DWD2_(1){_(2)}	**(Single Read)**
BC_DWD2_(1){_(2)}	**(Cyclic Read)**
(1) = Type of diagnosis window	[1 = CNC error, 2 = sequence errors, 3 = general errors, 4 = messages, 10 = start preconditions, 11 = warnings, 12 = setup diagnosis]
(2) = Module number	[1...99] ! only for window type 1 -4 !

Response Structure

The following table shows the general structure of the "DWD1" and "DWD2" FI commands. The number of lines depends on the number of messages pending. Different columns are valid according to the type of diagnosis.

If there are no messages, the number of lines is 0.

Line 1...n	**Column 1**	**...**	**Column 14**

Meaning of the Columns

1 =	Message text	[ASCII characters]
2 =	Time stamp day	[mm.dd.yyyy]
3 =	Time stamp hour	[hh:mm:ss]
4 =	Reference text available	[YES, NO]
5 =	Type of diagnosis	[1 = ProVi, 2 = SFC, 3 = MTC-NC, 4 = MTA-NC]
6 =	Message number	[ASCII characters]

7 =	Message ID	[ASCII characters] (DWORD, decimal) (ProVi)
8 =	Mechanism number	[0..31] (MTC-NC) [0] (MTA-NC)
9 =	2 byte additional information	[ASCII characters] (MTC NC)
10 =	Message group	[1...9999] (MTA-NC)
11 =	SFC entity name	[ASCII characters]
12 =	NC note	[ASCII characters] (MTC NC)
13 =	Analysis of criteria available	[YES, NO] (ProVi, SFC)
14 =	Message HTML file	[ASCII characters] (ProVi, MTC-NC)

Example DWD1 All diagnostic messages from module 3 in control unit 0.

There are two messages.

FI command		**00_BR_DWD1_4_3**
Line	**Column**	**Answer**
1	1	Guard not closed
	2	01.27.2000
	3	14:56:32
	4	YES
	5	1
	6	34
	7	43923028
	8	
	9	
	10	
	11	
	12	
	13	YES
	14	
2	1	Station waiting until tool-change command has ended.
	2	01.27.2000
	3	15:03:10
	4	YES
	5	3
	6	79
	7	
	8	1
	9	0
	10	
	11	
	12	
	13	NO
	14	

Example DWD2 First diagnostic message from module 3 in control unit 0.

There are two messages.

FI command		**00_BR_DWD2_4_3**
Line	**Column**	**Answer**
1	1	Guard not closed
	2	01.27.2000
	3	14:56:32
	4	YES
	5	1
	6	34
	7	43923028
	8	
	9	
	10	
	11	
	12	
	13	YES
	14	

Reference to Literature See chapter entitled "Literature" [13].

9.13 Component Information for a System Error: ECI

MWSX device group

Designation **ECI** **E**rror **C**omponent **I**nformation

Explanation When a system error is present, this command is used to define which controller component is causing the error. Here, PLC components are differentiated from general controller components (e.g. CNC, Synax, MTA, …).

FI command **BR_ECI1** **(Single Read)**

Response Structure The response to the "ECI1" FI command consists of one line with two columns.

Line 1	**Column 1**	**Column 2**

Value Range/Meaning of Columns

1 =	PLC component information	[0 = There is NO system error at the PLC) 1 = There is a system error at the PLC]
2 =	General information on controller components	[0 = There is NO system error at the general control component [0 = There is a system error at the general control component; the following applies: 2 = CNC component 3 = SYNAX component 4 = VISUAL-MOTION component 5 = MTA component 6 = TRANS 200 component]

Example ECI1 There is a system error present in device 0 (ISP200-P-G2)) which is caused by the PLC component.

FI command		**00_BR_ECI1**
Line	**Column**	**Answer**
1	1	1
1	2	0

9.14 Existing Errors: EDE

MWSX device groups

Designation **EDE** **E**xisting **D**iagnosis **E**rror

Explanation Whether or not errors exist in a control unit or in a module is queried. These can be step chain errors, NC errors, MTA 200 errors or ProVi errors.

FI command Query whether there are errors in this control unit.

BR_EDE1 **(Single Read)**

BC_EDE1 **(Cyclic Read)**

Response Structure The following table shows the general structure of the "EDE1" FI command.

Line 1	**Column 1**

Meaning of the Columns 1 = Error exists [YES, NO]

Example EDE1 Do errors exist in control unit 0?

FI command		**00_BR_EDE1**
Line	**Column**	**Answer**
1	1	YES

FI command Query whether or not errors exist in a specific module.

BR_EDE2_(1) **(Single Read)**

BC_EDE2_(1) **(Cyclic Read)**

(1) = Module number [1...99]

Response Structure The following table shows the general structure of the "EDE2" FI command.

Line 1	**Column 1**

Meaning of the Columns 1 = Error exists [YES, NO]

Example EDE2 Do errors exist in module 1 on control unit 0?

FI command		**00_BR_EDE2_2**
Line	**Column**	**Answer**
1	1	NO

9.15 Existing Diagnosis Window: EDW

MWSX device groups

Designation **EDW** **E**xisting **D**iagnosis **W**indow

Explanation Which types of diagnosis window exist is queried.

FI command Output all types of diagnosis window.

BR_EDW1 **(Single Read)**

Response Structure The following table shows the general structure of the "EDW1" FI command. The number of lines depends on the number of types of window existing.

Line 0...n	Column 1	Column 2

Meaning of the Columns

1 = Type of diagnosis window	[1 = CNC error, 2 = sequence errors, 3 = general errors, 4 = messages, 10 = start requirements, 11 = warnings, 12 = setup diagnosis]
2 = Module number	[ASCII characters] 0 = Diagnosis window type does not belong to any module

Example EDW1 All types of diagnosis window in control unit 0.

There are three diagnosis windows.

FI command		**00_BR_EDW1**
Line	**Column**	**Answer**
1	1	10
	2	0
2	1	1
	2	3
3	1	2
	2	3

FI command Output all diagnosis window types for a module.

BR_EDW2_(1) **(Single Read)**

(1) = Module number [1...99]

Response Structure The following table shows the general structure of the "EDW2" FI command. The number of lines depends on the number of types of window existing.

Line 0...n	Column 1	Column 2

Meaning of the Columns

1 = Type of diagnosis window	[1 = CNC error, 2 = sequence errors, 3 = general errors, 4 = messages]
2 = Module number	[ASCII characters] 0 = Diagnosis window type does not belong to any module

Example EDW2

All types of diagnosis window in Module 3, Control unit 0.

There are two diagnosis windows.

FI command		**00_BR_EDW2_3**
Line	**Column**	**Answer**
1	1	1
	2	3
2	1	2
	2	3

FI command

Query a specific type of diagnosis window.

BR_EDW3_(1){_(2)}	**(Single Read)**
(1) = Type of diagnosis window	[1 = CNC error, 2 = sequence errors, 3 = general errors, 4 = messages, 10 = start requirements, 11 = warnings, 12 = setup diagnosis]
(2) = Module number	[1...99] ! only for window type 1 -4 !

Response Structure

The following table shows the general structure of the "EDW3" FI command.

Line 1	**Column 1**

Meaning of the Columns

1 = Type of diagnosis window exists	[YES, NO]

Example EDW3

Query whether or not an NC error window exists in module 3, control unit 0.

FI command		**00_BR_EDW3_1_3**
Line	**Column**	**Answer**
1	1	YES

Reference to Literature

See chapter literature [13].

9.16 Existing PLC Diagnoses: EPD

MWSX device groups

Designation

EPD **E**xisting **P**LC **D**iagnosis

Explanation

Which PLC diagnostic types exist is queried. Depending on the FI command, specific types are queried or else the diagnostic types for a device or a module are output together.

FI command

Query which PLC diagnostic types are available on a control unit.

BR_EPD1 **(Single Read)**

Response Structure

The following table shows the general structure of the "EPD1" FI command.

Line 1	**Column 1-3**

Meaning of the Columns

1 = Start requirement exists	[YES, NO]
2 = Warning exists	[YES, NO]
3 = Setup diagnosis exists	[YES, NO]

Example EPD1

Query PLC diagnostic types in control unit 0.

FI command		00_BR_EPD1
Line	**Column**	**Answer**
1	1	YES
	2	NO
	3	YES

FI command

Query which PLC diagnostic types are available in a module.

BR_EPD2_(1)	**(Single Read)**
(1) = Module number	[1...99]

Response Structure

The following table shows the general structure of the "EPD2" FI command.

Line 1	Column 1-3

Meaning of the Columns

1 = Messages exist	[YES, NO]
2 = Errors exist	[YES, NO]
3 = Step chains exist	[YES, NO]

Example EPD2

Query the PLC diagnostic types in Module 2 on Control unit 0.

FI command		00_BR_EPD2_2
Line	**Column**	**Answer**
1	1	NO
	2	YES
	3	YES

FI command

Query a specific PLC diagnostic type.

BR_EPD3_(1){_(2)}	**(Single Read)**
(1) = Message type	[1 = error, 2 = messages, 3 = SFC, 10 = warnings, 11 = start requirements, 12 = setup diagnosis]
(2) = Module number	[1...99] ! only for message type 1 -3!

Response Structure

The following table shows the general structure of the "EPD3" FI command.

Line 1	Column 1

Meaning of the Columns

1 = Diagnosis type exists	[YES, NO]

Example EPD3

Are there any messages in module 4 in control unit 0?

FI command		00_BR_EPD3_2_4
Line	**Column**	**Answer**
1	1	YES

9.17 Existing ProVi Types: EPT

MWSX device groups

Designation **EPT** **E**xisting **P**roVi **T**ypes

Explanation Which ProVi types are programmed in the current PLC program is queried. The data is returned in a suitable form for the message texts of the small control panels. There is no need to define modules in Moduldef.ini.

FI command Output all ProVi types.

BR_EPT1 **(Single Read)**

Response Structure The following table shows the general structure of the "EPT1" FI command. The number of lines depends on the number of ProVi types existing.

Line 0...n	Column 1	Column 2

Meaning of the Columns

1 = Type [11 = error, 12 = messages,
20 = start requirements,
21 = warnings, 22 = setup diagnosis]

2 = Index [ASCII characters]

Example EPT1 All ProVi types in control unit 0.

There are three diagnosis windows.

FI command		**00_BR_EPT1**
Line	**Column**	**Answer**
1	1	20
	2	0
2	1	11
	2	3
3	1	12
	1	3

9.18 Error Status: EST

MWSX device groups

Designation **EST** **E**rror **ST**ate

Explanation Queries the error state of a variable.

FI command Query the frozen error state of a variable.

BR_EST1!(1)!(2)	**(Single Read)**
BC_EST1!(1)!(2)	**(Cyclic Read)**
(1) = Error ID	[ASCII characters] (DWORD, decimal)
(2) = Variable name	[ASCII characters]

Note: The separator "!" is used in this command.

Response Structure The following table shows the general structure of the "EXD1" FI command.

Line 1	Column 1

Meaning of the Columns 1 = Error state

WinPCL - Example EST Read the value of WinPCL variable "IB_EXT24" in WinPCL program "Prog", at device address 00.

Exception:
The WinPCL variable "IB_EXT24" is declared in the WinPCL Program "Prog" as BOOL.

FI command		**00_BR_EST1!5892855!:Prog.IB_EXT24**
Line	**Column**	**Answer**
1	1	1

9.19 Execution Display: EXD

MWSX device groups

Designation **EXD** **EX**ecution **D**isplay

Explanation Information for displaying the execution of a movement is output.

FI command Query the execution of a step or of an action.

BR_EXD1!(1)!(2)!(3)	**(Single Read)**
BC_EXD1!(1)!(2)!(3)	**(Cyclic Read)**
(1) = SFC entity name	[ASCII characters]
(2) = Step or action name	[ASCII - characters]
(3) = Behaviour of mode	[1 = all modes, 2 = manual mode]

Note: The separator "!" is used in this command.

Response Structure

The following table shows the general structure of the "EXD1" FI command.

Line 1	Column 1

Meaning of the Columns

1 = Execution	[1 = can be executed, 0 = cannot be executed]

Example EXD1

Query the execution of the step "open" for the chain "clamp" in control unit 0 for all modes.

FI command		**00_BR_EXD1!Station03A.Clamp!Open!1**
Line	**Column**	**Answer**
1	1	1

FI command

Query whether the condition analysis (control image) of a step chain is enabled.

BR_EXD2!(1) **(Single Read)**

(1) = SFC entity name [ASCII characters]

Note: The separator "!" is used in this command.

Response Structure

The following table shows the general structure of the "EXD2" FI command.

Line 1	Column 1

Meaning of the Columns

1 = Enabled	[1 = enabled, 0 = not enabled]

Example EXD2

Query whether the condition analysis of the "clamp" chain has been enabled.

FI command		**00_BR_EXD2!Station03A.Clamp**
Line	**Column**	**Answer**
1	1	1

9.20 Read Reference Name of a PLC Variable: MAR

MWSX device groups

Designation **MAR** **M**ap **A**bsolute PCL-**R**eference

PLC Explanation The absolute reference name of a symbolic PLC variable is read out.

FI command

Read the absolute reference name of a PLC variable.

BR_MAR_(1) **(Single Read)**

BC_MAR_(1) **(Cyclic Read)**

(1) = Identifier of the PLC variable

PLC – Example MAR

Read the absolute reference name of the PLC variable with the identifier "abref" at device address 00.

Assumption:
The PLC variable with the identifier "abref" is of the type "INTEGER".

FI command		**00_BR_MAR_abref**
Line	**Column**	**Answer**
1	1	%M100.0

WinPlc Explanation

The absolute reference name of a symbolic WinPlc PLC variable with program entity is read out.

Read the absolute reference name of a WinPlc PLC variable.

FI command

BR_MAR1_(1)	**(Single Read)**
BC_MAR_(1)	**(Cyclic Read)**

(1) = Identifier of the PLC variable

Win PLC - Example MAR

Read the absolute reference name of the Win PLC variable with the identifier "Prog.abref" at device address 00.

Assumption:
The Win PLC variable with the identifier "Prog.abref" is of the type "INTEGER" and is present in Win PLC program entity "Prog".

FI command		**00_BR_MAR1_:Prog.abref**
Line	**Column**	**Answer**
1	1	%M100.0

Reference to Literature

See chapter entitled "Literature" [30].

9.21 Device Data of the Module Configuration: MCD

MWSX device groups

Designation

MCD **M**odule **C**onfiguration: **D**evice Information

Explanation

All device data for module configuration is read out from the "Moduldef.ini" file which is stored in the "[LW]:\Program Files\Indramat\MTGUI\CustomData\Resource" directory on standard installation. The device data is in the sections [DeviceAddrX], whereby "X" stands for the configured device addresses.

FI command

Read out device data within the module configuration of the MWSX device group.

BR_MCD1	**(Single Read)**
BC_MCD1	**(Cyclic Read)**
BB_MCD1	**(Break Cyclic Read)**

Note: The "MCD1" FI command refers to all devices within the MWSX device group. Therefore, any valid device address can be indicated in the command line (see example MCD1).

Response Structure

The following table shows the general structure of the response to the "MCD1" FI command. The number of lines depends on the number of configured devices. Each line consists of four columns for the device address as well as PLC-FB names for providing setup diagnostics, warning messages and start requirements.

Value Range of the Columns

1 = Device address	[0...15]
2 = PLC-FB name for the setup diagnostics	[max. 9 ASCII characters]
3 = PLC-FB name for the warning messages	[max. 9 ASCII characters]
4 = PLC-FB name for the start requirements	[max. 9 ASCII characters]

Example MCD1

Read all device data of the module configuration

Assumption:
The following devices have been configured in the MWSX device group:

- Device address 01 (ISP200-P)
- Device address 03 (ISP200-R)

FI command	**03_BR_MCD1**			
Answer				
Line	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	01	PVSetup_1	PVWarn_1	PVStart_1
2	03	PVSetup_3	PVWarn_3	PVStart_3

Reference to Literature

See chapter entitled "Literature" [36].

9.22 Module Data of the Module Configuration: MCM

MWSX device groups

Designation

MCM **M**odule **C**onfiguration: **M**odule Information

Explanation

All module data of a particular device is read out from the "Moduldef.ini'" file. This file is located in the directory "[LW]:\Program Files\Indramat\MTGUI\CustomData\Resource" and contains the data for all module configurations. The module data is located in sections [DeviceAddrX\ModulY], whereby "X" stands for the device addressed and "Y" for the configured module numbers.

FI command

Read-out module data from the module configuration with respect to a device from the MWSX device group.

BR_MCM1 **(Single Read)**

BC_MCM1 **(Cyclic Read)**

BB_MCM1 **(Break Cyclic Read)**

Response Structure

The following table shows the general structure of the response to the "MCM1" FI command. The number of lines depends on the number of configured modules of a device. Each line consists of four columns for the module number, module name and PLC-FB names for general module errors and module messages.

Line 1	**Column 1**	**...**	**Column 4**

Value Range of the Columns

1 = Module number	[0...99]
2 = Module name	[max. 28 ASCII characters]
3 = PLC-FB name for general module errors	[max. 9 ASCII characters]
4 = PLC-FB name for module messages	[max. 9 ASCII characters]

Example MCM1

Read the module data of device 03 from the module configuration:

Assumption:
The following modules have been defined:

- Module number 5
- Module number 7

FI command	**03_BR_MCM1**			
Answer				
Line	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	5	Module 5 - Milling	PVError_5	PVMsg_5
2	7	Module 7 - Drilling	PVError_7	PVMsg_7

Reference to Literature See chapter entitled "Literature" [36].

9.23 SFC Data of the Module Configuration: MCS

MWSX device groups

Designation **MCS** **M**odule **C**onfiguration: **S**FC Information

Explanation All SFC data of a particular module is read out from the "Moduldef.ini" file. This file is located in the directory "[LW]:\Program Files\Indramat\MTGUI\CustomData\Resource" and contains the data for all module configurations. The SFC data is located in sections [DeviceAddrX\ModulY\Sfc], whereby "X" stands for the device addressed and "Y" for the selected module number.

FI command Read out the SFC data with respect to the module of a device from the module configuration of the MWSX device group.

BR_MCS1_(1)	**(Single Read)**
BC_MCS1_(1)	**(Cyclic Read)**
BB_MCS1_(1)	**(Break Cyclic Read)**
(1) = Module number	[0...99]

Response Structure The number of lines depends on the number of configured Indrastep step chains for a device. Each line contains a column for the name of the Indrastep step chains.

Value Range of the Column 1 = Name of the Indrastep step chain [format W.X.Y.Z]

Format W.X.Y.Z	**Value Range**
W	Max. 9 ASCII characters
X	Max. 9 ASCII characters ! OPTIONAL !
Y	Max. 9 ASCII characters ! OPTIONAL !
Z	Max. 9 ASCII characters ! OPTIONAL !

Example MCS1

Read the name of the Indrastep step chain of module 5 from device 03 of the module configuration.

Assumption:
The following Indrastep step chains have been defined:

- ISFB_1
- FB_US.ISFB_3
- FB_US.ISFB_3.SW1
- FB_US.ISFB_3.SW1.ABBA

FI command		**03_BR_MCS1_5**
Line	**Column**	**Answer**
1	1	ISFB_1
2	1	FB_US.ISFB_3
3	1	FB_US.ISFB_3.SW1
4	1	FB_US.ISFB_3.SW1.ABBA

9.24 Downloading Message Texts: MFD

MWSX device groups

Designation **MFD** **M**essage **F**iles **D**ownload

FI command

This is used to load the message texts into the device indicated. These message texts are required for small devices. The following message texts are transmitted, depending on the type of device:

- system error messages
- transmission error messages, and/or
- mechanism messages.

Note: This FI command is an FI job!

BW_MFD1 **(Single Write)**

Response Structure

The response to the "MFD1" FI command consists of three lines, each with one column. The meaning of the elements is as follows:

Line 1 = Job ID [01...20] (refer to chapter entitled "FI Commands for the MPCX Device Group", IFJ).

Line 2 = FI command [string, in accordance to chapter entitled "Elements of the FI Command"]

Line 3 = FI job error code (see chapter entitled "Error Codes")

Example MFD1

Load message texts into the device with device address 00.

FI command		**00_BW_MFD1**
Line	**Column**	**Answer**
1	1	01
2	1	00_BW_MFD1
3	1	0

9.25 Reading Machine Key Information: MKS

MWSX device groups

Designation **MKS** **M**achine **K**ey **S**tatus

Explanation Current machine key information can be read for the selected device.

FI command Read machine key information for selected device.

BR_MKS **(Single Read)**

BC_MKS **(Cyclic Read)**

BB_MKS **(Break Cyclic Read)**

Response Structure The following table shows the general structure of the response to the FI command "MKS".

Line 1	Column 1	Column 2

Value Range/Meaning of Columns

1 =	Information of machine key	[4 byte in HEX coding]
2 =	Information valid?	[0 = not valid, 1=valid]

Example MKS Read the current machine key information for device 0.

FI command		00_BR_MKS
Line	**Column**	**Answer**
1	1	00000000
	2	0

9.26 Writing the GUI-SK Block: MKT

MWSX device groups

Designation **MKT** **M**achine **K**ey **T**able

Explanation Writes the GUI-SK16 block in the PLC.

FI command Write GUI-SK16 block.

BW_MKT1_(1)	**(Single Write)**
(1) = List of the 48 PLC variables for writing the GUI-SK16 block.	A distinction is made between the following cases: 1. Clear GUI-SK16 block. 2. Write the GUI-SK16 block with the 48 PLC variables, filling gaps with $SPACE.

Response Structure (P_ACK) is returned following successful transmission.

Line 1	Column 1

Value Range/Meaning of the Columns

1 =	Successfully completed	(P_ACK)

1. Example MKT1

Clear GUI-SK16 block:

FI command		Value to be written: $EMPTY 00_BW_MKT1
Line	**Column**	**Answer**
1	1	(P_ACK)

2. Example MKT1

Write GUI-SK16 block:

FI command		Value to be written: $EMPTY SPSVAR1,SPSVAR2,$SPACE,... 00_BW_MKT1
Line	**Column**	**Answer**
1	1	(P_ACK)

FI command

Write the GUI-SK16 block, writing only those PLC variables which are defined in the current PLC program. All undefined PLC variables are automatically replaced by $SPACE and returned as a partial result (column 2).

BW_MKT2_(1)	(Single Write)
(1) = List of the 48 PLC variables for writing the GUI-SK16 block.	A distinction is made between the following cases: 1. Clear GUI-SK16 block: BW_MKT2 $EMPTY 2. Write the GUI-SK16 block with the 48 PLC variables, filling gaps with $SPACE: BW_MKT1 SPSVAR1,SPSVAR2, $SPACE,$SPACE,....

Response Structure

After successful transmission, one line with two columns is returned.

Line 1	Column 1	Column 2

Value Range/Meaning of Columns

1 =	Status report	[0 = at least 1 PLC variable in the current PLC program is NOT defined 1 = ALL PLC variables could be written]
2 =	List of the NON-defined PLC variables in the current PLC program	[-- = ALL PLC variables could be written, or else list of the PLC variables that could not be written.] The individual PLC variables are separated by a comma.

Example MKT2

Write GUI-SK16 component with 48 PLC variables, while the PLC variables SPSVAR11 and SPSVAR12 are NOT defined in the current PLC program.

FI command		Value to be written: SPSVAR1,SPSVAR2,...SPSVAR48 00_BW_MKT2
Line	**Column**	**Answer**
1	1	(P_ACK)
	2	--

Extended information

The variables are divided into 3 groups of 16 variables each and have the following meaning:

1. Variables 1 - 16: Machine function keys
2. Variables 17 - 32: Status pressed
3. Variables 33 - 48: Status shining

Note: When, for example, only the first 8 M keys are used, the telegram will contain only these 8 PLC variables. The other 40 variables need not be defined in the transmission parameter.

When certain areas, e.g. of M keys, are left unused, they must be filled up with '$SPACE' up to the next variable.

9.27 Read System Messages: MSG

MWSX device groups

Designation **MSG** **M**e**S**sa**G**e

Explanation Reading of system messages

FI command Message

CC_MSG_(1) **(Cyclic Read)**

(1) = SYS-Message number

Note: Exists only as a cyclic command

Response Structure The response of the FI command 'MSG' consists of the system message data.

Example MSG 00_CC_MSG_64 (64 = MSG_SYSERRGEN)

FI command		**00_CC_MSG_64/3**
Line	**Column**	**Answer**
1	1	00

Restriction The following system messages:

SYS Message	SYS Message number
MSG_PCLUPDBEG	52
MSG_PARUPDBEG	24
MSG_FWAUPDBEG	82

These commands cannot be used with the following programs:

- Bosch Rexroth OPC server
- Bosch Rexroth DDE server

9.28 Reading the Firmware/Monitor Identification: MTC

MWSX device groups

Designation **MTC** **MT-C**NC Slot Software Version

FI command This command is used to read the firmware identification from the various control components (slot numbers).

Note: For the time this FI command is executed, the internal FI communication interlocks (fast timeout monitoring, offline operation, etc.) are switched off.

FI command

BR_MTC_(1)	**(Single Read)**
BC_MTC_(1)	**(Cyclic Read)**
(1) = Slot number	[1=CNC, 2=SIO, 3=PLC, 4=APR1 5=APR2, 6=APR3, 7=APR4]

Response Structure The following table shows the general structure of the response to the FI command "MTC". A line of 1 column is output.

Line 1	Column 1

Value Range/Meaning of Columns

1 = Firmware identification string [max. 16 ASCII characters]

Example MTC Read the firmware identification of slot number 1 (CPU) of device 00.

FI command		00_BR_MTC_1
Line	Column	Answer

Explanation With the FI command "BR_MTC1", the monitor versions of the various components (CNC, PLC, APR) can be read out.

FI command **BR_MTC1** **(Single Read)**

Response Structure The response to the "BR_MTC1" FI command consists of six lines with four columns. One line is returned for each potential component (CNC, PLC, APR1-4).

Line 1.0.6	Column 1	Column 2	Column 3	Column 4

Value Range/Meaning of Columns

1 =	Line number	[1..6]
2 =	Component information	[CNC=NC component SPS=PLC component APR=APR component
3 =	Monitor version – "old" format	
4 =	Monitor version – "new" format	

Example MTC1 Read the monitor versions for device 0.

FI command		**00_BR_MTC1**
Line	**Column**	**Answer**
1	1	1
	2	CNC
	3	--
	4	--
2	1	2
	2	PLC
	3	MON-PMK 09.05/0705.02.01
	4	FWC-CONTROL-MON-06V00-NN
3	1	3
	2	APR
	3	--
	4	--
4	1	4
	2	APR
	3	--
	4	--
5	1	5
	2	APR
	3	--
	4	--
6	1	6
	2	APR
	3	--
	4	--

9.29 ProVi Diagnosis Data: PDD

MWSX device groups

Designation **PDD** **P**rovi **D**iagnosis **D**ata

Explanation Data for ProVi criteria analysis is output.

FI command Output of files to indicate the detail in the editor.

BR_PDD1_(1)_(2){_(3)}	**(Single Read)**
(1) = Message ID	[ASCII characters]
(2) = Message type	[1 = error, 2 = messages, 10 = warnings, 11 = start requirements, 12 = setup diagnosis]
(3) = Module number	[1...99] ! only for message type 1 -2!

Response Structure

The following table shows the general structure of the PDD1 FI command.

Line 1	Column 1	...	Column 5

Meaning of the Columns

1 = POU ID	[ASCII characters]
2 = Detail morpheme	[ASCII characters] (DWORD, decimal)
3 = Error ID	[ASCII characters] (DWORD, decimal)
4 = POE entity name	[ASCII characters]
5 = Nw ID (network ID)	[ASCII characters]

Example PDD1

Indication of data of a ProVi error with ID 43923028 from module 3 in control unit 0.

FI command		**00_BR_PDD1_43923028_1_1**
Line	**Column**	**Answer**
1	1	STATION_1_2
	2	98243823
	3	34985304
	4	Station2.Module3
	5	43493454

FI command

Output the I/O addresses to display a detail.

BR_PDD2_(1)_(2){_(3)}	**(Single Read)**
(1) = Message ID	[ASCII characters]
(2) = Message type	[1 = error, 2 = messages, 10 = warnings, 11 = start requirements, 12 = setup diagnosis]
(3) = Module number	[1...99] ! only for message type 1 -2!

Response Structure

The following table shows the general structure of the PDD2 FI command.

Line 1-n	Column 1	Column 2

Meaning of the Columns

1 = Variable morpheme	[ASCII characters] (DWORD, decimal)
2 = I/O address	[ASCII characters]

Example PDD2

Query of the I/O addresses of a ProVi error with ID 43923028 from module 3 in control unit 0.

Three variables have an I/O address.

FI command		**00_BR_PDD2_43923028_1_1**
Line	**Column**	**Answer**
1	1	98243823
	2	%I3.2.0
2	1	40923423
	2	%Q23.21.7
3	1	34985304
	2	%I100.3.5

FI command

Determine the multilingual comments for displaying a detail.

BR_PDD3_(1)_(2){_(3)}	**(Single Read)**
(1) = Message ID	[ASCII characters]
(2) = Message type	[1 = error, 2 = messages, 10 = warnings, 11 = start requirements, 12 = setup diagnosis]
(3) = Module number	[1...99] ! only for message type 1 -2!

Response Structure

The following table shows the general structure of the PDD3 FI command.

Line 1-n	Column 1	Column 2

Meaning of the Columns

1 = Comment morpheme	[ASCII characters] (DWORD, decimal)
2 = New comment	[ASCII characters]

Example PDD3

Query of the comments for indication of a ProVi error with ID 43923028 from module 3 in control unit 0.

Two comments are replaced by another text.

FI command		**00_BR_PDD3_43923028_1_1**
Line	**Column**	**Answer**
1	1	98243823
	2	Clamp open
2	1	40923423
	2	Clamp closed

FI command

Query of the status of a certain message

BR_PDD4_(1)_(2){_(3)}	**(Single Read)**
(1) = Message number	[ASCII characters]
(2) = Message type	[1 = error, 2 = messages, 10 = warnings, 11 = start requirements, 12 = setup diagnosis]
(3) = Module number	[1...99] ! only for message type 1 -2!

Response Structure

The following table shows the general structure of the PDD4 FI command.

Line 1-n	Column 1	Column 2

Meaning of the Columns

1 = Message is present	[YES, NO]
2 = Criteria analysis exists	[YES, NO]

Example PDD4

Query of the status of a ProVi error, number 1001 from module 3 in control 0.

This message is not present at the moment, and there is a criteria analysis.

FI command		**00_BR_PDD4_1001_1_1**
Line	**Column**	**Answer**
1	1	NO
	2	YES

FI command

Determination of the MessageID of a certain message

BR_PDD5!(1)!(2)!(3)!(4){!(5)}	**(Single Read)**
(1) = POU entity name	[ASCII characters]
(2) = Nw ID	[ASCII characters]
(3) = Message number	[ASCII characters]
(4) = Message type	[1 = error, 2 = messages, 10 = warnings, 11 = start requirements, 12 = setup diagnosis]
(5) = Module number	[1...99] ! only for message type 1 -2!

Note: The separator "!" is used in this command.

Response Structure

The following table shows the general structure of the PDD5 FI command.

Line 1-n	Column 1	...	Column 3

Meaning of the Columns

1 = Message ID	[ASCII characters] (DWORD, decimal)
2 = Message is present	[YES, NO]
3 = Criteria analysis exists	[YES, NO]

Example PDD5

Determination of the MessageID of a ProVi error, number 1001 from module 25.40 mm control 0.

Assumption:
This message is not present at the moment, and there is a criteria analysis.

FI command		**00_BR_PDD5!Station2.Modul3!43493454!1001!1!1**
Line	**Column**	**Answer**
1	1	240872342
	2	NO
	3	YES

9.30 Reading the Size of the PLC Memory: PMI

MWSX device group

Designation **PMI** **P**LC **M**emory **I**nformation

Explanation The current size of the PLC memory is read out.

FI command **CR_PMI** **(Single Read)**

Response Structure One line with two values in BYTE is output:

1. Total memory
2. Free memory available now.

Line 1	Column 1	Column 2

Example PMI Read the current size of the PLC memory at device address 00.

FI command		**00_CR_PMI**
Line	**Column**	**Answer**
1	1	123456
	2	3210

9.31 Issuing SYS Messages Specific to the PCL: PSM

MWSX device groups

Designation **PSM** **P**CL **S**ys **M**essage

Explanation Issues the most important SYS messages regarding the PCL programming interface – required for remote programming.

Note:
The appropriate device address is passed as the write value.

It allows the following SYS messages to be initiated:

- start of PCL download,
- end of PCL download,
- start of PLC online edit,
- end PLC online edit,
- start of PCL declaration change, and
- end of PCL declaration change.

FI command Issue the most important PCL SYS messages.

BW_PSM1_(1) **(Single Write)**

(1) = Requested SYS message [1= start of PCL download
2= end of PCL download
3= start of PCL online edit
4= end of PCL online edit
5= start of PCL declaration change
6= end of PCL declaration change]

Value to be written
device address

Response Structure

The following table shows the general structure of the response to the FI command "PSM1".

Line 1	Column 1	...	Column 8

Value Range/Meaning of Columns

1 =	Status report	[READY=SYS message has been correctly acknowledged by the WIN32 applications] [ERROR=SYS message has NOT been acknowledged by a WIN32 application within the pre-set time]
2 =	Task name (LogInIf name)	[Task name that has triggered the SYS message]
3 =	SYS message number	[contains the issued SYS message number]
4 =	Acknowledgement time	[contains the pre-set acknowledgement time]
5 =	Reference information	[contains, where applicable, the additional information transferred as a write value]
6 =	Length of reference information	[0 where NO reference information has been transferred]
7 =	Where applicable, LOG channel of the FI that has NOT acknowledged	[-- = acknowledgements have been completed in time or the LOG channel number of the WIN32 application that has NOT acknowledged in time]
8 =	Where applicable, task name that has NOT acknowledged in time	[-- = acknowledgements have been completed in time or the task name that has NOT acknowledged in time]

Example PSM1

Issue the SYS message Beginning PCL Download. The reference information, device address 00, is also transferred as a write value.

FI command		**XX_BW_PSM1_1**
Line	**Column**	**Answer**
1	1	READY
	2	WINPCL.EXE
	3	14
	4	30000
	5	00
	6	2
	7	--
	8	--

9.32 Edit PROVI Message Files: PVA

MWSX device groups

Designation **PVA** **PROV**I-Messages **A**ccess

Explanation This write command creates PROVI message files. With this write value, it is possible to decide whether the PROVI messages are to be generated according to the current PLC project, or selectively.

FI command **BW_PVA1** **(Single Write)**

Note: This command is an FI job command.

Value to be written

No write value exists	PROVI message files according to the current PLC project.
Write value exists	List of the requested PROVI message files (separated by a comma) according to the format: [PROVI-Diag-type: module number] Example: 01:01,01:02,02:02

Note: The value to be written is passed to the "acValue" parameter as an ASCII value in the "DataTransfer" routine.

Response Structure The response to the "BW_PVA1" FI command consists of three lines, each with one column. The meaning of the elements is as follows:

- Line 1 = Job ID [01...20]
  (see Chapter "FI Commands for the MPCX Device Group", IFJ).
- Line 2 = FI command
  [String, in accordance with Chapter "Elements of the FI Command"]
- Line 3 = FI Job Error Code (see Chapter "Error Codes")

...

Example PVA1 No write value is passed, i.e. the PROVI message files are generated according to the current PLC project.

FI command		**00_BW_PVA1**
Line	**Column**	**Answer**
1	1	01
2	1	00_BW_PVA1
3	1	0

Explanation The read command returns the most significant information on the created PROVI message files.

FI command **BR_PVA1** **(Single Read)**

....

Response Structure The following table shows the general construction of the answer of the FI command BR_PVA1. For each available PROVI message file, 1 line with 10 columns each is created.

Line 1...n	**Column 1**	**...**	**Column 10**

Value Range/Meaning of Columns

1 =	PROVI diagnosis type	[1..20]
2 =	PROVI diagnosis type designation	[The following designations can be returned: StartCondition, Error, Message, Warning, Setup]
3 =	Module number	[1..99]
4 =	PROVI diagnosis type and module number	[PROVI diagnosis type: module number, see write value for BW_PVA2]
5 =	Complete name of the PROVI message text file	[max. 200 ASCII characters]
6 =	Memory required for PROVI messages in the control	[figure in ASCII format]
7 =	Complete name of the PROVI index file	[max. 200 ASCII characters]
8 =	Memory required for PROVI index files in the control	[figure in ASCII format]
9 =	Total memory (text+index) required in the control	[figure in ASCII format]
10 =	Total memory for ALL PROVI files (text+index) required in the control	[figure in ASCII format]

Example PVA1

The most significant information of 2 available PROVI message files are returned.

FI command		**00_BR_PVA1_1**
Line	**Column**	**Answer**
1	1	1
	2	Error
	3	1
	4	01:01
	5	D:\Programs\indramat\Mtgui\Project_000\ Programmdata\Device_000\Diag\De\ERROR 1.TXT
	6	1345
	7	D:\Program Files\indramat\Mtgui\Project_000\ Programmdata\Device_000\Diag\De\ERROR 1.IDX
	8	234
	9	1579
	10	4491
2	1	2
	2	Message
	3	1
	4	02:01
	5	D:\Programs\indramat\Mtgui\Project_000\ Programmdata\Device_000\Diag\De\ERROR 1.TXT
	6	2456

	7	D:\Programs\indramat\Mtgui\Project_000\ Programmdata\Device_000\Diag\De\ERROR 1.TXT
	8	456
	9	2912
	10	4491

Explanation

This write command transmits PROVI message files into the selected device. Through the write value, it is possible to chose whether ALL or only the PROVI messages selected via the write value are to be transmitted.

FI command

BW_PVA2 **(Single Write)**

Note: This command is an FI job command.

Value to be written

No write value exists	All PROVI message files are transmitted into the selected device
Write value exists	List of the requested PROVI message files (separated by a comma) according to the format: [PROVI-Diag-type: module number] Example: 01:01,01:02,02:02

Note: The value to be written is passed to the "acValue" parameter as an ASCII value in the "DataTransfer" routine.

Response Structure

The response to the "BW_PVA2" FI command consists of three lines, each with one column. The meaning of the elements is as follows:

- Line 1 = Job ID [01...20]
  (see Chapter "FI Commands for the MPCX Device Group", IFJ).
- Line 2 = FI command
  [String, in accordance with Chapter "Elements of the FI Command"]
- Line 3 = FI Job Error Code (see Chapter "Error Codes")

....

Example PVA1

No write value is passed, i.e. all PROVI message files should be transmitted.

FI command		**00_BW_PVA2**
Line	**Column**	**Answer**
1	1	01
2	1	00_BW_PVA2
3	1	0

9.33 Formatted Input / Output of PLC Variables: PVF

MWSX device groups

Designation **PVF** **P**LC **V**ariable **F**ormatted

Explanation Formatted reading and writing of PLC variables, arrays and structures.

FI command Read PLC variables.

CR_PVF_(1)	**(Single Read)**
CC_PVF_(1)	**(Cyclic Read)**
CB_PVF_(1)	**(Break Cyclic Read)**
(1) = Identifier of the PLC variable	[acc. to declaration part of the PLC]

Response Structure One line with one column is output for single variables. For array and structure variables, one line per element is output, depending on the number of elements.

Line 1...n:	Column 1

n = number of elements.

Note: Only defined PLC variables can be read and written. Addressing a non-declared variable results in an error message. A PLC variable can only be read if its data length does not exceed 240 byte. (Refer also to chapter on "Programming" and "Guidelines").

Value Ranges ANSI / ASCII The value range of the response depends on the data type of the variable read. The following table indicates the range in which the results string is to be expected when reading out a single variable and into which C-data type this string can be converted without loss of information:

Data Type	Value Range	Can be converted to C-data type
BOOL	[0;1]	unsigned char
SINT	[-128...127]	char
INT	[-32768...32767]	short
DINT	[2147483648...2147483647]	long
USINT	[0...255]	unsigned char
UINT	[0...65535]	unsigned short
UDINT	[0...4294967295]	unsigned long
BYTE	[0x00...0xFF]	unsigned char
WORD	[0x0000....0xFFFF]	unsigned short
DWORD;	[0x00000000...0xFFFFFFFF]	unsigned long
TIME	[0...4294967295]	unsigned long (msec)
CHAR	[$00...$20,!...~,$7F...$FF]	char
STRING	<String> whereby <String> string is a character string with a maximum of as many characters as are declared for the string in the PLC	Char[xx+1]] +1 i.e. room for the zero byte
REAL	[-3.402823567E+38...3.402823567E+38]	Float

Note: An empty string is identified by two single inverted commas: ' ' (do not confuse with the double inverted commas ")!

All single variables can be part of array and structure variables. The value ranges maintain their validity, even when within structured data types.

Binary Value Range

The value range of the response depends on the data type of the variable read. The following table indicates the value range in which to expect the binary value of a single variable and how many bytes are included in the binary byte sequence:

Data Type	Value Range	Length (bytes)
BOOL	[00_H...01_H]	1
SINT	[80_H...$7F_H$] i.e. –128...127	1
INT	[8000_H (-32768)...$7FFF_H$ (32767)]	2
DINT	[80000000_H (-2147483648)... $7FFFFFFF_H$ (2147483647)]	4
USINT	[00_H (0)...FF_H (255)]	1
UINT	[00_H (0)...$FFFF_H$ (65535)]	2
UDINT	[0...4294967295]	4
BYTE	[0x00...0xFF]	1
WORD	[0x0000....0xFFFF]	2
DWORD;	[0x00000000...0xFFFFFFFF]	4
TIME	[0...4294967295]	4
CHAR	[$00...$20,!...~,$7F...$FF]	1
STRING	<String> whereby <String> string is a character string with a maximum of as many characters as are declared for the string in the PLC	XX+1
REAL	[-3.402823567E+38...3.402823567E+38]	4

Note: Binary array and structure elements are joined together without any spaces between (1-byte alignment).

PLC - Example 1 PVF

Read the value of the PLC variable "STK_TXT" in ASCII format from device address 00.

Assumption:
The "STK_TXT" variable is declared as STRING in the PLC program.

FI command		**00_CR_PVF_STK_TXT/1**
Line	**Column**	**Answer**
1	1	Repeat counter

WinPCL - Example 1 PVF

Read the value of WinPCL variable "STK_TXT" in ASCII format in WinPCL program "Prog" at device address 00.

Assumption:
The WinPCL variable "STK_TXT" is declared in WinPCL program entity "Prog" as STRING.

FI command		**00_CR_PVF_:Prog.STK_TXT/1**
Line	**Column**	**Answer**
1	1	Repeat counter

PLC - Example 2 PVF

Read the value of the PLC array "BEG_END" in ANSI format from device address 00.

Assumption:
The "BEG_END" variable is declared as BYTE with 2 elements in the PLC program.

FI command		**00_CR_PVF_BEG_END/3**
Line	**Column**	**Answer**
1	1	0x00
2	1	0x1F

WinPCL - Example 2 PVF

Read the value of WinPCL array "BEG_END" in ANSI format in WinPCL program "Prog" at device address 00.

Assumption:
The WinPCL variable "BEG_END" is declared in WinPCL program entity "Prog" as BYTE with two elements.

FI command		**00_CR_PVF_:Prog.BEG_END/3**
Line	**Column**	**Answer**
1	1	0x00
2	1	0x1F

PLC - Example 3 PVF

Read the value of the PLC structure "MSTRCT" in ASCII format from device address 00.

Assumption:
The "MSTRCT" variable is declared as a structure in the PLC program as follows:

```
TYP STRUCT
        T1      BOOL
        T2      CHAR
        T3      STRING[16]
        T4      TIME
END
```

FI command		**00_CR_PVF_MSTRCT/1**
Line	**Column**	**Answer**
1	1	0
2	1	A
3	1	ROBOT AXIS X
4	1	2000

WinPCL - Example 3 PVF

Read the value of WinPCL structure "MSTRCT" in ASCII format in WinPCL program "Prog" at device address 00.

Assumption:
The WinPCL variable "MSTRCT" is declared as a structure in WinPCL program entity "Prog" as follows:

```
TYP STRUCT
        T1      BOOL
        T2      CHAR
        T3      STRING[16]
        T4      TIME
END
```

FI command		00_CR_PVF_:Prog.MSTRCT/1
Line	**Column**	**Answer**
1	1	0
2	1	A
3	1	ROBOT AXIS X
4	1	2000

FI command

Write PLC variable.

CW_PVF_(1)	**(Single Write)**
(1) = Identifier of the PLC variable	[acc. to declaration part of the PLC]

Value to be written

Value of data element	[see value ranges]

Note: The value to be written is passed to the "acValue" parameter in the "DataTransfer" routine. The data code of the value is passed to the parameter "ValType".

Response Structure

One line is output with a column for acknowledgement of whether or not the FI command has been executed successfully.

(P_ACK) = **P**ositive **ACK**nowledge	Data element has been set

Value Range of the value to be written in ANSI / ASCII Format

The value ranges agree for the most part with the ANSI / ASCII result-value ranges during read access. ANSI umlauts are thereby converted into ASCII umlauts. Only ASCII umlauts are stored in the control unit. For deviations to this, please refer to the following note:

Note: Strings are enclosed by two single inverted commas ' ' , e.g. 'drill'.

Special characters can be indicated in accordance with DIN-1131 by a $ sign.

The following are used:

$'	'	
$$	$	
$R	\r	(Carriage Return)
$L	\n	(Linefeed)
$P	\f	(Form feed)
$T	\t	(Tab)
$xx	xx refers to a character written as a hexadecimal value, e.g. $20 (space)	

Array and structure elements are separated by a space.

Value Range of the Value to be written in Binary Format

The value ranges agree with the binary result-value range during read access. For deviations to this, please refer to the following note:

PLC - Example 4 PVF

Write into the PLC variable "STK_TXT" at device address 00. The value is passed in ANSI format.

Assumption:
The "STK_TXT" variable is declared as STRING in the PLC program.

FI command		**00_CW_PVF_STK_TXT/3**
Line	**Column**	**Answer**
1	1	(P_ACK)

Value to be written:

Value of data element	'item counter'
Data code	/3

WinPCL - Example 4 PVF

Write into the WinPCL variable "STK_TXT" in WinPCL program "Prog" at device address 00. The value is passed in ANSI format.

Assumption:
The WinPCL variable "STK_TXT" is declared in WinPCL program entity "Prog" as STRING.

FI command		**00_CW_PVF_:Prog.STK_TXT/3**
Line	**Column**	**Answer**
1	1	(P_ACK)

Value to be written:

Value of data element	'item counter'
Data code	/3

PLC - Example 5 PVF

Write into the PLC byte array "BEG_END" at device address 00. The value is passed in ANSI format.

Assumption:
The "BEG_END" variable is declared as a BYTE array with 2 elements in the PLC program.

FI command		**00_CR_PVF_BEG_END/3**
Line	**Column**	**Answer**
1	1	(P_ACK)

Value to be written:

Value of data element	0x20 0x3f
Data code	/3

WinPCL - Example 5 PVF

Write into the WinPCL byte array "BEG_END" in WinPCL program "Prog" at device address 00. The value is passed in ANSI format.

Assumption:
The WinPCL variable "BEG_END" is declared in WinPCL program entity "Prog" as BYTE with two elements.

FI command		**00_CW_PVF_:Prog.BEG_END/3**
Line	**Column**	**Answer**
1	1	(P_ACK)

Value to be written:

Value of data element	0x20 0x3f
Data code	/3

PLC - Example 6 PVF

Write the value of element T3 of the PLC structure "MSTRCT" at device address 00. The string "COUNTER" is transferred in binary format.

Assumption:
The "MSTRCT" variable is declared as a structure in the PLC program as follows:

```
TYP STRUCT
        T1      BOOL
        T2      CHAR
        T3      STRING[16]
        T4      TIME
END
```

FI command		00_CW_PVF_MSTRCT.T3/2
Line	**Column**	**Answer**
1	1	(P_ACK)

Value to be written:

Value of data element	Binary sequence: 43 4F 55 4E 54 45 52 00
Data code	/2

WinPCL - Example 6 PVF

Write the value of element T3 of the WinPCL structure "MSTRCT" at device address 00. The string "COUNTER" is transferred in binary format.

Assumption:
The WinPCL variable "MSTRCT" is declared as a structure in WinPCL program entity "Prog" as follows:

```
TYP STRUCT
        T1      BOOL
        T2      CHAR
        T3      STRING[16]
        T4      TIME
END
```

FI command		00_CW_PVF_:Prog.MSTRCT.T3/2
Line	**Column**	**Answer**
1	1	(P_ACK)

Value to be written:

Value of data element	Binary sequence: 43 4F 55 4E 54 45 52 00
Data code	/2

PLC - Example 7 PVF

Write the value of the PLC structure "MSTRCT" from the structure "mstrct" previously stored in the C program at device address 00.

Assumption:
The "MSTRCT" variable is declared as a structure in the PLC program as follows:

```
TYP STRUCT
        T1      BOOL
        T2      CHAR
        T3      STRING[16]
        T4      TIME
END
```

To exchange binary data in a C program, the following "C" data type can be used:

```
#pragma pack(1)    //Write all elements
                   //without spaces next to each other.
typeder struct
{
      unsigned char T1;
      char         T2;
      char         T3[17]; //Space for zero byte
      unsigned long T4;
} Tymstrct;        // Declare structure
Tymstrct mstrct;   // Apply structure
```

FI command		**00_CW_PVF_MSTRCT/2**
Line	**Column**	**Answer**
1	1	(P_ACK)

Value to be written: address of the C structure.

Value of data element	&mstrct
Data code	/2

WinPCL - Example 7 PVF

Write the value of the WinPCL structure "MSTRCT" from the structure "mstrct" previously stored in the C program at device address 00.

Assumption:
The WinPCL variable "MSTRCT" is declared as a structure in WinPCL program entity "Prog" as follows:

```
TYP STRUCT
      T1      BOOL
      T2      CHAR
      T3      STRING[16]
      T4      TIME
END
```

To exchange binary data in a C program, the following "C" data type can be used:

```
#pragma pack(1)    //Write all elements
                   //without spaces next to each other.
typeder struct
{
      unsigned char T1;
      char         T2;
      char         T3[17]; //Space for zero byte
      unsigned long T4;
} Tymstrct;        // Declare structure
Tymstrct mstrct;   // Apply structure
```

FI command		**00_CW_PVF_:Prog.MSTRCT/2**
Line	**Column**	**Answer**
1	1	(P_ACK)

Value to be written: address of the C structure.

Value of data element	&mstrct
Data code	/2

9.34 ProVi Messages: PVM

MWSX device groups

Designation **PVM** **Pro**Vi **M**essages

Explanation ProVi messages are output. These messages are assigned to a particular message type or module.

FI command Output all ProVi messages.

For optimization reasons, not all data will be applied, therefore the diagnosis server must be informed that the data is required (see ADW).

BR_PVM1_(1){_(2)} **(Single Read)**

BC_PVM1_(1){_(2)} **(Cyclic Read)**

(1) = Message type [1 = error, 2 = messages, 10 = warnings,
11 = start requirements,
12 = setup diagnosis]

(2) = Module number [1...99] ! only for message type 1 -2!

Output first ProVi messages.

BR_PVM2_(1){_(2)} **(Single Read)**

BC_PVM2_(1){_(2)} **(Cyclic Read)**

(1) = Message type [1 = error, 2 = messages, 10 = warnings,
11 = start requirements,
12 = setup diagnosis]

(2) = Module number [1...99] ! only for message type 1 -2!

Response Structure The following table shows the general structure of the FI commands "PVM1" and "PVM2". The number of lines depends on the number of messages pending.

If there are no messages, the number of lines is 0.

Line 1...n	**Column 1**	**...**	**Column 8**

Meaning of the Columns

1 = Message text	[ASCII characters]
2 = Message number	[ASCII characters]
3 = Time stamp day	[mm.dd.yyyy]
4 = Time stamp time	[hh:mm:ss]
5 = Message ID	[ASCII characters] (DWORD, decimal)
6 = Reference text exists	[YES, NO]
7 = Criteria analysis exists	[YES, NO]
8 = Message HTML file	[ASCII characters]

Example PVM1

All ProVi errors from module 3 in control unit 0. There are two messages:

FI command		**00_BR_PVM1_1_3**
Line	**Column**	**Answer**
1	1	Guard not closed
	2	34
	3	01.27.2000
	4	14:56:32
	5	43923028
	6	YES
	7	NO
	8	
2	1	Oil pressure too low
	2	124
	3	01.27.2000
	4	15:03:10
	5	98234039
	6	NO
	7	YES
	8	

Example PVM2

The first ProVi error from module 3 in control unit 0.

There are two messages:

FI command		**00_BR_PVM2_1_3**
Line	**Column**	**Answer**
1	1	Guard not closed
	2	34
	3	01.27.2000
	4	14:56:32
	5	43923028
	6	YES
	7	NO
	8	

FI command

Output the reference information of a ProVi message.

BR_PVM3_(1)_(2){_(3)} **(Single Read)**

(1) = Message ID [ASCII characters]

(2) = Message type [1 = error, 2 = messages,
10 = warnings,
11 = start requirements,
12 = setup diagnosis]

(3) = Module number [1...99] ! only for message type 1 -2!

Response Structure

The following table shows the general structure of the "PVM3" FI command.

Line 1	**Column 1**	**...**	**Column 16**

Meaning of the Columns

1 =	Message text	[ASCII characters]
2 =	Message number	[ASCII characters]
3 =	Error category	[ASCII characters] (empty no category)
4 =	Time stamp day	[mm.dd.yyyy]
5 =	Time stamp hour	[hh:mm:ss]
6 =	Reference text available	[YES, NO]
7 =	Reference text	[ASCII characters]
8 =	Message ID	[ASCII characters] (DWORD, decimal)
9 =	Diagnosis source	[ASCII characters] (PLC, CNC)
10 =	POE name	[ASCII characters]
11 =	Detail name	[ASCII characters] (empty implementation)
12 =	Detail type	[1 = action block, 3 = transition, 4 = implementation]
13 =	Network number	[ASCII characters]
14 =	Variable name	[ASCII characters]
15 =	POU entity name	[ASCII characters]
16 =	POU type	[2 = program, 3 = function block]
17 =	Analysis of criteria available	[YES, NO]
18 =	Message HTML file	[ASCII characters]
19 =	Reference info HTML file	[ASCII characters]

Example PVM3

Reference text of a ProVi error with ID 43923028 from module 3 in control unit 0.

FI command		**00_BR_PVM3_43923028_1_3**
Line	**Column**	**Answer**
1	1	Guard not closed
	2	34
	3	1
	4	01.27.2000
	5	14:56:32
	6	YES
	7	Oil pressure too low Oil pipe leaking or insufficient oil.
	8	43923028
	9	PLC
	10	MODULE3
	11	
	12	4
	13	34
	14	EschutzT
	15	Station2.Module3
	16	3

	17	NO
	18	
	19	D:\Program Files\Indramat\MtGui\Project_000\ProgramData\HMTL\DE\Error34.html

FI command

One after the other, all active ProVi messages are output. In the result, one line each is returned. After expiry of the set time, the next message is returned. The clock frequency can be set via the last parameter. This value can only be set once for the PC. The value transmitted last is always the valid value. Default setting is one second.

BR_PVM4_(1){_(2)_(3)}	**(Single Read)**
BC_PVM4_(1){_(2)_(3)}	**(Cyclic Read)**
(1) = Message type	[1 = error, 2 = messages, 10 = warnings, 11 = start requirements, 12 = setup diagnosis]
(2) = Module number	[1...99] ! only for message type 1 -2!
(3) = Clock frequency	[ASCII characters] Time in ms

Response Structure

The following table shows the general structure of the "PVM4" FI command.

If there are no messages, the number of lines is 0.

Line 1	Column 1	...	Column 8

Meaning of the Columns

1 = Message text	[ASCII characters]
2 = Message number	[ASCII characters]
3 = Time stamp day	[mm.dd.yyyy]
4 = Time stamp time	[hh:mm:ss]
5 = Message ID	[ASCII characters] (DWORD, decimal)
6 = Reference text available	[YES, NO]
7 = Criteria analysis exists	[YES, NO]
8 = Message index (1 = 1. message)	[ASCII characters]
9 = Message HTML file	[ASCII characters]

Example PVM1

ProVi errors from module 3 in control unit 0.

The 2nd message is being output. The clock frequency is to be 2 seconds.

FI command		**00_BR_PVM4_1_3_2000**
Line	**Column**	**Answer**
1	1	Oil pressure too low
	2	124
	3	01.27.2000
	4	15:03:10
	5	98234039
	6	NO
	7	YES
	8	2
	9	

9.35 Download of PLC Retain Variables: PVR

MWSX device groups

Designation **PVR** **P**LC **V**ariable **R**etain Backup

Explanation Download of PLC retain variables.

FI command **BW_PVR1!(1)** **(Single Write)**

(1) = Download file with path details.

Note: File and path details must be enclosed in inverted commas.
The separator "!" is used in this command.

Response Structure The response to the "PVR1" FI command consists of three lines, each with one column. The meaning of the elements is as follows:

- Line 1 = Job ID [01...20]
  (see Chapter "FI Commands for the MPCX Device Group: IFJ").
- Line 2 = FI command
  [String, in accordance with Chapter "Elements of the FI Command"]
- Line 3 = FI Job Error Code
  (see Chapter "Error Codes")

Example PVR1 00_BW_PVR1!"D:\Program Files\Indramat\Mtgui\Temp\download.ini"/3

FI command		**00_BW_PVR1!"D:\Program Files\Indramat\Mtgui\Temp\download.ini"/3**
Line	**Column**	**Answer**
1	1	01
2	1	00_BW_PVR1!"D:\Program Files\Indramat\Mtgui\Temp\download.ini" /3
3	1	0

Structure of Download File The structure of the "download.ini" file used in this example corresponds to an Ini file in Windows.

Note: Care must be taken in the use of upper and lower case letters.

9.36 Upload of PLC Retain Variables: PVR

MWSX device groups

Designation **PVR** **P**LC **V**ariable **R**etain Backup

Explanation PLC retain variables are uploaded via all active processes.

FI command **BR_PVR1!(1)** **(Single Read)**

(1) = Upload file with path details

Note: Enclose file and path details in inverted commas.
The separator "!" is used in this command.

Response Structure

The response to the "PVR1" FI command consists of three lines, each with one column. The meaning of the elements is as follows:

- Line 1 = Job ID [01...20]
  (see Chapter "FI Commands for the MPCX Device Group", IFJ).
- Line 2 = FI command
  [String, in accordance with Chapter "Elements of the FI Command"]
- Line 3 = FI Job Error Code (see Chapter "Error Codes")

Example PVR

00_BR_PVR1_"D:\Program Files\Indramat\Mtgui\Temp\Upload.ini"/3

FI command		**00_BR_PVR1!"D:\Program Files\Indramat\Mtgui\ Temp\upload.ini"/3**
Line	**Column**	**Answer**
1	1	01
2	1	00_BR_PVR1!"D:\Program Files\Indramat\Mtgui\ Temp\upload.ini" /3
3	1	0

Structure of Upload File

The upload file is structured in the Windows – "Ini" format structure.

Note: Care must be taken in the use of upper and lower case letters.

9.37 Reading the PLC Variable Declaration: PVT

MWSX device groups

Designation **PVT** **P**LC **V**ariable **T**ype

Explanation

A PLC variable has a particular type. To evaluate complex variables such as structures and arrays, their components and types must be read out.

Refer also to PVF, Reading Structured PLC Variables.

FI command

Read the PLC variable type.

BR_PVT_(1) **(Single Read)**

(1) = Identifier of the PLC variable [acc. to declaration part of the PLC]

Response Structure

One line with 2 columns is output for each element of the variables.

Line 1...n:	**Column 1**	**Column 2**

n = number of elements.

Value Range/Meaning of Columns

1 = Identifier of the PLC variable [acc. to declaration part of the PLC]

2 = Type [see value range PVF]

Examples:
PLC: Reading of a variable

Assumption:
The "TEST" variable is declared as WORD in the PLC program.

FI command	**00_BR_PVT_TEST**	
Answer		
Line	**Column 1 (Name)**	**Name**
1	TEST	WORD

WinPCL: Reading a Variable

Assumption:
The WinPCL variable "TEST" is declared as WORD in WinPCL program "Prog".

FI command	**00_BR_PVT_:Prog.TEST**	
Answer		
Line	**Column 1 (Name)**	**Name**
1	TEST	WORD

PLC: Reading a Structure

Assumption:
The "TEST1" variable is declared as STRUCT in the PLC program.

```
STRUCT
        E1      BOOL
        E2      INT
        E3      SINT
END
```

FI command	**00_BR_PVT_TEST1**	
Answer		
Line	**Column 1**	**Column 2**
1	TEST1.E1	BOOL
2	TEST1.E2	INT
3	TEST1.E3	SINT

WinPCL: Reading a Structure

Assumption:

The WinPCL variable "TEST1" is declared as STRUCT in WinPCL program "Prog".

```
STRUCT
        E1      BOOL
        E2      INT
        E3      SINT
END
```

FI command	**00_BR_PVT_:Prog.TEST1**	
Answer		
Line	**Column 1**	**Column 2**
1	TEST1.E1	BOOL
2	TEST1.E2	INT
3	TEST1.E3	SINT

PLC: Reading an Array

Assumption:
The "TEST2" variable is declared as ARRAY in the PLC program.

ARRAY [
    0 .. 3
] OF   BOOL

FI command	**00_BR_PVT_TEST2**	
Answer		
Line	**Column 1**	**Column 2**
1	TEST2[0]	BOOL
2	TEST2[1]	BOOL
3	TEST2[2]	BOOL
4	TEST2[3]	BOOL

WinPCL: Reading an Array

Assumption:
The WinPCL variable "TEST2" is declared as ARRAY in WinPCL program "Prog".

ARRAY [
    0 .. 3
] OF   BOOL

FI command	**00_BR_PVT_:Prog.TEST2**	
Answer		
Line	**Column 1**	**Column 2**
1	TEST2[0]	BOOL
2	TEST2[1]	BOOL
3	TEST2[2]	BOOL
4	TEST2[3]	BOOL

PLC: Reading an Array of a Structure

Assumption:
The "TEST3" variable is declared as ARRAY in the PLC program.

ARRAY [
    0 .. 1
] OF   STRUCT1,

where STRUCT1 is declared as follows:

STRUCT
    E1   BOOL
    E2   INT
    E3   SINT
END

END

FI command	**00_BR_PVT_TEST3**	
Answer		
Line	**Column 1**	**Column 2**
1	TEST3[0].E1	BOOL
2	TEST3[0].E2	INT
3	TEST3[0].E3	SINT
1	TEST3[1].E1	BOOL
2	TEST3[1].E2	INT
3	TEST3[1].E3	SINT

WinPCL: Reading an Array of a Structure

Assumption:
The WinPCL variable "TEST3" is declared as ARRAY in WinPCL program "Prog".

```
ARRAY [
          0 .. 1
] OF      STRUCT1,
```

where STRUCT1 is declared as follows:

```
STRUCT
          E1        BOOL
          E2        INT
          E3        SINT
END
```

FI command	**00_BR_PVT_:Prog.TEST3**	
Answer		
Line	**Column 1**	**Column 2**
1	TEST3[0].E1	BOOL
2	TEST3[0].E2	INT
3	TEST3[0].E3	SINT
1	TEST3[1].E1	BOOL
2	TEST3[1].E2	INT
3	TEST3[1].E3	SINT

Assumption:
The data types are output according to IEC1131.

See also command PVF.

9.38 SFC Diagnosis Data: SDD

MWSX device group

Designation **SDD** **S**FC **D**iagnosis **D**ata

Explanation Data for step chain diagnosis is output. Depending on the FI command this data can concern disrupted steps, actions, transitions or a definite ID to display the action or transition.

FI command Output the disrupted step of a step chain.

BR_SDD1!(1)!(2) **(Single Read)**

(1) = Module number [1...99]

(2) = SFC entity name [ASCII characters]

Note: The separator "!" is used in this command.

Response Structure The following table shows the general structure of the FI command "SDD1".

Line 1	**Column 1**	**...**	**Column 7**

Meaning of the Columns

1 = Step name [ASCII characters]

2 = Detail type [1 = action block,
2 =action network, 3 = transition]

3 = Detail name [ASCII characters]

4 = POU ID [ASCII characters]

Rexroth
Bosch Group

5 = Detail morpheme	[ASCII characters] (DWORD, decimal)
6 = Error ID	[ASCII characters] (DWORD, decimal)
7 = POU entity name	[ASCII characters]

Example SDD1

Query disrupted step of the "clamp" chain in module 3 in control unit 0.

FI command		**00_BR_SDD1!3!Station03A.Clamp**
Line	**Column**	**Answer**
1	1	Open
	2	1
	3	Aopen
	4	SFC_1_2
	5	98243823
	6	34985304
	7	Station2.Module3

FI command

Output the faulty action, monitor error or transition of a disrupted step.

BR_SDD2!(1)!(2)!(3)	**(Single Read)**
(1) = Module number	[1...99]
(2) = SFC entity name	[ASCII characters]
(3) = Step name	[ASCII characters]

Note: The separator "!" is used in this command.

Response Structure

The following table shows the general structure of the FI command "SDD2".

Line 1	Column 1	...	Column 6

Meaning of the Columns

1 = Detail type	[1 = action block, 2 =action network, 3 = transition]
2 = Detail name	[ASCII characters]
3 = POU ID	[ASCII characters]
4 = Detail morpheme	[ASCII characters] (DWORD, decimal)
5 = Error ID	[ASCII characters] (DWORD, decimal)
6 = POU entity name	[ASCII characters]

Example SDD2

Query faulty action of the disrupted step "open" of the "clamp" chain in module 3 in control unit 0.

FI command		**00_BR_SDD1!3!Station03A.Clamp_Open**
Line	**Column**	**Answer**
1	1	1
	2	AOpen
	3	SFC_1_2
	4	98243823
	5	34985304
	6	Station2.Module3

FI command

Output the definite ID to display the action, monitor error or transition.

BR_SDD3!(1)!(2)!(3)!(4)	**(Single Read)**
(1) = Module number	[1...99]

(2) = SFC entity name [ASCII characters]

(3) = Detail type [1 = action block, 2 = action network, 3 = transition]

(4) = Detail name [ASCII characters]

Note: The separator "!" is used in this command.

Response Structure

The following table shows the general structure of the FI command "SDD3".

Line 1	Column 1	...	Column 4

Meaning of the Columns

1 = POU ID	[ASCII characters]
2 = Detail morpheme	[ASCII characters] (DWORD, decimal)
3 = Error ID	[ASCII characters] (DWORD, decimal)
4 = POU entity name	[ASCII characters]

Example SDD3

Query ID to display the action "aOpen" of the "clamp" chain in module 3 in control unit 0.

FI command		**00_BR_SDD3!3!Station03A.Clamp!1!aOpen**
Line	**Column**	**Answer**
1	1	SFC_1_2
	2	98243823
	3	34985304
	4	Station2.Module3

FI command

Output the I/O addresses to display a detail.

BR_SDD4!(1)!(2)!(3)!(4) **(Single Read)**

(1) = Module number [1...99]

(2) = SFC entity name [ASCII characters]

(3) = Detail type [1 = action block, 2 = action network, 3 = transition]

(4) = Detail name [ASCII characters]

Note: The separator "!" is used in this command.

Response Structure

The following table shows the general structure of the FI command "SDD4".

Line 1-n	Column 1	Column 2

Meaning of the Columns

1 = Variable morpheme	[ASCII characters] (DWORD, decimal)
2 = I/O address	[ASCII characters]

Example SDD4

Query I/O addresses to display the action "aOpen" of the "clamp" chain in module 3 in control unit 0.

Three variables have an I/O address.

FI command		**00_BR_SDD4!3!Station03A.Clamp!1!aOpen**
Line	**Column**	**Answer**
1	1	98243823
	2	%I3.2.0
2	1	40923423
	2	%Q23.21.7
3	1	34985304
	2	%I100.3.5

FI command

Determine the multilingual comments for displaying a detail.

BR_SDD5!(1)!(2)!(3)!(4) **(Single Read)**

(1) = Module number [1...99]

(2) = SFC entity name [ASCII characters]

(3) = Detail type [1 = action block,
2 =action network,
3 = transition]

(4) = Detail name [ASCII characters]

Note: The separator "!" is used in this command.

Response Structure

The following table shows the general structure of the FI command "SDD5".

Line 1-n	Column 1	Column 2

Meaning of the Columns

1 = Comment morpheme	[ASCII characters] (DWORD, decimal)
2 = New comment	[ASCII characters]

Example SDD5

Query comments to display the action "aOpen" of the "clamp" chain in module 3 in control unit 0.

Two comments are replaced by another text.

FI command		**00_BR_SDD5!3!Station03A.Clamp!1!aOpen**
Line	**Column**	**Answer**
1	1	98243823
	2	Clamp open
2	1	40923423
	2	Clamp closed

FI command

Output the action that has not been performed, or the transition of a step calculated based on the online status.

BR_SDD6!(1)!(2)!(3) **(Single Read)**

(1) = Module number [1...99]

(2) = SFC entity name [ASCII characters]

(3) = Step name [ASCII characters]

Note: The separator "!" is used in this command.

Response Structure

The following table shows the general structure of the FI command "SDD6".

Line 1	Column 1	...	Column 6

Meaning of the Columns

1 = Detail type	[1 = action block, 3 = transition]
2 = Detail name	[ASCII characters]
3 = POU ID	[ASCII characters]
4 = Detail morpheme	[ASCII characters] (DWORD, decimal)
5 = Error ID	[ASCII characters] (DWORD, decimal)
6 = POU entity name	[ASCII characters]

Example SDD6

Query the action that has not been performed for the step "open" of the "clamp" chain in module 3 in control unit 0.

FI command		**00_BR_SDD6!3!Station03A.Clamp_Open**
Line	**Column**	**Answer**
1	1	1
	2	AOpen
	3	SFC_1_2
	4	98243823
	5	34985304
	6	Station2.Module3

FI command

Determine the module number of a step chain.

BR_SDD7!(1)	**(Single Read)**
(2) = SFC instances name	[ASCII characters]

Note: The separator "!" is used in this command.

Response Structure

The following table shows the general structure of the FI command "SDD7".

Line 1	Column 1

Meaning of the Columns

1 = Module number [1...99]

Example SDD7

Inquiry of the module of the chain "clamp".

FI command		**00_BR_EXD1!Station03A.Clamp**
Line	**Column**	**Answer**
1	1	3

9.39 Set the Device Status Information: SDS

MWSX device groups

Designation **SDS** **S**et **D**evice **S**tatus

Explanation

By this command, the device status information can be set; here, the configuration file IND_DEV.INI is adjusted as well.

Note: When this command is transmitted, the following system messages are generated:
MSG_DEVICEOFF or MSG_DEVICE_ON !

FI command

With this command, the device status information of **ALL** defined devices can be set.

BW_SDS1_(1)	**(Single Write)**
(1) = Device status information to be set	0 = Device status information OFF 1 = Device status information ON

Response Structure

The following table shows the general structure of the response to the "SDS1" FI command.

Line 1	**Column 1**

Value Range/Meaning of Columns

1 =	Status report	[(P_ACK)]

Example SDS1

Set device status information to OFF for **ALL** defined devices.

FI command		**00_BW_SDS1_0**
Line	**Column**	**Answer**
1	1	(P_ACK)

FI command

With this command, the device status information for a selected device can be set.

BW_SDS2_(1)	**(Single Write)**
(1) = Device status information to be set	0 = Device status information OFF 1 = Device status information ON

Response Structure

The following table shows the general structure of the response to the "SDS2" FI command.

Line 1	**Column 1**

Value Range/Meaning of Columns

1 =	Status report	[(P_ACK)]

Example: SDS2

Set device status information to OFF for the selected device 00.

FI command		**00_BW_SDS2_0**
Line	**Column**	**Answer**
1	1	(P_ACK)

9.40 Setting the FI Exclusive Mode: SEM

MWSX device group

Designation **SEM** **S**et FI **E**xclusive **M**ode

Explanation

This command is used to activate FI Exclusive mode for the selected device address.

FI Exclusive mode: In this mode, **ALL** the processes logged in at the FI – **with the exception** of the process issuing the SEM command – are blocked from data communication with this device address. This mode is used for example for data communication which must NOT be interrupted by other data requests. However, it is **imperative** that this FI Exclusive mode is deleted once more through the DEM command.

FI command **BW_SEM1** **(Single Write)**

Response Structure The following table shows the general structure of the response to the FI command "BW_SEM1". A line of 1 column is output.

Line 1	Column 1

Value Range/Meaning of Columns 1 = Status message (P_ACK) (P_ACK)

Example SEM1 Activate FI Exclusive mode for device address 0.

FI command		**00_BW_SEM1**
Line	**Column**	**Answer**
1	1	(P_ACK)

9.41 Sequencer Data: SFD

MWSX device groups

Designation **SFD** **SFC D**ata

Explanation Data for a step chain is outputted. Depending on the FI command this can concern a step chain comment, POE name, step comment, maximum time, action / transition / monitor error name (comment), qualifier and time value.

FI command Query the data for a step chain.

BR_SFD1!(1)!(2)	**(Single Read)**
(1) = Module number	[1...99]
(2) = SFC entity name	[ASCII characters]

Note: The separator "!" is used in this command.

Response Structure The following table shows the general structure of the "SFD1" FI command.

Line 1	Column 1	Column 2

Meaning of the Columns

1 = Step chain comment	[ASCII characters]
2 = POE name	[ASCII characters]

Example SFD1

Query data of the "clamp" chain in module 3 in control unit 0.

FI command		**00_BR_SFD1!3!Station03A.Clamp**
Line	**Column**	**Answer**
1	1	Clamping device
	2	CLAMP

FI command

Query the data of a step.

BR_SFD2!(1)!(2)!(3)	**(Single Read)**
(1) = Module number	[1...99]
(2) = SFC entity name	[ASCII characters]
(3) = Step name	[ASCII characters]

Note: The separator "!" is used in this command.

Response Structure

The following table shows the general structure of the "SFD2" FI command. The number of lines depends on the number of actions and transitions.

If there are no details the line number is 1.

Line 1	**Column 1**	**...**	**Column 3**
Line 2...n:	**Column 1**	**...**	**Column 6**

Meaning of the Columns

Line 1

1 = Step comment	[ASCII characters]
2 = Maximum time	[ASCII characters]
3 = Minimum time	[ASCII characters]

Line 2...n:

1 = Detail type	[1 = action block, 3 = transition]
2 = Name	[ASCII characters]
3 = Comment	[ASCII characters]
4 = Boolean variable	[YES, NO]
5 = Qualifier	[ASCII characters]
6 = Time value	[ASCII characters]

Example SFD2

Data for the step "Open" in the "clamp" chain in module 3 on control unit 0.

FI command		**00_BR_SFD2!3!Station03A.Clamp!Open**
Line	**Column**	**Answer**
1	1	Open clamping device
	2	T#5s
	3	
2	1	1
	2	aOpen
	3	Clamp open
	4	NO
	5	D
	6	T#3s

FI command		**00_BR_SFD2!3!Station03A.Clamp!Open**
Line	**Column**	**Answer**
3	1	3
	2	tOpen
	3	Clamping device is open
	4	NO
	5	
	6	

FI command

Output the data for a detail.

BR_SFD3!(1)!(2)!(3)!(4)	**(Single Read)**
(1) = Module number	[1...99]
(2) = SFC entity name	[ASCII characters]
(3) = Detail type	[1 = action block, 2 = action network, 3 = transition]
(4) = Detail name	[ASCII characters]

Note: The separator "!" is used in this command.

Response Structure

The following table shows the general structure of the "SFD3" FI command.

Line 1	**Column 1**	**Column 2**

Meaning of the Columns

1 = Comment	[ASCII characters]
2 = Boolean variable	[YES, NO]

Example SFD3

Data for the action "aOpen" in the "clamp" chain in module 3 in control unit 0.

FI command		**00_BR_SFD3!3!Station03A.Clamp!aOpen**
Line	**Column**	**Answer**
1	1	Clamp open
	2	NO

9.42 Sequencer Messages: SFE

MWSX device groups

Designation **SFE** **SFC E**rror

Explanation The sequencer messages of a module are output.

FI command

Output all SFC messages.

For reasons of optimization, not all data is called up. Accordingly, the Diagnosis server must be informed that the data is required (see ADW).

BR_SFE1_(1)	**(Single Read)**
BC_SFE1_(1)	**(Cyclic Read)**
(1) = Module number	[1...99]

Output first SFC messages.

BR_SFE2_(1)	**(Single Read)**

BC_SFE2_(1) **(Cyclic Read)**

(1) = Module number [1...99]

Response Structure

The following table shows the general structure of the FI commands "SFE1" and "SFE2". The number of lines depends on the number of messages pending.

If there are no messages, the number of lines is 0.

Line 1...n:	Column 1	...	Column 7

Meaning of the Columns

1 = Message text	[ASCII characters]
2 = SFC entity name	[ASCII characters]
3 = Step name	[ASCII characters]
4 = Time stamp day	[mm.dd.yyyy]
5 = Time stamp time	[hh:mm:ss]
6 = Type of error	[1 = time error, 2 = monitor error, 3 = monitor event]
7 = Is there condition analysis?	[YES, NO]

Example SFD1

All SFC messages from module 2 in control unit 0.

There are two messages:

FI command		**00_BR_SFE1_2**
Line	**Column**	**Answer**
1	1	TIME ERROR: Chain: chucking Step: up malfunction
	2	Station03A.Clamp
	3	Open
	4	01.27.2000
	5	11:56:32 AM
	6	1
	7	YES
2	1	ASSY ERROR: Chain: drilling Step: down malfunction
	2	Station02A.Drill
	3	Down
	4	01.27.200
	5	13:03:12
	6	2
	7	NO

Example SFE2

First SFC message from module 2 in control unit 0.

There are two messages.

FI command		**00_BR_SFE2_2**
Line	**Column**	**Answer**
1	1	TIME ERROR: Chain: chucking Step: up malfunction
	2	Station03A.Clamp
	3	Open
	4	01.27.2000
	5	14:56:32
	6	1
	7	YES

9.43 Sequencer Mode: SFM

MWSX device groups

Designation **SFM** **S**F**C** **M**ode

Explanation Queries step chain mode.

FI command Query the mode of a step chain.

BR_SFM1!(1)!(2)	**(Single Read)**
BC_SFM1!(1)!(2)	**(Cyclic Read)**
(1) = Module number	[1...99]
(2) = SFC entity name	[ASCII characters]

Note: The separator "!" is used in this command.

Response Structure The following table shows the general structure of the "SFM1" FI command.

Line 1	Column 1

Meaning of the Columns

1 = Mode [1 = time error, 2 = monitor error, 3 = monitor event, 10 = stop, 11 = auto, 12 = manual, 13 = jog]

Example SFM1 Query mode of the "clamp" chain in module 3 in control unit 0.

FI command		**00_BR_SFM1!3!Station03A.Clamp**
Line	**Column**	**Answer**
1	1	1

9.44 Software Installation Data: SID

MWSX device groups

Designation **SID** **S**oftware **I**nstallation **D**ata

Explanation Information is returned regarding installation. This information includes installation paths, context information, the software versions used, DLL mode, plus service pack and release information.

FI command Read-in the installation data and/or the software version data

BR_SID1 (Single Read)

Response Structure

One line with 16 columns is output for the returned values.

Line 1	Column 1	...	Column 16

Meaning of the Columns

1 = Basic directory	[EXE files of the BOF]
2 = FI installation directory	[FI directory]
3 = Data directory	[in accordance with BOF]
4 = GBO version	[from INDRAMAT.ini]
5 = IF-DLL mode	[from INDRAMAT.ini]
6 = IF version	[from INDRAMAT.ini - from DLL mode 400]
7 = Service package info	[from INDRAMAT.ini - from DLL mode 420]
8 = Release info	[from INDRAMAT.ini - from DLL mode 420]
9 = IF-Build-Info	[in accordance with Build process]
10 = Current context name	[in accordance with the installation]
11 = Physical installation path	[in accordance with the installation]
12 = Complete IF version indication string	
13 = WinPCL build number	[in accordance with WinPCL]
14 = Version number of the PLC compiler	[in accordance with WinPCL]
15 = Version number of the PLC linker	[in accordance with WinPCL]
16 = Version number of the PLC data basis	[in accordance with WinPCL]
17 = Platform version	

Example SID1

Return information on the current installation.

FI command		00_BR_SID1
Line	**Column**	**Answer**
1	1	--
	2	D:\Program Files\Indramat\MTGUI\Bin
	3	--
	4	005-22Vxx
	5	07.20
	6	07V00
	7	--
	8	--
	9	Build 3124 Mar 6 2003 08:53:55
	10	MTGUI_0-23T01 B3327
	11	D:\Program Files\Indramat\MTGUI\
	12	FI: 07V00 DLL-Mode: 07.20 Build 3124 Mar 6 2003 08:53:55
	13	347.15.4.11
	14	771

	15	515
	16	78
	17	Platform: 02V01 Build: 3214

Note: Refer to FI command "PHD" of the MPCX for working with absolute paths.

9.45 PLC Long Identification: SLI

MWSX device groups

Designation **SLI** PLC (**S**PS) **L**ong **I**dentification

Explanation Returns the unit data from the PLC long identification.

FI command Read PLC long identification.

BR_SLI **(Single Read)**

Response Structure One line with 15 columns is output for the returned values.

Line 1	Column 1	Column...	Column 15

Value Range/Meaning of the Columns

1 =	device address	[00...15]
2 =	program number	[01...99]
3 =	Project name	[max. 8 ASCII characters]
4 =	Program name	[max. 8 ASCII characters]
5 =	User name	[acc. to password entry]
6 =	Program length	[bytes]
7 =	Compilation time	[LONG] (coded in long value)
8 =	Compilation date	[8 ASCII characters]
9 =	Compilation time	[8 ASCII characters]
10 =	Download time	[LONG] (coded in long value)
11 =	Download date	[8 ASCII characters]1
12 =	Download time	[8 ASCII characters]
13 =	Version of PLC long identification	[LONG]
14 =	RUN flags	[HEX value]
15 =	Compiler info	[LONG]

Example SLI Read the unit data from the PLC long identification.

FI command		00_BR_SLI
Line	**Column**	**Answer**
1	1	02
	2	01
	3	--
	4	MOT12
	5	TEST
	6	17672
	7	630163960
	8	15.12.99
	9	17:15:48
	10	630163961
	11	15.12.99
	12	17:15:50
	13	2
	14	0x0000
	15	13

Reference to Literature see chapter entitled "Literature" [30].

9.46 Requesting Watch List Allocations: WLA

MWSX device groups

Designation **WLA** **W**atch **L**ist **A**llocation

Explanation Requests free watch list allocations. A maximum of ten free watch list allocations can be requested with one FI command.

BR_WLA1_(1)	**(Single Read)**
(1) =Number of requested free watch list allocations	The required number of free watch list allocations is identified here. The allowed value range: 1..10.

Response Structure The following table shows the general structure of the response to the FI command "WLA1".

Line 1	Column 1	...	Column n

Value Range/Meaning of Columns

1 =	1. free watch list allocation	Value range: 1..15
2 =	2. free watch list allocation	Value range: 1..15
3 =	3. free watch list allocation	Value range: 1..15
n =	nth free watch list allocation	Value range: 1..15

Example WLA1

Request four free watch list allocations.

Assumption:
Watch list allocations 3 and 5 are already assigned!

FI command		**00_BR_WLA1_4**
Line	**Column**	**Answer**
1	1	0
	2	1
	3	2
	4	4

9.47 Freeing Watch List Allocations: WLF

MWSX device groups

Designation **WLF** **W**atch **L**ist **F**ree

Explanation Previously requested watch list allocations are freed again.

FI command Free ALL assigned watch list allocations for the selected device.

BR_WLF1 **(Single Read)**

Note: The FI command "WLF1" frees ALL assigned watch list allocations, including those of other WIN32 applications.

Response Structure The following table shows the general structure of the response to the FI command "WLF1".

Line 1	Column 1	...	Column n

Value Range/Meaning of Columns

1 =	1. freed watch list allocation	Value range: 1..15
2 =	2. freed watch list allocation	Value range: 1..15
3 =	3. freed watch list allocation	Value range: 1..15
n =	nth freed watch list allocation	Value range: 1..15

Example WLF1

Free ALL assigned watch list allocations.

Assumption:
The following watch list numbers have been allocated: 0,1,2,3.

FI command		**00_BR_WLF1**
Line	**Column**	**Answer**
1	1	0
	2	1
	3	2
	4	3

FI command Free the required watch list allocations for a selected device.

BR_WLF2_(1)_{(2)..(10)}	**(Single Read)**
(1)..(10) = List of watch list allocations to be released	A maximum of 10 watch list allocations can be transferred here to be freed again.

Response Structure

The following table shows the general structure of the response to the FI command "WLF2".

Line 1	Column 1	...	Column n

Value Range/Meaning of Columns

1 =	1. freed watch list allocation	Value range: 1..15
2 =	2. freed watch list allocation	Value range: 1..15
3 =	3. freed watch list allocation	Value range: 1..15
n =	nth freed watch list allocation	Value range: 1..15

Example WLF2

Free required watch list allocations:
Assumption: Watch list allocations 0,3,4, and 8 have first been requested using the FI command "WLA1".

FI command		00_BR_WLF2_0_3_4_8
Line	**Column**	**Answer**
1	1	0
	2	3
	3	4
	4	8

10 FI Commands MWAX Device Group (MTA 200)

The FI Commands described in this chapter are valid for the MWAX device group. In this device group, the following types are listed as well as possible device addresses:

Group	Accompanying Types	Address
MWAX	MTA200-P (MTA 200-controller)	[00...63]

Note: Please note that the device address must be set before the respective FI command, e.g. "00_BR_ASM1" (refer also here to the chapter "Elements of the FI Command").

10.1 Current Axis Speed (Spindle Speed): AAS

MWAX device group

Designation **AAS** Actual **A**xis **S**peed

Explanation The spindle speed of a selected axis is read out. The "AAS2" FI command refers to the physical axis number.

FI command Output the current axis speed of the selected device related to the physical axis number.

BR_AAS2_(1) **(Single Read)**

(1) = Physical axis number [1..0.16, according to settings of the system parameters]

Response Structure The following table shows the general structure of the response to the "AAS2" FI command. One line is output with 4 columns for the axis designation, spindle speed, unit and – for compatibility reasons – once more the axis speed.

Line 1	Column 1	Column 2	Column 3	Column 4

Meaning of the Columns

1 = Axis name	[according to settings of axis parameters]
2 = Speed	[to one decimal place]
3 = Unit	1/min
4 = Speed	[as in column 2]

Note: If the selected axis is not defined, or if it is no spindle, then the response in all columns is [--].

Example AAS2 Read the speed of axis A. Axis A is defined as a spindle.

FI command	**00_BR_AAS2_1**			
Answer				
Line	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	A	5000.0	1/min	5000.0

10.2 Reading the Current NC Databases: ADB

MWAX device group

Designation **ADB** Get **A**ctual **D**ata **B**ase

Explanation This command is used to read out a list of the currently set up NC databases.

FI command Read a list of databases.

BR_ADB1 **(Single Read)**

Response Structure The following table shows the general structure of the response to the FI command "ADB1".

Line 1...n	Column 1	Column 2	Column 3

Value Range/Meaning of Columns

Line 1	1	Number of all databases
	2	Number of user databases
	3	Number of system databases
Line 2 - n	1	Database number
	2	Database name
	3	Database path

Example ADB1 Read out all set up databases.

FI command		**00_BR_ADB1**
Line	**Column**	**Answer**
1	1	5
	2	2
	3	3
2	1	1
	2	Default user path
	3	D:\Program Files\Indramat\MTGUI\ExternalApp\ANDRON\CustomData\Database\User1\
3	1	10
	2	NC_Local
	3	D:\Program Files\Indramat\MTGUI\ExternalApp\ANDRON\CustomData\Database\User1\
4	1	90
	2	Local control
	3	D:\Program Files\Indramat\MTGUI\ExternalApp\ANDRON\CustomData\Database\User1\
5	1	91
	2	Cycles
	3	D:\Program Files\Indramat\MTGUI\ExternalApp\ANDRON\SystemData\Database\Cycles\
6	1	92
	2	Tools
	3	D:\Program Files\Indramat\MTGUI\ExternalApp\ANDRON\SystemData\Database\Tools\

10.3 MTA 200 Messages: ADM

MWAX device group

Designation **ADM** MT**A** 200 **M**essages

Explanation MTA 200 NC messages are output. These messages are assigned to a specific module and message type.

FI command Output all MTA 200 messages.

For reasons of optimization, not all data is called up. Accordingly, the Diagnosis server must be informed that the data is required (see ADW).

BR_ADM1_(1)_(2)	(Single Read)
BC_ADM1_(1)_(2)	(Cyclic Read)
(1) = Message type	[1 = error, 2 = messages]
(2) = Module number	[1...99]

Output of first MTA 200 message.

BR_ADM2_(1)_(2)	(Single Read)
BC_ADM2_(1)_(2)	(Cyclic Read)
(1) = Message type	[1 = error, 2 = messages]
(2) = Module number	[1...99]

Response Structure The following table shows the general structure of the FI commands "ADM1" and "ADM2". The number of lines depends on the number of messages pending.

If there are no messages, the number of lines is 0.

Line 1...n	Column 1	...	Column 6

Meaning of the Columns

1 = Message text	[ASCII characters]
2 = Message number	[0..32768]
3 = Time stamp day	[mm.dd.yyyy]
4 = Time stamp time	[hh:mm:ss]
5 = Message group	[1..9999]
6 = Reference text exists	[YES, NO]

Example ADM1

All MTA 200 errors from module 3 in control unit 0.

There are two messages:

FI command		**00_BR_ADM1_1_3**
Line	**Column**	**Answer**
1	1	24 volt supply absent
	2	1002
	3	01.27.2000
	4	14:56:32
	5	12
	6	YES
2	1	Program stop
	2	152
	3	01.27.2000
	4	15:03:10
	5	13
	6	NO

Example ADM2

The first MTA 200 error from module 3 in control unit 0.

There are two messages:

FI command		**00_BR_ADM2_1_3**
Line	**Column**	**Answer**
1	1	24 volt supply absent
	2	1002
	3	01.27.2000
	4	14:56:32
	5	12
	6	YES

FI command

Output the additional information of a MTA 200 message.

BR_ADM3_(1)_(2)_(3)	**(Single Read)**
(1) = Module number	[1...99]
(2) = Message number	[0..32768]
(3) = Message group	[1..9999]

Response Structure

The following table shows the general structure of the "ADM3" FI command.

Line 1	**Column 1**	**...**	**Column 7**

Meaning of the Columns

1 = Message text	[ASCII characters]
2 = Message number	[0..32768]
3 = Time stamp day	[mm.dd.yyyy]
4 = Time stamp time	[hh:mm:ss]
5 = Message group	[1..9999]
6 = Additional text exists	[YES, NO]
7 = Additional text	[ASCII characters]

Example ADM3 Additional text of an MTA 200 error in module 3 in control unit 0.

FI command		**00_BR_ADM3_3_1002_12**
Line	**Column**	**Answer**
1	1	24 volt supply absent
	2	1002
	3	01.27.2000
	4	14:56:32
	5	12
	6	YES
	7	Switch on voltage

10.4 Active Diagnosis Window ADW

MWAX device group

Designation **ADW** Active **D**iagnosis **W**indow

Explanation Indicates the window types for which data is required.

For improved performance and with some diagnosis windows, not all the data is called up each time a diagnosis is performed, but only when the data is actually required.

Through this FI command, the diagnosis server can be informed that the data of the respective window type is required.

This command has to be issued at least every 3 seconds while the data is required. If this command is not issued any more, calling-up of all data will stop.

FI command Indicates the window types for which data is required.

BW_ADW1_(1){_(2)}	**(Single Write)**
(1) = Type of diagnosis window	[1 = CNC error, 2 = sequence errors, 3 = general errors, 4 = messages, 10 = start requirements, 11 = warnings, 12 = setup diagnosis]
(2) = Module number	[1...99] ! only for window type 1 -4!

Example ADW1 Call up data for CNC error in controller 0 module 1.

FI command	**00_BW_ADW1_1_1**

10.5 Current Feedrate: AFR

MWAX device group

Designation **AFR** Actual **F**eed **R**ate

Explanation The current value of the feed rate is output. In an NC program, the details of the feedrate are expressed by means of a feedrate value.

FI command The current value of the feedrate is output.

Using the optional second parameter it is possible to pre-select conversion of the result into mm or inches.

BR_AFR_(1){_(2)}	**(Single Read)**
(1) = CNC process number	[0]
(2) = Required measurement system (opt.)	[mm, inch]

Note: Only "0" can be input as a CNC process number. This is necessary for reasons of compatibility of the FI command.

Response Structure

The following table shows the general structure of the response to the FI command "AFR". One line with three columns is output for the identifier, the current value of the feedrate and the unit.

Line 1	Column 1	...	Column 3

Value Range/Meaning of Columns

1 = Identifier	[F = feedrate]
2 = Value	[Accuracy: one decimal place]
3 = Unit	[depending on the settings of the basic programming unit, e.g. the preset command]

Example AFR

Reads the current feedrate in CNC process 0 of device address 00.

FI command	**00_BR_AFR_0**		
Answer			
Line	**Column 1**	**Column 2**	**Column 3**
1	F	30000.0	[mm/min]

Example AFR

Reads the current feedrate in CNC process 0 of device address 00. The displayed value is to be converted into inch/min:

FI command	**00_BR_AFR_0_inch**		
Answer			
Line	**Column 1**	**Column 2**	**Column 3**
1	F	1181.1	[inch/min]

10.6 Active Mechanism Messages: AMM

MWAX device group

Designation **AMM** Active **M**echanism **M**essages

Explanation

Messages regarding active mechanism errors and mechanism diagnostics are output. These messages are assigned to a particular mechanism or process. Depending on the FI command, the device address, device name, mechanism number, mechanism name, type of message, message source, messages group, message number and messages text are all output.

FI command

Output mechanism messages currently pending.

BR_AMM7	**(Single Read)**
BC_AMM7	**(Cyclic Read)**
BB_AMM7	**(Break Cyclic Read)**

Note: The "AMM7" FI command refers only to devices within the MWAX device group. You should therefore make sure that only MTA devices are addressed via the system address.

Response Structure

The following table shows the general structure of the response to the FI command "AMM7". The answer consists of one up to a maximum of n=512 lines, each with 11 columns. The order of the individual error messages is oriented towards the time stamp, i.e. the oldest (triggering) error message is inserted into the first line. The maximum content for a result may not exceed 56 kbyte.

Line 1...n:	Column 1	Column...	Column 11

Value Range/Meaning of the Columns

1 =	Device address	[00...15]
2 =	Device name	[max. 32 ASCII characters]
3 =	Mechanism number	[0, default value always 0]
4 =	Mechanism name	[max. 28 ASCII characters, default value always the MTA process]
5 =	Type of message	[F = Fault/Error, D = Diagnosis]
6 =	Message source	[CNC, PLC, default value always "CNC"]
7 =	Message group	[1...9999]
8 =	Message number	[0...32768]
9 =	Message text	[max. 1024 ASCII characters]
10 =	Additional text	[X = exists, - - = does not exist, Default value - - does not exist (compatibility with Bosch Rexroth control units)]
11 =	2 bytes of additional information for the message number	[is required to resolve the information "@", default value "0" (compatibility with Bosch Rexroth control units)]

Example AMM7 Read the current mechanism messages of device address 3 (MTA 200).

FI command		**00_BR_AMM7**
Line	**Column**	**Answer**
1	1	03
	2	Crankshaft grinding machine
	3	0
	4	MTA process
	5	F
	6	CNC
	7	1
	8	5
	9	Programming error
	10	- -
	11	0
2	1	03
	2	Crankshaft grinding machine
	3	0
	4	MTA process
	5	F
	6	CNC
	7	1
	8	6
	9	Cycle point error
	10	- -

	11	0
3	1	03
	2	Crankshaft grinding machine
	3	0
	4	MTA process
	5	F
	6	CNC
	7	1
	8	19
	9	Emergency stop with immediate stop
	10	- -
	11	0

10.7 Actual (Current) Position of an Axis: APO

MWAX device group

Designation **APO** Actual **A**xis **PO**sition

Explanation The actual position of a selected axis is read out. The FI command "APO1" returns the position of an axis, related to the code of the axis meaning. On the other hand, the FI command "APO2" returns the position of an axis, related to the physical axis number.

FI command Output the position of the selected axis of the device specified, related to the code of the axis meaning.

Using the optional fourth parameter it is possible to pre-select conversion of the result into mm or inches.

CR_APO1_(1)_(2)_(3){_(4)}	**(Single Read)**
CC_APO1_(1)_(2)_(3){_(4)}	**(Cyclic Read)**
CB_APO1_(1)_(2)_(3){_(4)}	**(Break Cyclic Read)**
(1) = NC process number	[0]
(2) = Axis meaning	[1...16, the axis meaning corresponds to the physical axis number]
(3) = System of coordinates	[1 = machine coordinates 2 = program coordinates 3 = relative coordinates]
(4) = Required measurement system (opt.)	[mm, inch]

FI command Output the position of the selected axis of the device specified, related to the physical axis number.

Using the optional third parameter it is possible to pre-select conversion of the result into mm or inches.

CR_APO2_(1)_(2){_(3)}	**(Single Read)**
CC_APO2_(1)_(2){_(3)}	**(Cyclic Read)**
CB_APO2_(1)_(2){_(3)}	**(Break Cyclic Read)**
(1) = Physical axis number	[1..0.16, according to settings of the system parameters]
(2) = System of coordinates	[1 = machine coordinates

2 = program coordinates
3 = relative coordinates]

(3) = Required measurement system [mm, inch] (opt.)

Response Structure

The following table shows the general structure of the response to the FI commands "APO1" and "APO2". One line is output with 4 columns for the axis designation, position, unit and the position limited to "indicated decimal places".

Line 1	Column 1	Column 2	Column 3	Column 4

Value Range/Meaning of Columns

1 = Axis name	[according to settings of axis parameters]
2 = Position	[according to settings of process parameters]
3 = Unit	[according to settings of process parameters: mm, inch]
4 = Position	[as in column 2]

Note: If the selected axis is not defined then the response in all columns is [--].

Example APO1

Read the current position of the Z axis in machine coordinates in NC process 0 of device address 00. Values are displayed in the basic measurement system.

FI command	**00_CR_APO1_0_3_1**			
Answer				
Line	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	Z	-1.2345	[mm]	-1.2345

Example APO1

Read the current position of the Z axis in machine coordinates in NC process 0 of device address 00. Values are displayed in inches.

FI command	**00_CR_APO1_0_3_1_inch**			
Answer				
Line	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	Z	-0.0486	[inch]	-0.0486

Example APO2

Reads the current position of the Z axis (physical axis number = 3) in machine coordinates for the device address 00. The values are indicated in the basic measuring system.

FI command	**00_CR_APO2_3_1**			
Answer				
Line	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	Z1	-1.2345	[mm]	-1.2345

Reference to Literature

See chapter entitled "Literature" [16].

10.8 Active System Error Messages: ASM

MWAX device group

Designation **ASM** Active **S**ystem **M**essages

Explanation The active device information (system errors, device statuses) that affect the functioning of the entire electrical device are output. Depending on the FI command, the device address, device name, message number, type of message, short text and additional text are all output. Access to device information (system errors, device statuses) only refers to the PLC part (ISP 200).

FI command Output the device information (system errors, device statuses) currently pending for all active devices from the MWAX device group.

BR_ASM1	**(Single Read)**
BC_ASM1	**(Cyclic Read)**
BB_ASM1	**(Break Cyclic Read)**

Note: The "ASM1" FI command refers to all devices within this device group. Therefore, any valid device address can be indicated in the command line (see example "ASM1").

Response Structure The following table shows the general structure of the response to the FI command "ASM1". The number of lines (1 .. n=15) depends on the number of defined devices. Each line consists of 7 columns for the device address, device name, message number, message status, short text and indication of whether there is an additional text for this device information (system errors, device statuses).

Line 1...n	Column 1	...	Column 8

Value Range/Meaning of Columns

1 =	Device address	[00...15]
2 =	Device name	[max. 32 ASCII characters]
3 =	Message number	[0...150]
4 =	Type of message	[F = fault/error, D = diagnosis]
5 =	Short text	[max. 54 ASCII characters]
6 =	Reference text	[x= exists, -- = does not exist]
7 =	2 bytes of additional information for the message number	is required to resolve the information "@" (see ASM5)
8 =	File name for additional information	e.g. in HTML format

Example ASM1 Read the current device information (system errors, device statuses) of all defined devices within the MWAX device group.

Assumption:
The following three devices are defined:

- Device address 01,
- Device address 07 and
- Device address 10.

FI command		07_BR_ASM1
Line	**Column**	**Answer**
1	1	01
	2	Drill center
	3	71
	4	F
	5	PLC battery voltage too low.
	6	X
	7	0
	8	
2	1	07
	2	Milling center 1
	3	74
	4	F
	5	SLM time monitoring
	6	X
	7	0
	8	
3	1	10
	2	Milling center 2
	3	1
	4	D
	5	Error has been corrected.
	6	X
	7	0
	8	

FI command

Output the currently pending device information (system errors, device statuses) of the selected device from the MWAX device group.

BR_ASM2	**(Single Read)**
BC_ASM2	**(Cyclic Read)**
BB_ASM2	**(Break Cyclic Read)**

Response Structure

The following table shows the general structure of the response to the FI command "ASM2". The response consists of a line of 7 columns for the device address, device name, message number, message status, short text and indication of whether there is an additional text for this device information (system errors, device statuses).

Line 1...n	Column 1	...	Column 7

Value Range/Meaning of Columns

1 =	Device address	[00...15]
2 =	Device name	[max. 32 ASCII characters]
3 =	Message number	[0...150]
4 =	Type of message	[F = fault/error, D = diagnosis]
5 =	Short text	[max. 54 ASCII characters]
6 =	Additional text	[x= exists, -- = does not exist]

7 =	2 bytes of additional information for the message number	is required to resolve the information "@" (see ASM5)
8 =	File name for additional information	e.g. in HTML format

Example ASM2

Read the current device information (system errors, device statuses) of device address 01.

Assumption:
The following three devices are defined:

- Device address 01
- Device address 07 and
- Device address 10

FI command		01_BR_ASM2
Line	**Column**	**Answer**
1	1	01
	2	Drill center
	3	71
	4	F
	5	PLC battery voltage too low.
	6	X
	7	0
	8	

FI command

Output current device information (system errors, device statuses) of the device listed from the MWAX device group.

BR_ASM3_(1) **(Single Read)**

BC_ASM3_(1) **(Cyclic Read)**

BB_ASM3_(1) **(Break Cyclic Read)**

(1) = Selection list for a max. of 10 MWAX devices [00_01_02_ ... _15]

Response Structure

The following table shows the general structure of the response to the FI command "ASM3". The number of lines (1 .. n=15) depends on the number of listed MWAX devices. Each line consists of 7 columns for the device address, device name, message number, message status, short text and indication of whether there is an additional text for this device information (system errors, device statuses).

Line 1...n	Column 1	...	Column 7

Value Range/Meaning of Columns

1 =	Device address	[00...15]
2 =	Device name	[max. 32 ASCII characters]
3 =	Message number	[0...150]
4 =	Type of message	[F = fault/error, D = diagnosis]
5 =	Short text	[max. 54 ASCII characters]
6 =	Reference text	[x= exists, -- = does not exist]
7 =	2 bytes of additional information for the message number	is required to resolve the information "@" (see ASM5)
8 =	File name for additional information	e.g. in HTML format

Example ASM3

Read the current device information (system errors, device statuses) for the selected MWAX devices.

Assumption:
The following devices addresses are defined:

- Device address 01,
- Device address 07 and
- Device address 10.

FI command		**01_BR_ASM3_01_10**
Line	**Column**	**Answer**
1	1	01
	2	Drill center
	3	71
	4	F
	5	PLC battery voltage too low.
	6	X
	7	0
	8	
2	1	10
	2	Milling center 2
	3	1
	4	D
	5	Error has been corrected.
	6	X
	7	0
	8	

FI command

Output current device information (system errors, device statuses) of all defined devices (in accordance with the system configuration) from the MWAX device group.

BR_ASM4_(1)	**(Single Read)**
BC_ASM4_(1)	**(Cyclic Read)**
BB_ASM4_(1)	**(Break Cyclic Read)**
(1) = Device group	[MTRX, MWCX, MWSX, MWAX]

Response Structure

The following table shows the general structure of the response to the FI command "ASM4". The number of lines (1 .. n=15) depends on the number of defined MWAX devices. Each line consists of 7 columns for the device address, device name, message number, message status, short text and indication of whether there is an additional text for this device information (system errors, device statuses).

Line 1...n	**Column 1**	**...**	**Column 7**

Value Range/Meaning of Columns

1 =	Device address	[00...15]
2 =	Device name	[max. 32 ASCII characters]
3 =	Message number	[0...150]
4 =	Type of message	[F = fault/error, D = diagnosis]
5 =	Short text	[max. 54 ASCII characters]
6 =	Reference text	[x= exists, -- = does not exist]

7 =	2 bytes of additional information for the message number	is required to resolve the information "@" (see ASM5)
8 =	File name for additional information	e.g. in HTML format

Example ASM4

Read the current device information (system errors, device statuses) of all defined devices within the MWAX device group.
Assumption:
The following devices are defined:

- Device address 01 and
- Device address 10.

FI command		**01_BR_ASM4_MWAX**
Line	**Column**	**Answer**
1	1	01
	2	Drill center
	3	71
	4	F
	5	PLC battery voltage too low.
	6	X
	7	0
	8	
2	1	10
	2	Milling center 2
	3	1
	4	D
	5	Error has been corrected.
	6	X
	7	0
	8	

FI command

Output the additional text for the currently pending device information (system errors, device statuses) related to the device and the message number.

BR_ASM5_(1)_(2)	**(Single Read)**
(1) = Message number	[0...150]

(2) = 2 bytes of additional information for the message number

Response Structure

The following table shows the general structure of the response to the FI command "ASM5". The response consists of a line with 5 columns for device address, device name, message number and additional text.

Line 1...n	Column 1	...	Column 5

Value Range/Meaning of Columns

1 =	Device address	[00...15]
2 =	Device name	[max. 32 ASCII characters]
3 =	Message number	[0...150]
4 =	Type of message	[F = fault/error, D = diagnosis]
5 =	Reference text	[max. 14 lines with a max. 78

		characters/line]
6 =	File name for additional information	e.g. in HTML format

Example ASM5

Read the additional text relating to the device information (system errors, device statuses) with message number 74 of device address 01.

FI command		**01_BR_ASM5_74_0**
Line	**Column**	**Answer**
1	1	01
	2	Drill center
	3	74
	4	F
	5	Replace the SLM module on the PLC card (slot 3).
	6	

Reference to Literature

See chapter entitled "Literature" [13].

10.9 Reading and Writing CMOS RAM ASCII Parameters: CMA

MWAX device group

Designation **CMA** **CM**OS RAM **A**SCII Parameter

Explanation

CMOS RAM ASCII parameters can be read and written. The parameters always start with the index 0. The number of the available CMOS parameters is set in the EEPROM parameters of the MWAX device group:

aAxx.090 CMOS ASCII Variable (max. 80).

FI command

Read CMOS RAM ASCII parameters.

CR_CMA_(1) **(Single Read)**

(1) = CMOS RAM ASCII parameter numbers [0..79]

Response Structure

One line with one column is output for the value of the selected CMOS RAM ASCII parameter.

Example Read CMA Parameter

Read the value of the CMOS RAM ASCII parameter with the number 0 at device address 00.

FI command		**00_CR_CMA_0**
Line	**Column**	**Answer**
1	1	Waiting for tool change

FI command

Write CMOS RAM ASCII parameters.

CW_CMA_(1) **(Single Write)**

(1) = CMOS RAM ASCII parameter numbers [0..79]

Value to be written

Value of the parameter [ASCII characters]

Note: The value to be written is passed to the "acValue" parameter in the "DataTransfer" routine.

Response Structure

The return value of the "DataTransfer" routine is [0] if the write procedure has been successfully completed. In the event of an error, more information in the form of a general error result line can be requested by the routine "ReadGroupItem" (also refer to the chapter "Error Codes" and "General Error Result Line").

Example Write CMA Parameter

Write "Waiting for tool change" in the CMOS RAM ASCII parameter numbered 0 at device address 00.

FI command	**Value to be written: Waiting for tool change 00_CW_CMA_0**
Value to be written	Waiting for tool change

10.10 Creating MI Import Data: CMD

MWAX device group

Designation **CMD** **C**reate **M**I Import **D**ata

Explanation

Creation of the data the Message Integrator requires for data import.

FI command

Creation of the data of all ProVi messages.

BR_CMD1 **(Single Read)**

Response Structure

The command does not return any answer. If no error is signaled, the respective files have been generated.

FI command

Creation of the data of a certain ProVi message type.

BR_CMD2_(1){_(2)} **(Single Read)**

(1) = Message type [1 = error, 2 = messages, 10 = warnings,
11 = start requirements,
12 = setup diagnosis]

(2) = Module number [1...99] ! only for message type 1 -2!

Response Structure

The command does not return any answer. If no error is signaled, the respective files have been generated.

FI command

Creation of the data of all step chain messages.

BR_CMD3 **(Single Read)**

Response Structure

The command does not return any answer. If no error is signaled, the respective files have been generated.

FI command

Creation of the data of a certain step chain register.

BR_CMD4!(1) **(Single Read)**

(1) = Register name [ASCII characters]

Note: The separator "!" is used in this command.

Response Structure

The command does not return any answer. If no error is signaled, the respective files have been generated.

10.11 Reading and Writing CMOS RAM Floating Point Parameters: CMF

MWAX device group

Designation **CMF** **CM**OS RAM **F**loating point Parameter

Explanation CMOS RAM Floating Point parameters can be read and written.

FI command Read CMOS RAM Floating Point parameters.

CR_CMF_(1) **(Single Read)**

(1) = CMOS RAM Floating Point parameter numbers [0..79]

Response Structure One line with one column is output for the value of the selected CMOS RAM Floating Point parameter. The parameters always start with the index 0. The number of the available CMOS parameters is set in the EEPROM parameters of the MWAX device group:

aAxx.092 CMOS Floating point Variable (max. 80).

Example Read CMF Parameters Read the value of the CMOS RAM Floating Point parameter numbered 1 at device address 00.

FI command		**00_CR_CMF_1**
Line	**Column**	**Answer**
1	1	4711.0123

FI command Write CMOS RAM Floating Point parameters.

CW_CMF_(1) **(Single Write)**

(1) = CMOS RAM Floating Point parameter numbers [0..79]

Value to be written Value of the parameter [Type: floating point]

Note: The value to be written is passed to the "acValue" parameter in the "DataTransfer" routine.

Response Structure The return value of the "DataTransfer" routine is [0] if the write procedure has been successfully completed. In the event of an error, more information in the form of a general error result line can be requested by the routine "ReadGroupItem" (also refer to the chapter "Error Codes" and "General Error Result Line").

Example Write CMF Parameter Write the value [4711.0123] in the CMOS RAM Floating Point parameter numbered 1 at device address 00.

FI command	**Value to be written: 4711.0123** **00_CW_CMF_1**
Value to be written	(P_ACK)

10.12 Read and Write CMOS RAM Integer Parameters: CMI

MWAX device group

Designation **CMI** **CM**OS RAM **I**nteger Parameter

Explanation CMOS RAM Integer parameters can be read and written. The parameters always start with the index 0. The number of the available CMOS parameters is set in the EEPROM parameters of the MWAX device group:

aAxx.091 CMOS Integer Variable (max. 80).

FI command Read CMOS RAM Integer parameters.

CR_CMI_(1) **(Single Read)**

(1) = CMOS RAM integer parameter numbers [0..79]

Response Structure One line with one column is output for the value of the selected CMOS RAM integer parameter.

Example Read CMI Parameters Read the value of the CMOS RAM Integer parameter numbered 2 at device address 00.

FI command		**00_CR_CMI_2**
Line	**Column**	**Answer**
1	1	120270

FI command Write CMOS RAM Integer parameters.

CW_CMI_(1) **(Single Write)**

(1) = CMOS RAM integer parameter numbers [0..79]

Value to be written Value of the parameter [Type: integer]

Note: The value to be written is passed to the "acValue" parameter in the "DataTransfer" routine.

Response Structure The return value of the "DataTransfer" routine is [0] if the write procedure has been successfully completed. In the event of an error, more information in the form of a general error result line can be requested by the routine "ReadGroupItem" (also refer to the chapter "Error Codes" and "General Error Result Line").

Example Write CMI Parameter Write the value [120270] in the CMOS RAM Integer parameter numbered 2 at device address 00.

FI command		**Value to write: 120270** **00_CW_CMI_2**
Line	**Column**	**Answer**
1	1	(P_ACK)

10.13 Converting an NC Program: CNP

MWAX device group

Designation **CNP** **C**onvert **N**C **P**rogram

Explanation An NC record is converted and prepared for processing. After conversion, the generated IPD files are entered into the database.

FI command Conversion of the NC record program, the NC record program being specified directly with its full path. Before conversion, the source file is also entered into the database.

BW_CNP1_(1)_(2)_(3) **(Single Write)**

(1) = Database number [1..32]

(2) = Complete name of NC record program

(3) = Database name of the NC record program

Note: If a "0" is entered instead of the database name, the file name of the source file is used for the designation of the database.

Response Structure The response to the "CNP1" FI command consists of three lines, each with one column. The meaning of the elements is as follows:

- Line 1 = Job ID [01...20]
  (see Chapter 7.1 "FI Commands for the MPCX Device Group", IFJ).
- Line 2 = FI command
  [String, in accordance with Chapter 6.1 "Elements of the FI Command"]
- Line 3 = FI Job Error Code
  (see Chapter 8 "Error Codes")

Note: File and path details must be enclosed in inverted commas.

Example CNP1 The NC record program "test.txt" is converted and entered into database 1.

FI command		**00_BW_CNP1_1_"D:\NC_PRG\test.txt"_"nc_prog"**
Line	**Column**	**Answer**
1	1	01
2	1	00_BW_CNP1_1_"D:\NC_PRG\test.txt"
3	1	0

FI command Conversion of the NC record program, the NC record program being already entered into the database. The database name of the NC record program is specified.

BW_CNP2_(1)_(2) **(Single Write)**

(1) = Database number [1..32]

(2) = Database name of the NC record program

Response Structure

The response to the "CNP2" FI command consists of three lines, each with one column. The meaning of the elements is as follows:

- Line 1 = Job ID [01...20]
  (see Chapter 7.1 "FI Commands for the MPCX Device Group", IFJ).
- Line 2 = FI command
  [String, in accordance with Chapter 6.1 "Elements of the FI Command"]
- Line 3 = FI Job Error Code
  (see Chapter 8 "Error Codes")

Note: File names must be enclosed in inverted commas.

Example CNP2

The NC record program with the database name "nc_prog" is converted and entered into database 1.

FI command		**00_BW_CNP2_1_"nc_prog"**
Line	**Column**	**Answer**
1	1	01
2	1	00_BW_CNP2_1_"nc_prog"
3	1	0

JOB commands

The "CNP" FI command is a JOB command. That means that the FI command is started, initialized, and finished directly. The actual processing of the command takes place in a parallel task. Accordingly, the main program does not wait for completion of the order, and other tasks can be continued in the main program. Obviously, it cannot be assumed that the actual task (conversion of the NC record program) has been completed at the end of the command.

For handling the job, the FI commands BFJ, DFJ, and IFJ are available to interrupt or quit the job and to read out the current job process (see chapter 7.1 "FI Commands for the MPCX Device Group").

JOB information with IFJ

Structure of the return of the IFJ command (see Chapter 7.1 "FI Commands for the MPCX Device Group", IFJ).

While the job is executed, the following three groups of situations may occur which can be differentiated by means of the result column 4 of the IFJ command.

Job is running (RUN)

In this case, columns 12 and 14 are important.
12 = Lines which have been converted.
14 = Additional information as a token number
  101 = Conversion started
  102 = NC source file is entered into the database (with CNP1)
  103 = Conversion is running; in this case, the line number is increased
  104 = Result file is entered into the database
  105 = Conversion completed; in this case, column 12 will indicate the number of NC lines of the NC record program

Job finished without errors (READY)

Here, column 12 may be important which indicates the number of NC lines of the NC record program.

Job finished with errors (READY)

In this case, columns 15 and 16 are important.
15 = Error number (see chapter "Error codes")
16 = Additional information as a token number
  110 = Error which should not occur; in this case, column 15 must be evaluated.

111 = Abortion by the operator via FI command "BFJ".
112 = Abortion by the system, because too many errors have occurred during conversion; continuing the conversion would not make sense. A maximum of 50 errors or warnings are permissible in the NC program.
113 = Conversion has been fully completed, but there is at least one error in the NC program. The program will not be entered into the database.

114 = Conversion has been fully completed, but there is at least one warning in the NC program. The program in entered into the database and is ready for processing.

Note: If NC program errors or warnings have been identified during conversion, they are saved to the file "NcConvert_Error.err" which is in the "...CustomData\Resource" directory. The first number is the line number, then follows a space, and then the error text.

10.14 Current Diagnosis Information: CPI

MWAX device group

Designation **CPI** **C**urrent **P**rocess **I**nformation

Explanation The following information is returned for the MTA:

1 = Device address
2 = Device name
3 = Number of mechanism (always: 0)
4 = Name of mechanism (always: MTA process)
5 = Type of message
6 = Source of message (always: CNC)
7 = Type of message (always: I)
8 = Message number
9 = Message text
10 = Additional text available Yes/No (always: –)
11 = 2 byte additional information (always: 0)
12 = HTML file name (always: empty string)
13 = NC notification number (always: 0000)
14 = NC notification text (always: --)

FI command The current diagnosis information for the selected MTA is to be returned. If there is more than one piece of diagnosis information, the indicated message text (column 9) is framed with "<<<" and ">>>".

BR_CPI1 **(Single Read)**
BC_CPI1 **(Cyclic Read)**
BB_CPI1 **(Break Cyclic Read)**

Response Structure The following table shows the general structure of the response to the FI command "CPI1". The response consists of one line with 14 columns.

Line 1	Column 1	...	Column 14

Value Range/Meaning of Columns

1 =	Device address	[00...63]
2 =	Device name	[max. 32 ASCII characters]
3 =	Mechanism number	[always: 0]
4 =	Mechanism name	[always: MTA process]
5 =	Type of message	[F = fault/error, D = diagnosis]
6 =	Message source	[always: CNC]
7 =	Type of message	[always: I = internal]
8 =	Message number	[0...32768]
9 =	Message text	[max. 1024 ASCII characters]
10 =	Reference text	[always: -- = not available]
11 =	2 byte additional information for the message number	[always: 0]
12 =	File name for additional information for message text	[always: empty string]
13 =	NC notification number	[always:0000]
14 =	NC notification	[always:--]

Note: The MTA has NO mechanisms and NO NC notifications.

Example CPI1

Read the diagnosis information of the MTA with the address 03. The MTA returns only one piece of diagnosis information.

FI command		**03_BR_CPI1**
Line	**Column**	**Answer**
1	1	03
	2	Drill center
	3	0
	4	MTA process
	5	D
	6	CNC
	7	I
	8	504
	9	No program loaded up.
	10	--
	11	0
	12	
	13	0000
	14	--

Example CPI1

Read the diagnosis information of the MTA with the address 03. The MTA returns more than one piece of diagnosis information. This is visible from the message text (<<<Message text>>>).

FI command		**03_BR_CPI1**
Line	**Column**	**Answer**
1	1	03
	2	Drill center
	3	0
	4	MTA process
	5	D
	6	CNC
	7	I
	8	504
	9	<<<No program loaded up>>>
	10	--
	11	0
	12	
	13	0000
	14	--

10.15 Trigger Control Reset: CRT

MWAX device group

Designation **CRT** **Control-Reset**

Explanation

The control reset allows the selected device to be reset after a system error. If there is no system error at the selected device then the job is ignored.

CAUTION

Carrying out a reset completely re-initializes the device.
During initialization, communication is temporarily interrupted (inherent to design).

FI command **BW_CRT** **(Single Write)**

Value to be written Trigger reset 0

Note: The value to be written is passed to the "acValue" parameter in the "DataTransfer" routine.

Response Structure

The return value of the "DataTransfer" routine is [0] if the write procedure has been successfully completed. In the event of an error, more information in the form of a general error result line can be requested by the routine "ReadGroupItem" (refer here to chapter 8, "Error Codes" and "General Error Result Line").

Example CRT Trigger a control reset on the selected device.

FI command		**Value to be written: 0** **00_CW_CRT**
Line	**Column**	**Answer**
1	1	(P_ACK)
Value to be written		0

Reference to Literature See chapter entitled "Literature" [26].

10.16 Checking of the Virtual Axis: CVA

MWAX device group

Designation **CVA** **C**heck **V**irtual **A**xis

Explanation This command is used to check whether the requested drive address is a virtual axis.

FI command **BR_CVA1_(1)** **(Single Read)**

(1) = Requested drive address [1..32] with MTC systems
[1..16] with MTA systems

Response Structure The response to the "BR_CVA1" FI command consists of one line with one column.

Line 1	Column 1

Value Range/Meaning of Columns

1 =	Information on whether the selected drive is a virtual axis	[0 = Virtual axis 1 = Real axis

Example CVA1 Check whether drive 1 at device 0 is a real or a virtual axis. The axis in question is a real axis.

FI command		**00_BR_CVA1_1**
Line	**Column**	**Answer**
1	1	1

10.17 Device Axis Configuration Parameter: DAC

MWAX device group

Designation **DAC** **D**evice **A**xis **C**onfiguration Parameter

Explanation The configuration of the device axes that are configured in the active machine parameter record is read out. The following belong to the configuration data of the device axes: axis number, corresponding process, assigned processes, type of axis, APR number, APR axis number, main axis meaning, secondary axis meaning, main axis name, secondary axis name and corresponding axis number.

FI command Output the current parameters of all configured device axes.

BR_DAC1 **(Single Read)**

Response Structure The following table shows the general structure of the response to the "DAC1" FI command. The number of answer lines [1..0.16] depends on the number of configured device axes. Each line consists of 11 columns.

Line 1...n:	Column 1	...	Column 11

Note: If there is no active machine parameter record in the device then the answer lines are not applicable.

Value Range/Meaning of Columns

1 =	Physical axis number	[1...16]
2 =	CNC process number	[always 0]
3 =	Assigned processes	[always 0]
4 =	Type of axis	[81 = digital linear axis 82 = digital rotary axis 83 = digital main spindle 89 = digital Modulo axis] Please note: The values are returned in HEX code
5 =	APR number	[always 1]
6 =	APR axis number	[always 1]
7 =	Main axis meaning	[1..16; the following definition applies: 1 = A axis 2 = X axis 3 = Z axis 4 = Y axis 5 = B axis 6 = C axis 7 = D axis 8 = E axis 9 = X‘ axis 10 = Y‘ axis 11= P axis 12 = Q axis 13 = R axis 14 = U axis 15 = V axis 16 = W axis]
8 =	Secondary axis meaning	[see column 7]
9 =	Main axis name	[according to settings of axis parameters]
10 =	Secondary axis name	[according to settings of axis parameters]
11 =	Assigned axis number	[1..16]

Example DAC1 Reads the current parameters of all configured device axes of the active machine parameter record of device address 00.

Assumption:
The following three device axes have been defined:

- Digital linear axis (axis number 1)
- Digital linear axis (axis number 2)
- Digital main spindle (axis number 3).

FI command		**00_BR_DAC1**
Line	**Column**	**Answer**
1	1	1
	2	0
	3	0
	4	81
	5	1
	6	1
	7	1
	8	1
	9	X1
	10	X1
	11	1
2	1	2
	2	0
	3	0
	4	81
	5	1
	6	1
	7	2
	8	2
	9	Y1
	10	Y1
	11	2
3	1	3
	2	0
	3	0
	4	83
	5	1
	6	1
	7	3
	8	3
	9	S3
	10	S3
	11	3

10.18 Downloading the anlog-C Program and Ident-File into the NC: DAP

MWAX device group

Designation **DAP** **D**ownload **a**nlog-C **P**rogram

Explanation This command is used to write an anlog-C program and a respective Ident-file into the NC. After successful execution, the program will be ready for processing.

FI command Download the anlog-C Program and Ident-File into the NC.

BW_DAP1_(1)_(2)_(3) **(Single Write)**

(1) = Database number	[1..32, the selectable databases must be entered into the NC configurator.]
(2) = anlog-C program	[A maximum of 24 characters is possible]
(3) = Ident-file	[A maximum of 24 characters is possible]

Response Structure After successful execution, the FI command will return the answer (P_ACK).

Example DAP1 Write the anlog-C program "TEST_MOVE" and the respective Ident-file "PARAM_MOVE1" from the database1 into the NC.

FI command		**00_BW_DAP1_1_"TEST_MOVE"_"PARAM_MOVE1"**
Line	**Column**	**Answer**
1	1	(P_ACK)

FI command Download the NC program into the NC.

BW_DAP2_(1)_(2) **(Single Write)**

(1) = Database number	[1..32, the selectable databases must be entered into the NC configurator.]
(2) = NC program	[A maximum of 24 characters is possible]

Response Structure After successful execution, the FI command will return the answer (P_ACK).

Example DAP1 Write the NC program "NC_MOVE" from database1 into the NC.

FI command		**00_BW_DAP2_1_"NC_MOVE"**
Line	**Column**	**Answer**
1	1	(P_ACK)

10.19 Reading Device Component Information: DCI

MWAX device group

Designation **DCI** **D**evice **C**omponent **I**nformation

Explanation The current device component information is read out of the device. From the device component information, the user is provided with information on the components the addressed device is equipped with, and the firmware each component contains. The command will not file if no access to firmware is possible (e.g. while the device is in monitor mode). Instead, the failed access is reported through the firmware access status.

FI command Read the device component information.

BR_DCI1 **(Single Read)**

Response Structure The following table shows the general structure of the response to the FI command "BR_DCI1". For each device component available in the device, one line is returned. Each line consists of 11 columns.

FI command		**00_BR_DCI1**
Line	**Column**	**Answer**
1	1	PCB type
	2	Configured component type
	3	Detected component type
	4	Firmware access status; i.e. has an error occurred accessing the firmware, Yes/No? Valid range of values [YES/NO] In case of an error, the error cause can be defined from one the two following columns.
	5	Error class on accessing firmware identification: (see Error Class Definition under General Error Result Line)
	6	Error code on accessing firmware identification: (see Error Code Definition under Error Codes)
	7	Firmware identification
	8	Firmware version
	9	Firmware release
	10	Is the component address in column 11 a sub-address, Yes/No? Valid range of values [YES/NO]
	11	Component address
2	1	PCB type
	..	..
	11	Component address

Example DCI1 At device address 00, read out the current device component information.

FI command		**00_BR_DCI1**
Line	**Column**	**Answer**
1	1	CPU
	2	MTC-P
	3	MTC-P
	4	NO
	5	0
	6	0
	7	CPU06/0006-23V10
	8	23
	9	10
	10	NO
	11	1
2	1	PLC
	2	MTS-P01.2
	3	MTS-P01.2
	4	NO
	5	0
	6	0
	7	PLC06S-M05-06V05
	8	06
	9	05
	10	NO
	11	3
3	1	APR
	2	--
	3	APR-P
	4	NO
	5	0
	6	0
	7	APR06/0003-23T06
	8	23
	9	06
	10	NO
	11	4

Example DCI1 while booting is blocked

While booting is blocked (i.e. while the device is in monitor mode), read out the current device component information at device address 00.

FI command		**00_BR_DCI1**
Line	**Column**	**Answer**
1	1	CPU
	2	MTC-P
	3	MTC-P
	4	YES
	5	1
	6	2082
	7	--
	8	--
	9	--
	10	NO
	11	1
2	1	PLC
	2	MTS-P01.2
	3	MTS-P01.2
	4	YES
	5	1
	6	2082
	7	--
	8	--
	9	--
	10	NO
	11	3
3	1	APR
	2	--
	3	APR-P
	4	YES
	5	1
	6	2082
	7	--
	8	--
	9	--
	10	NO
	11	4

10.20 Device Configuration Parameter: DCP

MWAX device group

Designation **DCP** **D**evice **C**onfiguration **P**arameter

Explanation The device configuration parameters that are entered in the active machine parameter record as well as in the "IND_DEV.INI" file are output. The configuration parameters of the device include the device address, the device name, device type, mechanism number, mechanism name, and the process types.

FI command Output the configuration parameters of all defined devices.

BR_DCP1 **(Single Read)**

Note: The "DCP1" FI command refers to all defined devices. Therefore, any valid device address can be indicated in the command line (see example DCP1).

Response Structure The following table shows the general structure of the response to the "DCP1" FI command. The response consists of a maximum of n=512 lines (n=16 devices x 32 mechanisms = 512), each with 7 columns.

Line 1...n:	Column 1	...	Column 7

Note: If no active machine parameter record exists in the device, then the columns [1...7] for the respective device are not applicable.

Value Range/Meaning of Columns

1 =	Device address	[00...63]
2 =	Device name	[max. 32 ASCII characters]
3 =	Device Type	[MTC200-P-G2, MTC200-R-G2, MTVNC, MTRA-P, MTRA-R]
4 =	Mechanism number	[0...31]
5 =	Mechanism name	[max. 28 ASCII characters]
6 =	Process type	[1= internal, 2 = external process]
7 =	Process type	[1 = NC process, 2 = PLC process]

Example DCP1 Read the device configuration parameters of all defined devices.

Assumption:
Three devices have been defined

- Device address 00 (MTC200-R-G2)
- Device address 01 (MTC200-P-G2) and
- Device address 02 (MTC200-P-G2)

FI command		**00_BR_DCP1**
Line	**Column**	**Answer**
1	1	00
	2	Rotary transfer machine
	3	MC200-R-G2
	4	1
	5	Master
	6	1
	7	2
2	1	01
	2	0
	3	MTC200-P-G2
	4	0
	5	Milling machine 01
	6	1
	7	1
3	1	02
	2	0
	3	MTC200-P-G2
	4	1
	5	Milling machine 02
	6	1
	7	1

FI command Output the configuration parameters of the selected device.

BR_DCP2 **(Single Read)**

Response Structure The following table shows the general structure of the response to the "DCP2" FI command. The response consists of a line with 7 columns.

Line 1	**Column 1**	**...**	**Column 7**

Note: If no active machine parameter record exists in the device, then the columns [1...7] for the respective device are not applicable.

Value Range/Meaning of Columns

1 =	Device address	[00...15]
2 =	Device name	[max. 32 ASCII characters]
3 =	Device Type	[MTC200-P-G2, MTC200-R-G2, MTVNC, MTRA-P, MTRA-R]
4 =	Mechanism number	[0...31]
5 =	Mechanism name	[max. 28 ASCII characters]
6 =	Process type	[1= internal, 2 = external process]
7 =	Process type	[1 = NC process, 2 = PLC process]

Example DCP2

Read the device configuration parameters of the selected device (device address 01).

Assumption:
Three devices have been defined

- Device address 00 (MTCNC)
- Device address 01 (MTC200-P)
- Device address 02 (MTC200-P)

FI command		**01_BR_DCP2**
Line	**Column**	**Answer**
1	1	01
	2	0
	3	MTC200-P
	4	0
	5	Milling machine 01
	6	1
	7	1

Reference to Literature

See chapter entitled "Literature" [28].

10.21 Setting the Communication Timeout Time DCT

MWAX device group

Designation

DCT **D**evice **C**ommunication **T**imeout

Explanation

By means of this command, the timeout time for the selected device is set dynamically (timeout time in ms).

FI command

BW_DCT1_(1) **(Single Write)**

(1) = requested timeout time in ms

Response Structure

The response to the "DCT1" FI command consists of one line with one column.

Line 1	**Column 1**

Value Range/Meaning of Columns

1 = Status message (P_ACK) (P_ACK)

Example DCT1

For the device 00, the timeout time is set 1500 ms.

FI command		**00_BW_DCT1_1500**
Line	**Column**	**Answer**
1	1	(P_ACK)

FI command

With this command, the timeout time for the selected device can be reset to default value.

BW_DCT2 **(Single Write)**

Response Structure

The response to the "DCT2" FI command consists of one line with one column.

Line 1	**Column 1**

Value Range/Meaning of Columns

1 = Status message (P_ACK) (P_ACK)

Example DCP2

For the device 00, the timeout time is reset to the default value.

FI command		**00_BW_DCT2**
Line	**Column**	**Answer**
1	1	(P_ACK)

10.22 Deleting the FI Exclusive Mode: DEM

MWAX device group

Designation

DEM **D**elete FI **E**xclusive **M**ode

Explanation

This command is used to deactivate FI Exclusive mode for the selected device address.

FI Exclusive mode: In this mode, **ALL** the processes logged in at the FI – **with the exception** of the process issuing the SEM command – are blocked from data communication with this device address. This mode is used for example for data communication which must NOT be interrupted by other data requests. However, it is **imperative** that this FI Exclusive mode is deleted once more through the DEM command.

FI command

BW_DEM1 **(Single Write)**

Response Structure

The following table shows the general structure of the response to the FI command "BW_DEM1". A line of 1 column is output.

Line 1	**Column 1**

Value Range/Meaning of Columns

1 = Status message (P_ACK) (P_ACK)

Example DEM1

Deactivate the FI Exclusive mode for device address 0. The FI Exclusive mode for device address 0 has previously been activated by the SEM command.

FI command		**00_BW_DEM1**
Line	**Column**	**Answer**
1	1	(P_ACK)

10.23 Database Functions: DFH

MWAX device group

Designation **DFH** **D**atabase**F**ile **H**andling

Explanation This command is used to delete, copy, or rename files in the NC database.

FI command Database functions

BW_DFH1_(1)_(2)_(3)_(4)_(5)_(6)} **(Single Write)**

(1) = Function type [1...3,
1 = Delete file
2 = Rename file
3 = Copy file
4 = Paste file]

(2) = Overwrite [0...1,
0 = no
1 = yes]

(3) = Database number [1..32, the selectable databases must be entered into the NC configurator.]

(4) = Group of files [1...8,
1 = Ident-files
3 = anlog-C programs
5 = NC record programs
6 = IPD programs
7 = Tool data
8 = Ident-files (default)]

(5) = Source file [max. 24 characters possible]

(6) = Target file [max. 24 characters possible]

Response Structure One line is output with a column for acknowledgement of whether or not the FI command has been executed successfully.

(P_ACK) = **P**ositive **ACK**nowledge Data element has been set

Example DFH1 The ident-file in database1 is renamed from "Test" to "Processing".

Value to be written not relevant, not evaluated

FI command		**00_BW_DFH1_2_1_1_1_"Test"_"Processing"**
Line	**Column**	**Answer**
1	1	(P_ACK)

10.24 Static/Dynamic Device Information: DIF

MWAX device group

Designation **DIF** **D**evice **I**n**F**ormation

Explanation Static device information and network information is read according to the "IND_DEV.INI" and „FAR_DEV.INI" files.

FI command Reading of the static device information and network information of a selected device.

BR_DIF1	**(Single Read)**
BC_DIF1	**(Cyclic Read)**
BB_DIF1	**(Break Cyclic Read)**

Response Structure The following table shows the general structure of the response to the "DIF1" FI command. The response consists of one line with 24 columns.

Line 1	**Column 1**	**...**	**Column 24**

Value Range/Meaning of Columns

1 =	Local/far device address	[00..63]
2 =	Device name	IND_DEV.INI entry: DeviceName=
3 =	Device type	IND_DEV.INI entry: DeviceType=
4 =	PLC support	IND_DEV.INI entry: PLC=
5 =	Device status	IND_DEV.INI entry: DeviceStatus=
6 =	Assignment of a simulation pair	IND_DEV.INI entry: DeviceAssign=
7 =	Device mode	IND_DEV.INI entry: MtvncMode=
8 =	Communication channel	IND_DEV.INI entry: [CommAddrX]
9 =	Description of the communication channel	IND_DEV.INI entry: CommStr=
10 =	Timeout value	IND_DEV.INI entry: Timeout=
11 =	Device group	(see Chapter 6.1 "Identifier")
12 =	PLC component type	IND_DEV.INI entry: Component type1=
13 =	CNC component type	IND_DEV.INI entry: Component type2=
14 =	Device log	IND_DEV.INI entry: DeviceProtocol=
15 =	Device simulation	IND_DEV.INI entry: DeviceSimulation=
16 =	Not yet assigned	[--]
17 =	Not yet assigned	[--]
18 =	Not yet assigned	[--]
19 =	Not yet assigned	[--]
20 =	Network ON/OFF	[ON = Network active OFF = No network active]
21 =	Network name	Max. 28 ASCII characters
22 =	PC number	[00..99,XX]
23 =	PC name	Max. 255 ASCII characters
24 =	Local device address	[00..63]

Example DIF1 Read the static device information and network information of device 1 while the network is active.

FI command		**01_BR_DIF1**
Line	**Column**	**Answer**
1	1	01
	2	Grinding station right side
	3	MTA200-P-G2
	4	YES
	5	ON
	6	NO
	7	OFF
	8	4
	9	SHM,3,TCON
	10	3500
	11	MWAX
	12	MTS-R
	13	NONE
	14	CNC
	15	OFF
	16	--
	17	--
	18	--
	19	--
	20	ON
	21	PC network 1
	22	29
	23	BTV20-STATION-LEFT
	24	01

Explanation The dynamic device information and network information is read. The current data identifications are made available from the selected controller.

FI command Reading of the dynamic device information and network information of a selected device.

BR_DIF2	**(Single Read)**
BC_DIF2	**(Cyclic Read)**
BB_DIF2	**(Break Cyclic Read)**

Response Structure The answer consists of 23 lines, each line having a specific meaning.

Line 1	Static device information
Line 2	Firmware information
Line 3	Current parameter set
Line 4	Current PLC program
Line 5	Current machine data set
Line 6	Current NC package for memory A
Line 7	Current NC package for memory B

Line 8	Current cycle package
Line 9	Current NC program name for process 0
Line 10	Current NC program name for process 1
Line 11	Current NC program name for process 2
Line 12	Current NC program name for process 3
Line 13	Current NC program name for process 4
Line 14	Current NC program name for process 5
Line 15	Current NC program name for process 6
Line 16	Current tool list for process 0
Line 17	Current tool list for process 1
Line 18	Current tool list for process 2
Line 19	Current tool list for process 3
Line 20	Current tool list for process 4
Line 21	Current tool list for process 5
Line 22	Current tool list for process 6
Line 23	Current I/O configuration table

Meaning of line 1

Line 1 returns the most significant static device information and network information and consists of 18 columns.

Line 1	Column 1	...	Column 18

Value Range/Meaning of Columns

1 =	Line number	[1]
2 =	Status information	Contains the information whether or not the subsequent data in this line is valid; the following applies: [0 = Data is invalid – further column results [--] 1 = Data is valid]
3 =	Local/far device address	[00..63]
4 =	Device name	According to device configuration
5 =	Device type	According to device configuration
6 =	PLC Components	According to device configuration
7 =	CNC components	According to device configuration
8 =	Device group	(see Chapter 6.1 "Identifier")
9 =	Device status	According to device configuration ON = DeviceStatus ON OFF = DeviceStatus OFF
10 =	Current device status	ON = Device ONLINE OFF = Device OFFLINE
11 =	Not yet assigned	[--]
12 =	Not yet assigned	[--]
13 =	Not yet assigned	[--]
14 =	Network ON/OFF	[ON = Network active OFF = No network active]
15 =	Network name	Max. 28 ASCII characters
16 =	PC number	[00..99,XX]
17 =	PC name	Max. 255 ASCII characters
18 =	Local device address	[00..63]

Meaning of line 2

Returns the firmware versions of the existing controller components. Each line consists of 8 columns.

Line 2	Column 1	...	Column 8

Value Range/Meaning of Columns

1 =	Line number	[2]
2 =	Status information	Contains the information whether or not the subsequent data in this line is valid; the following applies: [0 = Data is invalid – further column results [--] 1 = Data is valid]
3 =	Firmware version of the CNC component	Designation according to convention
4 =	Firmware version of the PLC component	Designation according to convention
5 =	Firmware version of the 1.APR component	Designation according to convention
6 =	Firmware version of the 2.APR component	Designation according to convention
7 =	Firmware version of the 3.APR component	Designation according to convention
8 =	Firmware version of the 4.APR component	Designation according to convention

Meaning of line 3 Returns the identification of the current parameter set and consists of 6 columns.

Line 3	Column 1	...	Column 6

Value Range/Meaning of Columns

1 =	Line number	[3]
2 =	Status information	Contains the information whether or not the subsequent data in this line is valid; the following applies: [0 = Data is invalid – further column results [--] 1 = Data is valid]
3 =	Index of the parameter set	[01..99]
4 =	Designation of the parameter set	Max. 32 ASCII characters
5 =	Date string	Date of generation/modification
6 =	Time string	Time of generation/modification

Meaning of line 4 Returns the identification of the current PLC program and consists of 6 columns.

Line 4	Column 1	...	Column 6

Value Range/Meaning of Columns

1 =	Line number	[4]
2 =	Status information	Contains the information whether or not the subsequent data in this line is valid; the following applies: [0 = Data is invalid – further column results [--] 1 = Data is valid]
3 =	Index of the PLC program	Always [00]
4 =	PLC resource name = PLC program name	Max. 32 ASCII characters
5 =	Date string	Date of generation/modification
6 =	Time string	Time of generation/modification

Meaning of line 5

Returns the identification of the current machine data set and consists of 6 columns.

Line 5	Column 1	...	Column 6

Value Range/Meaning of Columns

1 =	Line number	[5]
2 =	Status information	Contains the information whether or not the subsequent data in this line is valid; the following applies: [0 = Data is invalid – further column results [--] 1 = Data is valid]
3 =	Index of the machine data set	[01..99]
4 =	Designation of the machine data set	Max. 32 ASCII characters
5 =	Date string	Date of generation/modification
6 =	Time string	Time of generation/modification

Meaning of line 6

Returns the identification of the current NC package in memory A and consists of 6 columns.

Line 6	Column 1	...	Column 6

Value Range/Meaning of Columns

1 =	Line number	[6]
2 =	Status information	Contains the information whether or not the subsequent data in this line is valid; the following applies: [0 = Data is invalid – further column results [--] 1 = Data is valid]
3 =	Index of the NC package in memory A	[01..99]
4 =	Designation of the NC package in memory A	Max. 32 ASCII characters
5 =	Date string	Date of generation/modification
6 =	Time string	Time of generation/modification

Meaning of line 7

Returns the identification of the current NC package in memory B and consists of 6 columns.

Line 7	Column 1	...	Column 6

Value Range/Meaning of Columns

1 =	Line number	[7]
2 =	Status information	Contains the information whether or not the subsequent data in this line is valid; the following applies: [0 = Data is invalid – further column results [--] 1 = Data is valid]
3 =	Index of the NC package in memory B	[01..99]
4 =	Designation of the NC package in memory B	Max. 32 ASCII characters
5 =	Date string	Date of generation/modification
6 =	Time string	Time of generation/modification

Meaning of line 8

Returns the identification of the current cycle package and consists of 6 columns.

Line 8	Column 1	...	Column 6

Value Range/Meaning of Columns

1 =	Line number	[8]
2 =	Status information	Contains the information whether or not the subsequent data in this line is valid; the following applies: [0 = Data is invalid – further column results [--] 1 = Data is valid]
3 =	Index of the cycle package	[01..99]
4 =	Designation of the cycle package	Max. 32 ASCII characters
5 =	Date string	Date of generation/modification
6 =	Time string	Time of generation/modification

Meaning of the lines 9 - 15

These lines return information on the current NC program for the processes 0..6 and consist of 8 columns each.

Line 9..15	Column 1	...	Column 8

Value Range/Meaning of Columns

1 =	Line number	[9..15]
2 =	Status information	Contains the information whether or not the subsequent data in this line is valid; the following applies: [0 = Data is invalid – further column results [--] 1 = Data is valid]
3 =	Process number	[00..06]
4 =	Process name	Max. 40 ASCII characters
5 =	Current NC memory	[A,B]
6 =	Current NC program number	[01..99]
7 =	Current NC program name	Max. 32 ASCII characters
8 =	Current NC block	

Meaning of the lines 16 -22

These lines return information on the current tool lists for the processes 0..6 and consist of 12 columns each.

Line 16.0.22	Column 1	...	Column 12

Value Range/Meaning of Columns

1 =	Line number	[16..22]
2 =	Status information	Contains the information whether or not the subsequent data in this line is valid; the following applies: [0 = Data is invalid – further column results [--] 1 = Data is valid]
3 =	Process number	[00..06]
4 =	Process name	Max. 40 ASCII characters
5 =	Tool list index	Always [00]
6 =	Name of the tool list	Max. 32 ASCII characters
7 =	Date string	Date of generation/modification
8 =	Time string	Time of generation/modification

9 =	Tool magazine type	[MAGAZINE] [TURRET]
10 =	Number of spindles	[0..4]
11 =	Number of grippers	[0..4]
12 =	Number of magazine locations	[0..999]

Meaning of line 23

Returns the identification of the current I/O configuration list and consists of 6 columns.

Line 23	Column 1	...	Column 6

Value Range/Meaning of Columns

1 =	Line number	[23]
2 =	Status information	Contains the information whether or not the subsequent data in this line is valid; the following applies: [0 = Data is invalid – further column results [--] 1 = Data is valid]
3 =	Index of the I/O configuration list	[01..99]
4 =	Designation of the I/O configuration list	Max. 32 ASCII characters
5 =	Date string	Date of generation/modification
6 =	Time string	Time of generation/modification

Example DIF2

Read the dynamic device information of device 1. It is an MTA200-P-G2, and it is active in a network.

FI command		**01_BR_DIF2**
Line	**Column**	**Answer**
1	1	1
	2	1
	3	01
	4	Grinding station center
	5	MTA200-P-G2
	6	MTS-P01.2
	7	NONE
	8	MWAX
	9	ON
	10	ON
	11	--
	12	--
	13	--
	14	ON
	15	PC network 1
	16	29
	17	BTV20-RIGHT
	18	01

2	1	2
	2	1
	3	AND/MAIN02-10V09
	4	PLC05S-A05-06T03
	5	--
	6	--
	7	--
	8	--
3	1	3
	2	1
	3	01
	4	Parameter set gearbox 2
	5	07.04.03
	6	09:45:42
4	1	4
	2	1
	3	00
	4	PLC program gearbox 2
	5	13.10.02
	6	09:45:34
5	1	5
	2	0
	3	--
	4	--
	5	--
	6	--
6	1	6
	2	0
	3	--
	4	--
	5	--
	6	--
7	1	7
	2	0
	3	--
	4	--
	5	--
	6	--

8	1	8
	2	0
	3	--
	4	--
	5	--
	6	--
9	1	9
	2	1
	3	00
	4	MTA-PROCESS
	5	--
	6	--
	7	--
	8	--
10	1	10
	2	0
	3	--
	4	--
	5	--
	6	--
	7	--
	8	--
11	1	11
	2	0
	3	--
	4	--
	5	--
	6	--
	7	--
	8	--
12	1	12
	2	0
	3	--
	4	--
	5	--
	6	--
	7	--
	8	--

13	1	13
	2	0
	3	--
	4	--
	5	--
	6	--
	7	--
	8	--
14	1	14
	2	0
	3	--
	4	--
	5	--
	6	--
	7	--
	8	--
15	1	15
	2	0
	3	--
	4	--
	5	--
	6	--
	7	--
	8	--
16	1	16
	2	0
	3	--
	4	--
	5	--
	6	--
	7	--
	8	--
	9	--
	10	--
	11	--
	12	--

17	1	17
	2	0
	3	--
	4	--
	5	--
	6	--
	7	--
	8	--
	9	--
	10	--
	11	--
	12	--
18	1	18
	2	0
	3	--
	4	--
	5	--
	6	--
	7	--
	8	--
	9	--
	10	--
	11	--
	12	--
19	1	19
	2	0
	3	--
	4	--
	5	--
	6	--
	7	--
	8	--
	9	--
	10	--
	11	--
	12	--

20	1	20
	2	0
	3	--
	4	--
	5	--
	6	--
	7	--
	8	--
	9	--
	10	--
	11	--
	12	--
21	1	21
	2	0
	3	--
	4	--
	5	--
	6	--
	7	--
	8	--
	9	--
	10	--
	11	--
	12	--
22	1	22
	2	0
	3	--
	4	--
	5	--
	6	--
	7	--
	8	--
	9	--
	10	--
	11	--
	12	--
23	1	23
	2	0
	3	--
	4	--
	5	--
	6	--

FI command Returns information on which controller data is currently available in the selected device.

BR_DIF3 **(Single Read)**

BC_DIF3 **(Cyclic Read)**

Response Structure The answer consists of 1 line with 14 columns, each column having a specific meaning.

1 =	Active parameter set available	Yes/No
2 =	NC package memory A available	Yes/No
3 =	NC package memory B available	Yes/No
4 =	NC zero points memory A available	Yes/No
5 =	NC zero points memory B available	Yes/No
6 =	NC events available	Yes/No
7 =	NC variables available	Yes/No
8 =	NC D-corrections available	Yes/No
9 =	NC cycles available	Yes/No
10 =	Active machine data record available	Yes/No
11 =	PLC retain variables available	Yes/No
12 =	Tool lists available	Yes/No
13 =	Drive parameters available	Yes/No
14 =	I/O configuration list available	Yes/No

Line 1	Column 1	...	Column 14

Value Range/Meaning of Columns 1 = Controller data available Yes/NO [YES,NO]

Example DIF3 Return information on which controller data is currently available in the selected device.

FI command		**01_BR_DIF3**
Line	**Column**	**Answer**
1	1	YES
	2	NO
	3	NO
	4	NO
	5	NO
	6	NO
	7	NO
	8	NO
	9	NO
	10	NO
	11	YES
	12	NO
	13	YES
	14	NO

10.25 Long ID of PLC Data Block: DIS

MWAX device group

Designation **DIS** **D**ata **I**dentification **S**tring

Explanation

Reads the long identification (directory entries) of MTA 200/PLC data records. Included in the directory entries are the number of the entry in the directory, the name, length and date and time of creation and/or details of the last time the respective data record was changed. The long identifications of the following MTA 200/PLC data records are output:

- MTA 200 parameter record (FI command: DIS1)
- PLC program (FI command: DIS2)

FI command

Output the directory entries of the valid NC parameter record in the selected device.

BR_DIS1 **(Single Read)**

BC_DIS1 **(Cyclic Read)**

BB_DIS1 **(Break Cyclic Read)**

Response Structure

The following table shows the general structure of the response to the "DIS1" FI command. The response consists of a line with five columns.

Line 1	Column 1	...	Column 5

Value Range/Meaning of Columns

1 =	Number in MTA 200 parameter directory	[01...99]
2 =	Name of the MTA 200 parameter record	[max. 32 ASCII characters]
3 =	Length of the MTA 200 parameter record	[byte]
4 =	Date of creation/last change to MTA 200 parameter record	[DD.MM.YY]
5 =	Time of creation/last change to MTA 200 parameter record	[HH:MM:SS]

Note: If there is no valid MTA 200 parameter record in the selected device then all columns contain [--] . This command can also be used when the selected device is in OFFLINE mode (DeviceStatus=OFF).

Example DIS1

Read the directory entries of the MTA 200 parameter record at device address 00.
Assumption:
There is a valid MTA 200 parameter record in the selected device.

FI command		**00_BR_DIS1**
Line	**Column**	**Answer**
1	1	01
	2	KEY1
	3	3579
	4	16.05.99
	5	10:41:08

Reference to Literature

See chapter entitled "Literature" [29].

FI command

BR_DIS2 **(Single Read)**
BC_DIS2 **(Cyclic Read)**
BB_DIS2 **(Break Cyclic Read)**

Response Structure

The following table shows the general structure of the response to the "DIS2" FI command. The response consists of a line with six columns.

Line 1	Column 1	...	Column 6

Value Range/Meaning of Columns

1 =	Number in PLC directory	[01...99]
2 =	Name of the PLC program	[max. 8 ASCII characters]
3 =	Length of the PLC program	[byte]
4 =	Date of creation/last change to PLC program	[DD.MM.YY]
5 =	Time of creation/last change to the PLC program	[HH:MM:SS]
6 =	Date of creation/last change to PLC program	[DD.MM.YYYY]

Note: If there is no valid NC package in the selected NC memory then all columns contain [--] .

Example DIS2

Read the directory entries of the PLC program at address 00.
Assumption:
There is a valid PLC program in the selected device.

FI command		**00_BR_DIS2**
Line	**Column**	**Answer**
1	1	01
	2	KEY1
	3	20018
	4	10.05.99
	5	12:42:00
	6	10.05.1999

Reference to Literature

see chapter entitled "Literature" [30].

10.26 Downloading/Uploading of Drive Parameters: DPA

MWAX device group

Designation **DPA** **D**rive **P**arameter **A**ccess

Explanation Drive parameter data records are downloaded by means of a download file. This download command is an FI job.

Structure of Download File The structure of the download file corresponds to that of a Windows Ini file.

Summary:

Section [COMMONDATA]

Contains general information on the generation of this file.

DeviceAddr=Device address for which the drive parameters have been collected.
DeviceName=Device name
DeviceType=Device type
DriveAddrList=List of the drive addresses contained in this file.
MTGUIVersion=GUI version used to generate this download file.
SaveDate=Date when this file was generated (e.g. through an UPLOAD process).
SaveTime=Time of the day when this file was generated.
SaveElementCode=Contains bit-coded information on which Sercos data elements (see SPA commands) are available in this file.
SaveType=Contains the information on which Sercos parameters are available in this file. This concerns the following Sercos parameters:
- according to the list from S-0-0017
- according to the list from S-0-0192

Section [DESCRIPTION]

Contains a brief description of the keys under the section [DRIVExx:X-Y-ZZZZ].
xx=Drive address
X=Sercos data type (S=standard data,P=product data)
Y=Parameter set (0..7)
ZZZZ=Data block no. (0..4095)

Section [DRIVExx]

Contains the required drive data.

DriveType=Drive typs
Max_P_Number=Max. data block number for the product data
Max_P_Set=Max.parameter set for the product data
Max_S_Number=Max. data block number for the standard data
Max_S_Set=Max.parameter set for the standard data

Section [DRIVExx:X-Y-ZZZZ]

Contains the SERCOS parameter data.

001=Number of data lines for the SERCOS operating date
002=Name of the SERCOS parameter
003=Attributes of the SERCOS parameter
004=Unit of the SERCOS parameter
005=Min. input value of the SERCOS parameter
006= Max. input value of the SERCOS parameter
007.001-007.XXX=Data lines for the SERCOS operating date
008=Data status of the SERCOS parameter

BW_DPA1_(1)_(2)_{(3)} **(Single Write)**

(1) = Defines which drives are to be downloaded	0 = Drives according to the current parameter set > 0 = Requested drive address [1.0.16]
(2) = Complete download file name	Download file according to the preset structure
(3) = Optional parameter; defines bit-coded controller information. Format: WORD in HEX code 0xYYZZ If this parameter does not exist, the following default setting is used: - Writing according to list S-0-0192 - Write only operating date - Attribute comparison is performed	**High-Byte (0xYY)** defines according to which list the SERCOS parameters are written; the following applies: 0x00 = acc. to S-0-0192 0x01 = acc. to S-0-0017 0x02 = acc. to INI-File (not yet implemented !!) **Low-Byte (0xZZ)** defines which SERCOS elements are to be written; here, only the attributes (0x04) and the operating date (0x40) can be written. **EXCEPTION:** If the highest bit has been set in this word, attribute comparison is switched off during download!!

Response Structure

The response to the "DPA1" FI command consists of three lines, each with one column. The meaning of the elements is as follows:

- Line 1 = Job ID [01...20]
  (see Chapter 7.1 "FI Commands for the MPCX Device Group", IFJ).
- Line 2 = FI command
  [String, in accordance with Chapter 6.1 "Elements of the FI Command"]
- Line 3 = FI Job Error Code
  (see Chapter 8 "Error Codes")

Note: File and path details must be enclosed in inverted commas.

Example DPA1

The SERCOS parameters stored in the download file D:\SERCPAR.DAT are to be transferred into the parameterized drives of device 0.

As there is no optional 3rd parameter, the default setting is used.

Writing according to list S-0-0192

Write only operating date

Attribute comparison is performed

FI command		**00_BW_DPA1_0_"D:\SERCPAR.DAT"**
Line	**Column**	**Answer**
1	1	01
2	1	00_BW_ DPA1_0_"D:\SERCPAR.DAT"
3	1	0

Explanation

Reads the SERCOS parameters from the drives and saves them to the upload file. This upload command is an FI job.

Structure of upload file

The structure of the upload file corresponds to that of a Windows Ini file. The structure is identical with that of a download file.

BR_DPA1_(1)_(2)_{(3)}	**(Single Read)**
(1) = defines which drives are to be saved	0 = Drives according to the current parameter set > 0 = Requested drive address [1.0.16]
(2) = Complete upload file name	
(3) = Optional parameter; defines bit-coded controller information. Format: WORD in HEX code 0xYYZZ If this parameter does not exist, the following default setting is used: - Reading according to list S-0-0192 - Read attribute and operating date	**High-Byte (0xYY)** defines according to which list the SERCOS parameters are read; the following applies: 0x00 = acc. to S-0-0192 0x01 = acc. to S-0-0017 0x02 = acc. to INI-File (not yet implemented !!) **Low-Byte (0xZZ)** defines which SERCOS elements are to be read; the following applies: 0x01 = Date status 0x02 = Name 0x03 = Attribute 0x08 = Unit 0x10 = Min. input value 0x20 = Max. input value 0x40 = Operating date The corresponding bits can be OR'd, e.g. operating date (0x40) and unit (0x08) produces OR'd 0x48.

Response Structure

The response to the "DPA1" FI command consists of three lines, each with one column. The meaning of the elements is as follows:

- Line 1 = Job ID [01...20]
  (see Chapter 7.1 "FI Commands for the MPCX Device Group", IFJ).
- Line 2 = FI command
  [String, in accordance with Chapter 6.1 "Elements of the FI Command"]
- Line 3 = FI Job Error Code
  (see Chapter 8 "Error Codes")

Note: File and path details must be enclosed in inverted commas.

Example DPA1

The SERCOS parameters which are then to be stored in the upload file D:\SERCPAR.DAT, are to be saved from the parameterized drives of device 0. Data storage is to be performed according to list S-0-0017.

During this process, the SERCOS elements:

- Data status
- Name
- Attribute
- Unit
- Min. input value
- Max. input value
- Operating date

are to be saved.

FI command		**00_BR_DPA1_0_"D:\ SERCPAR .DAT_0x017F"**
Line	**Column**	**Answer**
1	1	01
2	1	00_BR_DPA1_0_"D:\SERCPAR.DAT_0x017F"
3	1	0

Explanation

Reads the log file generated during download of the drive parameters. With the "BR_DPA2" FI command described in the following, this file is read out subsequently to indicate download errors.

Note: File and path details must be enclosed in inverted commas.

BR_DPA2_(1) **(Single Read)**

(1) = Complete download file name

Response Structure

The response to the "DPA2" command consists of n lines, each with 9 columns. One line is provided for each drive.

Line n	**Column 1**	**...**	**Column 9**

Value Range/Meaning of Columns

1 =	Drive address	[1..16]
2 =	Download status	[READY] = Successful download for the drive [WARNING] = Download failed for at least 1 SERCOS parameter [ERROR] = Download failed for the drive
3 =	Error Text	[--] = No error text available; otherwise, the error text for the failed drive download
4 =	Current drive firmware – acc. to S-0-0030	
5 =	Drive firmware – acc. to the download file	
6 =	Number of SERCOS parameters missing in the download file	[0] = All required SERCOS parameters are available (> 0] = Number of missing SERCOS parameters
7 =	List of missing SERCOS parameters	[--] = No missing SERCOS parameters; otherwise, the list of missing SERCOS parameters, the character ‚|‘ (0x7C) being used as a separator, e.g.:S-0-0009|P-0-0096.
8 =	Number of SERCOS parameters which could NOT be loaded	[0] = All required SERCOS parameters could be loaded (> 0] = Number of missing SERCOS parameters which could NOT be loaded
9 =	List of SERCOS parameters which could NOT be loaded	[--] = No unloadable SERCOS parameters; otherwise, the list of unloadable SERCOS parameters, the character ‚|‘ (0x7C) being used as a separator, e.g.:S-0-0006|S-0-0359.

Example DPA2

SERCOS parameters have been transferred into drives 1 and 2 of device 0. In drive 2, the SERCOS parameters S-0-0006 and S-0-0359 are missing, and attribute comparison has failed for the SERCOS parameters S-0-0393, P-0-0099. and P-0-0260.

FI command		**00_BR_DPA2_"D:\SERCPAR.DAT"**
Line	**Column**	**Answer**
1	1	1
	2	READY
	3	--
	4	HSM1.1-SSE-03V25
	5	HSM1.1-SSE-03V22
	6	0
	7	--
	8	0
	9	--
2	1	2
	2	WARNING
	3	--
	4	HSM1.1-SSE-03V25
	5	HSM1.1-SSE-03V22
	6	2
	7	S-0-0006|S-0-0359
	8	3
	9	S-0-0393|P-0-0099|P-0-0260

Explanation

Reads the detailed information from the log file generated during download of the drive parameters. The "BR_DPA3" FI command described in the following is used to supply the detailed error information of the individual drive in plain text.

Note: File and path details must be enclosed in inverted commas.

BR_DPA3_(1)_(2)_(3)	**(Single Read)**
(1) = Requested drive address	[1..16]
(2) = Controller information on whether the detailed information on missing or unloadable SERCOS parameters is requested	[0] = Information on the missing SERCOS parameters [1] = Information on the unloadable SERCOS parameters
(3) = Complete download file name	

Response Structure

The response to the "DPA3" command consists of n lines, each with 9 columns. There is one line for each missing or unloadable SERCOS parameter.

Line n	Column 1	Column 2

Value Range/Meaning of Columns

1 =	SERCOS parameter designation	According to SERCOS specification, e.g.: S-0-0009
2 =	Error Text	

Example DPA3

Detailed information on the missing SERCOS parameters of drive 2 (device 0) is to be requested.

FI command		00_BR_DPA3_2_0_"D:\SERCPAR.DAT"
Line	**Column**	**Answer**
1	1	S-0-0006
	2	The SERCOS operating date is NOT available in the drive parameter download file.
2	1	S-0-0359
	2	The SERCOS operating date is NOT available in the drive parameter download file.

Explanation

This command is used to read out drive addresses and the respective axis type available in the transferred drive data download file.

Note: File and path details must be enclosed in inverted commas.

BR_DPA4_(1) **(Single Read)**

(1) = Complete download file name

Response Structure

The response to the "DPA4" command consists of n lines, each with 4 columns. One line is provided for each drive.

Line n	Column 1	...	Column 4

Value Range/Meaning of Columns

1 =	Drive address	[1..16]
2 =	Type of axis	[see Chapter "Data Tables"]
3 =	Date of data generation	Contains the date of generation of the drive data
4 =	Time of data generation	Contains the time of generation of the drive data

Example DPA4

In the drive data download file, the SERCOS data for drives 1,2,5,8 are stored.

FI command		00_BR_DPA4_"D:\SERCPAR.DAT"
Line	**Column**	**Answer**
1	1	1
	2	81
	3	04.06.2003
	4	14:16:23
2	1	2
	2	82
	3	04.06.2003
	4	14:16:23
3	1	5
	2	81
	3	04.06.2003
	4	14:16:23

4	1	8
	2	83
	3	04.06.2003
	4	14:16:23

10.27 Reading the Device Status Information: DSI

MWAX device group

Designation **DSI** **D**evice **S**tatus **I**nformation

Explanation This enables the most important device status information to be read. The following information is returned:

Type of information	**Status**	**Statement**
System error information		Yes/No
Information on mechanism error		Yes/No
Machine key information		4 Byte HEX
Machine key information	valid	Yes/No
Machine status information		4 Byte HEX
Sercans information		4 Byte HEX
Parameter download	running	Yes/No
PLC download	running	Yes/No
Firmware download	running	Yes/No
Offline/Online information		
Device simulation	switched on	Yes/No
Device status information		ON/ OFF
Communication channel defined		Yes/No
PLC components available		Yes/No
Monitor mode	active	Yes/No

FI command Read out device status information for ALL defined devices.

BR_DSI1 **(Single Read)**

BC_DSI1 **(Cyclic Read)**

BB_DSI1 **(Break Cyclic Read)**

Note: The "DSI1" FI command refers to all devices within this device group. Therefore, any valid device address can be indicated in the command line (see example DSI1). The FI device polling mechanism **MUST** be switched on (see system configurator)!

Response Structure The following table shows the general structure of the response to the "DSI1" FI command.

Line 1...n	**Column 1**	**...**	**Column 11**

Value Range/Meaning of Columns

1 =	Device address	[00...63]
2 =	System error information	[0 = there is no system error 1 = there is a system error]
3 =	Information on mechanism error	[0 = there is no mechanism error 1 = there is a mechanism error]
4 =	Machine key information	[4 byte in HEX coding]
5 =	Is machine key information valid?	[0 = not valid, 1=valid]
6 =	Machine status information	[4 byte in HEX coding]
7 =	Sercans information	[4 byte in HEX coding]
8 =	Is parameter download active?	[0 = parameter download not running 1 = parameter download running]
9 =	Is PLC download active?	[0 = PLC download not running 1 = PLC download running]
10 =	Is firmware download active?	[0 = PLC download not running 1 = PLC download running]
11 =	Offline/Online information	[0 = device connection interrupted 1 = device connection O.K.]
12 =	Device simulation switched on?	[0 = NO Simulation mode 1 = simulation mode]
13 =	Current device status information	[0 = Device status=OFF 1 = Device status=ON]
14 =	Communication channel defined?	[0 = NO communication channel 1 = Communication channel defined]
15 =	PLC components available ?	[0 = NO PLC component 1 = PLC component (DOS-Pcl) 2 = PLC component (WIN-Pcl)]
16 =	Monitor mode	[0 = NO monitor mode active 1 = Monitor mode active]

Example DSI1 Read the current device status information.

Assumption:

The following devices addresses are defined:

- Device address 01 (MTA200-P)
- Device address 03 (MTA200-P)

FI command		**01_BR_DSI1**
Line	**Column**	**Answer**
1	1	01
	2	0
	3	0
	4	00000000
	5	0
	6	00000000
	7	00000000
	8	0
	9	0
	10	0
	11	1
	12	0
	13	1
	14	1
	15	2
	16	0
2	1	03
	2	1
	3	0
	4	00000000
	5	0
	6	00000000
	7	00000000
	8	0
	9	0
	10	0
	11	1
	12	0
	13	1
	14	1
	15	2
	16	0

FI command Read out device status information for a selected device.

BR_DSI2	**(Single Read)**
BC_DSI2	**(Cyclic Read)**
BB_DSI2	**(Break Cyclic Read)**

Response Structure

The following table shows the general structure of the response to the "DSI2" FI command.

Line 1...n	Column 1	...	Column 11

Value Range/Meaning of Columns

1 =	Device address	[00...63]
2 =	System error information	[0 = there is no system error 1 = there is a system error]
3 =	Information on mechanism error	[0 = there is no mechanism error 1 = there is a mechanism error]
4 =	Machine key information	[4 byte in HEX coding]
5 =	Machine key information valid?	[0 = not valid, 1=valid]
6 =	Machine status information	[4 byte in HEX coding]
7 =	Sercans information	[4 byte in HEX coding]
8 =	Is parameter download active?	[0 = parameter download not running 1 = parameter download running]
9 =	Is PLC download active?	[0 = PLC download not running 1 = PLC download running]
10 =	Is firmware download active?	[0 = PLC download not running 1 = PLC download running]
11 =	Offline/Online information	[0 = device connection interrupted 1 = device connection O.K.]
12 =	Device simulation switched on?	[0 = NO Simulation mode 1 = simulation mode]
13 =	Current device status information	[0 = Device status=OFF 1 = Device status=ON]
14 =	Communication channel defined?	[0 = NO communication channel 1 = Communication channel defined]
15 =	PLC components available ?	[0 = NO PLC component 1 = PLC component (DOS-Pcl) 2 = PLC component (WIN-Pcl)]
16 =	Monitor mode	[0 = NO monitor mode active 1 = Monitor mode active]

Example DSI2 Read the current device status information for the selected device.

FI command		**00_BR_DSI2**
Line	**Column**	**Answer**
1	1	00
	2	0
	3	0
	4	00000000
	5	0
	6	00000000
	7	00000000
	8	0
	9	0
	10	0
	11	1
	12	0
	13	1
	14	1
	15	2
	16	0

10.28 Distance to Go of Axis Movement: DTG

MWAX device group

Designation **DTG** **D**istance **T**o **G**o

Explanation The distance to go of the movement of a selected axis is output. The FI command "DTG1" returns the distance to go of an axis, related to the code of the axis meaning. The FI command "DTG2", on the other hand, returns the distance to go of an axis, related to the physical axis number.

FI command Output the distance to go of the selected axis of the device specified, related to the code of the axis meaning.

Using the optional fourth parameter it is possible to pre-select conversion of the result into mm or inches.

CR_DTG1_(1)_(2)_(3){_(4)}	**(Single Read)**
CC_DTG1_(1)_(2)_(3){_(4)}	**(Cyclic Read)**
CB_DTG1_(1)_(2)_(3){_(4)}	**(Break Cyclic Read)**
(1) = NC process number	[0]
(2) = Axis meaning	[1...16, the axis meaning corresponds to the physical axis number]
(3) = System of coordinates	[1 = machine coordinates 2 = program coordinates 3 = relative coordinates]
(4) = Required measurement system (opt.)	[mm, inch]

FI command Output the distance to go of the movement of the selected axis of the device specified related to the physical axis number.

Using the optional third parameter it is possible to pre-select conversion of the result into mm or inches.

CR_DTG2_(1)_(2){_(3)}	**(Single Read)**
CC_DTG2_(1)_(2){_(3)}	**(Cyclic Read)**
CB_DTG2_(1)_(2){_(3)}	**(Break Cyclic Read)**
(1) = Physical axis number	[1..0.16, according to settings of the system parameters]
(2) = System of coordinates	[1 = machine coordinates 2 = program coordinates 3 = relative coordinates]
(3) = Required measurement system (opt.)	[mm, inch]

Response Structure

The following table shows the general structure of the response to the FI commands "DTG1" and "DTG2". One line is output with 4 columns for the axis designation, distance to go, unit and the distance to go limited to "indicated decimal places".

Line 1	Column 1	...	Column 4

Value Range/Meaning of Columns

1 = Axis name	[according to settings of axis parameters]
2 = Distance to go	[according to settings of process parameters]
3 = Unit	[mm, inch]
4 = Distance to go	[as column 2]

Note: If the specified axis or a spindle is not defined in the selected NC process then the answer in all columns is [--].

Example DTG1

Read the distance to go of the movement of the Z axis in machine coordinates in NC process 0 of device address 00.

FI command	**00_CR_DTG1_0_3_1**			
Answer				
Line 1	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	Z	-1.2345	[mm]	-1.2345

Example DTG1

Read the distance to go of the movement of the Z axis in machine coordinates in NC process 0 of device address 00. Values are displayed in inches.

FI command	**00_CR_DTG1_0_3_1_inch**			
Answer				
Line 1	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	Z	-0.0486	[inch]	-0.0486

Example DTG2

Read the distance to go of the movement of the Z axis (physical axis number = 3) in machine coordinates at the device address 00.

FI command	**00_CR_DTG2_3_1**			
Answer				
Line 1	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	Z	-1.2345	[mm]	-1.2345

Reference to Literature

See chapter entitled "Literature" [16].

10.29 Device Type and Accompanying Components: DTY

MWAX device group

Designation **DTY** **D**evice **TY**pe

Explanation The device type and the accompanying components of the selected device address are output.

FI command **BR_DTY1** **(Single Read)**

BC_DTY1 **(Cyclic Read)**

Response Structure The following table shows the general structure of the response to the "DTY1" FI command. A line with three columns for the device type is output as well as the name of the first device component and the name of the second device component.

Line 1	Column 1	...	Column 3

Value Range/Meaning of Columns

1 =	Device Type	(see Chapter "Identifier")
2 =	Component type1	IND_DEV.INI-Entry: Componenttype1=
3 =	Component type 2	IND_DEV.INI-Entry: Componenttype2=

Example DTY1 Output the device type and the accompanying components of device address 00.

FI command	00_BR_DTY1		
Answer			
Line	**Column 1**	**Column 2**	**Column 3**
1	MTA200-P	MTS-P	MTC-P

10.30 Diagnosis Window Data: DWD

MWAX device group

Designation **DWD** **D**iagnosis **W**indow **D**ata

Explanation Diagnostic messages are output. The data are edited in such a way that they can be output directly in the diagnosis overview, i.e., where applicable, different types of diagnosis, such as a ProVi message and a process message, are returned simultaneously.

FI command Output all diagnostic messages.

For reasons of optimization, not all data is called up. Accordingly, the Diagnosis server must be informed that the data is required (see ADW).

BR_DWD1_(1){_(2)} **(Single Read)**

BC_DWD1_(1){_(2)} **(Cyclic Read)**

(1) = Type of diagnosis window	[1 = NC error, 2 = sequence errors, 3 = general errors, 4 = messages, 10 = start preconditions, 11 = warnings, 12 = setup diagnosis]
(2) = Module number	[1...99] ! only for window type 1 -4 !

Output first diagnostic messages.

BR_DWD2_(1){_(2)} **(Single Read)**

BC_DWD2_(1){_(2)}	**(Cyclic Read)**
(1) = Type of diagnosis window	[1 = NC error, 2 = sequence errors, 3 = general errors, 4 = messages, 10 = start preconditions, 11 = warnings, 12 = setup diagnosis]
(2) = Module number	[1...99] ! only for window type 1 -4 !

Response Structure

The following table shows the general structure of the "DWD1" and "DWD2" FI commands. The number of lines depends on the number of messages pending. Different columns are valid according to the type of diagnosis.

If there are no messages, the number of lines is 0.

Line 1...n	Column 1	...	Column 14

Meaning of the Columns

1 =	Message text	[ASCII characters]
2 =	Time stamp day	[mm.dd.yyyy]
3 =	Time stamp hour	[hh:mm:ss]
4 =	Reference text available	[YES, NO]
5 =	Type of diagnosis	[1 = ProVi, 2 = SFC, 3 = MTC-NC, 4 = MTA-NC]
6 =	Message number	[ASCII characters]
7 =	Message ID	[ASCII characters] (DWORD, decimal) (ProVi)
8 =	Mechanism number	[0..31] (MTC-NC) [0] (MTA-NC)
9 =	2 byte additional information	[ASCII characters] (MTC NC)
10 =	Message group	[1...9999] (MTA-NC)
11 =	SFC entity name	[ASCII characters]
12 =	NC note	[ASCII characters] (MTC NC)
13 =	Analysis of criteria available	[YES, NO] (ProVi, SFC)
14 =	Message HTML file	[ASCII characters] (ProVi, MTC-NC)

Example DWD1 All diagnostic messages from module 3 in control unit 0.

There are two messages:

FI command		**00_BR_DWD1_4_3**
Line	**Column**	**Answer**
1	1	Guard not closed
	2	01.27.2000
	3	14:56:32
	4	YES
	5	1
	6	34
	7	43923028
	8	
	9	
	10	
	11	
	12	
	13	YES
	14	
2	1	Station waiting until tool-change command has ended.
	2	01.27.2000
	3	15:03:10
	4	YES
	5	3
	6	79
	7	
	8	1
	9	0
	10	
	11	
	12	
	13	YES
	14	NO

Example DWD2 First diagnostic message from module 3 in control unit 0.

There are two messages:

FI command		**00_BR_DWD2_4_3**
Line	**Column**	**Answer**
1	1	Guard not closed
	2	01.27.2000
	3	14:56:32
	4	YES
	5	1
	6	34
	7	43923028
	8	
	9	
	10	
	11	
	12	
	13	YES
	14	

Reference to Literature See chapter entitled "Literature" [13].

10.31 Existing MTA 200 Diagnoses: EAD

MWAX device group

Designation **EAD** **E**xisting MT**A** 200 **D**iagnosis

Explanation Which MTA 200 diagnostic types exist is queried. Depending on the FI command, specific types are queried or else the diagnostic types for one module are output together.

FI command Query which MTA 200 diagnostic types are available in a module.

BR_EAD1_(1)	**(Single Read)**
(1) = Module number	[1...99]

Response Structure The following table shows the general structure of the "EAD1" FI command.

Line 1	**Column 1-2**

Meaning of the Columns

1 = Messages exist	[YES, NO]
2 = Errors exist	[YES, NO]

Example EAD1 Query the MTA 200 diagnostic types in Module 2 on Control unit 0.

FI command		**00_BR_EAD1_2**
Line	**Column**	**Answer**
1	1	NO
	2	YES

FI command Query a specific MTA 200 diagnostic type.

BR_EAD2_(1)_(2)	**(Single Read)**
(1) = Message type	[1 = error, 2 = messages]
(2) = Module number	[1...99]

Response Structure

The following table shows the general structure of the "EAD2" FI command.

Line 1	Column 1

Meaning of the Columns

1 = Diagnosis type exists [YES, NO]

Example EAD2

Are there any messages in module 4 in control unit 0?

FI command		**00_BR_EAD2_2_4**
Line	**Column**	**Answer**
1	1	YES

10.32 Component Information for a System Error: ECI

MWAX device group

Designation

ECI **E**rror **C**omponent **I**nformation

Explanation

When a system error is present, this command is used to define which controller component is causing the error. Here, PLC components are differentiated from general controller components (e.g. CNC, Synax, MTA, …).

FI command

BR_ECI1 **(Single Read)**

Response Structure

The response to the "ECI1" FI command consists of one line with two columns.

Line 1	Column 1	Column 2

Value Range/Meaning of Columns

1 =	PLC component information	[0 = There is NO system error at the PLC) 1 = There is a system error at the PLC]
2 =	General information on controller components	[0 = There is NO system error at the general control component [0 = There is a system error at the general control component; the following applies: 2 = CNC component 3 = SYNAX component 4 = VISUAL-MOTION component 5 = MTA component 6 = TRANS 200 component]

Example ECI1

There is a system error present in device 0 MTA200-P (VM-P) which is caused by the PLC component.

FI command		**00_BR_ECI1**
Line	**Column**	**Answer**
1	1	1
1	2	0

10.33 Existing errors: EDE

MWAX device group

Designation **EDE** **E**xisting **D**iagnosis **E**rror

Explanation Whether or not errors exist in a control unit or in a module is queried. These can be sequencer errors, NC errors, MTA 200 errors or ProVi errors.

FI command Query whether there are errors in this control unit.

BR_EDE1	**(Single Read)**
BC_EDE1	**(Cyclic Read)**

Response Structure The following table shows the general structure of the "EDE1" FI command.

Line 1	Column 1

Meaning of the Columns

1 = Error exists	[YES, NO]

Example EDE1 Do errors exist in control unit 0?

FI command		00_BR_EDE1
Line	**Column**	**Answer**
1	1	YES

FI command Query whether or not errors exist in a specific module.

BR_EDE2_(1)	**(Single Read)**
BC_EDE2_(1)	**(Cyclic Read)**
(1) = Module number	[1...99]

Response Structure The following table shows the general structure of the "EDE2" FI command.

Line 1	Column 1

Meaning of the Columns

1 = Error exists	[YES, NO]

Example EDE2 Do errors exist in Module 1 on Control unit 0?

FI command		00_BR_EDE2_2
Line	**Column**	**Answer**
1	1	NO

10.34 Existing Diagnosis Window: EDW

MWAX device group

Designation **EDW** **E**xisting **D**iagnosis **W**indow

Explanation Which types of diagnosis window exist is queried.

FI command Output all types of diagnosis window.

BR_EDW1 **(Single Read)**

Response Structure The following table shows the general structure of the "EDW1" FI command. The number of lines depends on the number of types of window existing.

Line 0...n	Column 1	Column 2

Meaning of the Columns

1 = Type of diagnosis window	[1 = NC error, 2 = sequence errors, 3 = general errors, 4 = messages, 10 = start preconditions, 11 = warnings, 12 = setup diagnosis]
2 = Module number	[ASCII characters] 0 = Diagnosis window type does not belong to any module

Example EDW1 All types of diagnosis window in control unit 0.

There are three diagnosis windows:

FI command		**00_BR_EDW1**
Line	**Column**	**Answer**
1	1	10
	2	0
2	1	1
	2	3
3	1	2
	2	3

FI command Output all diagnosis window types for a module.

BR_EDW2_(1) **(Single Read)**

(1) = Module number [1...99]

Response Structure The following table shows the general structure of the "EDW2" FI command. The number of lines depends on the number of types of window existing.

Line 0...n	Column 1	Column 2

Meaning of the Columns

1 = Type of diagnosis window	[1 = NC error, 2 = sequence errors, 3 = general errors, 4 = messages]
2 = Module number	[ASCII characters] 0 = Diagnosis window type does not belong to any module

Example EDW2 All types of diagnosis window in Module 3, Control unit 0.

There are two diagnosis windows.

FI command		**00_BR_EDW2_3**
Line	**Column**	**Answer**
1	1	1
	2	3
2	1	2
	2	3

FI command

Query a specific type of diagnosis window.

BR_EDW3_(1){_(2)}	**(Single Read)**
(1) = Type of diagnosis window	[1 = NC error, 2 = sequence errors, 3 = general errors, 4 = messages, 10 = start preconditions, 11 = warnings, 12 = setup diagnosis]
(2) = Module number	[1...99] ! only for window type 1 -4 !

Response Structure

The following table shows the general structure of the "EDW3" FI command.

Line 1	**Column 1**

Meaning of the Columns

1 = Type of diagnosis window exists [YES, NO]

Example EDW3

Query whether or not a NC error window exists in module 3, control unit 0.

FI command		**00_BR_EDW3_1_3**
Line	**Column**	**Answer**
1	1	YES

10.35 Existing ProVi Types: EPT

MWAX device group

Designation

EPT **E**xisting **P**roVi **T**ypes

Explanation

Which ProVi types are programmed in the current PLC program is queried. The data is returned in a suitable form for the message texts of the small control panels. There is no need to define modules in Moduldef.ini.

FI command

Output all ProVi types.

BR_EPT1 **(Single Read)**

Response Structure

The following table shows the general structure of the "EPT1" FI command. The number of lines depends on the number of ProVi types existing.

Line 0...n	**Column 1**	**Column 2**

Meaning of the Columns

1 = Type	[11 = error, 12 = messages, 20 = start requirements, 21 = warnings, 22 = setup diagnosis]
2 = Index	[ASCII characters]

Example EPT1

All ProVi types in control unit 0.

There are three diagnosis windows.

FI command		**00_BR_EPT1**
Line	**Column**	**Answer**
1	1	20
	2	0
2	1	11
	2	3
3	1	12
	1	3

10.36 Error Status: EST

MWAX device group

Designation **EST** **E**rror **ST**ate

Explanation Queries the error state of a variable.

FI command Query the frozen error state of a variable.

BR_EST1!(1)!(2)	**(Single Read)**
BC_EST1!(1)!(2)	**(Cyclic Read)**
(1) = Error ID	[ASCII characters] (DWORD, decimal)
(2) = Variable name	[ASCII characters]

Note: The separator "!" is used in this command.

Response Structure

The following table shows the general structure of the "EXD1" FI command.

Line 1	**Column 1**

Meaning of the Columns 1 = Error state

WinPCL - Example EST

Read the value of WinPCL variable "IB_EXT24" in WinPCL program "Prog", at device address 00.

Suggestion:
The WinPCL variable "IB_EXT24" is declared as BOOL in the WinPCL program "Prog".

FI command		**00_BR_EST1!5892855!:Prog.IB_EXT24**
Line	**Column**	**Answer**
1	1	1

10.37 Execution Display: EXD

MWAX device group

Designation **EXD** **EX**ecution **D**isplay

Explanation Information for displaying the execution of a movement is output.

FI command Query the execution of a step or of an action.

BR_EXD1!(1)!(2)!(3)	**(Single Read)**
BC_EXD1!(1)!(2)!(3)	**(Cyclic Read)**
(1) = SFC entity name	[ASCII characters]
(2) = Step or action name	[ASCII - characters]
(3) = Behaviour of mode	[1 – all modes, 2 – manual mode]

Note: The separator "!" is used in this command.

Response Structure The following table shows the general structure of the "EXD1" FI command.

Line 1	Column 1

Meaning of the Columns

1 = Execution	[1 – can be executed, 0 – cannot be executed]

Example EXD1 Query the execution of the step "open" for the chain "clamp" in control unit 0 for all modes.

FI command		**00_BR_EXD1!Station03A.Clamp!Open!1**
Line	**Column**	**Answer**
1	1	1

FI command Query whether the condition analysis (control image) of a step chain is enabled.

BR_EXD2!(1)	**(Single Read)**
(1) = SFC entity name	[ASCII characters]

Note: The separator "!" is used in this command.

Response Structure The following table shows the general structure of the "EXD2" FI command.

Line 1	Column 1

Meaning of the Columns

1 = Enabled	[1 - enabled, 0 – not enabled]

Example EXD2 Query whether the condition analysis of the "clamp" chain has been enabled.

FI command		**00_BR_EXD2!Station03A.Clamp**
Line	**Column**	**Answer**
1	1	1

10.38 Read a File List from the Database: GDF

MWAX device group

Designation **GDF** **G**et **D**atabase-**F**ilelist

Explanation This command is used to read a file list of a defined file type out of a database. If the optional fifth parameter is not written, the language-independent range is selected in all cases.

FI command Read a file list.

BR_GDF1_(1)_(2)_(3)_(4)_{(5)} **(Single Read)**

(1) =	Database number	[1..32, the selectable databases must be entered into the NC configurator.]
(2) =	Group of files	[1...8, 1 = Ident-files 2 = Menus 3 = anlog-C programs 4 = Images 5 = NC record programs 6 = IPD programs 7 = Tool data 8 = Ident-files (default)]
(3) =	File type	[0...49999, 0 = all files (independent of type) 1...49999 = selected file type]
(4) =	Order	[0...3, 0 = No order 1 = Name order 2 = File type order 3 = Date order]
(5) =	Language (optional)	[0...6, 0 = Language-independent range 1 = English 2 = German 3 = French 4 = Italian 5 = Spanish 6 = Swedish]

Response Structure The following table shows the general structure of the response to the FI command "GDF1".

Line 1...n	Column 1	Column 2

Value Range/Meaning of Columns

Line 1 =	1	Number of files
	2	--
Line 2...n =	1	File name
	2	File type

Example GDF1

Read all the files from the "anlog-C programs" group out of database 1. The result is ordered according to names, and the language-independent range is used.

FI command		**00_BR_GDF1_1_3_0_1**
Line	**Column**	**Answer**
1	1	3
	2	--
2	1	FIRST_PROGRAM
	2	12000
3	1	PRG_EXAMPLE
	2	1234
4	1	TEST_MOVE
	2	25000

10.39 Read the Current File Information out of the NC: GMF

MWAX device group

Designation

GMF **G**et active **M**ain-**F**ileinformation

Explanation

This command is used to read out all files which are active in the NC.

FI command

Read the active file information.

BR_GMF1 **(Single Read)**

Response Structure

The following table shows the general structure of the response to the FI command "GMF1".

Line 1..6	**Column 1**

Value Range/Meaning of Columns

Line 1 = Current anlog-C program
Line 2 = Current ident file
Line 3 = Current NC Program
Line 4 = Start of anlog-C program
Line 5 = Start of ident file
Line 6 = Start of NC Program

Example GMF1

Read the current file information out of the NC.

FI command		**00_BR_GMF1**
Line	**Column**	**Answer**
1	1	TEST_MOVE
2	1	PARAM_MOVE3
3	1	DRESSER_GRINDINGWHEEL
4	1	TEST_MOVE
5	1	--
6	1	PARAM_MOVE1

10.40 Global Process Parameter: GPP

MWAX device group

Designation **GPP** **G**lobal **P**rocess **P**arameter

Explanation The global process parameters of the active machine parameter record of the selected device from the MWAX device group is read out. This includes the programmable and actually displayed digits after the decimal point for the displacement, the name of the MTA process, and the basic programming unit.

FI command Output of the configuration of the global process parameters of the defined MTA process of the active machine parameter record.

BR_GPP1 **(Single Read)**

Response Structure The following table shows the general structure of the response to the "GPP1" FI command. The response consists of one line with 6 columns.

Line 1	Column 1	...	Column 6

Value Range/Meaning of Columns

1 = MTA process number [always 0]

2 = Name of the MTA process [always MTA-PROCESS]

3 = Basic programming unit [mm, inch]

4 = Programmed number of positions after decimal point [always 5]

5 = Displayed positions after the decimal point [always 4]

6 = Reserved [always --]

Example GPP1 Read the global process parameters in the MTA process of the active machine parameter record of device address 00.

FI command		**00_BR_GPP1**
Line	**Column**	**Answer**
1	1	0
	2	MTA-PROCESS
	3	[mm]
	4	5
	5	4
	6	--

FI command Output the configuration of the global process parameters of the active machine parameter record of the selected device related to the MTA process.

BR_GPP2_(1) **(Single Read)**

(1) = MTA process number [always 0]

Response Structure The following table shows the general structure of the response to the "GPP2" FI command. The response consists of one line with 6 columns.

Line 1	Column 1	...	Column 6

Value Range/Meaning of Columns

1 =	MTA process number	[always 0]
2 =	Name of the MTA process	[always MTA-PROCESS]
3 =	Basic programming unit	[mm, inch]
4 =	Programmed number of positions after decimal point	[always 5]
5 =	Displayed positions after the decimal point	[always 4]
6 =	Reserved	[always --]

Example GPP2 Read the global process parameters in the MTA process of the active machine parameter record of device address 00.

FI command		**00_BR_GPP2_0**
Line	**Column**	**Answer**
1	1	0
	2	MTA-PROCESS
	3	[mm]
	4	5
	5	4
	6	--

10.41 Read or Write Hand-Parameters: HPF

MWAX device group

Designation **HPF** **H**and-**P**arameter **F**lotingpoint

Explanation This command is used to read and/or write floatingpoint hand-parameters.

FI command Read a floatingpoint hand-parameter out of the NC.

BR_HPF1_(1) **(Single Read)**

(1) = Parameter number [0...255]

Response Structure The response of the FI command is the value of the requested parameter.

Line 1	**Column 1**

Value Range/Meaning of Columns 1 = Value to be read

Example HPF1 Read the floating point hand-parameter 5 out of the NC.

FI command		**00_BR_HPI1_5"**
Line	**Column**	**Answer**
1	1	123,78

FI command Write a floating point hand-parameter into the NC.

BW_HPF1_(1) **(Single Write)**

(1) = Parameter number [0...255]

Response Structure After successful execution, the FI command will return the answer (P_ACK).

Example HPF1 Write the floating point hand-parameter 5 into the NC.

FI command		**Value to be written: 123,78** **00_BW_HPF1_5**
Line	**Column**	**Answer**
1	1	(P_ACK)

10.42 Read or Write Hand-Parameters: HPI

MWAX device group

Designation **HPI** **H**and-**P**arameter **I**nteger

Explanation This command is used to read and/or write integer hand-parameters.

FI command Read an integer hand-parameter out of the NC.

BR_HPI1_(1) **(Single Read)**

(1) = Parameter number [0...255]

Response Structure The response of the FI command is the value of the requested parameter.

Line 1	**Column 1**

Value Range/Meaning of Columns 1 = Value to be read

Example HPI1 Read the integer hand-parameter 5 out of the NC.

FI command		**00_BR_HPI1_5"**
Line	**Column**	**Answer**
1	1	67

FI command Write an integer hand-parameter into the NC.

BW_HPI1_(1) **(Single Write)**

(1) = Parameter number [0...255]

Response Structure After successful execution, the FI command will return the answer (P_ACK).

Example HPI1 Write the integer hand-parameter 5 into the NC.

FI command		**Value to be written: 67** **00_BW_HPI1_5**
Line	**Column**	**Answer**
1	1	(P_ACK)

10.43 Read the Parameter Range out of an Ident File: IFR

MWAX device group

Designation **IFR** **I**dent **F**ile **R**ange

Explanation This command is used to read parameters of any kind out of an ident file. Up to 10 parameter ranges can be defined.

FI command Read parameter ranges out of an ident file.

BR_IFR1_(1)_(2)_(3)
{_(4)_(5)_(6)_(7)_(8)
_(9)_(10)_(11)_(12)} **(Single Read)**

(1) = Database number [1..32, the selectable databases must be entered into the NC configurator.]

(2) = Ident file name [A maximum of 24 characters is possible]

(3) - (12) =
Parameter range
[I = Integer
F = Float
A = Ascii
Range from 0-2047
"-" = Separators between start and end

e. g. I25-40, F120-200, A2050-2100

Note: A maximum of 200 parameters may be read out with one FI command.

Response Structure The response of the FI command is a two-dimensional box with the requested parameters.

Value Range/Meaning of Columns

Line1 .. n	Column 1	Column 2	Column 3
1 .. n	Parameter type 0 = Integer 1 = Float 2 = Ascii	Parameter number	Parameter value

Example IFR1 From database 1 and ident file "TESTIF", read integer parameters 10-30, 150-160 and float parameters 250-255 and Ascii parameters 50-65.

FI command		00_BR_IFR1_1_"TESTIF"_I10-30_I150-160_ F250-255_A50-65
Line	**Column**	**Answer**
1	1	0
1	2	10
1	3	123
...	...	...
n	1	2
n	2	65
n	3	"test"

10.44 Read or Write ASCII Parameters from or to an Ident File: IPA

MWAX device group

Designation **IPA** **I**dent file **P**arameter **A**SCII

Explanation This command is used to read ASCII parameters from an ident file or write them into an ident file.

FI command Read an ASCII parameter from an ident file.

BR_IPA1_(1)_(2)_(3) **(Single Read)**

(1) = Parameter number	[0...4095]
(2) = Database number	[1..32, the selectable databases must be entered into the NC configurator.]
(3) = Ident file name	[A maximum of 24 characters is possible]

Response Structure The response of the FI command is the value of the requested parameter.

Line 1	Column 1

Value Range/Meaning of Columns 1 = Value to be read

Example IPA1 From database 1 and ident file "TESTIF", read ASCII parameter 10.

FI command		**00_BR_IPA1_10_1_"TESTIF"**
Line	**Column**	**Answer**
1	1	Hello World

FI command Write an ASCII parameter into an ident file.

BW_IPA1_(1)_(2)_(3)_(4) **(Single Write)**

(1) = Parameter number	[0...4095]
(2) = Database number	[1..32, the selectable databases must be entered into the NC configurator.]
(3) = Ident file name	[A maximum of 24 characters is possible]
(4) = Ident file type	[1...49999]

Response Structure After successful execution, the FI command will return the answer (P_ACK).

Example IPA1 Write the ASCII parameter 10 into the "TESTIF" ident file of database 1. If the ident file does not exist yet, a new one is generated with the file type "1234".

FI command		**Value to be written: Hello World** **00_BW_IPA1_10_1_"TESTIF"_1234**
Line	**Column**	**Answer**
1	1	(P_ACK)

10.45 Read or Write Floating Point Parameters from or to an Ident File: IPF

MWAX device group

Designation **IPF** **I**dent file **P**arameter **F**LP

Explanation This command is used to read floating point parameters from an ident file or write them into an ident file.

FI command Read a floating point parameter from an ident file.

BR_IPF1_(1)_(2)_(3) **(Single Read)**

(1) = Parameter number [0...4095]

(2) = Database number [1..32, the selectable databases must be entered into the NC configurator.]

(3) = Ident file name [A maximum of 24 characters is possible]

Response Structure The response of the FI command is the value of the requested parameter.

Line 1	Column 1

Value Range/Meaning of Columns 1 = Value to be read

Example IPF1 From database 1 and ident file "TESTIF", read FLP parameter 10.

FI command		**00_BR_IPF1_10_1_"TESTIF"**
Line	**Column**	**Answer**
1	1	12,45

FI command Write an FLP parameter into an ident file.

BW_IPF1_(1)_(2)_(3)_(4) **(Single Write)**

(1) = Parameter number [0...4095]

(2) = Database number [1..32, the selectable databases must be entered into the NC configurator.]

(3) = Ident file name [A maximum of 24 characters is possible]

(4) = Ident file type [1...49999]

Response Structure After successful execution, the FI command will return the answer (P_ACK).

Example IPF1 Write the FLP parameter 10 into the "TESTIF" ident file of database 1. If the ident file does not exist yet, a new one is generated with the file type "1234".

FI command		**Value to be written: 12,45** **00_BW_IPF1_10_1_"TESTIF"_1234**
Line	**Column**	**Answer**
1	1	(P_ACK)

10.46 Read or Write Integer Parameters from or to an Ident File: IPI

MWAX device group

Designation **IPI** **I**dent file **P**arameter **I**nteger

Explanation This command is used to read integer parameters from an ident file or write them into an ident file.

FI command Read integer parameters out of an ident file.

BR_IPI1_(1)_(2)_(3) **(Single Read)**

(1) = Parameter number	[0...4095]
(2) = Database number	[1..32, the selectable databases must be entered into the NC configurator.]
(3) = Ident file name	[A maximum of 24 characters is possible]

Response Structure The response of the FI command is the value of the requested parameter.

Line 1	Column 1

Value Range/Meaning of Columns 1 = Value to be read

Example IPI1 From database 1 and ident file "TESTIF", read integer parameter 10.

FI command		**00_BR_IPI1_10_1_"TESTIF"**
Line	**Column**	**Answer**
1	1	67

FI command Write an integer parameter into an ident file.

BW_IPI1_(1)_(2)_(3)_(4) **(Single Write)**

(1) = Parameter number	[0...4095]
(2) = Database number	[1..32, the selectable databases must be entered into the NC configurator.]
(3) = Ident file name	[A maximum of 24 characters is possible]
(4) = Ident file type	[1...49999]

Response Structure After successful execution, the FI command will return the answer (P_ACK).

Example IPI1 Write the integer parameter 10 into the "TESTIF" ident file of database 1. If the ident file does not exist yet, a new one is generated with the file type "1234".

FI command		**Value to be written: 67** **00_BW_IPI1_10_1_"TESTIF"_1234**
Line	**Column**	**Answer**
1	1	(P_ACK)

10.47 Module Assignment of a Process: MAP

MWAX device group

Designation **MAP** **Mod**ule **A**ssign of **P**rocess

Explanation The module to which a particular process is assigned is read from the "Moduldef.ini'" file. This file is located in the directory "[LW]:\Program Files\Indramat\MTGUI\CustomData\Resource" and contains the data for all module configurations. The process data is located in three sections:

[DeviceAddrX\ModulY\Process]

whereby "X" stands for the device addressed and "Y" for the configured module numbers.

FI command Determine the module to which the process belongs. Information is read out from the module configuration of the MWAX device group.

BR_MAP1_(1)	**(Single Read)**
BC_MAP1_(1)	**(Cyclic Read)**
BB_MAP1_(1)	**(Break Cyclic Read)**

1 = Mechanism number [0]

Response Structure The following table shows the general structure of the response to the "MAP1" FI command. One line with one column is output for the module number that has been determined.

Line 1	Column 1

Value Range of the Column 1 = Module number [0...99]

Example MAP1 Read the module number which is assigned to NC process number 0 from the module configuration.

Assumption:
The module to which NC process 0 is assigned has module number 5.

FI command		**00_BR_MAP1_0**
Line	**Column**	**Answer**
1	1	5

Reference to Literature See chapter entitled "Literature" [36].

10.48 Read Reference Name of a PLC Variable: MAR

MWAX device group

Designation **MAR** **M**ap **A**bsolute PCL-**R**eference

PLC Explanation The absolute reference name of a symbolic PLC variable is read out.

FI command Read the absolute reference name of a PLC variable.

BR_MAR_(1) **(Single Read)**

BC_MAR_(1) **(Cyclic Read)**

(1) = Identifier of the PLC variable

Response Structure The following table shows the general structure of the response to the FI command "MAR". One line with one column is output for the reference name that has been determined.

Line 1	Column 1

Meaning of the Column 1 = Identifier of the PLC variable

PLC – Example MAR Read the absolute reference name of the PLC variable with the identifier "abref" at device address 00.

<u>Assumption:</u>
The PLC variable with the identifier "abref" is of the type "INTEGER".

FI command		**00_BR_MAR_abref**
Line	**Column**	**Answer**
1	1	%M100.0

WinPCL Explanation The absolute reference name of a symbolic WinPCL PLC variable with program entity is read out.

FI command Read the absolute reference name of a WinPCL PLC variable.

BR_MAR1_(1) **(Single Read)**

BC_MAR_(1) **(Cyclic Read)**

(1) = Identifier of the PLC variable

WinPCL - Example MAR1 Read the absolute reference name of the WinPCL variable with the identifier "Prog.abref" at device address 00.

<u>Assumption:</u>
The WinPLC variable with the identifier "Prog.abref" is of the type "INTEGER" and is present in WinPCL program entity "Prog".

FI command		**00_BR_MAR1_:Prog.abref**
Line	**Column**	**Answer**
1	1	%M100.0

Reference to Literature See chapter entitled "Literature" [30].

10.49 Device Data of the Module Configuration: MCD

MWAX device group

Designation **MCD** **M**odule **C**onfiguration: **D**evice Information

Explanation All device data for module configuration is read out from the "Moduldef.ini" file which is stored in the "[LW]:\Program Files\Indramat\MTGUI\CustomData\Resource" directory on standard installation. The device data is in the sections [DeviceAddrX], whereby "X" stands for the configured device addresses.

FI command Read-out device data within the module configuration of the MWAX device group.

BR_MCD1	**(Single Read)**
BC_MCD1	**(Cyclic Read)**
BB_MCD1	**(Break Cyclic Read)**

Note: The "MCD1" FI command refers to all devices within the MWAX device group. Therefore, any valid device address can be indicated in the command line (see example "MCD1").

Response Structure The following table shows the general structure of the response to the "MCD1" FI command. The number of lines depends on the number of configured devices. Each line consists of four columns for the device address as well as PLC-FB names for providing setup diagnostics, warning messages and start requirements.

Line 1	Column 1	...	Column 4

Value Range of the Columns

1 = Device address	[0...15]
2 = PLC-FB name for the setup diagnostics	[max. 9 ASCII characters]
3 = PLC-FB name for the warning messages	[max. 9 ASCII characters]
4 = PLC-FB name for the start requirements	[max. 9 ASCII characters]

Example MCD1 Read all device data of the module configuration

Assumption:
The following devices have been configured in the MWAX device group:

- Device address 01 (MTA200-P)
- Device address 03 (MTA200-R)

FI command	**03_BR_MCD1**			
Answer				
Line	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	01	PVSetup_1	PVWarn_1	PVStart_1
2	03	PVSetup_3	PVWarn_3	PVStart_3

Reference to Literature See chapter entitled "Literature" [36].

10.50 Module Data of the Module Configuration: MCM

MWAX device group

Designation **MCM** **M**odule **C**onfiguration: **M**odule Information

Explanation All module data for module configuration is read out from the "Moduldef.ini" file which is stored in the "[LW]:\Program Files\Indramat\MTGUI\CustomData\Resource" directory following standard installation. This file contains all module configuration data. The module data is located in sections [DeviceAddrX\ModulY], whereby "X" stands for the device addressed and "Y" for the configured module numbers.

FI command Read module data from the module configuration with respect to a device from the MWAX device group.

BR_MCM1	**(Single Read)**
BC_MCM1	**(Cyclic Read)**
BB_MCM1	**(Break Cyclic Read)**

Response Structure The following table shows the general structure of the response to the "MCM1" FI command. The number of lines depends on the number of configured modules of a device. Each line consists of four columns for the module number, module name and PLC-FB names for general module errors and module messages.

Line 1	Column 1	...	Column 4

Value Range of the Columns

1 = Module number	[0...99]
2 = Module name	[max. 28 ASCII characters]
3 = PLC-FB name for general module errors	[max. 9 ASCII characters]
4 = PLC-FB name for module messages	[max. 9 ASCII characters]

Example MCM1 Read the module data of device 03 from the module configuration:

Assumption:
The following modules have been defined:

- Module number 5
- Module number 7

FI command	**03_BR_MCM1**			
Answer				
Line	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	5	Module 5 – Milling	PVError_5	PVMsg_5
2	7	Module 7 - Drilling	PVError_7	PVMsg_7

Reference to Literature See chapter entitled "Literature" [36].

10.51 Process Data of the Module Configuration: MCP

MWAX device group

Designation **MCP** **M**odule **C**onfiguration: **P**rocess Information

Explanation All process data of a certain module is read out from the "Moduldef.ini" file which is stored in the "[LW]:\Program Files\Indramat\MTGUI\CustomData\Resource" directory following standard installation. This file contains all module configuration data. The process data is located in sections [DeviceAddrX\ModulY\Process], whereby "X" stands for the device addressed and "Y" for the selected module number.

BR_MCP1_(1)	**(Single Read)**
BC_MCP1_(1)	**(Cyclic Read)**
BB_MCP1_(1)	**(Break Cyclic Read)**
(1) = Module number	[0...99]

Response Structure The response to the FI command "MCP1" consists of one up to a maximum number of n=32 lines with 1 column for the number of the NC process or of the external mechanisms.

Line 1...32	Column 1

Value Range of the Column 1 = Mechanism number [0]

Example MCP1 Read the NC process number of module 5 of device 03 of the module configuration.

Assumption:
The following NC processes are defined:

- NC process number 0

FI command		00_BR_MCP1_5
Line	**Column**	**Answer**
1	1	1
2	1	0

Reference to Literature See chapter entitled "Literature" [36].

10.52 SFC Data of the Module Configuration: MCS

MWAX device group

Designation **MCS** **M**odule **C**onfiguration: **S**FC Information

Explanation All SFC data of a certain module is read out from the "Moduldef.ini" file which is stored in the "[LW]:\Program Files\Indramat\MTGUI\CustomData\Resource" directory following standard installation. This file contains all module configuration data. The SFC data is located in sections [DeviceAddrX\ModulY\Sfc], whereby "X" stands for the device addressed and "Y" for the selected module number.

FI command Read the SFC data with respect to the module of a device from the module configuration of the MWAX device group.

BR_MCS1_(1)	**(Single Read)**
BC_MCS1_(1)	**(Cyclic Read)**
BB_MCS1_(1)	**(Break Cyclic Read)**
(1) = Module number	[0...99]

Response Structure The number of lines depends on the number of configured Indrastep step chains for a device. Each line contains a column for the name of the Indrastep step chains.

Value Range of the Column 1 = Name of the Indrastep step chain [format W.X.Y.Z]

Format W.X.Y.Z	Value Range
W	Max. 9 ASCII characters
X	Max. 9 ASCII characters ! OPTIONAL !
Y	Max. 9 ASCII characters ! OPTIONAL !
Z	Max. 9 ASCII characters ! OPTIONAL !

Example MCS1 Read the name of the Indrastep step chain of module 5 from device 03 of the module configuration.

Assumption:
The following Indrastep step chains have been defined:

- ISFB_1
- FB_US.ISFB_3
- FB_US.ISFB_3.SW1
- FB_US.ISFB_3.SW1.ABBA

FI command		**03_BR_MCS1_5**
Line	**Column**	**Answer**
1	1	ISFB_1
2	1	FB_US.ISFB_3
3	1	FB_US.ISFB_3.SW1
4	1	FB_US.ISFB_3.SW1.ABBA

Reference to Literature See chapter entitled "Literature" [36].

10.53 Inputting an NC Record: MDI

MWAX device group

Designation **MDI** **M**anual **D**ata **I**nput

FI command Input an NC record for direct execution in manual mode.

BW_MDI1_(1) **(Single Write)**

(1) = MDI status [0...1
0 = Deactivate MDI
1 = Activate MDI]

Value to be written NC record

Note: More than one NC record can be written into the NC. In this case, the individual NC records must be separated by the line feed key. On activation of the MDI operation, the transferred NC record is converted and written into the NC. The value to be written is not evaluated when the MDI operation is deactivated.

Response Structure One line is output with a column for acknowledgement of whether or not the FI command has been executed successfully.

(P_ACK) = **P**ositive **ACK**nowledge Data element has been set

Example MDI1 Write an NC record into the NC which is to be processed directly.

Precondition The NC must be operative; it must not be in "Automatic mode active" or "Automatic mode interrupt" status; and there must be no error pending.

FI command		**Value to be written:** **G01 X1 50.45 Y1 35.456 F 1000** **00_BW_MDI1_1**
Line	**Column**	**Answer**
1	1	(P_ACK)

Example MDI1 Deactivate MDI mode.

Precondition The NC must be operative; it must not be in "Automatic mode active" or "Automatic mode interrupt" status; and there must be no error pending.

Value to be written not relevant, not evaluated

FI command		**00_BW_MDI1_0**
Line	**Column**	**Answer**
1	1	(P_ACK)

10.54 Downloading Message Texts: MFD

MWAX device group

Designation **MFD** **M**essage **F**iles **D**ownload

FI command This is used to load the message texts into the device indicated. These message texts are required for small devices. The following message texts are transmitted, depending on the type of device:

- system error messages
- transmission error messages
- mechanism messages

Note: This FI command is an FI job!

BW_MFD1 **(Single Write)**

Response Structure The response to the "MFD1" FI command consists of three lines, each with one column. The meaning of the elements is as follows:

Line 1 = Job ID	[01...20] (refer to chapter entitled "FI Commands for the MPCX Device Group", IFJ).
Line 2 = FI command	[string, in accordance to chapter entitled "Elements of the FI Command"]
Line 3 = FI job error code	(see chapter entitled "Error Codes")

Example MFD1 Load message texts into the device with device address 00.

FI command		**00_BW_MFD1**
Line	**Column**	**Answer**
1	1	01
2	1	00_BW_MFD1
3	1	0

10.55 Reading or writing the Status of the MM_INCH Fixed Path Keys: MIS

MWAX device group

Designation **MIS** **M**M_**I**NCH-**S**tatus

Explanation This command is used to read or write the status of the fixed path keys (jogging). The fixed path keys are evaluated accordingly as mm, or as 10th of an Inch.

FI command Read the status of the fixed path keys:

BR_MIS1 **(Single Read)**

Response Structure The following table shows the general structure of the response to the "MIS1" FI command.

Line 1	Column 1

Value Range/Meaning of Columns

1 = Status mm_inch	mm Inch

Example MIS1 Read the status of the fixed path keys of the NC. It is set to mm.

FI command		**00_BR_MIS1**
Line	**Column**	**Answer**
1	1	mm

Example MIS1 Read the status of the fixed path keys of the NC. It is set to Inch.

FI command		**00_BR_MIS1**
Line	**Column**	**Answer**
1	1	inch

FI command Write the status of the fixed path keys.

BW_MIS1_(1) **(Single Write)**

(1) = New status [mm, inch]

Response Structure After successful execution, the FI command will return the answer (P_ACK).

Example MIS1 Set the unit of the fixed path keys to Inch.

FI command		**00_BW_MIS1_inch**
Line	**Column**	**Answer**
1	1	(P_ACK)

10.56 Reading Machine Key Information : MKS

MWAX device group

Designation **MKS** **M**achine **K**ey **S**tatus

Explanation Current machine key information can be read for the selected device.

FI command Read machine key information for selected device.

BR_MKS	**(Single Read)**
BC_MKS	**(Cyclic Read)**
BB_MKS	**(Break Cyclic Read)**

Response Structure The following table shows the general structure of the response to the FI command "MKS".

Line 1	Column 1	Column 2

Value Range/Meaning of Columns

1 =	Machine key information	[4 byte in HEX coding]
2 =	Information valid?	[0 = not valid, 1=valid]

Example MKS Read the current machine key information for device 0.

FI command		**00_BR_MKS**
Line	**Column**	**Answer**
1	1	00000000
	2	0

10.57 Writing the GUI-SK Block: MKT

MWAX device group

Designation **MKT** **M**achine **K**ey **T**able

Explanation Writes the GUI-SK16 block in the PLC.

FI command Write GUI-SK16 block.

BW_MKT1_(1)	**(Single Write)**
(1) = List of the 48 PLC variables for writing the GUI-SK16 block.	The following cases are to be differentiated: 1.Delete the GUI-SK16 block: 2.Write the GUI-SK16 block with the 48 PLC variables, filling gaps with $SPACE.

Response Structure (P_ACK) is returned following successful transmission.

Line 1	Column 1

Value Range/Meaning of the Columns

1 =	Successfully completed	(P_ACK)

1. Example MKT1 1.Clear GUI-SK16 block:

FI command		**Value to be written: $EMPTY** **00_BW_MKT1**
Line	**Column**	**Answer**
1	1	(P_ACK)

2. Example MKT1

Write GUI-SK16 block:

FI command		**Value to be written: $EMPTY** **SPSVAR1,SPSVAR2,$SPACE,...** **00_BW_MKT1**
Line	**Column**	**Answer**
1	1	(P_ACK)

FI command

Write the GUI-SK16 block, writing only those PLC variables which are defined in the current PLC program. All undefined PLC variables are automatically replaced by $SPACE and returned as a partial result (column 2).

BW_MKT2_(1)		**(Single Write)**
(1) =	List of the 48 PLC variables for writing the GUI-SK16 block.	The following cases are to be differentiated: 1. Delete the GUI-SK-16 block: BW_MKT2 $EMPTY 2.Write the GUI-SK16 block with the 48 PLC variables, filling gaps with $SPACE: BW_MKT1 SPSVAR1,SPSVAR2, $SPACE,$SPACE,....

Response Structure

After successful transmission, one line with two columns is returned.

Line 1	**Column 1**	**Column 2**

Value Range/Meaning of Columns

1 =	Status report	[0 = at least 1 PLC variable in the current PLC program is NOT defined 1 = ALL PLC variables could be written]
2 =	List of the NON-defined PLC variables in the current PLC program	[-- = ALL PLC variables could be written, or else list of the PLC variables that could not be written.] The individual PLC variables are separated by a comma.

Example MKT1

Write GUI-SK16 component with 48 PLC variables, while the PLC variables SPSVAR11 and SPSVAR12 are NOT defined in the current PLC program.

FI command		**Value to be written:** **SPSVAR1,SPSVAR2,...SPSVAR48** **00_BW_MKT1**
Line	**Column**	**Answer**
1	1	(P_ACK)

Extended information

The variables are divided into 3 groups of 16 variables each and have the following meaning:

1. Variables 1 - 16: Machine function keys
2. Variables 17 - 32: Status pressed
3. Variables 33 - 48: Status shining

Notes: When, for example, only the first 8 M keys are used, the telegram will contain only these 8 PLC variables. The other 40 variables need not be defined in the transmission parameter.

When certain areas, e.g. of M keys, are left unused, they must be filled up with '$SPACE' up to the next variable.

10.58 Read System Messages: MSG

MWAX device group

Designation **MSG** **M**e**S**sa**G**e

Explanation Reading of system messages

FI command Message

CC_MSG_(1) **(Cyclic Read)**

(1) = SYS-Message number

Note: Exists only as a cyclic command

Response Structure The response of the FI command 'MSG' consists of the system message data.

Example MSG 00_CC_MSG_64 (64 = MSG_SYSERRGEN)

FI command		**00_CC_MSG_64/3**
Line	**Column**	**Answer**
1	1	00

Restriction

- The following system messages:

SYS Message	SYS Message number
MSG_PCLUPDBEG	52
MSG_PARUPDBEG	24
MSG_FWAUPDBEG	82

These commands cannot be used with the following programs:

- Bosch Rexroth OPC Server
- Bosch Rexroth DDE server

10.59 Reading the Firmware/Monitor Identification: MTC

MWAX device group

Designation **MTC** **MT-C**NC Slot Software Version

FI command This command is used to read the firmware identification from the various control components (slot numbers).

Note: For the time this FI command is executed, the internal FI communication interlocks (fast timeout monitoring, offline operation, etc.) are switched off.

FI command

BR_MTC_(1) **(Single Read)**

BC_MTC_(1) **(Cyclic Read)**

(1) = Slot number [1 = Andron firmware version,
2 = NOT assigned,
3 = PLC firmware version]

Response Structure The following table shows the general structure of the response to the FI command "MTC". A line of 1 column is output.

Line 1	Column 1

Value Range/Meaning of Columns 1 = Firmware identification string [max. 16 ASCII characters]

Example MTC Read the firmware identification of slot number 1 (CPU) of device 00.

FI command		**00_BR_MTC_1**
Line	**Column**	**Answer**
1	1	CPU01/0004-20V00

Explanation With the FI command "BR_MTC1", the monitor versions of the various components (CNC, PLC, APR) can be read out.

FI command **BR_MTC1** **(Single Read)**

Response Structure The response to the "BR_MTC1" FI command consists of six lines with four columns. One line is returned for each potential component (CNC, PLC, APR1-4).

Line 1.0.6	Column 1	Column 2	Column 3	Column 4

Value Range/Meaning of Columns

1 = Line number [1..6]

2 = Component information [CNC=NC component
SPS=PLC component
APR=APR component

3 = Monitor version – "old" format

4 = Monitor version – "new" format

Example MTC1 Read the monitor versions for device 0.

FI command		**00_BR_MTC1**
Line	**Column**	**Answer**
1	1	1
	2	CNC
	3	--
	4	--
2	1	2
	2	PLC
	3	MON-PMK 09.05/0705.02.01
	4	FWC-CONTROL-MON-06V00-NN
3	1	3
	2	APR
	3	--
	4	--
4	1	4
	2	APR
	3	--
	4	--
5	1	5
	2	APR
	3	--
	4	--
6	1	6
	2	APR
	3	--
	4	--

10.60 Reading NC Parameters: NPA

MWAX device group

Designation **NPA** NC-**PA**rameter

FI command Read a parameter line.

BR_NPA1_(1)_(2) **(Single Read)**

(1) = Parameter record number	[1..99]
(2) = Parameter number	[aA00.000..aCxx.120]

Response Structure The following table shows the general structure of the response to the FI command "NPA1". One line is output with 3 columns for the identifier, the value and the name respectively.

Line 1	Column 1	Column 2	Column 3

Value Range/Meaning of Columns

1 = Identifier	Parameter ID [max. 32 ASCII characters]
2 = Value	[ASCII text]
3 = Name	[unit, related to the value or empty]

Example NPA1 Return the parameter line from parameter record 10 with parameter number aB00.007.

Assumption:
Parameter record 10 has been created and process 00 has been defined. Here, the following information is to be found:
Max. past acceleration 75 mm/sec^2.

FI command		00_BR_NPA1_10_aB00.007
Line	**Column**	**Answer**
1	1	AB00.007
	2	75
	3	mm/sec^2

FI command Read out several parameter lines from a parameter record.

BR_NPA2_(1)_(2)_(3) **(Single Read)**

(1) = Parameter record number	[1..99]
(2) = Parameter number [from]	[aA00.000..aCxx.120]
(3) = Parameter number [to]	[aA00.000..aCxx.120]

Response Structure The following table shows the general structure of the response to the FI command "NPA2". As many lines as are requested are output, each with three columns for the identifier, the value and the name respectively.

Line 1...n:	Column 1	...	Column 3

Value Range/Meaning of Columns

1 = Identifier	[max. 32 ASCII characters]
2 = Value	[ASCII text]
3 = Name	[unit, related to the value or empty]

Example NPA2 Return the parameter lines from parameter record 10 of parameter number aA00.000 to parameter number aA00.001.

Assumption:
Parameter record 10 has been created and contains the following information in this location:

FI command		**00_BR_NPA2_10_aA00.000_aA00.001**
Line	**Column**	**Answer**
1	1	aA00.000
	2	Master
	3	
2	1	aA00.001
	2	Process 1
	3	

Explanation It is possible to read a list with a maximum of 10 parameters of the same type (system parameters, process parameters or axis parameters).

FI command Read NC parameters for a selected device.

BR_NPA5_(1)_(2)_{(3)..(12)} **(Single Read)**

(1) = Parameter type	1 = System parameter 2 = Process parameter 3 = Axis parameter
(2) = Process number or axis number	If "system parameter" has been selected as the type of parameter, then this parameter is NOT evaluated – set to 0.
(3)... ..(12) = List of requested parameters	A maximum of 10 parameters of the same type may be listed here. Please take the parameter number from the general description of parameters for the control unit.

Response Structure The following table shows the general structure of the response to the FI command "NPA5".

Line 1...n	Column 1	...	Column 3

Value Range/Meaning of Columns

1 =	Parameter number	Parameter number that has been requested.
2 =	Parameter value	Data setup – see general description of parameters.
3 =	Parameter unit	Data setup – see general description of parameters.

Example NPA5 NC parameter request for system parameters 0,52,53.

FI command		**00_BR_NPA5_1_0_0_52_53**
Line	**Column**	**Answer**
1	1	0
	2	Master
	3	--
2	1	52
	2	0
	3	--
3	1	53
	2	1
	3	--

Reference to Literature See chapter entitled "Literature" [38].

10.61 Parameter Download: PAA

MWAX device group

Designation **PAA** **PA**rameter **A**ccess

Explanation Complete parameter records are downloaded by means of a download file.

FI command Parameter download command whereby the parameter download file is directly indicated.

BW_PAA2_(1) **(Single Write)**

(1) = Complete parameter download file name

Response Structure The response to the "PAA2" FI command consists of three lines, each with one column. The meaning of the elements is as follows:

- Line 1 = Job ID [01...20]
  (see Chapter "FI Commands for the MPCX Device Group", IFJ).
- Line 2 = FI command
  [String, in accordance with Chapter "Elements of the FI Command"]
- Line 3 = FI Job Error Code
  (see Chapter "Error Codes")

Note: File and path details must be enclosed in inverted commas.

Example PAA2 00_BW_PAA2_"D:\DOWNLOAD.DAT"

FI command		**00_BW_PAA2_"D:\DOWNLOAD.DAT"**
Line	**Column**	**Answer**
1	1	01
2	1	00_BW_PAA2_"D:\DOWNLOAD.DAT"
3	1	0

Structure of Download File The structure of the download file corresponds to that of a Windows Ini file. Bosch Rexroth’s own description in V20_Param_08_Definitions_Parameter_Download_01.doc is recommended for a more detailed account of the structure of the download file.

Summary:

Section [ID_PARAMETER]
Information concerning parameter identification.

Section [ID_SYSTEM]
Information concerning system parameter identification.

Section [DATA_SYSTEM]
Listing of system parameter data.

Section [ID_PROCESSX]
Information concerning process parameter identification.

Section [DATA_PROCESSX]
Listing of process parameter data.

Section [ID_AXISX]
Information concerning axis parameter identification.

Section [DATA_AXISX]
Listing of axis parameter data.

10.62 ProVi Diagnosis Data: PDD

MWAX device group

Designation **PDD** **P**rovi **D**iagnosis **D**ata

Explanation Data for ProVi criteria analysis is output.

FI command Output of files to indicate the detail in the editor.

BR_PDD1_(1)_(2){_(3)}	**(Single Read)**
(1) = Message ID	[ASCII characters]
(2) = Message type	[1 = error, 2 = messages, 10 = warnings, 11 = start requirements, 12 = setup diagnosis]
(3) = Module number	[1...99] ! only for message type 1 -2!

Response Structure The following table shows the general structure of the PDD1 FI command.

Line 1	Column 1	...	Column 5

Meaning of the Columns

1 = POU ID	[ASCII characters]
2 = Detail morpheme	[ASCII characters] (DWORD, decimal)
3 = Error ID	[ASCII characters] (DWORD, decimal)
4 = POU entity name	[ASCII characters]
5 = Nw ID (network ID)	[ASCII characters]

Example PDD1 Indication of data of a ProVi error with ID 43923028 from module 3 in control unit 0.

FI command		**00_BR_PDD1_43923028_1_1**
Line	**Column**	**Answer**
1	1	STATION_1_2
	2	98243823
	3	34985304
	4	Station2.Module3
	5	43493454

FI command Output the I/O addresses to display a detail.

BR_PDD2_(1)_(2){_(3)}	**(Single Read)**
(1) = Message ID	[ASCII characters]
(2) = Message type	[1 = error, 2 = messages, 10 = warnings, 11 = start requirements, 12 = setup diagnosis]
(3) = Module number	[1...99] ! only for message type 1 -2!

Response Structure The following table shows the general structure of the PDD2 FI command.

Line 1-n	Column 1	Column 2

Meaning of the Columns

1 = Variable morpheme	[ASCII characters] (DWORD, decimal)
2 = I/O address	[ASCII characters]

Example PDD2

Query of the I/O addresses of a ProVi error with ID 43923028 from module 3 in control unit 0.

Three variables have an I/O address.

FI command		**00_BR_PDD2_43923028_1_1**
Line	**Column**	**Answer**
1	1	98243823
	2	%I3.2.0
2	1	40923423
	2	%Q23.21.7
3	1	34985304
	2	%I100.3.5

FI command

Determine the multilingual comments for displaying a detail.

BR_PDD3_(1)_(2){_(3)}	**(Single Read)**
(1) = Message ID	[ASCII characters]
(2) = Message type	[1 = error, 2 = messages, 10 = warnings, 11 = start requirements, 12 = setup diagnosis]
(3) = Module number	[1...99] ! only for message type 1 -2!

Response Structure

The following table shows the general structure of the PDD3 FI command.

Line 1-n	**Column 1**	**Column 2**

Meaning of the Columns

1 = Comment morpheme	[ASCII characters] (DWORD, decimal)
2 = New comment	[ASCII characters]

Example PDD3

Query of the comments for indication of a ProVi error with ID 43923028 from module 3 in control unit 0.

Two comments are replaced by another text.

FI command		**00_BR_PDD3_43923028_1_1**
Line	**Column**	**Answer**
1	1	98243823
	2	Clamp open
2	1	40923423
	2	Clamp closed

FI command

Query of the status of a certain message

BR_PDD4_(1)_(2){_(3)}	**(Single Read)**
(1) = Message number	[ASCII characters]
(2) = Message type	[1 = error, 2 = messages, 10 = warnings, 11 = start preconditions, 12 = setup diagnosis]
(3) = Module number	[1...99] ! only for message type 1 -2!

Response Structure

The following table shows the general structure of the PDD4 FI command.

Line 1-n	**Column 1**	**Column 2**

Meaning of the Columns

1 = Message is present [YES, NO]

2 = Criteria analysis exists [YES, NO]

Example PDD4

Query of the status of a ProVi error, number 1001 from module 3 in control 0.

This message is not present at the moment, and there is a criteria analysis.

FI command		**00_BR_PDD4_1001_1_1**
Line	**Column**	**Answer**
1	1	NO
	2	YES

FI command

Determination of the MessageID of a certain message

BR_PDD5!(1)!(2)!(3)!(4){!(5)} **(Single Read)**

(1) = POU entity name [ASCII characters]

(2) = Nw ID [ASCII characters]

(3) = Message number [ASCII characters]

(4) = Message type [1 = error, 2 = messages, 10 = warnings, 11 = start requirements, 12 = setup diagnosis]

(5) = Module number [1...99] ! only for message type 1 -2!

Note: The separator "!" is used in this command.

Response Structure

The following table shows the general structure of the PDD5 FI command.

Line 1-n	**Column 1**	**...**	**Column 3**

Meaning of the Columns

1 = Message ID [ASCII characters] (DWORD, decimal)

2 = Message is present [YES, NO]

3 = Criteria analysis exists [YES, NO]

Example PDD5

Determination of the MessageID of a ProVi error, number 1001 from module 25.40 mm control 0.

Assumption:
This message is not present at the moment, and there is a criteria analysis.

FI command		**00_BR_PDD5!Station2.Modul3!43493454!1001!1!1**
Line	**Column**	**Answer**
1	1	240872342
	2	NO
	3	YES

10.63 Programmed Feedrate: PFR

MWAX device group

Designation **PFR** **P**rogrammed **F**eed **R**ate

Explanation The programmed value of the feed rate is output. In an NC program, the details of the feedrate are expressed by means of a feedrate value.

FI command The programmed value of the feedrate is output.

Using the optional second parameter it is possible to pre-select conversion of the result into mm or inches.

BR_PFR_(1){_(2)} **(Single Read)**

(1) = CNC process number [0]

(2) = Required measurement system (opt.) [mm, inch]

Note: Only "0" can be input as a CNC process number. This is necessary for reasons of compatibility of the FI command.

Response Structure The following table shows the general structure of the response to the FI command "PFR". One line with three columns is output for the identifier, the current value of the feedrate and the unit.

Line 1	Column 1	...	Column 3

Value Range/Meaning of Columns

1 = Identifier	[F = feedrate]
2 = Value	[Accuracy: one decimal place]
3 = Unit	[depending on the settings of the basic programming unit, e.g. the preset command]

Example PFR Reads the current feedrate in CNC process 0 of device address 00.

FI command	**00_BR_PFR_0**		
Answer			
Line	**Column 1**	**Column 2**	**Column 3**
1	F	30000.0	[mm/min]

Example PFR Reads the current feedrate in CNC process 0 of device address 00. The displayed value is to be converted into inch/min:

FI command	**00_BR_PFR_0_inch**		
Answer			
Line	**Column 1**	**Column 2**	**Column 3**
1	F	1181.1	[inch/min]

10.64 Reading the Size of the PLC Memory: PMI

MWAX device group

Designation **PMI** **P**lc **M**emory **I**nformation

Explanation The current size of the PLC memory is read out.

FI command **CR_PMI** **(Single Read)**

Response Structure One line with two values in BYTE is output:

1. Total memory
2. Free memory available now.

Line 1	Column 1	Column 2

Example PMI Read the current size of the PLC memory at device address 00.

FI command		00_CR_PMI
Line	**Column**	**Answer**
1	1	123456
	2	3210

10.65 Actual (Current) Information on Position of All Axes: POI

MWAX device group

Designation **POI** **PO**sition **I**nformation

Explanation The current position information for all axes are read. The FI command "POI1" returns all necessary data for indicating the position.

FI command

BR_POI1_(1){_(2)}	**(Single Read)**
BC_POI1_(1){_(2)}	**(Cyclic Read)**
BB_POI1_(1){_(2)}	**(Break Cyclic Read)**
(1) = updated position information	[0...31, 1 = axis has been homed 2 = machine coordinates 4 = program coordinates 8 =relative coordinates 16 = distance to go] all combinations are possible!
(2) = Required measurement system (opt.)	[mm, inch]

Response Structure The following table shows the general structure of the response to the FI command "POI1". 16 lines with 8 columns are returned for axis type, axis name, axis has been homed, position values in the various systems of coordinates, distance to go, and unit.

Line 1	Column 1	Column 2	Column 3	Column 4	Column 5	Column 6	Column 7	Column 8
Line 2								
...								
Line 16								

Value range/Meaning of lines

Line = axis meaning
[1 = A axis, 2 = X axis
3 = Z axis, 4 = Y axis
5 = B axis, 6 = C axis
7 = D axis, 8 = E axis
9 = X' axis, 10 = Y' axis
11 = P axis, 12 = Q axis
13 = R axis, 14 = U axis
15 = V axis, 16 = W axis]

Value Range/Meaning of Columns

1 = Axis type	[0 = axis not defined 1 = Linear axis 2 = rotary axis 3 = Modulo axis 4 = main spindle]
2 = Axis name	[according to settings of axis parameters]
3 = Axis has been homed	[0 = axis has not been homed 1 = axis has been homed]
4 = Machine coordinates	
5 = Program coordinates	
6 = Relative coordinates	
7 = Distance to go	
8 = Required measurement system (opt.)	[mm, inch]

Note: If an axis is not defined then the response in all columns is [--].

Example POI1

Read for all axes: axis type, axis name, machine coordinates, program coordinates, distance to go, and unit. Values are displayed in the basic measurement system.

Assumption:
The axes X, Y, Z, C, B and X' are defined.

<table>
<tr><th colspan="3">FI command</th><th colspan="6">00_BR_POI1_22</th></tr>
<tr><th colspan="9">Answer</th></tr>
<tr><th>Line</th><th>Column 1</th><th>Column 2</th><th>Column 3</th><th>Column 4</th><th>Column 5</th><th>Column 6</th><th>Column 7</th><th>Column 8</th></tr>
<tr><td>1</td><td>0</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td></tr>
<tr><td>2</td><td>1</td><td>X</td><td>--</td><td>148.0000</td><td>123.0000</td><td>--</td><td>0.0000</td><td>[mm]</td></tr>
<tr><td>3</td><td>1</td><td>Z</td><td>--</td><td>23.4548</td><td>56.0000</td><td>--</td><td>0.0000</td><td>[mm]</td></tr>
<tr><td>4</td><td>1</td><td>Y</td><td>--</td><td>0.0000</td><td>0.0000</td><td>--</td><td>0.0000</td><td>[mm]</td></tr>
<tr><td>5</td><td>2</td><td>B</td><td>--</td><td>180.0000</td><td>180.0000</td><td>--</td><td>16.0000</td><td>[deg]</td></tr>
<tr><td>6</td><td>2</td><td>C</td><td>--</td><td>270.0000</td><td>90.0000</td><td>--</td><td>0.0000</td><td>[deg]</td></tr>
<tr><td>7</td><td>0</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td></tr>
<tr><td>8</td><td>0</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td></tr>
<tr><td>9</td><td>4</td><td>X’</td><td>--</td><td>0.0000</td><td>0.0000</td><td>--</td><td>0.0000</td><td>[%]</td></tr>
<tr><td>10</td><td>0</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td></tr>
<tr><td>11</td><td>0</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td></tr>
<tr><td>12</td><td>0</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td></tr>
<tr><td>13</td><td>0</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td></tr>
<tr><td>14</td><td>0</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td></tr>
<tr><td>15</td><td>0</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td></tr>
<tr><td>16</td><td>0</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td></tr>
</table>

10.66 Reading the NC Record Display: PPS

MWAX device group

Designation **PPS** Part **P**rogram **S**equence

Explanation This command is used to read out 5 NC records: the current one, the two previous ones, and the two following ones.

FI command Reading the NC record display.

BR_PPS7_(1)	**(Single Read)**
(1) = Activate record display	[0...1 0 = Deactivate 1 = Activate]

Response Structure The following table shows the general structure of the response to the FI command "PPS7".

Line1..5	Column 1

Value Range/Meaning of Columns

Line 1 = Two NC records before the current record
Line 2 = One NC record before the current record
Line 3 = Current NC record
Line 4 = One NC record after the current record
Line 5 = Two NC records after the current record

Example PPS7 Read the current NC recorded display

FI command		**00_BR_PPS7_1**
Line	**Column**	**Answer**
1	1	G1 X81.68497 Y26.17664
2	1	G3 X80.8039 Y24.35733 I81.40607 J25.18871
3	1	G2 X81.10019 Y24.12975 I76.90702 J18.9772
4	1	X81.73686 Y23.4987 I77.28121 J19.64001
5	1	G3 X83.76677 Y21.46255 I101.64243 J41.31321

10.67 Setting of the Preset Data: PSD

MWAX device group

Designation **PSD** Set **PreS**et **D**ata

FI command Apart from the "Set absolute 0" function, which allows for the coordinate system of the PLC controller to be set to 0, the preset function makes it possible to shift the coordinate system of the controller within the machine. As a precondition, the operating mode PRESET must have been activated in the parameters.

BW_PSD1_(1)	**(Single Write)**
(1) = Axis number preset	[0...16 0 = Deactivate preset for all axes 1..16 = Activate axis number for preset]

Value to be written Set pass value for preset

Note: The preset value corresponds to the vector between machine coordinate system and controller coordinate system. If "0" is passed as a parameter, the preset data for all axes is deleted. If the preset value is to be deleted for only one axis, "0" must be written as a pass value (vector).

Response Structure One line is output with a column for acknowledgement of whether or not the FI command has been executed successfully.

(P_ACK) = **P**ositive **ACK**nowledge	Data element has been set

Example PSD1 Set the preset value of the X axis to 120. The machine coordinates of the X axis are e.g. 30; after the "preset" function, the controller coordinates of the X axis will be 150.

Precondition The NC must be operative; it must not be in "Automatic mode active" or "Automatic mode interrupt" status; and there must be no error pending.

FI command		**Value to be written: 120.0** **00_BW_PSD1_2**
Line	**Column**	**Answer**
1	1	(P_ACK)

Example PSD1 Deletes the preset data of all axes.

Precondition The NC must be operative; it must not be in "Automatic mode active" or "Automatic mode interrupt" status; and there must be no error pending.

Value to be written not relevant, not evaluated

FI command		**00_BW_PSD1_0**
Line	**Column**	**Answer**
1	1	(P_ACK)

10.68 Issuing SYS Messages Specific to the PCL: PSM

MWAX device group

Designation **PSM** PCL **S**ys **M**essage

Explanation Issues the most important SYS messages regarding the PCL programming interface – required for remote programming.

Note:
The appropriate device address is passed as the write value.

It allows the following SYS messages to be initiated:

- start of PCL download,
- end of PCL download,
- start of PCL online edit,
- end of PCL online edit,
- start of PCL declaration change.
- end of PCL declaration change.

FI command

Issue the most important PCL SYS messages.

BW_PSM1_(1)	**(Single Write)**
(1) =Required SYS message	[1= start of PCL download 2= end of PCL download 3= start of PCL online edit 4= end of PCL online edit 5= start of PCL declaration change 6= end of PCL declaration change]

Value to be written: Device address

Response Structure

The following table shows the general structure of the response to the FI command "PSM1".

Line 1	Column 1	...	Column 8

Value Range/Meaning of Columns

1 =	Status report	[READY=SYS message has been correctly acknowledged by the WIN32 applications] [ERROR=SYS message has NOT been acknowledged by a WIN32 application within the pre-set time]
2 =	Task name (LogInlf name)	[Task name that has triggered the SYS message]
3 =	SYS message number	[contains the issued SYS message number]
4 =	Acknowledgement time	[contains the pre-set acknowledgement time]
5 =	Reference information	[contains, where applicable, the additional information transferred as a write value]
6 =	Length of the reference information	[0 where NO additional information has been transferred]
7 =	Where applicable, LOG channel of the FI that has NOT acknowledged	[-- = acknowledgements have been completed in time or the LOG channel number of the WIN32 application that has NOT acknowledged in time]
8 =	Where applicable, task name that has NOT acknowledged in time.	[-- = acknowledgements have been completed in time or the task name that has NOT acknowledged in time]

Example PSM1

Issue the SYS message Beginning PCL Download. The additional information, device address 00, is also transferred as a write value.

FI command		**value to be written: 00** **XX_BW_PSM1_1**
Line	**Column**	**Answer**
1	1	READY
	2	WINPCL.EXE
	3	14
	4	30000
	5	00
	6	2
	7	--
	8	--

10.69 Edit PROVI Message Files: PVA

MWAX device group

Designation **PVA** **P**ROVI-Messages **A**ccess

Explanation This write command creates PROVI message files. With this write value, it is possible to decide whether the PROVI messages are to be generated according to the current PLC project, or selectively.

FI command **BW_PVA1** **(Single Write)**

Note: This command is an FI job command.

Value to be written

No write value exists	PROVI message files according to the current PLC project.
Write value exists	List of the requested PROVI message files (separated by a comma) according to the format: [PROVI-Diag-type: module number] Example: 01:01,01:02,02:02

Note: The value to be written is passed to the "acValue" parameter as an ASCII value in the "DataTransfer" routine.

Response Structure The response to the "BW_PVA1" FI command consists of three lines, each with one column. The meaning of the elements is as follows:

- Line 1 = Job ID [01...20]
  (see Chapter "FI Commands for the MPCX Device Group", IFJ).
- Line 2 = FI command
  [String, in accordance with Chapter "Elements of the FI Command"]
- Line 3 = FI Job Error Code (see Chapter "Error Codes")

....

Example PVA1 No write value is passed, i.e. the PROVI message files are generated according to the current PLC project.

FI command		**00_BW_PVA1**
Line	**Column**	**Answer**
1	1	01
2	1	00_BW_PVA1
3	1	0

Explanation The read command returns the most significant information on the created PROVI message files.

FI command **BR_PVA1** **(Single Read)**

....

Response Structure The following table shows the general construction of the answer of the FI command BR_PVA1. For each available PROVI message file, 1 line with 10 columns each is created.

Line 1...n	**Column 1**	**...**	**Column 10**

Value Range/Meaning of Columns

1 =	PROVI diagnostic type	[1..20]
2 =	PROVI diagnosis type designation	[The following designations can be returned: StartCondition, Error, Message, Warning, Setup]
3 =	Module number	[1..99]
4 =	PROVI diagnosis type and module number	[PROVI diagnosis type: module number, see write value for BW_PVA2]
5 =	Complete name of the PROVI message text file	[max. 200 ASCII characters]
6 =	Memory required for PROVI messages in the control	[figure in ASCII format]
7 =	Complete name of the PROVI index file	[max. 200 ASCII characters]
8 =	Memory required for PROVI index files in the control	[figure in ASCII format]
9 =	Total memory (text+index) required in the control	[figure in ASCII format]
10 =	Total memory for ALL PROVI files (text+index) required in the control	[figure in ASCII format]

Example PVA1

The most significant information of 2 available PROVI message files are returned.

FI command		**00_BR_PVA1_1**
Line	**Column**	**Answer**
1	1	1
	2	Error
	3	1
	4	01:01
	5	D:\Program Files\indramat\Mtgui\Project_000\ Program Data\Device_000\Diag\De\ERROR 1.TXT
	6	1345
	7	D:\Program Files\indramat\Mtgui\Project_000\ Program Data\Device_000\Diag\De\ERROR 1.IDX
	8	234
	9	1579
	10	4491
2	1	2
	2	Message
	3	1
	4	02:01
	5	D:\Program Files\indramat\Mtgui\Project_000\ Program Data\Device_000\Diag\De\ERROR 1.TXT
	6	2456

FI command		**00_BR_PVA1_1**
Line	**Column**	**Answer**
	7	D:\Program Files\indramat\Mtgui\Project_000\ Program Data\Device_000\Diag\De\ERROR 1.TXT
	8	456
	9	2912
	10	4491

Explanation

This write command transmits PROVI message files into the selected device. Through the write value, it is possible to chose whether ALL or only the PROVI messages selected via the write value are to be transmitted.

FI command

BW_PVA2 **(Single Write)**

Note: This command is an FI job command.

Value to be written

No write value exists	All PROVI message files are transmitted into the selected device
Write value exists	List of the requested PROVI message files (separated by a comma) according to the format: [PROVI-Diag-type: module number] Example: 01:01,01:02,02:02

Note: The value to be written is passed to the "acValue" parameter as an ASCII value in the "DataTransfer" routine.

Response Structure

The response to the "BW_PVA2" FI command consists of three lines, each with one column. The meaning of the elements is as follows:

- Line 1 = Job ID [01...20]
  (see Chapter "FI Commands for the MPCX Device Group", IFJ).
- Line 2 = FI command
  [String, in accordance with Chapter "Elements of the FI Command"]
- Line 3 = FI Job Error Code (see Chapter "Error Codes")

....

Example PVA2

No write value is passed, i.e. all PROVI message files should be transmitted.

FI command		**00_BW_PVA2**
Line	**Column**	**Answer**
1	1	01
2	1	00_BW_PVA2
3	1	0

10.70 Formatted Input / Output of PLC Variables: PVF

MWAX device group

Designation **PVF** **P**LC **V**ariable **F**ormatted

Explanation Formatted reading and writing of PLC variables, arrays and structures.

FI command Read PLC variables.

CR_PVF_(1)	**(Single Read)**
CC_PVF_(1)	**(Cyclic Read)**
CB_PVF_(1)	**(Break Cyclic Read)**
(1) = Identifier of the PLC variable	[acc. to declaration part of the PLC]

Response Structure One line with one column is output for single variables. For array and structure variables, one line per element is output, depending on the number of elements.

Line 1...n:	Column 1

n = number of elements.

Note: Only defined PLC variables can be read and written. Addressing a non-declared variable results in an error message. A PLC variable can only be read if its data length does not exceed 240 byte (refer also to chapter "Programming" and "Guidelines").

Value Ranges ANSI / ASCII The value range of the response depends on the data type of the variable read. The following table indicates the range in which the results string is to be expected when reading out a single variable and into which C-data type this string can be converted without loss of information:

Data Type	Value Range	Can be converted to C-data type
BOOL	[0;1]	unsigned char
SINT	[-128...127]	char
INT	[-32768...32767]	short
DINT	[2147483648...2147483647]	long
USINT	[0...255]	unsigned char
UINT	[0...65535]	unsigned short
UDINT	[0...4294967295]	unsigned long
BYTE	[0x00...0xFF]	unsigned char
WORD	[0x0000....0xFFFF]	unsigned short
DWORD	[0x00000000...0xFFFFFFFF]	unsigned long
TIME	[0...4294967295]	unsigned long (msec)
CHAR	[$00...$20,!...~,$7F...$FF]	char
STRING	<String> whereby <String> string is a character string with a maximum of as many characters as are declared for the string in the PLC	Char[xx+1]] +1 i.e. room for the zero byte
REAL	[-3.402823567E+38...3.402823567E+38]	Float

Note: An empty string is identified by two single inverted commas: ' ' (do not confuse with the double inverted commas ")!

All single variables can be part of array and structure variables. The value ranges maintain their validity, even when within structured data types.

Binary Value Range

The value range of the response depends on the data type of the variable read. The following table indicates the value range in which to expect the binary value of a single variable and how many bytes are included in the binary byte sequence:

Data Type	Value Range	Length (bytes)
BOOL	[00_H...01_H]	1
SINT	[80_H...$7F_H$] i.e. –128...127	1
INT	[8000_H (-32768)...$7FFF_H$ (32767)]	2
DINT	[80000000_H (-2147483648)... $7FFFFFFF_H$ (2147483647)]	4
USINT	[00_H (0)...FF_H (255)]	1
UINT	[00_H (0)...$FFFF_H$ (65535)]	2
UDINT	[0...4294967295]	4
BYTE	[0x00...0xFF]	1
WORD	[0x0000....0xFFFF]	2
DWORD	[0x00000000...0xFFFFFFFF]	4
TIME	[0...4294967295]	4
CHAR	[$00...$20,!...~,$7F...$FF]	1
STRING	<String> whereby <String> string is a character string with a maximum of as many characters as are declared for the string in the PLC	XX+1
REAL	[-3.402823567E+38...3.402823567E+38]	4

Note: Binary array and structure elements are joined together without any spaces between (1-byte alignment).

PLC - Example 1 PVF

Read the value of the PLC variable "STK_TXT" in ASCII format from device address 00.

Assumption:
The "STK_TXT" variable is declared as STRING in the PLC program.

FI command		**00_CR_PVF_STK_TXT/1**
Line	**Column**	**Answer**
1	1	Repeat counter

WinPCL - Example 1 PVF

Read the value of WinPCL variable "STK_TXT" in ASCII format in WinPCL program "Prog" at device address 00.

Assumption:
The WinPCL variable "STK_TXT" is declared in WinPCL program "Prog" as STRING.

FI command		**00_CR_PVF_:Prog.STK_TXT/1**
Line	**Column**	**Answer**
1	1	Repeat counter

PLC - Example 2 PVF

Read the value of the PLC array "BEG_END" in ANSI format from device address 00.

Assumption:
The "BEG_END" variable is declared as BYTE with 2 elements in the PLC program.

FI command		**00_CR_PVF_BEG_END/3**
Line	**Column**	**Answer**
1	1	0x00
2	1	0x1F

WinPCL - Example 2 PVF

Read the value of WinPCL array "BEG_END" in ANSI format in WinPCL program "Prog" at device address 00.

Assumption:
The WinPCL variable "BEG_END" is declared in WinPCL program "Prog" as BYTE with two elements.

FI command		**00_CR_PVF_:Prog.BEG_END/3**
Line	**Column**	**Answer**
1	1	0x00
2	1	0x1F

PLC - Example 3 PVF

Read the value of the PLC structure "MSTRCT" in ASCII format from device address 00.

Assumption:
The "MSTRCT" variable is declared as a structure in the PLC program as follows:

```
TYP STRUCT
        T1      BOOL
        T2      CHAR
        T3      STRING[16]
        T4      TIME
END
```

FI command		**00_CR_PVF_MSTRCT/1**
Line	**Column**	**Answer**
1	1	0
2	1	A
3	1	ROBOT AXIS X
4	1	2000

WinPCL - Example 3 PVF

Read the value of WinPCL structure "MSTRCT" in ASCII format in WinPCL program "Prog" at device address 00.

Assumption:
The WinPCL variable "MSTRCT" is declared as a structure in WinPCL program "Prog" as follows:

```
TYP STRUCT
        T1      BOOL
        T2      CHAR
        T3      STRING[16]
        T4      TIME
END
```

FI command		**00_CR_PVF_:Prog.MSTRCT/1**
Line	**Column**	**Answer**
1	1	0
2	1	A
3	1	ROBOT AXIS X
4	1	2000

FI command

Write PLC variable.

CW_PVF_(1)	**(Single Write)**
(1) = Identifier of the PLC variable	[acc. to declaration part of the PLC]

Value to be written

Value of data element	[see value ranges]

Note: The value to be written is passed to the "acValue" parameter in the "DataTransfer" routine. The data code of the value is passed to the parameter "ValType".

Response Structure

One line is output with a column for acknowledgement of whether or not the FI command has been executed successfully.

(P_ACK) = **P**ositive **ACK**nowledge	Data element has been set

Value Range of the value to be written in ANSI / ASCII Format

The value ranges agree for the most part with the ANSI / ASCII result-value ranges during read access. ANSI umlauts are thereby converted into ASCII umlauts. Only ASCII umlauts are stored in the control unit. For deviations to this, please refer to the following note:

Note: Strings are enclosed by two single inverted commas ' ' , e.g. 'drill'.

Special characters can be indicated in accordance with DIN-1131 by a $ sign.

The following are used:

$'	'	
$$	$	
$R	\r	(Carriage Return)
$L	\n	(Linefeed)
$P	\f	(Form feed)
$T	\t	(Tab)
$xx	xx	refers to a character written as a hexadecimal value. e.g. $20 (space)

Array and structure elements are separated by a space.

Value Range of the Value to be written in Binary Format

The value ranges agree with the binary result-value range during read access. For deviations to this, please refer to the following note:

PLC - Example 4 PVF

Write into the PLC variable "STK_TXT" at device address 00. The value is passed in ANSI format.

Assumption:
The "STK_TXT" variable is declared as STRING in the PLC program.

FI command		**00_CW_PVF_STK_TXT/3**
Line	**Column**	**Answer**
1	1	(P_ACK)

Value to be written:

Value of data element	'item counter'
Data code	/3

WinPCL - Example 4 PVF

Write into the WinPCL variable "STK_TXT" in WinPCL program "Prog" at device address 00. The value is passed in ANSI format.

Assumption:
The WinPCL variable "STK_TXT" is declared in WinPCL program "Prog" as STRING.

FI command		**00_CW_PVF_:Prog.STK_TXT/3**
Line	**Column**	**Answer**
1	1	(P_ACK)

Value to be written:

Value of data element	'item counter'
Data code	/3

PLC - Example 5 PVF

Write into the PLC byte array "BEG_END" at device address 00. The value is passed in ANSI format.

Assumption:
The "BEG_END" variable is declared as a BYTE array with 2 elements in the PLC program.

FI command		**00_CR_PVF_BEG_END/3**
Line	**Column**	**Answer**
1	1	(P_ACK)

Value to be written:

Value of data element	0x20 0x3f
Data code	/3

WinPCL - Example 5 PVF

Write into the WinPCL byte array "BEG_END" in WinPCL program "Prog" at device address 00. The value is passed in ANSI format.

Assumption:
The WinPCL variable "BEG_END" is declared in WinPCL program "Prog" as BYTE with two elements.

FI command		**00_CW_PVF_:Prog.BEG_END/3**
Line	**Column**	**Answer**
1	1	(P_ACK)

Value to be written:

Value of data element	0x20 0x3f
Data code	/3

PLC - Example 6 PVF

Write the value of element T3 of the PLC structure "MSTRCT" at device address 00. The string "COUNTER" is output in binary format.

Assumption:
The "MSTRCT" variable is declared as a structure in the PLC program as follows:

```
TYP STRUCT
        T1      BOOL
        T2      CHAR
        T3      STRING[16]
        T4      TIME
END
```

FI command		**00_CW_PVF_MSTRCT.T3/2**
Line	**Column**	**Answer**
1	1	(P_ACK)

Value to be written:

Value of data element	Binary sequence: 43 4F 55 4E 54 45 52 00
Data code	/2

WinPCL - Example 6 PVF

Write the value of element T3 of the WinPCL structure "MSTRCT" at device address 00. The string "COUNTER" is output in binary format.

Assumption:
The WinPCL variable "MSTRCT" is declared as a structure in WinPCL program "Prog" as follows:

```
TYP STRUCT
        T1      BOOL
        T2      CHAR
        T3      STRING[16]
        T4      TIME
END
```

FI command		**00_CW_PVF_:Prog.MSTRCT.T3/2**
Line	**Column**	**Answer**
1	1	(P_ACK)

Value to be written:

Value of data element	Binary sequence: 43 4F 55 4E 54 45 52 00
Data code	/2

PLC - Example 7 PVF

Write the value of the PLC structure "MSTRCT" from the structure "mstrct" previously stored in the C program at device address 00.

Assumption:
The "MSTRCT" variable is declared as a structure in the PLC program as follows:

```
TYP STRUCT
        T1      BOOL
        T2      CHAR
        T3      STRING[16]
        T4      TIME
END
```

To exchange binary data in a C program, the following "C" data type can be used:

```
#pragma pack(1)   //Write all elements
                  //without spaces next to each other.
typeder struct
{
      unsigned char T1;
      char         T2;
      char         T3[17]; //Space for zero byte
      unsigned long T4;
} Tymstrct;       // Declare structure
Tymstrct mstrct;  // Apply structure
```

FI command		**00_CW_PVF_MSTRCT/2**
Line	**Column**	**Answer**
1	1	(P_ACK)

Value to be written: address of the C structure.

Value of data element	&mstrct
Data code	/2

WinPCL - Example 7 PVF

Write the value of the WinPCL structure "MSTRCT" from the structure "mstrct" previously stored in the C program at device address 00.

Assumption:
The WinPCL variable "MSTRCT" is declared as a structure in WinPCL program "Prog" as follows:

```
TYP STRUCT
      T1     BOOL
      T2     CHAR
      T3     STRING[16]
      T4     TIME
END
```

To exchange binary data in a C program, the following "C" data type can be used:

```
#pragma pack(1)   //Write all elements
                  //without spaces next to each other.
typeder struct
{
      unsigned char T1;
      char         T2;
      char         T3[17]; //Space for zero byte
      unsigned long T4;
} Tymstrct;       // Declare structure
Tymstrct mstrct;  // Apply structure
```

FI command		**00_CW_PVF_:Prog.MSTRCT/2**
Line	**Column**	**Answer**
1	1	(P_ACK)

Value to be written: address of the C structure.

Value of data element	&mstrct
Data code	/2

10.71 ProVi Messages: PVM

MWAX device group

Designation **PVM** **P**ro**V**i **M**essages

Explanation ProVi messages are output. These messages are assigned to a particular message type or module.

FI command Output all ProVi messages.

For reasons of optimization, not all data is called up. Accordingly, the Diagnosis server must be informed that the data is required (see ADW).

BR_PVM1_(1){_(2)} **(Single Read)**

BC_PVM1_(1){_(2)} **(Cyclic Read)**

(1) = Message type [1 = error, 2 = messages,
10 = warnings,
11 = start requirements,
12 = setup diagnosis]

(2) = Module number [1...99] ! only for message type 1 -2!

Output first ProVi messages.

BR_PVM2_(1){_(2)} **(Single Read)**

BC_PVM2_(1){_(2)} **(Cyclic Read)**

(1) = Message type [1 = error, 2 = messages,
10 = warnings,
11 = start requirements,
12 = setup diagnosis]

(2) = Module number [1...99] ! only for message type 1 -2!

Response Structure The following table shows the general structure of the FI commands "PVM1" and "PVM2". The number of lines depends on the number of messages pending.

If there are no messages, the number of lines is 0.

Line 1...n	Column 1	...	Column 8

Meaning of the Columns

1 = Message text	[ASCII characters]
2 = Message number	[ASCII characters]
3 = Time stamp day	[mm.dd.yyyy]
4 = Time stamp time	[hh:mm:ss]
5 = Message ID	[ASCII characters] (DWORD, decimal)
6 = Reference text exists	[YES, NO]
7 = Criteria analysis exists	[YES, NO]
8 = Message HTML file	[ASCII characters]

Example PVM1

All ProVi errors from module 3 in control unit 0.

There are two messages.

FI command		**00_BR_PVM1_1_3**
Line	**Column**	**Answer**
1	1	Guard not closed
	2	34
	3	01.27.2000
	4	14:56:32
	5	43923028
	6	YES
	7	NO
	8	
2	1	Oil pressure too low
	2	124
	3	01.27.2000
	4	15:03:10
	5	98234039
	6	NO
	7	YES
	8	

Example PVM2

The first ProVi error from module 3 in control unit 0.

There are two messages:

FI command		**00_BR_PVM2_1_3**
Line	**Column**	**Answer**
1	1	Guard not closed
	2	34
	3	01.27.2000
	4	14:56:32
	5	43923028
	6	YES
	7	NO
	8	

FI command

Output the additional information of a ProVi message.

BR_PVM3_(1)_(2){_(3)}	**(Single Read)**
(1) = Message ID	[ASCII characters]
(2) = Message type	[1 = error, 2 = messages, 10 = warnings, 11 = start requirements, 12 = setup diagnosis]
(3) = Module number	[1...99] ! only for message type 1 -2!

Response Structure

The following table shows the general structure of the "PVM3" FI command.

Line 1	**Column 1**	**...**	**Column 16**

Meaning of the Columns

1 =	Message text	[ASCII characters]
2 =	Message number	[ASCII characters]
3 =	Error category	[ASCII characters] (empty no category)
4 =	Time stamp day	[mm.dd.yyyy]
5 =	Time stamp hour	[hh:mm:ss]
6 =	Additional text available	[YES, NO]
7 =	Additional text	[ASCII characters]
8 =	Message ID	[ASCII characters] (DWORD, decimal)
9 =	Diagnosis source	[ASCII characters] (PLC, CNC)
10 =	POU name	[ASCII characters]
11 =	Detail name	[ASCII characters] (empty implementation)
12 =	Detail type	[1 = action block, 3 = transition, 4 = implementation]
13 =	Network number	[ASCII characters]
14 =	Variable name	[ASCII characters]
15 =	POU entity name	[ASCII characters]
16 =	POU type	[2 = program, 3 = function block]
17 =	Analysis of criteria available	[YES, NO]
18 =	Message HTML file	[ASCII characters]
19 =	Reference info HTML file	[ASCII characters]

Example PVM3 Additional text of a ProVi error with ID 43923028 from module 3 in control unit 0.

FI command		**00_BR_PVM3_43923028_1_3**
Line	**Column**	**Answer**
1	1	Guard not closed
	2	34
	3	1
	4	01.27.2000
	5	14:56:32
	6	YES
	7	Oil pressure too low Oil pipe leaking or insufficient oil.
	8	43923028
	9	PLC
	10	MODULE3
	11	
	12	4
	13	34
	14	EschutzT
	15	Station2.Module3

FI command		**00_BR_PVM3_43923028_1_3**
Line	**Column**	**Answer**
	16	3
	17	NO
	18	
	19	D:\Program Files\Indramat\MtGui\Project_000\ ProgramData\HMTL\DE\Error34.html

FI command One after the other, all active ProVi messages are output. In the result, one line each is returned. After expiry of the set time, the next message is returned. The clock frequency can be set via the last parameter. This value can only be set once for the PC. The value transmitted last is always the valid value. Default setting is 1 second.

BR_PVM4_(1){_(2)_(3)}	**(Single Read)**
BC_PVM4_(1){_(2)_(3)}	**(Cyclic Read)**
(1) = Message type	[1 = error, 2 = messages, 10 = warnings, 11 = start requirements, 12 = setup diagnosis]
(2) = Module number	[1...99] ! only for message type 1 -2!
(3) = Clock frequency	[ASCII characters] Time in ms

Response Structure The following table shows the general structure of the "PVM4" FI command.

If there are no messages, the number of lines is 0.

Line 1	Column 1	...	Column 8

Meaning of the Columns

1 = Message text	[ASCII characters]
2 = Message number	[ASCII characters]
3 = Time stamp day	[mm.dd.yyyy]
4 = Time stamp time	[hh:mm:ss]
5 = Message ID	[ASCII characters] (DWORD, decimal)
6 = Additional text available	[YES, NO]
7 = Criteria analysis exists	[YES, NO]
8 = Message index (1 = 1. message)	[ASCII characters]
9 = Message HTML file	[ASCII characters]

Example PVM1

ProVi errors from module 3 in control unit 0.

The 2nd message is being output. The clock frequency is to be 2 seconds.

FI command		**00_BR_PVM4_1_3_2000**
Line	**Column**	**Answer**
1	1	Oil pressure too low
	2	124
	3	01.27.2000
	4	15:03:10
	5	98234039
	6	NO
	7	YES
	8	2
	9	

10.72 Download of PLC Retain Variables: PVR

MWAX device group

Designation **PVR** **P**LC **V**ariable **R**etain Backup

Explanation Download of PLC retain variables.

FI command **BW_PVR1!(1)** **(Single Write)**

(1) = Download file with path details.

Note: File and path details must be enclosed in inverted commas.
The separator "!" is used in this command.

Response Structure

The response to the "PVR1" FI command consists of three lines, each with one column. The meaning of the elements is as follows:

- Line 1 = Job ID [01...20]
  (see Chapter "FI Commands for the MPCX Device Group: IFJ").
- Line 2 = FI command
  [String, in accordance with Chapter "Elements of the FI Command"]
- Line 3 = FI Job Error Code (see Chapter "Error Codes")

Example PVR1

00_BW_PVR1!"D:\Program Files\Indramat\Mtgui\Temp\download.ini"/3

FI command		**00_BW_PVR1!"D:\Program Files\Indramat\Mtgui\Temp\download.ini" /3**
Line	**Column**	**Answer**
1	1	01
2	1	00_BW_PVR1!"D:\Program Files\Indramat\Mtgui\Temp\download.ini" /3
3	1	0

Structure of Download File

The structure of the "download.ini" file used in this example corresponds to an Ini file in Windows.

Note: Care must be taken in the use of upper and lower case letters.

10.73 Upload of PLC Retain Variables: PVR

MWAX device group

Designation **PVR** **P**LC **V**ariable **R**etain Backup

Explanation PLC retain variables are uploaded via all active processes.

FI command **BR_PVR1!(1)** **(Single Read)**

(1) = Upload file with path details

Note: Enclose file and path details in inverted commas.
The separator "!" is used in this command.

Response Structure The response to the "PVR1" FI command consists of three lines, each with one column. The meaning of the elements is as follows:

- Line 1 = Job ID [01...20]
 (see Chapter "FI Commands for the MPCX Device Group", IFJ).
- Line 2 = FI command
 [String, in accordance with Chapter "Elements of the FI Command"]
- Line 3 = FI Job Error Code (see Chapter "Error Codes")

Example PVR 00_BR_PVR1_"D:\Program Files\Indramat\Mtgui\Temp\Upload.ini"/3

FI command		**00_BR_PVR1!"D:\Program Files\Indramat\Mtgui\ Temp\upload.ini" /3**
Line	**Column**	**Answer**
1	1	01
2	1	00_BR_PVR1!"D:\Program Files\Indramat\Mtgui\ Temp\upload.ini" /3
3	1	0

Structure of Upload File The upload file is structured in the Windows – "Ini" format structure.

Note: Care must be taken in the use of upper and lower case letters.

10.74 Reading the PLC Variable Declaration: PVT

MWAX device group

Designation **PVT** **P**LC **V**ariable **T**ype

Explanation A PLC variable has a particular type. To evaluate complex variables such as structures and arrays, their components and types must be read out.

Refer also to PVF, Reading Structured PLC Variables.

FI command Read the PLC variable type.

BR_PVT_(1) **(Single Read)**

(1) = Identifier of the PLC variable [acc. to declaration part of the PLC]

Response Structure One line with 2 columns is output for each element of the variables.

Line 1...n:	Column 1	Column 2

n = number of elements.

Value Range/Meaning of Columns

(1) = Identifier of the PLC variable [acc. to declaration part of the PLC]

2 = Type [see value range PVF]

Examples:
PLC: Reading of a variable

Assumption:
The "TEST" variable is declared as WORD in the PLC program.

FI command	**00_BR_PVT_TEST**	
	Answer	
Line	**Column 1 (Name)**	**Name**
1	TEST	WORD

WinPCL: Reading a Variable

Assumption:
The WinPCL variable "TEST" is declared as WORD in WinPCL program "Prog".

FI command	**00_BR_PVT_:Prog.TEST**	
	Answer	
Line	**Column 1 (Name)**	**Name**
1	TEST	WORD

PLC: Reading a Structure

Assumption:
The "TEST1" variable is declared as STRUCT in the PLC program.

```
STRUCT
        E1      BOOL
        E2      INT
        E3      SINT
END
```

FI command	**00_BR_PVT_TEST1**	
	Answer	
Line	**Column 1**	**Column 2**
1	TEST1.E1	BOOL
2	TEST1.E2	INT
3	TEST1.E3	SINT

WinPCL: Reading a Structure

Assumption:
The WinPCL variable "TEST1" is declared as STRUCT in WinPCL program "Prog".

```
STRUCT
        E1      BOOL
        E2      INT
        E3      SINT
END
```

FI command	**00_BR_PVT_:Prog.TEST1**	
Answer		
Line	**Column 1**	**Column 2**
1	TEST1.E1	BOOL
2	TEST1.E2	INT
3	TEST1.E3	SINT

PLC: Reading an Array

Assumption:
The "TEST2" variable is declared as ARRAY in the PLC program.

```
ARRAY [
        0 .. 3
] OF    BOOL
```

FI command	**00_BR_PVT_TEST2**	
Answer		
Line	**Column 1**	**Column 2**
1	TEST2[0]	BOOL
2	TEST2[1]	BOOL
3	TEST2[2]	BOOL
4	TEST2[3]	BOOL

WinPCL: Reading an Array

Assumption:
The WinPCL variable "TEST2" is declared as ARRAY in WinPCL program "Prog".

```
ARRAY [
        0 .. 3
] OF    BOOL
```

FI command	**00_BR_PVT_:Prog.TEST2**	
Answer		
Line	**Column 1**	**Column 2**
1	TEST2[0]	BOOL
2	TEST2[1]	BOOL
3	TEST2[2]	BOOL
4	TEST2[3]	BOOL

PLC: Reading an Array of a Structure

Assumption:
The "TEST3" variable is declared as ARRAY in the PLC program.

```
ARRAY [
        0 .. 1
] OF    STRUCT1,
```

where STRUCT1 is declared as follows:

```
STRUCT
        E1      BOOL
        E2      INT
        E3      SINT
END
```

FI command	**00_BR_PVT_TEST3**	
Answer		
Line	**Column 1**	**Column 2**
1	TEST3[0].E1	BOOL
2	TEST3[0].E2	INT
3	TEST3[0].E3	SINT
1	TEST3[1].E1	BOOL
2	TEST3[1].E2	INT
3	TEST3[1].E3	SINT

WinPCL: Reading an Array of a Structure

Assumption:
The WinPCL variable "TEST3" is declared as ARRAY in WinPCL program "Prog".

```
ARRAY [
        0 .. 1
] OF    STRUCT1,
```

where STRUCT1 is declared as follows:

```
STRUCT
        E1      BOOL
        E2      INT
        E3      SINT
END
```

FI command	**00_BR_PVT_:Prog.TEST3**	
Answer		
Line	**Column 1**	**Column 2**
1	TEST3[0].E1	BOOL
2	TEST3[0].E2	INT
3	TEST3[0].E3	SINT
1	TEST3[1].E1	BOOL
2	TEST3[1].E2	INT
3	TEST3[1].E3	SINT

Assumption:
The data types are output according to IEC1131.

See also command PVF.

10.75 Logging in and off of the SERCOS Channel: SCO

MWAX device group

Designation **SCO** **S**ercos **CO**nnection

Explanation This command is used to open and close a SERCOS channel so that the drive parameters (e.g. S-0-0101) can be read and written. Simultaneous access of the NC and another application (e.g. integrated DriveTop) is not possible. If for example an NC program is running, the channel cannot be opened; and if the channel is opened, no NC program can be started.

FI command Logging in and off of the SERCOS Channel.

BW_SCO1	**(Single Write)**
Write value = 1	Log in
Write value = 0	Log off

Response Structure The following table shows the general structure of the response to the FI command "SCO1". A line of 18 columns is output.

Line 1	Column 1	...	Column 18

After log-in, the return value of the routine indicates whether log-in has been successful. If log-in has been successful, the current Sercos phase and the physically available axes are returned.

Element	Meaning
1	Log-in successful = 1 / unsuccessful = 0
2	Sercos phase (if log-in = 0, the value is --)
3	Axis no. 1 physically available = 1, otherwise = --
4	Axis no. 2 physically available = 1, otherwise = --
5	Axis no. 3 physically available = 1, otherwise = --
6	Axis no. 4 physically available = 1, otherwise = --
7	Axis no. 5 physically available = 1, otherwise = --
8	Axis no. 6 physically available = 1, otherwise = --
9	Axis no. 7 physically available = 1, otherwise = --
10	Axis no. 8 physically available = 1, otherwise = --
11	Axis no. 9 physically available = 1, otherwise = --
12	Axis no. 10 physically available = 1, otherwise = --
13	Axis no. 11 physically available = 1, otherwise = --
14	Axis no. 12 physically available = 1, otherwise = --
15	Axis no. 13 physically available = 1, otherwise = --
16	Axis no. 14 physically available = 1, otherwise = --
17	Axis no. 15 physically available = 1, otherwise = --
18	Axis no. 16 physically available = 1, otherwise = --

When logging off, the return value is always output as shown in the example below.

1. Example SCO1 Log-in of the SERCOS channel

FI command	**Value to be written: 1** **00_BW_SCO1**
Return value	1 4 -- 1 1 1 – 1 -- -- -- -- -- -- -- -- -- --

2. Example SCO1 Log-off of the SERCOS channel

FI command	Value to be written: 0 00_BW_SCO1
Return value	0 -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --

FI command The status of the SERCOS channel is read out.

BR_SCO1 (Single Read)

Response Structure The following table shows the general structure of the response to the FI command "SCO1". A line of 18 columns is output.

Line 1	Column 1	...	Column 18

Element 1 of the answer indicates whether the Sercos channel is free, or is being used by another application.

Element	Meaning
1	Logged in = 1 / Not logged in = 0
2	Sercos Phase
3	Axis no. 1 physically available = 1, otherwise = --
4	Axis no. 2 physically available = 1, otherwise = --
5	Axis no. 3 physically available = 1, otherwise = --
6	Axis no. 4 physically available = 1, otherwise = --
7	Axis no. 5 physically available = 1, otherwise = --
8	Axis no. 6 physically available = 1, otherwise = --
9	Axis no. 7 physically available = 1, otherwise = --
10	Axis no. 8 physically available = 1, otherwise = --
11	Axis no. 9 physically available = 1, otherwise = --
12	Axis no. 10 physically available = 1, otherwise = --
13	Axis no. 11 physically available = 1, otherwise = --
14	Axis no. 12 physically available = 1, otherwise = --
15	Axis no. 13 physically available = 1, otherwise = --
16	Axis no. 14 physically available = 1, otherwise = --
17	Axis no. 15 physically available = 1, otherwise = --
18	Axis no. 16 physically available = 1, otherwise = --

1. Example SCO1 Status of the SERCOS channel after successful log-in:

FI command	00_BR_SCO1
Return value	1 4 -- 1 1 1 -- 1 -- -- -- -- -- -- -- -- -- --

2. Example SCO1 Status of the SERCOS channel after unsuccessful log-in:

FI command	00_BR_SCO1
Return value	0 -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --

10.76 SFC Diagnosis Data: SDD

MWAX device group

Designation **SDD** S**F**C **D**iagnosis **D**ata

Explanation Data for step chain diagnosis is output. Depending on the FI command this data can concern disrupted steps, actions, transitions or a definite ID to display the action or transition.

FI command Output the disrupted step of a step chain.

BR_SDD1!(1)!(2)	**(Single Read)**
(1) = Module number	[1...99]
(2) = SFC entity name	[ASCII characters]

Note: The separator "!" is used in this command.

Response Structure The following table shows the general structure of the FI command "SDD1".

Line 1	Column 1	...	Column 7

Meaning of the Columns

1 = Step name	[ASCII characters]
2 = Detail type	[1 = action block, 2 =action network, 3 = transition]
3 = Detail name	[ASCII characters]
4 = POU ID	[ASCII characters]
5 = Detail morpheme	[ASCII characters] (DWORD, decimal)
6 = Error ID	[ASCII characters] (DWORD, decimal)
7 = POE entity name	[ASCII characters]

Example SDD1 Query disrupted step of the "clamp" chain in module 3 in control unit 0.

FI command		**00_BR_SDD1!3!Station03A.Clamp**
Line	**Column**	**Answer**
1	1	Open
	2	1
	3	Open
	4	SFC_1_2
	5	98243823
	6	34985304
	7	Station2.Module3

FI command Output the faulty action, monitor error or transition of a disrupted step.

BR_SDD2!(1)!(2)!(3)	**(Single Read)**
(1) = Module number	[1...99]
(2) = SFC entity name	[ASCII characters]
(3) = Step name	[ASCII characters]

Note: The separator "!" is used in this command.

Response Structure

The following table shows the general structure of the FI command "SDD2".

Line 1	Column 1	...	Column 6

Meaning of the Columns

1 = Detail type	[1 = action block, 2 =action network, 3 = transition]
2 = Detail name	[ASCII characters]
3 = POU ID	[ASCII characters]
4 = Detail morpheme	[ASCII characters] (DWORD, decimal)
5 = Error ID	[ASCII characters] (DWORD, decimal)
6 = POE entity name	[ASCII characters]

Example SDD2

Query faulty action of the disrupted step "open" of the "clamp" chain in module 3 in control unit 0.

FI command		**00_BR_SDD1!3!Station03A.Clamp_Open**
Line	**Column**	**Answer**
1	1	1
	2	Open
	3	SFC_1_2
	4	98243823
	5	34985304
	6	Station2.Module3

FI command

Output the definite ID to display the action, monitor error or transition.

BR_SDD3!(1)!(2)!(3)!(4)	**(Single Read)**
(1) = Module number	[1...99]
(2) = SFC entity name	[ASCII characters]
(3) = Detail type	[1 = action block, 2 = action network, 3 = transition]
(4) = Detail name	[ASCII characters]

Note: The separator "!" is used in this command.

Response Structure

The following table shows the general structure of the FI command "SDD3".

Line 1	Column 1	...	Column 4

Meaning of the Columns

1 = POU ID	[ASCII characters]
2 = Detail morpheme	[ASCII characters] (DWORD, decimal)
3 = Error ID	[ASCII characters] (DWORD, decimal)
4 = POE entity name	[ASCII characters]

Example SDD3

Query ID to display the action "aOpen" of the "clamp" chain in module 3 in control unit 0.

FI command		**00_BR_SDD3!3!Station03A.Clamp!1!aOpen**
Line	**Column**	**Answer**
1	1	SFC_1_2
	2	98243823
	3	34985304
	4	Station2.Module3

FI command

Output the I/O addresses to display a detail.

BR_SDD4!(1)!(2)!(3)!(4)	**(Single Read)**
(1) = Module number	[1...99]
(2) = SFC entity name	[ASCII characters]
(3) = Detail type	[1 = action block, 2 = action network, 3 = transition]
(4) = Detail name	[ASCII characters]

Note: The separator "!" is used in this command.

Response Structure

The following table shows the general structure of the FI command "SDD4".

Line 1-n	**Column 1**	**Column 2**

Meaning of the Columns

1 = Variable morpheme	[ASCII characters] (DWORD, decimal)
2 = I/O address	[ASCII characters]

Example SDD4

Query I/O addresses to display the action "aOpen" of the "clamp" chain in module 3 in control unit 0.

Three variables have an I/O address.

FI command		**00_BR_SDD4!3!Station03A.Clamp!1!aOpen**
Line	**Column**	**Answer**
1	1	98243823
	2	%I3.2.0
2	1	40923423
	2	%Q23.21.7
3	1	34985304
	2	%I100.3.5

FI command

Determine the multilingual comments for displaying a detail.

BR_SDD5!(1)!(2)!(3)!(4)	**(Single Read)**
(1) = Module number	[1...99]
(2) = SFC entity name	[ASCII characters]
(3) = Detail type	[1 = action block, 2 = action network, 3 = transition]
(4) = Detail name	[ASCII characters]

Note: The separator "!" is used in this command.

Response Structure

The following table shows the general structure of the FI command "SDD5".

Line 1-n	Column 1	Column 2

Meaning of the Columns

1 = Comment morpheme	[ASCII characters] (DWORD, decimal)
2 = New comment	[ASCII characters]

Example SDD5

Query comments to display the action "aOpen" of the "clamp" chain in module 3 in control unit 0.

Two comments are replaced by another text.

FI command		**00_BR_SDD5!3!Station03A.Clamp!1!aOpen**
Line	**Column**	**Answer**
1	1	98243823
	2	Clamp open
2	1	40923423
	2	Clamp closed

FI command

Output the action that has not been performed, or the transition of a step calculated based on the online status.

BR_SDD6!(1)!(2)!(3)	**(Single Read)**
(1) = Module number	[1...99]
(2) = SFC entity name	[ASCII characters]
(3) = Step name	[ASCII characters]

Note: The separator "!" is used in this command.

Response Structure

The following table shows the general structure of the FI command "SDD6".

Line 1	Column 1	...	Column 6

Meaning of the Columns

1 = Detail type	[1 = action block, 3 = transition]
2 = Detail name	[ASCII characters]
3 = POU ID	[ASCII characters]
4 = Detail morpheme	[ASCII characters] (DWORD, decimal)
5 = Error ID	[ASCII characters] (DWORD, decimal)
6 = POU entity name	[ASCII characters]

Example SDD6

Query the action that has not been performed for the step "open" of the "clamp" chain in module 3 in control unit 0.

FI command		**00_BR_SDD6!3!Station03A.Clamp_Open**
Line	**Column**	**Answer**
1	1	1
	2	Open
	3	SFC_1_2
	4	98243823
	5	34985304
	6	Station2.Module3

FI command

Determine the module number of a step chain.

BR_SDD7!(1) **(Single Read)**

(2) = SFC instances name [ASCII characters]

Note: The separator "!" is used in this command.

Response Structure

The following table shows the general structure of the FI command "SDD7".

Line 1	Column 1

Meaning of the Columns

1 = Module number [1...99]

Example SDD7

Inquiry of the module of the chain "clamp".

FI command		**00_BR_EXD1!Station03A.Clamp**
Line	**Column**	**Answer**
1	1	3

10.77 Setting of Device Status Information: SDS

MWAX device group

Designation **SDS** Set **D**evice **S**tatus

Explanation

By this command, the device status information can be set; here, the configuration file IND_DEV.INI is adjusted as well.

Note: When this command is transmitted, the following system messages are generated:
MSG_DEVICEOFF or MSG_DEVICE_ON !

FI command

With this command, the device status information of **ALL** defined devices can be set.

BW_SDS1_(1) **(Single Write)**

(1) = Device status information to be set
0 = Device status information OFF
1 = Device status information ON

Response Structure

The following table shows the general structure of the response to the "SDS1" FI command.

Line 1	Column 1

Value Range/Meaning of Columns

1 = Status report [(P_ACK)]

Example SDS1

Set device status information to OFF for **ALL** defined devices.

FI command		**00_BW_SDS1_0**
Line	**Column**	**Answer**
1	1	(P_ACK)

FI command

With this command, the device status information for a selected device can be set.

BW_SDS2_(1) **(Single Write)**

(1) = Device status information to be set
0 = Device status information OFF
1 = Device status information ON

Response Structure

The following table shows the general structure of the response to the "SDS2" FI command.

Line 1	Column 1

Value Range/Meaning of Columns

1 = Status report [(P_ACK)]

Example: SDS2

Set device status information to OFF for the selected device 00.

FI command		00_BW_SDS2_0
Line	**Column**	**Answer**
1	1	(P_ACK)

10.78 Setting the FI Exclusive Mode: SEM

MWAX device group

Designation

SEM **S**et FI **E**xclusive **M**ode

Explanation

This command is used to activate FI Exclusive mode for the selected device address.

FI Exclusive mode: In this mode, **ALL** the processes logged in at the FI – **with the exception** of the process issuing the SEM command – are blocked from data communication with this device address. This mode is used for example for data communication which must NOT be interrupted by other data requests. However, it is **imperative** that this FI Exclusive mode is deleted once more through the DEM command.

FI command

BW_SEM1 **(Single Write)**

Response Structure

The following table shows the general structure of the response to the FI command "BW_SEM1". A line of 1 column is output.

Line 1	Column 1

Value Range/Meaning of Columns

1 = Status message (P_ACK) (P_ACK)

Example SEM1

Activate FI Exclusive mode for device address 0.

FI command		00_BW_SEM1
Line	**Column**	**Answer**
1	1	(P_ACK)

10.79 Sequencer Data: SFD

MWAX device group

Designation **SFD** **SF**C **D**ata

Explanation Data for a step chain is outputted. Depending on the FI command this can concern a sequencer comment, POU name, step comment, maximum time, action / transition / monitor error name (comment), qualifier and time value.

FI command Query the data for a step chain.

BR_SFD1!(1)!(2)	**(Single Read)**
(1) = Module number	[1...99]
(2) = SFC entity name	[ASCII characters]

Note: The separator "!" is used in this command.

Response Structure The following table shows the general structure of the "SFD1" FI command.

Line 1	Column 1	Column 2

Meaning of the Columns

1 = Step chain comment	[ASCII characters]
2 = POU name	[ASCII characters]

Example SFD1 Query data of the "clamp" chain in module 3 in control unit 0.

FI command		**00_BR_SFD1!3!Station03A.Clamp**
Line	**Column**	**Answer**
1	1	Clamping device
	2	CLAMP

FI command Query the data of a step.

BR_SFD2!(1)!(2)!(3)	**(Single Read)**
(1) = Module number	[1...99]
(2) = SFC entity name	[ASCII characters]
(3) = Step name	[ASCII characters]

Note: The separator "!" is used in this command.

Response Structure The following table shows the general structure of the "SFD2" FI command. The number of lines depends on the number of actions and transitions.

If there are no details the line number is 1.

Line 1	Column 1	...	Column 3
Line 2...n:	Column 1	...	Column 6

Meaning of the Columns

Line 1

1 = Step comment	[ASCII characters]
2 = Maximum time	[ASCII characters]
3 = Minimum time	[ASCII characters]

Line 2...n:

1 = Detail type	[1 = action block, 3 = transition]
2 = Name	[ASCII characters]
3 = Comment	[ASCII characters]
4 = Boolean variable	[YES, NO]
5 = Qualifier	[ASCII characters]
6 = Time value	[ASCII characters]

Example SFD2

Data for the step "Open" in the "clamp" chain in module 3 on control unit 0.

FI command		**00_BR_SFD2!3!Station03A.Clamp!Open**
Line	**Column**	**Answer**
1	1	Open clamping device
	2	T#5s
	3	
2	1	1
	2	aOpen
	3	Clamp open
	4	NO
	5	D
	6	T#3s
3	1	3
	2	tOpen
	3	Clamping device is open
	4	NO
	5	
	6	

FI command

Output the data for a detail.

BR_SFD3!(1)!(2)!(3)!(4)	**(Single Read)**
(1) = Module number	[1...99]
(2) = SFC entity name	[ASCII characters]
(3) = Detail type	[1 = action block, 2 = action network, 3 = transition]
(4) = Detail name	[ASCII characters]

Note: The separator "!" is used in this command.

Response Structure

The following table shows the general structure of the "SFD3" FI command.

Line 1	Column 1	Column 2

Meaning of the Columns

1 = Comment	[ASCII characters]
2 = Boolean variable	[YES, NO]

Example SFD3

Data for the action "aOpen" in the "clamp" chain in module 3 on control unit 0.

FI command		**00_BR_SFD3!3!Station03A.Clamp!aOpen**
Line	**Column**	**Answer**
1	1	Clamp open
	2	NO

10.80 Sequencer Messages: SFE

MWAX device group

Designation **SFE** **SF**C **E**rror

Explanation The sequencer messages of a module are output.

FI command Output all SFC messages.

For reasons of optimization, not all data is called up. Accordingly, the Diagnosis server must be informed that the data is required (see ADW).

BR_SFE1_(1)	**(Single Read)**
BC_SFE1_(1)	**(Cyclic Read)**
(1) = Module number	[1...99]

Output first SFC messages.

BR_SFE2_(1)	**(Single Read)**
BC_SFE2_(1)	**(Cyclic Read)**
(1) = Module number	[1...99]

Response Structure The following table shows the general structure of the FI commands "SFE1" and "SFE2". The number of lines depends on the number of messages pending.

If there are no messages, the number of lines is 0.

Line 1...n:	**Column 1**	**...**	**Column 7**

Meaning of the Columns

1 = Message text	[ASCII characters]
2 = SFC entity name	[ASCII characters]
3 = Step name	[ASCII characters]
4 = Time stamp day	[mm.dd.yyyy]
5 = Time stamp time	[hh:mm:ss]
6 = Type of error	[1 = time error, 2 = monitor error, 3 = monitor event]
7 = Is there condition analysis?	[YES, NO]

Example SFE1

All SFC messages from module 2 in control unit 0.

There are two messages.

FI command		00_BR_SFE1_2
Line	**Column**	**Answer**
1	1	TIME ERROR: Chain: chucking Step: up malfunction
	2	Station03A.Clamp
	3	Open
	4	01.27.2000
	5	11:56:32 AM
	6	1
	7	YES
2	1	ASSY ERROR: Chain: drilling Step: down malfunction
	2	Station02A.Drill
	3	Down
	4	01.27.200
	5	13:03:12
	6	2
	7	NO

Example SFE2

First SFC message from module 2 in control unit 0.

There are two messages.

FI command		00_BR_SFE2_2
Line	**Column**	**Answer**
1	1	TIME ERROR: Chain: chucking Step: up malfunction
	2	Station03A.Clamp
	3	Open
	4	01.27.2000
	5	14:56:32
	6	1
	7	YES

10.81 Sequencer Mode: SFM

MWAX device group

Designation **SFM** **SFC** **M**ode

Explanation Queries step chain mode.

FI command Query the mode of a step chain.

BR_SFM1!(1)!(2)	**(Single Read)**
BC_SFM1!(1)!(2)	**(Cyclic Read)**
(1) = Module number	[1...99]
(2) = SFC entity name	[ASCII characters]

Note: The separator "!" is used in this command.

Response Structure

The following table shows the general structure of the "SFM1" FI command.

Line 1	Column 1

Meaning of the Columns

1 = Mode [1 = time error, 2 = monitor error, 3 = monitor event, 10 = stop, 11 = auto, 12 = manual, 13 = jog]

Example SFM1

Query mode of the "clamp" chain in module 3 in control unit 0.

FI command		00_BR_SFM1!3!Station03A.Clamp
Line	**Column**	**Answer**
1	1	1

10.82 Software Installation Data: SID

MWAX device group

Designation

SID **S**oftware **I**nstallation **D**ata

Explanation

Information is returned regarding installation. This information includes installation paths, the software version used, context information, DLL mode, plus service pack and release information.

FI command

Read-in the installation data and/or the software version data

BR_SID1 **(Single Read)**

BC_SID1 **(Cyclic Read)**

Response Structure

One line with 8 columns is output for the returned values.

Line 1	Column 1	...	Column 8

Meaning of the Columns

1 = Basic directory	[EXE files of the BOF]
2 = FI installation directory	[FI directory]
3 = Data directory	[in accordance with BOF]
4 = GBO version	[from INDRAMAT.ini]
5 = IF-DLL mode	[from INDRAMAT.ini]
6 = IF version	[from INDRAMAT.ini from DLL mode 400]
7 = Service package info	[from INDRAMAT.ini from DLL mode 420]
8 = Release info	[from INDRAMAT.ini from DLL mode 420]
9 = IF-Build-Info	[in accordance with Build process]
10 = Current context name	[in accordance with the installation]
11 = Physical installation path	[in accordance with the installation]
12 = Complete IF version indication string	
13 = WinPCL build number	[in accordance with WinPCL]
14 = Version number of the PLC compiler	[in accordance with WinPCL]
15 = Version number of the PLC linker	[in accordance with WinPCL]
16 = Version number of the PLC data basis	[in accordance with WinPCL]
17 = Platform version	

Example SID1 Return information on the current installation.

FI command		**00_BR_SID1**
Line	**Column**	**Answer**
1	1	--
	2	D:\Program Files\Indramat\MTGUI\Bin
	3	--
	4	005-22Vxx
	5	07.00
	6	07V00
	7	--
	8	--
	9	Build 3124 Mar 6 2003 08:53:55
	10	MTGUI_0-23T01 B3327
	11	D:\Program Files\Indramat\MTGUI\
	12	FI: 07V00 DLL-Mode: 07.20 Build 3124 Mar 6 2003 08:53:55
	13	347.15.4.11
	14	771
	15	515
	16	78
	17	Platform: 02V01 Build: 3214

Note: Refer to FI command "PHD" of the MPCX for working with absolute paths.

10.83 PLC Long Identification: SLI

MWAX device group

Designation **SLI** PLC (**S**PS) **L**ong **I**dentification

Explanation Returns the unit data from the PLC long identification.

FI command Read PLC long identification.

BR_SLI **(Single Read)**

Response Structure One line with 15 columns is output for the returned values.

Line 1	Column 1	Column...	Column 15

Value Range/Meaning of the Columns

1 =	Device address	[00...15]
2 =	Program number	[01...99]
3 =	Project name	[max. 8 ASCII characters]
4 =	Program name	[max. 8 ASCII characters]
5 =	User name	[acc. to password entry]
6 =	Program length	[bytes]
7 =	Compilation time	[LONG] (coded in long value)
8 =	Compilation date	[8 ASCII characters]
9 =	Compilation time	[8 ASCII characters]
10 =	Download time	[LONG] (coded in long value)
11 =	Download date	[8 ASCII characters]1
12 =	Download time	[8 ASCII characters]
13 =	Version of PLC long identification	[LONG]
14 =	RUN flags	[HEX value]
15 =	Compiler info	[LONG]

Example SLI Read the unit data from the PLC long identification.

FI command		**00_BR_SLI**
Line	**Column**	**Answer**
1	1	02
	2	01
	3	--
	4	MOT12
	5	TEST
	6	17672
	7	630163960
	8	15.12.99
	9	17:15:48
	10	630163961
	11	15.12.99
	12	17:15:50

Rexroth
Bosch Group

FI command		**00_BR_SLI**
Line	**Column**	**Answer**
	13	2
	14	0x0000
	15	13

Reference to Literature see chapter entitled "Literature" [30].

10.84 SERCOS Parameters: SPA

MWAX device group

Designation **SPA** **S**ERCOS **PA**rameter

Explanation A SERCOS drive parameter is read out or written. Each parameter consists of 7 elements, whereby any combination of elements can be selected by element coding.

FI command

BR_SPA1_(1)_(2)_(3) **(Single Read)**

BC_SPA1_(1)_(2)_(3) **(Cyclic Read)**

BB_SPA1_(1)_(2)_(3) **(Break Cyclic Read)**

BW_SPA1_(1)_(2)_(3) **(Single Write)**

(1) = Drive address [0...99]

(2) = Parameter No. in the format: X-Y-ZZZZ

(3) = Element coding [standard or advanced format]

Parameter No.

Format X-Y-ZZZZ	**Value range**
X	S = standard data P = product data
Y	[0..0.7] = parameter record
Z	[0...4095] = datablock no.

Element Coding Element coding in standard format allows individual elements, such as the operating date, to be requested. If several elements are to be read out in one request, then the element coding can be OR'd in advanced format, e.g. operating date (0x40) and unit (0x08) produces OR'd (0x48) → 48

The advanced format 0x80 has priority over 0x40.

Element	**Standard Format**	**Advanced Format**	**Format:**	**Example**
Data status	S	01H	Hexadecimal word	0x0000
Name	The marked section is then printed out.	02H	String	NC cycle time (TNcyc)
Attribute	A	04H	Hexadecimal double word	0x60110001
Unit	U	08H	String	µs
Min. input value	L	10H	Decimal word	2000
Max. input value	H	20H	Decimal word	20000
Operating date	D	40H	(see Displaying the Operating Date	
Operating date, when no list		80H		

Displaying the Operating Date The display of the operating date depends on the parameter number requested.

Decimal

Decimal values are given as floating points, e.g. 1.5. Leading spaces, zeros, plus and minus signs as well as trailing spaces are allowed.

Hexadecimal

Hexadecimal values are displayed by "0x...", e.g. 0x80. Up to a maximum of eight positions are allowed. Leading or trailing spaces are allowed. Leading additional zeros or plus and minus signs are not allowed.

Binary (max. 32 characters)

Leading or trailing spaces are allowed. The decimal point serves as separator:

e.g., 1111.0000.1010.1100.1111.0000.1010.1100

Note: Leading additional zeros or plus and minus signs are not allowed.

Lists of Variable Length

Lists always begin with two decimal numbers for the actual length and maximum length of the list. The length specification refers to the length of the list in the drive and therefore designates the number of bytes for storage (storage bytes). The number of elements in the list can be calculated using the attribute. The list elements are displayed according to the attribute. All parts of the list are separated from each other by a line feed ("\n").

Example:

Parameter S-0-0017, IDN list of all parameters

"400\n400\nS-0-0001\nS-0-0002\n..."

ASCII List

ASCII lists are a special form of variable length lists. The individual string characters are not separated by a line feed. When displaying the lists, a distinction is made between standard format and advanced format. In standard format, only the character string is returned, whereas in advanced format the actual length and the maximum length of the list (string) is also transmitted.

Example:

Parameter S-0-0030, operating date
Standard format: "DKC2.1-SSE-01V09"
Advanced format: "16\n16\nDKC2.1-SSE-01V09"

Note: When requesting SERCANS parameters the drive address can be anywhere within the range [0..254].

Response Structure

The following table shows the general structure of the response to the FI command "SPA1". Line 1 is output both when reading and when writing. Additional lines are only output when reading depending on the element coding.

Notes: If the element coding has been requested in standard format then the first line is not applicable.

Line 1 is a status line that either contains SERCOS / SERCANS errors or displays the successful processing of the FI command. If the command has been processed successfully, then columns 1 and 3 contain the value [0x0000].

In the first line, column 2 or column 4, the number of the drive is output that reports the SERCOS error or the global SERCANS error. Not all global SERCANS errors can be directly assigned to a drive. In this case, the single result "Drive No." is set to 0x0000.

Line	Column 1	Column 2	Column 3	Column 4
1	<SERCOS error>	<Drive No. SERCOS error>	<Global SERCANS error>	<Drive No. Global SERCANS error>
2	Read: 1. Element corresponding to the element coding.			
...	...			
n	Read: (n-1). Element corresponding to the element coding.			

Example SPA1 / read Read parameter S-0-0003 of the 3rd drive (element coding 0x48)

FI command	**00_BR_SPA1_3_S-0-0003_48**			
Answer				
Line	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	0x0000	0x0000	0x0000	0x0000
2	µs			
3	2000			

Example SPA1 / write Write the ID number P-0-0037 in parameter S-0-0305 of the 3rd drive (element coding 0x40).

Technical background:

Realtime status bit 1 is to be assigned the trigger status word of the oscilloscope function of a DIAX04 drive.

FI command	**Value to be written: P-0-0037** **00_BW_SPA1_3_S-0-0305_40**			
Answer				
Line	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	0x0000	0x0003	0x0000	0x0000

Reference to Literature See chapter entitled "Literature" [41].

See chapter entitled "Literature" [46].

Explanation A Sercos parameter of a drive is read out of the transferred Sercos data file. Each Sercos parameter consists of 7 elements, whereby any combination of elements can be selected by element coding.

FI command **BR_SPA3_(1)_(2)_(3)_(4)** **(Single Read)**

(1) = Drive address [1..32] with MTCNC systems
[1..16] with MTA systems

(2) = Parameter No. in the format: X-Y-ZZZZ

(3) = Element coding [standard or advanced format]

(4) = Complete Sercos data file name

Note: File and path details must be enclosed in inverted commas.

For a more detailed description of parameter no., element coding, displaying the operating date, see the description of the ‘SPA1’ command.

Response Structure The structure of the response data corresponds to the ‘SPA1’ command.

Example SPA3 Read the parameter S-0-0003 of the 3rd drive (element coding 0x48) of device 0 out of the Sercos data file D:\ SERCPAR .DAT.

FI command	**00_BR_SPA3_3_S-0-0003_48_"D:\SERCPAR.DAT"**			
Answer				
Line	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	0x0000	0x0000	0x0000	0x0000
2	µs			
3	2000			

Explanation A Sercos parameter of a drive is read out of the transferred Sercos data file and written into the addressed drive. Only the operating date can be written.

FI command **BR_SPA3_(1)_(2)_(3)_(4)** **(Single Read)**

(1) = Drive address [1..32] with MTCNC systems [1..16] with MTA systems

(2) = Parameter No. in the format: X-Y-ZZZZ

(3) = Element coding [standard or advanced format]

(4) = Complete Sercos data file name

Note: File and path details must be enclosed in inverted commas.

For a more detailed description of parameter no., element coding, displaying the operating date, see the description of the ‘SPA1’ command.

Response Structure The structure of the response data corresponds to the ‘SPA1’ command.

Example SPA3 Write the operating date of the Sercos parameter S-0-0305, which is saved in the Sercos data file, into drive 3 of device 0. The name of the Sercos data file is D:\SERCPAR.DAT.

FI command	**00_BR_SPA3_3_S-0-0003_48_"D:\SERCPAR.DAT"**			
Answer				
Line	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	0x0000	0x0003	0x0000	0x0000

Explanation The "SPA4" command is identical with the "SPA1" command. They only differ in one feature: with the "SPA4" command, the system makes up to 100 attempts to compensate any SERCOS error 0x13EA (SERCOS busy flag set).

FI command **BR_SPA1_(1)_(2)_(3)** **(Single Read)**

(1) = Drive address [1..16]

(2) = Parameter No. in the format: X-Y-ZZZZ

(3) = Element coding [standard or advanced format]

For a more detailed description of parameter no., element coding, displaying the operating date, see the description of the ‘SPA1’ command.

Response Structure The structure of the response data corresponds to the ‘SPA1’ command.

FI command **BW_SPA4_(1)_(2)_(3)** **(Single Write)**

(1) = Drive address [1..16]

0(2) = Parameter No. in the format: X-Y-ZZZZ

(3) = Element coding [standard or advanced format]

For a more detailed description of parameter no., element coding, displaying the operating date, see the description of the ‘SPA1’ command.

The write value is passed as with the "SPA1" command.

Response Structure The structure of the response data corresponds to the ‘SPA1’ command.

10.85 Active SERCOS Phase Switch-Over: SPH

MWAX device group

Designation **SPH** **S**ERCOS **PH**ase

Explanation All drives within a SERCOS ring are in the same communication phase. The phase status can be read-out or changed by this command.

FI command

BR_SPH	**(Single Read)**
BC_SPH	**(Cyclic Read)**
BB_SPH	**(Break Cyclic Read)**
BW_SPH	**(Single Write)**

Value to be written/ Result The phase conditions allowed are shown by the numbers [0...4].

Response Structure The following table shows the general structure of the response to the FI command "SPH". In the first line, column 2 or column 4, the number of the drive is output that reports the SERCOS error or the current system error. Not all current system errors can be directly allocated to a drive. In this case, the single result "Drive No." is set to 0x0000.

Line	Column 1	Column 2	Column 3	Column 4
1	<SERCOS error>	<Drive No. SERCOS error>	<Current system error>	<Drive no. that has caused the current system error>
2	Read: current phase Write: former phase			

Example: BR_SPH Read SERCOS phase: Read the active phase of the synax control at device address 00.

FI command	**00_BR_SPH**			
Answer				
Line	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	0x0000	0x0000	0x0000	0x0000
2	2			

Example: BW_SPH Write SERCOS Phase Switch-over the synax control (write) after phase 4; phase 2 is active.

FI command	**Value to be written: 4** **00_BW_SPH**			
Answer				
Line	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	0x0000	0x0000	0x0000	0x0000
2	2			

Reference to Literature See chapter entitled "Literature" [42].

10.86 Reading or Writing Tool Data Record: TDA

MWAX device group

Designation **TDA** **T**ool **DA**ta

Explanation A complete tool data record consisting of basic data and defined cutter data is read from or written into the control unit.

FI command Read the complete tool data record. For this FI command, the tool data record is addressed via the tool memory and the location number.

BR_TDA7_(1)_(2){_(3)} **(Single Read)**

(1) = Tool memory [M = magazine,
S = spindle,
G = gripper]

(2) = Location number [1...50]

(3) = Required measurement system (opt.) [mm, inch]

Using the optional third parameter it is possible to pre-select conversion of the result into mm or inches.

Response Structure The following table shows the general structure of the response to the FI command "BR_TDA7".

Column 1	Tool memory
Column 2	Tool location
Column 3	Tool number
Column 4	Tool name
Column 5	Total tool radius
Column 6	Tool radius offset
Column 7	Tool radius geometry
Column 8	Tool radius wear
Column 9	Total tool length
Column 10	Tool length offset
Column 11	Tool length geometry
Column 12	Tool length wear

Example TDA7

Read the complete tool data record of machine location 15. Radius and length are indicated in the basic programming unit.

FI command		00_BR_TDA7_M_15
Line	**Column**	**Answer**
1	1	M
	2	15
	3	4
	4	Miller D8
	5	10.0
	6	0.5
	7	9.4
	8	0.1
	9	100.0
	10	1.0
	11	99.0
	12	0.0

FI command

Write the complete tool data record. For this FI command, the tool data record is addressed via the tool memory and the location number.

BW_TDA7_(1)_(2){_(3)} **(Single Write)**

(1) = Tool memory [M = magazine,
S = spindle,
G = gripper]

(2) = Location number [1...50]

(3) = Required measurement system (opt.) [mm, inch]

Using the optional third parameter it is possible to pre-select conversion of the write value into mm or inches.

Values to be written

The values to be written are passed in a table.

Note: The "|" (= 0x7D) character is used as a separator between the data elements.

Code to be written

Data element	Value
1	Tool number
2	Tool name
3	Tool radius offset
4	Tool radius geometry
5	Tool radius wear
6	Tool length offset
7	Tool length geometry
8	Tool length wear

Example TDA7 Write the data of tool 4 to magazine location 15:

- Tool number 4
- Tool name "Miller D8"
- Radius offset 0.5 mm
- Radius geometry 9.4 mm
- Radius wear 0.1 mm
- Length offset 1.0 mm
- Length geometry 99.0 mm
- Length wear 0.0 mm

Radius and length are indicated in "mm".

FI command	**00_BW_TDA7_M_15_mm**
Values to be written **4|Fräser D8|0.5|9.4|0.1|1.0|99.0|0.0**	

Note: To delete a tool, "0" must be passed as tool number.

10.87 Read Tool Data from the Current Tool List: TDL

MWAX device group

Designation **TDL** Tool **D**ata **L**ist

Explanation All data of all defined tools are read for the current tool list.

FI command The parameters are used to define the tool list in more detail.

BR_TDL1_(1)_(2)_(3){_(4)}	**(Single Read)**
(1) = Number of spindle locations	[0...1]
(2) = Number of gripper locations	[0...2]
(3) = Number of magazine locations	[0...50]
(4) = Required measurement system (opt.)	[mm, inch]

Using the optional fourth parameter it is possible to pre-select conversion of the result into mm or inches.

Response Structure The following table shows the general structure of the response to the FI command "BR_TDL1".

Line	**Column 1**	**Column 2**	**Column 3**	**Column 4**	**Column 5-8**	**Column 9 - 12**
1 .. n	Tool magazine	Tool location	Tool number	Tool designation	Tool radii	Tool lengths

Example TDL1 Read the complete tool list with one spindle tool, two gripper locations, and 24 magazine locations.

FI command	**00_BR_TDL1_1_2_24**											
Answer												
Line	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**	**S9**	**S10**	**S11**	**S12**
1	M	2	12	Miller D6	3.0000	0.1000	2.9000	0.0000	80.0000	0.5000	79.4000	0.1000
2	M	6	2	Miller D8	4.0000	0.2000	3.8000	0.0000	90.0000	0.0000	89.9500	0.0500
3	M	12	6	Drill D6	3.0000	0.0000	3.0000	0.0000	80.0000	0.2000	79.8000	0.0000
4	M	15	1	Drill D8	4.0000	0.0000	4.0000	0.0000	90.0000	0.2000	89.8000	0.0000
5	S	1	10	Drill D4	2.0000	0.3000	1.7000	0.0000	70.0000	0.8000	69.1700	0.0300
6	G	2	15	Miller D4	2.0000	0.1000	1.9000	0.0000	70.0000	0.0000	69.8500	0.1500

Note: Locations without tools are not output.

10.88 Tool Data Record Elements: TLD

MWAX device group

Designation **TLD** **Too**L **D**ata

Explanation Returns elements of a tool in the tool magazine.

FI command **BR_TLD7_(1)_(2)_(3)_(4){_(5)}** **(Single Read)**

(1) = Tool number	[1...99999]
(2) = Index number	[1...1]
(3) = Data record	[0 = tool basic data, 1 = cutter data]
(4) = Data element	of the base data: [1..0.4] of the tool edge data: [1...8]
(5) = Required measurement system (opt.)	[mm, inch]

Using the optional fifth parameter it is possible to pre-select conversion of the result into mm or inches.

Response Structure The following table shows the general structure of the response to the FI command "BR_TLD7". One line with one column is output for the returned value.

Line 1	**Column 1**

Value Range/Meaning of the Columns

1..00.4 = requested tool basic data	1 = Tool magazine 2 = Tool location 3 = Tool number 4 = Tool name
1..00.8 = requested tool cutter data	1 = Total tool radius 2 = Tool radius offset 3 = Tool radius geometry 4 = Tool gradius wear 5 = Total tool length 6 = Tool length offset 7= Tool length geometry 8 = Tool length wear

Example TLD7

Read the designation (basic date 4) of tool no. 3.

FI command		**00_BR_TLD7_3_1_0_4**
Line	**Column**	**Answer**
1	1	Miller D8

FI command

Write single element of basic data or cutter data of a tool. Addressing via tool number + duplo number.

BW_TLD7_(1)_(2)_(3)_(4) {_(5)} **(Single Write)**

(1) = Tool number	[1...99999]
(2) = Index number	[1...1]
(3) = Data record	[0 = tool basic data, 1 = cutter data]
(4) = Data element	of the base data: [4] of the tool edge data: [2...4, 6...8]
(5) = Required measurement system (opt.)	[mm, inch]

With the basic tool data, only data element 4 (tool designation) can be used.

Using the optional fifth parameter it is possible to pre-select conversion of the write value into mm or inches.

Value to be written

4 = Tool basic data to be written	4 = Tool name
2...4, 6...8 = Tool edge data to be written	2 = Tool radius offset 3 = Tool radius geometry 4 = Tool radius wear 6 = Tool length offset 7 = Tool length geometry 8 = Tool length wear

Note: The value to be written is passed to the "acValue" parameter as an ASCII value in the "DataTransfer" routine.

Response Structure

One line is output with a column for acknowledgement of whether or not the FI command has been executed successfully.

(P_ACK) = **P**ositive **ACK**nowledge	Data element has been set

Example TLD7

Write the tool no. 3 / duplo no. 1 into tool edge 1 of data element 7 (length geometry). The value is preset as Inch.

FI command		**00_BW_TLD7_3_1_1_7_inch** **Value to be written: 6.5**
Line	**Column**	**Answer**
1	1	(P_ACK)

10.89 Torque of a Spindle: TQE

MWAX device group

Designation **TQE** **T**or**Q**u**E**

Explanation The torque at a selected axis is read out. The FI command "TQE2" returns the torque of an axis, related to the physical axis number.

FI command Output the torque at the selected axis of the device specified, related to the physical axis number.

BR_TQE2_(1) **(Single Read)**

(1) = Physical axis number [1...16]

Response Structure The following table shows the general structure of the response to the FI command "TQE2". One line with three columns is output for the name of the axis, the torque and the unit [%].

Line 1	Column 1	Column 2	Column 3

Value Range/Meaning of Columns

1 = Axis name	[according to settings of axis parameters]
2 = Torque	[to one decimal place]
3 = Unit	[%]

Note: If the selected axis is not defined, or if it is no spindle, then the response in all columns is [--].

Example TQE2 Read the torque at the A axis. Axis A is defined as a spindle.

FI command	00_BR_TQE2_1		
Answer			
Line	Column 1	Column 2	Column 3
1	A	-25.6	[%]

10.90 Requesting Watch List Allocations: WLA

MWAX device group

Designation **WLA** **W**atch **L**ist **A**llocation

Explanation Requests free watch list allocations. A maximum of ten free watch list allocations can be requested with one FI command.

BR_WLA1_(1) **(Single Read)**

(1) =	Number of the requested free watch list numbers	The required number of free watch list allocations is identified here. The allowed value range: 1…10

Response Structure The following table shows the general structure of the response to the FI command "WLA1".

Line 1	Column 1	...	Column n

Value Range/Meaning of Columns

1 =	1. free watch list allocation	Value range: 1..15
2 =	2. free watch list allocation	Value range: 1..15
3 =	3. free watch list allocation	Value range: 1..15
n =	nth free watch list allocation	Value range: 1..15

Example WLA1

Request four free watch list allocations.

Assumption:
Watch list allocations 3 and 5 are already assigned!

FI command		**00_BR_WLA1_4**
Line	**Column**	**Answer**
1	1	0
	2	1
	3	2
	4	4

10.91 Freeing Watch List Allocations: WLF

MWAX device group

Designation **WLF** **W**atch **L**ist **F**ree

Explanation Previously requested watch list allocations are freed again.

FI command Free ALL assigned watch list allocations for the selected device.

BR_WLF1 **(Single Read)**

Note: The FI command "WLF1" frees ALL assigned watch list allocations, including those of other WIN32 applications.

Response Structure The following table shows the general structure of the response to the FI command "WLF1".

Line 1	**Column 1**	**...**	**Column n**

Value Range/Meaning of Columns

1 =	1. freed watch list allocation	Value range: 1..15
2 =	2. freed watch list allocation	Value range: 1..15
3 =	3. freed watch list allocation	Value range: 1..15
n =	nth freed watch list allocation	Value range: 1..15

Example WLF1

Free ALL assigned watch list allocations.

Assumption:
The following watch list numbers have been allocated: 0,1,2,3

FI command		**00_BR_WLF1**
Line	**Column**	**Answer**
1	1	0
	2	1
	3	2
	4	3

FI command Free the required watch list allocations for a selected device.

BR_WLF2_(1)_{(2)..(10)}	**(Single Read)**
(1)..(10) = List of watch list allocations to be released	A maximum of 10 watch list allocations can be transferred here to be freed again.

Response Structure The following table shows the general structure of the response to the FI command "WLF2".

Line 1	Column 1	...	Column n

Value Range/Meaning of Columns

1 =	1. enabled watch list allocation	Value range: 1..15
2 =	2. enabled watch list allocation	Value range: 1..15
3 =	3. enabled watch list allocation	Value range: 1..15
n =	nth freed watch list allocation	Value range: 1..15

Example WLF2

Free required watch list allocations:
Assumption:
Watch list allocations 0,3,4, and 8 have first been requested using the FI command "WLA1".

FI command		**00_BR_WLF2_0_3_4_8**
Line	**Column**	**Answer**
1	1	0
	2	3
	3	4
	4	8

10.92 Data of the Zero Offset Table: ZOD

MWAX device group

Designation **ZOD** Zero **O**ffset **D**ata

Explanation

The zero-offset table data can be read and written. The zero offsets allow the origin of a coordinate axis to be shifted (offset) by a set value, related to the original position of the machine. The location of the machine zero point remains securely stored in the CNC controls and is not changed by the zero offset.

FI command

BR_ZOD7_(1)_(2){_(3)}	**(Single Read)**
(1) = Physical axis number	[1...16]
(2) = Offset type	[3...9 3 = General offset 4...9 = G54 to G59]
(3) = Required measurement system (opt.)	[mm, inch]

Using the optional third parameter it is possible to pre-select conversion of the result into mm or inches.

Response Structure

The following table shows the general structure of the response to the FI command "BR_ZOD7". One line with three columns is output for the returned values.

Line 1	Column 1	Column 2	Column 3

Value Range/Meaning of the Columns

1 = Axis name	[according to setting of axis parameters]
2 = Offset value	[mm or Inch, as defined in the settings]
3 = Unit	[mm or Inch, as defined in the settings]

Example ZOD7

Read the offset for G55 of the Z axis. Unit: Inches.

FI command	**00_BR_ZOD7_3_5_inch**		
Answer			
Line	**Column 1**	**Column 2**	**Column 3**
1	Z	25.0000	inch

FI command

Write a single offset value of an axis and a type.

BW_ZOD7_(1)_(2){_(3)} **(Single Write)**

(1) = Physical axis number [1...16]

(2) = Offset type [3...9
3 = General offset
4...9 = G54 to G59]

(3) = Required measurement system (opt.) [mm, inch]

Using the optional third parameter it is possible to pre-select conversion of the write value into mm or inches.

Value to be written

Zero offset

Note: The value to be written is passed to the "acValue" parameter as an ASCII value in the "DataTransfer" routine.

Response Structure

One line is output with a column for acknowledgement of whether or not the FI command has been executed successfully.

(P_ACK) = **P**ositive **ACK**nowledge Data element has been set

Example ZOD7

Write the zero offset G58 of the X axis with the unit "Inch".

FI command		**00_BW_ZOD7_2_8_mm** **Value to be written: 280.0**
Line	**Column**	**Answer**
1	1	(P_ACK)

FI command

Read all the offset table of all axes.

BR_ZOD8{_(1)} **(Single Read)**

(1) = Required measurement system (opt.) [mm, inch]

Using the optional first parameter it is possible to pre-select conversion of the result into mm or inches.

Response Structure

The following table shows the general structure of the response to the FI command "BR_ZOD8". 16 lines, each with 10 columns, are output.

Line 1	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**	**S9**	**S10**
Line 2										
...										
Line 16										

Value range/Meaning of lines

Line = physical axis number [according to setting of axis parameters]

1 = Axis type [0 = axis not defined
1 = Linear axis
2 = rotary axis
3 = Modulo axis
4 = main spindle]

2 = Axis name [according to settings of axis parameters]

3 = Unit [mm, Inch] for linear axes; otherwise, acc. to axis setting

4 = Offset value G54

5 = Offset value G55

6 = Offset value G56

7 = Offset value G57

8 = Offset value G58

9 = Offset value G59

10 = General offset

Note: If an axis is not defined then the response in all columns is "--".

Example ZOD8 Read all the zero table; the offset values are output in mm.

<table>
<tr><th colspan="5">FI command</th><th colspan="6">00_BR_ZOD8_mm</th></tr>
<tr><th colspan="11">Answer</th></tr>
<tr><th>Line</th><th>S1</th><th>S2</th><th>S3</th><th>S4</th><th>S5</th><th>S6</th><th>S7</th><th>S8</th><th>S9</th><th>S10</th></tr>
<tr><td>1</td><td>0</td><td></td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td></tr>
<tr><td>2</td><td>1</td><td>X</td><td>[mm]</td><td>13.0000</td><td>14.0000</td><td>0.0000</td><td>0.0000</td><td>0.0000</td><td>14.0000</td><td>5.0000</td></tr>
<tr><td>3</td><td>1</td><td>Z</td><td>[mm]</td><td>56.0000</td><td>23.4548</td><td>0.0000</td><td>0.0000</td><td>0.0000</td><td>23.4548</td><td>4.0000</td></tr>
<tr><td>4</td><td>1</td><td>Y</td><td>[mm]</td><td>27.0000</td><td>12.5000</td><td>0.0000</td><td>0.0000</td><td>0.0000</td><td>8.0000</td><td>0.0000</td></tr>
<tr><td>5</td><td>2</td><td>B</td><td>[deg]</td><td>18.0000</td><td>18.0000</td><td>0.0000</td><td>0.0000</td><td>0.0000</td><td>18.0000</td><td>0.0000</td></tr>
<tr><td>6</td><td>2</td><td>C</td><td>[deg]</td><td>90.0000</td><td>27.0000</td><td>0.0000</td><td>0.0000</td><td>0.0000</td><td>27.0000</td><td>0.0000</td></tr>
<tr><td>7</td><td>0</td><td></td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td></tr>
<tr><td>8</td><td>0</td><td></td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td></tr>
<tr><td>9</td><td>0</td><td></td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td></tr>
<tr><td>10</td><td>0</td><td></td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td></tr>
<tr><td>11</td><td>0</td><td></td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td></tr>
<tr><td>12</td><td>0</td><td></td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td></tr>
<tr><td>13</td><td>0</td><td></td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td></tr>
<tr><td>14</td><td>0</td><td></td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td></tr>
<tr><td>15</td><td>0</td><td></td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td></tr>
<tr><td>16</td><td>0</td><td></td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td><td>--</td></tr>
</table>

11 FI Commands - MSYX Device Group (SYNAX200)

The FI Commands described in this chapter are valid for the MSYX device group. In this device group, the following types are listed as well as possible addresses:

Group	Accompanying Types	Address
MSYX	SYNAX200-P, SYNAX200-R	[00...63]

Note: Please note that the device address must be set before the respective FI command, e.g. 00_CR_AAC_0 (refer also here to the chapter 2.1 "Elements of the FI Command").

Parameters for the MSYX device group are grouped together in the chapter entitled "Construction and Availability of the FI Commands", "Data Tables".

11.1 Active Diagnosis Window ADW

MSYX device group

Designation **ADW** Active **D**iagnosis **W**indow

Explanation Indicates the window types for which data is required.

For improved performance and with some diagnosis windows, not all the data is called up each time a diagnosis is performed, but only when the data is actually required.

Through this FI command, the diagnosis server can be informed that the data of the respective window type is required.

This command has to be issued at least every 3 seconds while the data is required. If this command is not issued any more, calling-up of all data will stop.

FI command Indicates the window types for which data is required.

BW_ADW1_(1){_(2)}	**(Single Write)**
(1) = Type of diagnosis window	[1 = CNC error, 2 = sequence errors, 3 = general errors, 4 = messages, 10 = start requirements, 11 = warnings, 12 = setup diagnosis]
(2) = Module number	[1...99] ! only for window type 1 -4!

Example ADW1 Call up data for CNC error in controller 0 module 1.

FI command	00_BW_ADW1_1_1

11.2 Determining the Actual (Current) System Error: ASE

MSYX device group

Designation **ASE** Actual System Error

Explanation The current system error is read out, whereby the answer 0x0000 indicates that the Synax device is functioning correctly.

FI command **CR_ASE** **(Single Read)**

Response Structure The following table shows the general structure of the response to the FI command "ASE". In line 1, column 4, the number of the drive is output that reports the current system error. Not all current system errors can be directly allocated to a drive. In this case, the single result "Drive No." is set to 0x0000.

Line 1	Column 1	...	Column 4

Value Range/Meaning of Columns

1 = 0x0000

2 = 0x0000

3 = Current system error

4 = Drive No.

Example ASE Reading the current system error returns LWL ring interrupted.

FI command		00_CR_ASE
Line	**Column**	**Answer**
1	1	0x0000
	2	0x0000
	3	0x8009
	4	0x0000

Reference to Literature See chapter entitled "Literature" [42].

11.3 Trigger Control Reset: CRT

MSYX device group

Designation **CRT** **C**ontrol **R**ese**T**

Explanation The control reset allows the selected device to be reset after a system error. If there is no system error at the selected device then the job is ignored.

CAUTION

Carrying out a reset completely re-initializes the device.
During initialization, communication is temporarily interrupted (inherent to design).

FI command **BW_CRT** **(Single Write)**

Value to be written Trigger reset 0

Note: The value to be written is passed to the "acValue" parameter in the "DataTransfer" routine.

Response Structure The return value of the "DataTransfer" routine is [0] if the write procedure has been successfully completed. In the event of an error, more information in the form of a general error result line can requested by the routine "ReadGroupItem" (refer here to chapter 8, "Error Codes" and "General Error Result Line").

Example CRT Trigger a control reset on the selected device.

FI command		**Value to be written: 0** **00_BW_CRT**
Line	**Column**	**Answer**
1	1	(P_ACK)

Reference to Literature See chapter entitled "Literature" [26].

11.4 Deleting the Actual (Current) System Error: CSE

MSYX device group

Designation **CSE** **C**lear **S**ystem **E**rror

Explanation An error reported by the Synax device is deleted again.

FI command **CW_CSE** **(Single Write)**

Value to be written: The contents of the value parameter is not evaluated.

Response Structure The following table shows the general structure of the response to the FI command "CSE". In line 1, column 4, the number of the drive is output that reports the current system error. Not all current system errors can be directly allocated to a drive. In this case, the single result "Drive No." is set to 0x0000.

Line 1	Column 1	...	Column 4

Value Range/Meaning of Columns

1 = 0x0000

2 = 0x0000

3 = Actual (current) system error

4 = Drive No.

Example CSE Deleting the actual (current) system error:

FI command		00_CW_CSE
Line	**Column**	**Answer**
1	1	0x0000
	2	0x0000
	3	0x0000
	4	0x0000

Reference to Literature see chapter entitled "Literature" [45].

11.5 Setting the Communication Timeout Time DCT

MSYX device group

Designation

DCT **D**evice **C**ommunication **T**imeout

Explanation

By means of this command, the timeout time for the selected device is set dynamically (timeout time in ms).

FI command

BW_DCT1_(1) **(Single Write)**

(1) = requested timeout time in ms

Response Structure

The response to the "DCT1" FI command consists of one line with one column.

Line 1	Column 1

Value Range/Meaning of Columns

1 = Status message (P_ACK) (P_ACK)

Example DCT1

For the device 00, the timeout time is set 1500 ms.

FI command		**00_BW_DCT1_1500**
Line	**Column**	**Answer**
1	1	(P_ACK)

FI command

With this command, the timeout time for the selected device can be reset to default value.

BW_DCT2 **(Single Write)**

Response Structure

The response to the "DCT2" FI command consists of one line with one column.

Line 1	Column 1

Value Range/Meaning of Columns

1 = Status message (P_ACK) (P_ACK)

Example DCP2

For the device 00, the timeout time is reset to the default value.

FI command		**00_BW_DCT2**
Line	**Column**	**Answer**
1	1	(P_ACK)

11.6 Deleting the FI Exclusive Mode: DEM

MSYX device group

Designation **DEM** **D**elete FI **E**xclusive **M**ode

Explanation This command is used to deactivate FI Exclusive mode for the selected device address.

FI Exclusive mode: In this mode, **ALL** the processes logged in at the FI – **with the exception** of the process issuing the SEM command – are blocked from data communication with this device address. This mode is used for example for data communication which must NOT be interrupted by other data requests. However, it is **imperative** that this FI Exclusive mode is deleted once more through the DEM command.

FI command **BW_DEM1** **(Single Write)**

Response Structure The following table shows the general structure of the response to the FI command "BW_DEM1". A line of 1 column is output.

Line 1	Column 1

Value Range/Meaning of Columns 1 = Status message (P_ACK) (P_ACK)

Example DEM1 Deactivate the FI Exclusive mode for device address 0. The FI Exclusive mode for device address 0 has previously been activated by the SEM command.

FI command		**00_BW_DEM1**
Line	**Column**	**Answer**
1	1	(P_ACK)

11.7 Static Device Information: DIF

MSYX device group

Designation **DIF** **D**evice **I**n**F**ormation

Explanation Static device information and network information is read according to the "IND_DEV.INI" and "FAR_DEV.INI" files.

FI command Reading of the static device information and network information of a selected device.

BR_DIF1	**(Single Read)**
BC_DIF1	**(Cyclic Read)**
BB_DIF1	**(Break Cyclic Read)**

Response Structure The following table shows the general structure of the response to the "DIF1" FI command. The response consists of one line with 24 columns.

Line 1	Column 1	...	Column 24

Value Range/Meaning of Columns

1 =	Local/far device address	[00..63]
2 =	Device name	IND_DEV.INI entry: DeviceName=
3 =	Device type	IND_DEV.INI entry: DeviceType=
4 =	PLC support	IND_DEV.INI entry: PLC=
5 =	Device status	IND_DEV.INI entry: DeviceStatus=
6 =	Assignment of a simulation pair	IND_DEV.INI entry: DeviceAssign=
7 =	Device mode	IND_DEV.INI entry: MtvncMode=
8 =	Communication channel	IND_DEV.INI entry: [CommAddrX]
9 =	Description of the communication channel	IND_DEV.INI entry: CommStr=
10 =	Timeout value	IND_DEV.INI entry: Timeout=
11 =	Device group	(see Chapter 6.1 "Identifier")
12 =	PLC component type	IND_DEV.INI entry: Component type1=
13 =	CNC component type	IND_DEV.INI entry: Component type2=
14 =	Device log	IND_DEV.INI entry: DeviceProtocol=
15 =	Device simulation	IND_DEV.INI entry: DeviceSimulation=
16 =	Not yet assigned	[--]
17 =	Not yet assigned	[--]
18 =	Not yet assigned	[--]
19 =	Not yet assigned	[--]
20 =	Network ON/OFF	[ON = Network active OFF = No network active]
21 =	Network name	Max. 28 ASCII characters
22 =	PC number	[00..99,XX]
23 =	PC name	Max. 255 ASCII characters
24 =	Local device address	[00..63]

Example DIF1

Read the static device information and network information of device 1 while the network is active.

FI command		**01_BR_DIF1**
Line	**Column**	**Answer**
1	1	01
	2	Print station right side
	3	SYNAX200-R-G2
	4	NO
	5	ON
	6	NO
	7	OFF
	8	4
	9	V24,COM2,19200,EVEN,RS232,TCOFF
	10	3500
	11	MSYX
	12	NONE
	13	NONE
	14	SIS
	15	OFF
	16	--
	17	--
	18	--
	19	--
	20	ON
	21	PC network 1
	22	29
	23	BTV20-STATION-LEFT
	24	01

11.8 Reading the Device Status Information: DSI

MSYX device group

Designation **DSI** Device **S**tatus **I**nformation

Explanation This enables the most important device status information to be read. The following information is returned:

Type of information	status	Statement
System error information		Yes/No
Mechanism error information		Yes/No
Machine key information		4 Byte HEX
Machine key information	valid	Yes/No
Machine status information		4 Byte HEX
Sercans information		4 Byte HEX
Parameter download	running	Yes/No
PLC download	running	Yes/No
Firmware download	running	Yes/No
Offline/Online information		Yes/No
Device simulation	switched on	Yes/No
Device status information		ON/ OFF
Communication channel defined		Yes/No
PLC components available		Yes/No
Monitor mode	active	Yes/No

FI command Read out device status information for ALL defined devices.

BR_DSI1 **(Single Read)**

BC_DSI1 **(Cyclic Read)**

BB_DSI1 **(Break Cyclic Read)**

Note: The "DSI1" FI command refers to all devices within this device group. Therefore, any valid device address can be indicated in the command line (see example DSI1). The FI device polling mechanism **MUST** be switched on (see system configurator)!

Response Structure The following table shows the general structure of the response to the "DSI1" FI command.

Line 1...n	Column 1	...	Column 11

Value Range/Meaning of Columns

1 =	Device address	[00...63]
2 =	System error information	[0 = there is no system error 1 = there is a system error]
3 =	Mechanism error information	[0 = there is no mechanism error 1 = there is a mechanism error]
4 =	Machine key information	[4 byte in HEX coding]
5 =	Machine key information valid?	[0 = not valid, 1=valid]
6 =	Machine status information	[4 byte in HEX coding]
7 =	Sercans information	[4 byte in HEX coding]

8 =	Is parameter download active?	[0 = parameter download not running 1 = parameter download running]
9 =	Is PLC download active?	[0 = PLC download not running 1 = PLC download running]
10 =	Is firmware download active?	[0 = PLC download not running 1 = PLC download running]
11 =	Offline/Online information	[0 = device connection interrupted 1 = device connection O.K.]
12 =	Device simulation switched on?	[0 = NO Simulation mode 1 = simulation mode]
13 =	Current device status information	[0 = Device status=OFF 1 = Device status=ON]
14 =	Communication channel defined?	[0 = NO communication channel 1 = Communication channel defined]
15 =	PLC components available ?	[0 = NO PLC component 1 = PLC component (DOS-Pcl) 2 = PLC component (WIN-Pcl)]
16 =	Monitor mode	[0 = NO monitor mode active 1 = Monitor mode active]

Example DSI1 Read the current device status information.

Assumption:
The following device addresses have been defined:

Device address 01 (SYNAX200-P)

Device address 03 (SYNAX200-R)

FI command		**01_BR_DSI1**
Line	**Column**	**Answer**
1	1	01
	2	0
	3	0
	4	00000000
	5	0
	6	00000000
	7	00000000
	8	0
	9	0
	10	0
	11	1
	12	0
	13	1
	14	1
	15	0
	16	0

2	1	03
	2	1
	3	0
	4	00000000
	5	0
	6	00000000
	7	00000000
	8	0
	9	0
	10	0
	11	1
	12	0
	13	1
	14	1
	15	0
	16	0

FI command

Read out device status information for a selected device.

BR_DSI2 **(Single Read)**

BC_DSI2 **(Cyclic Read)**

BB_DSI2 **(Break Cyclic Read)**

Response Structure

The following table shows the general structure of the response to the "DSI2" FI command.

Line 1	Column 1	...	Column 11

Value Range/Meaning of Columns

1 = Device address [00...63]

2 = System error information [0 = there is no system error
1 = there is a system error]

3 = Mechanism error information [0 = there is no mechanism error
1 = there is a mechanism error]

4 = Machine key information [4 byte in HEX coding]

5 = Machine key information valid? [0 = not valid, 1=valid]

6 = Machine status information [4 byte in HEX coding]

7 = Sercans information [4 byte in HEX coding]

8 = Is parameter download active? [0 = parameter download not running
1 = parameter download running]

9 = Is PLC download active? [0 = PLC download not running
1 = PLC download running]

10 = Is firmware download active? [0 = PLC download not running
1 = PLC download running]

11 = Offline/Online information [0 = device connection interrupted
1 = device connection O.K.]

12 = Device simulation switched on? [0 = NO Simulation mode
1 = simulation mode]

13 =	Current device status information	[0 = Device status=OFF 1 = Device status=ON]
14 =	Communication channel defined?	[0 = NO communication channel 1 = Communication channel defined]
15 =	PLC components available ?	[0 = NO PLC component 1 = PLC component (DOS-Pcl) 2 = PLC component (WIN-Pcl)]
16 =	Monitor mode	[0 = NO monitor mode active 1 = Monitor mode active]

Example DSI2 Read the current device status information for the selected device.

FI command		**00_BR_DSI2**
Line	**Column**	**Answer**
1	1	00
	2	0
	3	0
	4	00000000
	5	0
	6	00000000
	7	00000000
	8	0
	9	0
	10	0
	11	1
	12	0
	13	1
	14	1
	15	0
	16	0

11.9 Device Type and Accompanying Components: DTY

MSYX device group

Designation **DTY** **D**evice **TY**pe

Explanation The device type and the accompanying components of the selected device address are output.

FI command

BR_DTY1 **(Single Read)**

BC_DTY1 **(Cyclic_Read)**

Response Structure The following table shows the general structure of the response to the "DTY1" FI command. A line with three columns for the device type is output as well as the name of the first device component and the name of the second device component.

Line 1	Column 1	...	Column 3

Value Range/Meaning of Columns

1 =	Device Type	(see chapter entitled "Elements of the FI Command", and "Identifier")
2 =	Component type1	IND_DEV.INI entry: Component type1=
3 =	Component type 2	IND_DEV.INI entry: Component type2=

Example DTY1 Output the device type and the accompanying components of device address 00.

FI command	**00_BR_DTY1**		
Answer			
Line	**Column 1**	**Column 2**	**Column 3**
1	SYNAX200-R	NONE	PPC-R

11.10 Diagnosis Window Data: DWD

MSYX device group

Designation **DWD** **D**iagnosis **W**indow **D**ata

Explanation Diagnosis messages are output. The data is edited in such a way that they can be outputted directly in the diagnosis overview, i.e., where applicable, different types of diagnosis, such as ProVi and a process report, are returned simultaneously.

FI command Output all diagnosis messages.

For reasons of optimation, not all data is called up. Accordingly, the Diagnosis server must be informed that the data is required (see ADW).

BR_DWD1_(1){_(2)}	**(Single Read)**
BC_DWD1_(1){_(2)}	**(Cyclic Read)**
(1) = Type of diagnosis window	[1 = CNC error, 2 = sequence errors, 3 = general errors, 4 = messages, 10 = start preconditions, 11 = warnings, 12 = setup diagnosis]
(2) = Module number	[1...99] ! only for window type 1 -4 !

Output the first diagnosis messages.

BR_DWD2_(1){_(2)}	**(Single Read)**
BC_DWD2_(1){_(2)}	**(Cyclic Read)**
(1) = Type of diagnosis window	[1 = CNC error, 2 = sequence errors, 3 = general errors, 4 = messages, 10 = start preconditions, 11 = warnings, 12 = setup diagnosis]
(2) = Module number	[1...99] ! only for window type 1 -4!

Response Structure

The following table shows the general structure of the "DWD1" and "DWD2" FI commands. The number of lines depends on the number of messages pending. Different columns are valid according to the type of diagnosis.

If there are no messages, the number of lines is 0.

Line 1...n	Column 1	...	Column 12

Meaning of the Columns

1 =	Message text	[ASCII characters]
2 =	Time stamp day	[mm.dd.yyyy]
3 =	Time stamp hour	[hh:mm:ss]
4 =	Reference text available	[YES, NO]
5 =	Type of diagnosis	[1 = ProVi, 2 = SFC, 3 = MTC-NC, 4 = MTA-NC]
6 =	Message number	[ASCII characters]
7 =	Message ID	[ASCII characters] (DWORD, decimal) (ProVi)
8 =	Mechanism number	[0..31] (MTC-NC) [0] (MTA-NC)
9 =	2 byte additional information	[ASCII characters] (MTC NC)
10 =	Message group	[1...9999] (MTA-NC)
11 =	SFC entity name	[ASCII characters]
12 =	NC note	[ASCII characters] (MTC NC)
13 =	Analysis of criteria available	[YES, NO] (ProVi, SFC)
14 =	Message HTML file	[ASCII characters] (ProVi, MTC-NC)

Example DWD1 All diagnosis messages from module 3 in control unit 0.

There are two messages.

FI command		**00_BR_DWD1_4_3**
Line	**Column**	**Answer**
1	1	Guard not closed
	2	01.27.2000
	3	14:56:32
	4	YES
	5	1
	6	34
	7	43923028
	8	
	9	
	10	
	11	
	12	
	13	YES
	14	
2	1	Station waiting until tool-change command has ended.
	2	01.27.2000
	3	15:03:10
	4	YES
	5	3
	6	79
	7	
	8	1
	9	0
	10	
	11	
	12	
	13	NO
	14	

Example DWD2 First diagnostic message from module 3 in control unit 0.

There are two messages.

FI command		**00_BR_DWD2_4_3**
Line	**Column**	**Answer**
1	1	Guard not closed
	2	01.27.2000
	3	14:56:32
	4	YES
	5	1
	6	34
	7	43923028
	8	
	9	
	10	
	11	
	12	
	13	YES
	14	

Reference to Literature See chapter entitled "Literature" [13].

11.11 Component Information for a System Error: ECI

MSYX device group

Designation **ECI** **E**rror **C**omponent **I**nformation

Explanation When a system error is present, this command is used to define which controller component is causing the error. Here, PLC components are differentiated from general controller components (e.g. CNC, Synax, MTA, …).

FI command **BR_ECI1** **(Single Read)**

Response Structure The response to the "ECI1" FI command consists of one line with two columns.

Line 1	**Column 1**	**Column 2**

Value Range/Meaning of Columns

1 =	PLC component information	[0 = There is NO system error at the PLC) 1 = There is a system error at the PLC]
2 =	General information on controller components	[0 = There is NO system error at the general control component [0 = There is a system error at the general control component; the following applies: 2 = CNC component 3 = SYNAX component 4 = VISUAL-MOTION component 5 = MTA component 6 = TRANS 200 component]

Example ECI1

There is a system error present in device 0 (SYNAX200-P) which is caused by the PLC component.

FI command		**00_BR_ECI1**
Line	**Column**	**Answer**
1	1	1
1	2	0

11.12 Existing Errors: EDE

MSYX device group

Designation

EDE **E**xisting **D**iagnosis **E**rror

Explanation

Whether or not errors exist in a control unit or in a module is queried. These can be step chain errors, NC errors, MTA 200 errors or ProVi errors.

FI command

Query whether there are errors in this control unit.

BR_EDE1 **(Single Read)**

BC_EDE1 **(Cyclic Read)**

Response Structure

The following table shows the general structure of the "EDE1" FI command.

Line 1	**Column 1**

Meaning of the Columns

1 = Error exists [YES, NO]

Example EDE1

Do errors exist in control unit 0?

FI command		**00_BR_EDE1**
Line	**Column**	**Answer**
1	1	YES

FI command

Query whether or not errors exist in a specific module.

BR_EDE2_(1) **(Single Read)**

BC_EDE2_(1) **(Cyclic Read)**

(1) = Module number [1...99]

Response Structure

The following table shows the general structure of the "EDE2" FI command.

Line 1	**Column 1**

Meaning of the Columns

1 = Error exists [YES, NO]

Example EDE2

Do errors exist in module 1 on control unit 0?

FI command		**00_BR_EDE2_2**
Line	**Column**	**Answer**
1	1	NO

11.13 Existing Diagnosis Window: EDW

MSYX device group

Designation **EDW** **E**xisting **D**iagnosis **W**indow

Explanation Which types of diagnosis window exist is queried.

FI command Output all types of diagnosis window.

BR_EDW1 **(Single Read)**

Response Structure The following table shows the general structure of the "EDW1" FI command. The number of lines depends on the number of types of window existing.

Line 0...n	Column 1	Column 2

Meaning of the Columns

1 = Type of diagnosis window	[1 = CNC error, 2 = sequence errors, 3 = general errors, 4 = messages, 10 = start requirements, 11 = warnings, 12 = setup diagnosis]
2 = Module number	[ASCII characters] 0 = Diagnosis window type does not belong to any module

Example EDW1 All types of diagnosis window in control unit 0.

There are three diagnosis windows.

FI command		**00_BR_EDW1**
Line	**Column**	**Answer**
1	1	10
	2	0
2	1	1
	2	3
3	1	2
	2	3

FI command Output all diagnosis window types for a module.

BR_EDW2_(1) **(Single Read)**

(1) = Module number [1...99]

Response Structure The following table shows the general structure of the "EDW2" FI command. The number of lines depends on the number of types of window existing.

Line 0...n	Column 1	Column 2

Meaning of the Columns

1 = Type of diagnosis window	[1 = CNC error, 2 = sequence errors, 3 = general errors, 4 = messages]
2 = Module number	[ASCII characters] 0 = Diagnosis window type does not belong to any module

Example EDW2

All types of diagnosis window in Module 3, Control unit 0.

There are two diagnosis windows.

FI command		**00_BR_EDW2_3**
Line	**Column**	**Answer**
1	1	1
	2	3
2	1	2
	2	3

FI command

Query a specific type of diagnosis window.

BR_EDW3_(1){_(2)}	**(Single Read)**
(1) = Type of diagnosis window	[1 = CNC error, 2 = sequence errors, 3 = general errors, 4 = messages, 10 = start requirements, 11 = warnings, 12 = setup diagnosis]
(2) = Module number	[1...99] ! only for window type 1 -4!

Response Structure

The following table shows the general structure of the "EDW3" FI command.

Line 1	**Column 1**

Meaning of the Columns

1 = Type of diagnosis window exists	[YES, NO]

Example EDW3

Query whether or not a CNC error window exists in module 3, control unit 0.

FI command		**00_BR_EDW3_1_3**
Line	**Column**	**Answer**
1	1	YES

Reference to Literature

See chapter entitled "Literature" [13].

11.14 Process Activated Language: LNG

MSYX device group

Designation **LNG** Activated **L**a**NG**uage

Explanation The country code of the language activated for the selected device address is output.

FI command **BR_LNG** **(Single Read)**

Response Structure The response to the FI command "LNG" consists of one line with one column for the country code of the activated language.

Value Range of the Column 1 = Country code of the activated language [2 ASCII characters]

Country code	Language
EN	English (US/GB)
DE	German
ES	Spanish
IT	Italian
FR	French
HU	Hungarian
PT	Portuguese
SE	Swedish
CS	Czech

Example LNG Read the country code of the language activated at device address 00.

FI command		00_BR_LNG
Line	**Column**	**Answer**
1	1	SE

Explanation This command is used to define the language settings of the selected device.

FI command **BW_LNG_(1)** **(Single Read)**

(1) = Language to be set =[see the Windows language extension, e.g. DE, EN, …]

Response Structure The following table shows the general structure of the response to the FI command "BW_LNG". A line of 1 column is output.

Line 1	Column 1

Value Range/Meaning of Columns 1 = Status message (P_ACK) (P_ACK)

Example LNG Set the language setting of the device 0 to English.

FI command		00_BW_LNG_EN
Line	**Column**	**Answer**
1	1	(P_ACK)

11.15 Read System Messages: MSG

MSYX device group

Designation **MSG** **M**e**S**sa**G**e

Explanation Reading of system messages

FI command Message

CC_MSG_(1) **(Cyclic Read)**

(1) = SYS message numbers

Note: Exists only as a cyclic command

Response Structure The response of the FI command 'MSG' consists of the system message data.

Example MSG 00_CC_MSG_64 (64 = MSG_SYSERRGEN)

FI command		**00_CC_MSG_64/3**
Line	**Column**	**Answer**
1	1	00

Restriction The following system messages:

SYS Message	SYS Message number
MSG_PCLUPDBEG	52
MSG_PARUPDBEG	24
MSG_FWAUPDBEG	82

These commands cannot be used with the following programs:

- Bosch Rexroth OPC server
- Bosch Rexroth DDE server

11.16 Setting of Device Status Information: SDS

MSYX device group

Designation **SDS** **S**et **D**evice **S**tatus

Explanation By this command, the device status information can be set; here, the configuration file IND_DEV.INI is adjusted as well.

Note: When this command is transmitted, the following system messages are generated:
MSG_DEVICEOFF or MSG_DEVICE_ON !

FI command With this command, the device status information of **ALL** defined devices can be set.

BW_SDS1_(1)	**(Single Write)**
(1) = Device status information to be set	0 = Device status information OFF 1 = Device status information ON

Response Structure The following table shows the general structure of the response to the "SDS1" FI command.

Line 1	**Column 1**

Value Range/Meaning of Columns

1 =	Status report	[(P_ACK)]

Example SDS1 Set device status information to OFF for **ALL** defined devices.

FI command		**00_BW_SDS1_0**
Line	**Column**	**Answer**
1	1	(P_ACK)

FI command With this command, the device status information for a selected device can be set.

BW_SDS2_(1)	**(Single Write)**
(1) = Device status information to be set	0 = Device status information OFF 1 = Device status information ON

Response Structure The following table shows the general structure of the response to the "SDS2" FI command.

Line 1	**Column 1**

Value Range/Meaning of Columns

1 =	Status report	[(P_ACK)]

Example: SDS2 Set device status information to OFF for the selected device 00.

FI command		**00_BW_SDS2_0**
Line	**Column**	**Answer**
1	1	(P_ACK)

11.17 Setting the FI Exclusive Mode: SEM

MSYX device group

Designation **SEM** **S**et FI **E**xclusive **M**ode

Explanation This command is used to activate FI Exclusive mode for the selected device address.

FI Exclusive mode: In this mode, **ALL** the processes logged in at the FI – **with the exception** of the process issuing the SEM command – are blocked from data communication with this device address. This mode is used for example for data communication which must NOT be interrupted by other data requests. However, it is **imperative** that this FI Exclusive mode is deleted once more through the DEM command.

FI command **BW_SEM1** **(Single Write)**

Response Structure The following table shows the general structure of the response to the FI command "BW_SEM1". A line of 1 column is output.

Line 1	Column 1

Value Range/Meaning of Columns 1 = Status message (P_ACK) (P_ACK)

Example SEM1 Activate FI Exclusive mode for device address 0.

FI command		**00_BW_SEM1**
Line	**Column**	**Answer**
1	1	(P_ACK)

11.18 Software Installation Data: SID

MSYX device group

Designation **SID** **S**oftware **I**nstallation **D**ata

Explanation Information is returned regarding installation. This information includes the installation paths, the software version used, DLL mode, context information, plus service pack and release information.

FI command Read-in the installation data.

BR_SID1 **(Single Read)**

Response Structure One line with 8 columns is output for the returned values.

Line 1	Column 1	...	Column 16

Meaning of the Columns

1 = Basic directory	[EXE files of the DOS-BOF]
2 = FI installation directory	[FI directory]
3 = Data directory	[in accordance with DOS-BOF]
4 = GBO version	[from INDRAMAT.ini]
5 = IF-DLL mode	[from INDRAMAT.ini]
6 = IF version	[from INDRAMAT.ini from DLL mode 400]
7 = Service pack info	[from INDRAMAT.ini from DLL mode 420]
8 = Release info	[from INDRAMAT.ini from DLL mode 420]
9 = IF-Build-Info	[in accordance with Build process]
10 = Current context name	[in accordance with the installation]

11 = Physical installation path	[in accordance with the installation]
12 = Complete IF version indication string	
13 = WinPCL build number	[in accordance with WinPCL]
14 = Version number of the PLC compiler	[in accordance with WinPCL]
15 = Version number of the PLC linker	[in accordance with WinPCL]
16 = Version number of the PLC data basis	[in accordance with WinPCL]
17 = Platform version	

Example SID1 Return information on the current installation.

FI command		**00_BR_SID1**
Line	**Column**	**Answer**
1	1	--
	2	D:\Program Files\Indramat\MTGUI\Bin
	3	--
	4	005-22Vxx
	5	07.00
	6	07V00
	7	--
	8	--
	9	Build 3124 Mar 6 2003 08:53:55
	10	MTGUI_0-23T01 B3327
	11	D:\Program Files\Indramat\MTGUI\
	12	FI: 07V00 DLL-Mode: 07.20 Build 3124 Mar 6 2003 08:53:55
	13	347.15.4.11
	14	771
	15	515
	16	78
	17	Platform: 02V01 Build: 3214

Note: Refer to FI command "PHD" of the MPCX device group for working with absolute paths.

11.19 SERCOS Parameters: SPA

MSYX device group

Designation **SPA** **S**ERCOS **PA**rameter

Explanation A SERCOS drive parameter is output or written. Each parameter consists of 7 elements, whereby any combination of elements can be selected by element coding.

FI command

BR_SPA1_(1)_(2)_(3)	**(Single Read)**
BC_SPA1_(1)_(2)_(3)	**(Cyclic Read)**
BB_SPA1_(1)_(2)_(3)	**(Break Cyclic Read)**
BW_SPA1_(1)_(2)_(3)	**(Single Write)**
(1) = Drive address	[0...254]
(2) = Parameter No.	in format X-Y-ZZZZ
(3) = Element coding	[standard or advanced format]

Parameter No.

Format X-Y-ZZZZ	Value Range
X	S = standard data P = product data Y = SERCANS parameter
Y	[0..00.15] = parameter record
Z	[0...4095] = data block no.

Element Coding Element coding in standard format allows individual elements, such as the operating date, to be requested. If several elements are to be read out in one request, then the element coding can be OR'd in advanced format, e.g. operating date (0x40) and unit (0x08) produces OR'd (0x48) → 48

The advanced format 0x80 has priority over 0x40.

Element	Standard Format	Advanced Format	Format:	Example
Data status	S:	01H	Hexadecimal word	0x0000
Name	N	02H	String	NC cycle time (TNcyc)
Attribute	A	04H	Hexadecimal double word	0x60110001
Unit	U	08H	String	µs
Min. input value	L	10H	Decimal word	2000
Max. input value	H	20H	Decimal word	20000
Operating date	D	40H	(see Displaying the Operating Date	
Operating date, when no list		80H		

Displaying the Operating Date The display of the operating date depends on the parameter number requested.

Decimal Decimal values are given as floating points, e.g. 1.5. Leading spaces, zeros, plus and minus signs as well as trailing spaces are allowed.

Hexadecimal Hexadecimal values are displayed by "0x...", e.g. 0x80. Up to a maximum of eight positions are allowed. Leading or trailing spaces are allowed. Leading additional zeros or plus and minus signs are not allowed.

Binary (max. 32 characters)

Leading or trailing spaces are allowed. The decimal point serves as separator:

e.g., 1111.0000.1010.1100.1111.0000.1010.1100

Note: Leading additional zeros or plus and minus signs are not allowed.

Lists of Variable Length

Lists always begin with two decimal numbers for the actual length and maximum length of the list. The length specification refers to the length of the list in the drive and therefore designates the number of bytes for storage (storage bytes). The number of elements in the list can be calculated using the attribute. The list elements are displayed according to the attribute. All parts of the list are separated from each other by a line feed ("\n").

Example:

Parameter S-0-0017, IDN list of all parameters

"400\n400\nS-0-0001\nS-0-0002\n..."

ASCII List

ASCII lists are a special form of variable length lists. The individual string characters are not separated by a line feed. When displaying the lists, a distinction is made between standard format and advanced format. In standard format, only the character string is returned, whereas in advanced format the actual length and the maximum length of the list (string) is also transmitted.

Example:

Parameter S-0-0030, operation date

Standard format:	"DKC2.1-SSE-01V09"
Advanced format:	"16\n16\nDKC2.1-SSE-01V09"

Note: When requesting SERCANS parameters the drive address can be anywhere within the range [0..254].

Response Structure

The following table shows the general structure of the response to the FI command "SPA1". Line 1 is output both when reading and when writing. Additional lines are only output when reading depending on the element coding.

Note: If the element coding has been requested in standard format then the first line is not applicable.

Note: Line 1 is a status line that either contains SERCOS / SERCANS errors or displays the successful processing of the FI command. If the command has been processed successfully, then columns 1 and 3 contain the value [0x0000].

In the first line, column 2 or column 4, the number of the drive is output that reports the SERCOS error or the global SERCANS error. Not all global SERCANS errors can be directly assigned to a drive. In this case, the single result "Drive No." is set to 0x0000.

Line	Column 1	Column 2	Column 3	Column 4
1	<SERCOS error>	<Drive no. SERCOS error>	<Global SERCANS error>	<Drive No. Global SERCANS error>
2	Read: Element corresponding to the element coding.			
...	...			
n	Reading: (n-1). Element corresponding to the element coding.			

Example SPA1 / read Read parameter S-0-0003 of the 3rd drive (element coding 0x48)

FI command	**00_BR_SPA1_3_S-0-0003_48**			
Answer				
Line	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	0x0000	0x0000	0x0000	0x0000
2	µs			
3	2000			

Example SPA1 / write Write the ID number P-0-0037 in parameter S-0-0305 of the 3rd drive (element coding 0x40).

Technical background:

- Real time status bit 1 is to be assigned the trigger status word of the oscilloscope function of a DIAX04 drive.

FI command	**Value to be written: : P-0-0037** **00_BW_SPA1_3_S-0-0305_40**			
Answer				
Line	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	0x0000	0x0003	0x0000	0x0000

Reference to Literature See chapter entitled "Literature" [41].

See chapter entitled "Literature" [46].

11.20 Active SERCOS Phase Switch-Over: SPH

MSYX device group

Designation **SPH** **S**ERCOS **PH**ase

Explanation All drives within a SERCOS ring are in the same communication phase. The phase status can be read-out or changed by this command.

FI command

BR_SPH	**(Single Read)**
BC_SPH	**(Single Write)**
BB_SPH	**(Break Cyclic Read)**
BW_SPH	**(Single Write)**

Value to be written/ Result The phase conditions allowed are shown by the numbers [0...4].

Response Structure The following table shows the general structure of the response to the FI command "SPH". In the first line, column 2 or column 4, the number of the drive is output that reports the SERCOS error or the current system error. Not all current system errors can be directly allocated to a drive. In this case, the single result "Drive No." is set to 0x0000.

Line	Column 1	Column 2	Column 3	Column 4
1	<SERCOS error>	<Drive No. SERCOS error>	<Current system error>	<Drive no. that has caused the current system error>
2	Read: current phase Write: former phase			

Example BR_SPH Read SERCOS Phase Read the active phase of the synax control at device address 00.

FI command	**00_BR_SPH**			
		Answer		
Line	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	0x0000	0x0000	0x0000	0x0000
2	2			

Example BW_SPH Write SERCOS Phase Switch-over the synax control (write) after phase 4; phase 2 is active.

FI command	**Value to be written: 4** **00_BW_SPH**			
		Answer		
Line	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	0x0000	0x0000	0x0000	0x0000
2	2			

Reference to Literature See chapter entitled "Literature" [42].

12 FI Commands - MWYX Device Group (SYNAXISP200)

The FI Commands described in this chapter are valid for the MWYX device group. In this device group, the following types are listed as well as possible device addresses:

Group	Accompanying Types	Address
MWYX	SYNAXISP200-P-G2, SYNAXISP200-R-G2	[00...63]

Note: Please note that the device address must be set before the respective FI command, e.g. 00_BR_ASM1 (refer also here to the chapter 2.1 "Elements of the FI Command").

12.1 Active Diagnosis Window ADW

MWYX device group

Designation **ADW** Active **D**iagnosis **W**indow

Explanation Indicates the window types for which data is required.

For improved performance and with some diagnosis windows, not all the data is called up each time a diagnosis is performed, but only when the data is actually required.

Through this FI command, the diagnosis server can be informed that the data of the respective window type is required. This command has to be issued at least every 3 seconds while the data is required. If this command is not issued any more, calling-up of all data will stop.

FI command Indicates the window types for which data is required.

BW_ADW1_(1){_(2)}	**(Single Write)**
(1) = Type of diagnosis window	[1 = CNC error, 2 = sequence errors, 3 = general errors, 4 = messages, 10 = start requirements, 11 = warnings, 12 = setup diagnosis]
(2) = Module number	[1...99] ! only for window type 1 -4!

Example ADW1 Call up data for CNC error in controller 0 module 1.

FI command	**00_BW_ADW1_1_1**

12.2 Determining the Actual (Current) System Error: ASE

MSYX device group

Designation **ASE** Actual **S**ystem **E**rror

Explanation The current system error is read out, whereby the answer 0x0000 indicates that the Synax device is functioning correctly.

FI command **CR_ASE** **(Single Read)**

Response Structure The following table shows the general structure of the response to the FI command "ASE". In line 1, column 4, the number of the drive is output that reports the current system error. Not all current system errors can be directly allocated to a drive. In this case, the single result "Drive No." is set to 0x0000.

Line 1	Column 1	...	Column 4

Value Range/Meaning of Columns

1 = 0x0000

2 = 0x0000

3 = Current system error

4 = Drive No.

Example ASE Reading the current system error returns LWL ring interrupted.

FI command		00_CR_ASE
Line	**Column**	**Answer**
1	1	0x0000
	2	0x0000
	3	0x8009
	4	0x0000

Reference to Literature See chapter entitled "Literature" [42].

12.3 Active System Error Messages: ASM

MWYX device groups

Designation **ASM** Active **S**ystem **M**essages

Explanation The active device information is output (system errors, device statuses) that affect the functioning of the entire electrical device. Depending on the FI command, the device address, device name, message number, type of message, short text and reference text are all output.

FI command Output of the currently pending device information (system errors, device statuses) of all active devices from the MWYX device group.

BR_ASM2	**(Single Read)**
BC_ASM2	**(Cyclic Read)**
BB_ASM2	**(Break Cyclic Read)**

Response Structure The following table shows the general structure of the response to the FI command "ASM2". Each line consists of 7 columns for the device address, device name, message number, message status, short text and indication of whether there is an reference text for this device information.

Line 1..n	Column 1	...	Column 7

Value Range/Meaning of Columns

1 =	Device address	[00...15]
2 =	Device name	[max. 32 ASCII characters]
3 =	Message number	[0...150]
4 =	Type of message	[F = fault/error, D = diagnosis]
5 =	Short text	[max. 54 ASCII characters]
6 =	Reference text	[x= exists, -- = does not exist]
7 =	2 bytes of additional information for the message number	is required to resolve the information "@" (see ASM5)

Example ASM2 Read the current system error messages of device address 01.

Assumption: The following three devices are defined:

- Device address 01
- Device address 07 and
- Device address 10

FI-command		**01_BR_ASM2**
Line	**Column**	**Response**
1	1	01
	2	Drill center
	3	71
	4	F
	5	PLC battery voltage too low.
	6	X
	7	0

Reference to Literature see chapter entitled Literature [13].

12.4 Creating MI Import Data: CMD

MWYX device group

Designation **CMD** **C**reate **MI** Import **D**ata

Explanation Creation of the data the Message Integrator requires for data import.

FI command Creation of the data of all ProVi messages.

BR_CMD1 **(Single Read)**

Response Structure The command does not return any answer. If no error is signaled, the respective files have been generated.

FI Command Creation of the data of a certain ProVi message type.

BR_CMD2_(1){_(2)} **(Single Read)**

(1) = Message type [1 = error, 2 = messages, 10 = warnings,
11 = start requirements,
12 = setup diagnosis]

(2) = Module number [1...99] ! only for message type 1 -2!

Response Structure The command does not return any answer. If no error is signaled, the respective files have been generated.

FI command Creation of the data of all step chain messages.

BR_CMD3 **(Single Read)**

Response Structure The command does not return any answer. If no error is signaled, the respective files have been generated.

FI command Creation of the data of a certain step chain register.

BR_CMD4!(1) **(Single Read)**

(1) = Register name [ASCII character]

Note: The separator "!" is used in this command.

Response Structure The command does not return any answer. If no error is signaled, the respective files have been generated.

12.5 Trigger Control Reset: CRT

MWYX device groups

Designation **CRT** **Control-Reset**

Explanation The control reset allows the selected device to be reset after a system error. If there is no system error at the selected device then the job is ignored.

CAUTION

Resetting causes a complete reinitialization of device.
A temporary failure of communication occurred during initialization..

FI command **BW_CRT** **(Single Write)**

Value to be written Trigger reset 0

Note: The value to be written is passed to the "acValue" parameter in the "DataTransfer" routine.

Response Structure The return value of the "DataTransfer" routine is [0] if the write procedure has been successfully completed. In the event of an error, more information in the form of a general error result line can be requested by the routine "ReadGroupItem" (refer here to chapter 8, "Error Codes" and "General Error Result Line").

Example CRT Trigger a control reset on the selected device.

FI command	**Value to be written: 0** **00_BW_CRT**
Value to be written	0

Reference to Literature See chapter entitled "Literature" [26].

12.6 Deleting the Actual (Current) System Error: CSE

MSYX device group

Designation **CSE** **C**lear **S**ystem **E**rror

Explanation An error reported by the Synax device is deleted again.

FI command **CW_CSE** **(Single Write)**

Value to be written The contents of the value parameter is not evaluated.

Response Structure The following table shows the general structure of the response to the FI command "CSE". In line 1, column 4, the number of the drive is output that reports the current system error. Not all current system errors can be directly allocated to a drive. In this case, the single result "Drive No." is set to 0x0000.

Line 1	Column 1	...	Column 4

Value Range/Meaning of Columns

1 = 0x0000

2 = 0x0000

3 = Actual (current) system error

4 = Drive No.

Example CSE Deleting the actual (current) system error:

FI command		00_CW_CSE
Line	**Column**	**Answer**
1	1	0x0000
	2	0x0000
	3	0x0000
	4	0x0000

Reference to Literature see chapter entitled "Literature" [45].

12.7 Reading Device Component Information: DCI

MWYX device group

Designation **DCI** **D**evice **C**omponent **I**nformation

Explanation The current device component information is read out of the device. From the device component information, the user is provided with information on the components the addressed device is equipped with, and the firmware each component contains. The command will not file if no access to firmware is possible (e.g. while the device is in monitor mode). Instead, the failed access is reported through the firmware access status.

FI command Read the device component information.

BR_DCI1 **(Single Read)**

Response Structure The following table shows the general structure of the response to the FI command "BR_DCI1". For each device component available in the device, one line is returned. Each line consists of 11 columns.

FI command		**00_BR_DCI1**
Line	**Column**	**Response**
1	1	PCB type
	2	Configured component type
	3	Detected component type
	4	Firmware access status; i.e. has an error occurred accessing the firmware, Yes/No? Valid range of values [YES/NO] In case of an error, the error cause can be defined from one the two following columns.
	5	Error class on accessing firmware identification: (see Error Class Definition under General Error Result Line)
	6	Error code on accessing firmware identification: (see Error Code Definition under Error Codes)
	7	Firmware identification
	8	Firmware version
	9	Firmware release
	10	Is the component address in column 11 a sub-address, Yes/No? Valid range of values [YES/NO]
	11	Component address
2	1	PCB type
	..	..
	11	Component address

Example DCI1 At device address 00, read out the current device component information.

FI command		**00_BR_DCI1**
Line	**Column**	**Response**
1	1	CPU
	2	MTC-P
	3	MTC-P
	4	NO
	5	0
	6	0
	7	CPU06/0006-23V10
	8	23
	9	10
	10	NO
	11	1
2	1	PLC
	2	MTS-P01.2
	3	MTS-P01.2
	4	NO
	5	0
	6	0
	7	PLC06S-M05-06V05
	8	06
	9	05
	10	NO
	11	3
3	1	APR
	2	--
	3	APR-P
	4	NO
	5	0
	6	0
	7	APR06/0003-23T06
	8	23
	9	06
	10	NO
	11	4

Example DCI1 while booting is blocked

While booting is blocked (i.e. while the device is in monitor mode), read out the current device component information at device address 00.

FI command		**00_BR_DCI1**
Line	**Column**	**Response**
1	1	CPU
	2	MTC-P
	3	MTC-P
	4	YES
	5	1
	6	2082
	7	--
	8	--
	9	--
	10	NO
	11	1
2	1	PLC
	2	MTS-P01.2
	3	MTS-P01.2
	4	YES
	5	1
	6	2082
	7	--
	8	--
	9	--
	10	NO
	11	3
3	1	APR
	2	--
	3	APR-P
	4	YES
	5	1
	6	2082
	7	--
	8	--
	9	--
	10	NO
	11	4

12.8 Setting the Communication Timeout Time DCT

MWYX device groups

Designation **DCT** Device **C**ommunication **T**imeout

Explanation By means of this command, the timeout time for the selected device is set dynamically (timeout time in ms).

FI command **BW_DCT1_(1)** **(Single Write)**

(1) = requested timeout time in ms

Response Structure The response to the "DCT1" FI command consists of one line with one column.

Line 1	Column 1

Value Range/Meaning of Columns 1 = Status message (P_ACK) (P_ACK)

Example DCT1 For the device 00, the timeout time is set 1500 ms.

FI command		**00_BW_DCT1_1500**
Line	**Column**	**Answer**
1	1	(P_ACK)

FI command With this command, the timeout time for the selected device can be reset to default value.

BW_DCT2 **(Single Write)**

Response Structure The response to the "DCT2" FI command consists of one line with one column.

Line 1	Column 1

Value Range/Meaning of Columns 1 = Status message (P_ACK) (P_ACK)

Example DCP2 For the device 00, the timeout time is reset to the default value.

FI command		**00_BW_DCT2**
Line	**Column**	**Answer**
1	1	(P_ACK)

12.9 Deleting the FI Exclusive Mode: DEM

MWYX device group

Designation **DEM** **D**elete FI **E**xclusive **M**ode

Explanation This command is used to deactivate FI Exclusive mode for the selected device address.

FI Exclusive mode: In this mode, **ALL** the processes logged in at the FI – **with the exception** of the process issuing the SEM command – are blocked from data communication with this device address. This mode is used for example for data communication which must NOT be interrupted by other data requests. However, it is **imperative** that this FI Exclusive mode is deleted once more through the DEM command.

FI command **BW_DEM1** **(Single Write)**

Response Structure The following table shows the general structure of the response to the FI command "BW_DEM1". A line of 1 column is output.

Line 1	Column 1

Value Range/Meaning of Columns 1= Status message (P_ACK) (P_ACK)

Example DEM1 Deactivate the FI Exclusive mode for device address 0. The FI Exclusive mode for device address 0 has previously been activated by the SEM command.

FI command		**00_BW_DEM1**
Line	**Column**	**Response**
1	1	(P_ACK)

12.10 Static/Dynamic Device Information: DIF

MWYX device group

Designation **DIF** **D**evice **I**n**F**ormation

Explanation Static device information and network information is read according to the "IND_DEV.INI" and "FAR_DEV.INI" files.

FI command Reading of the static device information and network information of a selected device.

BR_DIF1	**(Single Read)**
BC_DIF1	**(Cyclic Read)**
BB_DIF1	**(Break Cyclic Read)**

Response Structure The following table shows the general structure of the response to the "DIF1" FI command. The response consists of one line with 24 columns.

Line 1	Column 1	...	Column 24

Value Range/Meaning of Columns

1 =	Local/far device address	[00..63]
2 =	Device name	IND_DEV.INI entry: DeviceName=
3 =	Device type	IND_DEV.INI entry: DeviceType=
4 =	PLC support	IND_DEV.INI entry: PLC=
5 =	Device status	IND_DEV.INI entry: DeviceStatus=
6 =	Assignment of a simulation pair	IND_DEV.INI entry: DeviceAssign=
7 =	Device mode	IND_DEV.INI entry: MtvncMode=
8 =	Communication channel	IND_DEV.INI entry: [CommAddrX]
9 =	Description of the communication channel	IND_DEV.INI entry: CommStr=
10 =	Timeout value	IND_DEV.INI entry: Timeout=
11 =	Device group	(see Chapter 6.1 "Identifier")
12 =	PLC component type	IND_DEV.INI entry: Component type1=
13 =	CNC component type	IND_DEV.INI entry: Component type2=
14 =	Device log	IND_DEV.INI entry: DeviceProtocol=
15 =	Device simulation	IND_DEV.INI entry: DeviceSimulation=
16 =	Not yet assigned	[--]
17 =	Not yet assigned	[--]
18 =	Not yet assigned	[--]
19 =	Not yet assigned	[--]
20 =	Network ON/OFF	[ON = Network active OFF = No network active]
21 =	Network name	Max. 28 ASCII characters
22 =	PC number	[00..99,XX]
23 =	PC name	Max. 255 ASCII characters
24 =	Local device address	[00..63]

Example DIF1 Read the static device information and network information of device 1 while the network is active.

FI commando		**01_BR_DIF1**
Line	**Column**	**Response**
1	1	01
	2	Handling station right side
	3	ISP200-R-G2
	4	YES
	5	ON
	6	NO
	7	OFF
	8	4
	9	V24,COM2,19200,EVEN,RS232,TCOFF
	10	3500
	11	MWMX
	12	MTS-R
	13	NONE
	14	CNC
	15	OFF
	16	--
	17	--
	18	--
	19	--
	20	ON
	21	PC network 1
	22	29
	23	BTV20-STATION-LEFT
	24	01

12.11 Long ID of PLC Data Block: DIS

MWYX device groups

Designation **DIS** Data **I**dentification **S**tring

Explanation Reads the long ID (directory entries) of the PLC program. Included in the directory entries are the number of the entry in the directory, the name, length and date and time of creation and/or details of the last time the respective data record was changed.

FI command

BR_DIS2	**(Single Read)**
BC_DIS2	**(Cyclic Read)**
BB_DIS2	**(Break Cyclic Read)**

Response Structure The following table shows the general structure of the response to the "DIS2" FI command. The response consists of a line with six columns.

Line 1	Column 1	...	Column 6

Value Range/Meaning of Columns

1 =	Number in PLC directory	[01...99]
2 =	Name of the PLC program	[max. 8 ASCII characters]
3 =	Length of the PLC program	[byte]
4 =	Date of creation/last change to PLC program	[DD.MM.YY]
5 =	Time of creation/last change to the PLC program	[HH:MM:SS]
6 =	Date of creation/last change to PLC program	[DD.MM.YYYY]

Note: If there is no valid NC package in the selected NC memory then all columns contain [--] .

Example DIS2 Read the directory entries of the PLC program at address 00.
Assumption:
There is a valid PLC program in the selected device.

FI command		**00_BR_DIS2**
Line	**Column**	**Answer**
1	1	01
	2	KEY1
	3	20018
	4	10.05.99
	5	12:42:00
	6	10.05.1999

Reference to Literature See chapter entitled "Literature" [30].

12.12 Reading the Device Status Information: DSI

MWYX device groups

Designation **DSI** Device Status Information

Explanation This allows the most important device status information to be read out. The following information is returned:

Type of information	Status	Statement
System error information		Yes/No
Mechanism error information		Yes/No
Machine key information		4 Byte HEX
Machine key information	valid	Yes/No
Machine status information		4 Byte HEX
Sercans information		4 Byte HEX
Parameter download	running	Yes/No
PLC download	running	Yes/No
Firmware download	running	Yes/No
Offline/Online information		
Device simulation	switched on	Yes/No
Device status information		ON/OFF
Communication channel defined		Yes/No
PLC components available		Yes/No
Monitor mode	active	Yes/No

FI command Read out device status information for ALL defined devices.

BR_DSI1 **(Single Read)**

BC_DSI1 **(Cyclic Read)**

BB_DSI1 **(Break Cyclic Read)**

Note: The "DSI1" FI command refers to all devices within this device group. Therefore, any valid device address can be indicated in the command line (see example DSI1). The FI device polling mechanism **MUST** be switched on (see system configurator)!

Response Structure The following table shows the general structure of the response to the "DSI1" FI command.

Line 1...n	Column 1	...	Column 11

Value Range/Meaning of Columns

1 =	device address	[00...63]
2 =	System error information	[0 = there is no system error 1 = there is a system error]
3 =	Mechanism error information	[0 = there is no mechanism error 0 = there is a mechanism error
4 =	Machine key information	[4 byte in HEX coding]
5 =	Machine key information valid?	[0 = not valid, 1=valid]
6 =	Machine status information	[4 byte in HEX coding]
7 =	Sercans information	[4 byte in HEX coding]

8 =	Is parameter download active?	[0 = parameter download not running 1 = parameter download running]
9 =	Is PLC download active?	[0 = PLC download not running 1 = PLC download running]
10 =	Is firmware download active?	[0 = PLC download not running 1 = PLC download running]
11 =	Offline/Online information	[0 = device connection interrupted 1 = device connection O.K.]
12 =	Device simulation switched on?	[0 = NO Simulation mode 1 = simulation mode]
13 =	Current device status information	[0 = Device-Status=OFF 1 = Device-Status=ON]
14 =	Communication channel defined?	[0 = NO communication channel 1 = Communication channel defined]
15 =	PLC components available ?	[0 = NO PLC component 1 = PLC component (DOS-Pcl) 2 = PLC component (WIN-Pcl)]
16 =	Monitor mode	[0 = NO monitor mode active 1 = Monitor mode active]

Example DSI1 Read the current device status information.

Assumption:
The following devices addresses are defined:

- Device address 01 (SYNAXISP200-P-G2)
- Device address 03 (SYNAXISP200-R-G2)

FI command		**01_BR_DSI1**
Line	**Column**	**Answer**
1	1	01
	2	0
	3	0
	4	00000000
	5	0
	6	00000000
	7	00000000
	8	0
	9	0
	10	0
	11	1
	12	0
	13	1
	14	1
	15	2
	16	0

2	1	03
	2	1
	3	0
	4	00000000
	5	0
	6	00000000
	7	00000000
	8	0
	9	0
	10	0
	11	1
	12	0
	13	1
	14	1
	15	2
	16	0

FI command Read out device status information for a selected device.

BR_DSI2 **(Single Read)**

BC_DSI2 **(Cyclic Read)**

BB_DSI2 **(Break Cyclic Read)**

Response Structure The following table shows the general structure of the response to the "DSI2" FI command.

Line 1...n	Column 1	...	Column 11

Value Range/Meaning of Columns

1 =	device address	[00...63]
2 =	System error information	[0 = there is no system error 1 = there is a system error]
3 =	Mechanism error information	[0 = there is no mechanism error 1 = there is a mechanism error]
4 =	Machine key information	[4 byte in HEX coding]
5 =	Is machine key information valid?	[0 = not valid, 1=valid]
6 =	Machine status information	[4 byte in HEX coding]
7 =	Sercans information	[4 byte in HEX coding]
8 =	Is parameter download active?	[0 = parameter download not running 1 = parameter download running]
9 =	Is PLC download active?	[0 = PLC download not running 1 = PLC download running]
10 =	Is firmware download active?	[0 = PLC download not running 1 = PLC download running]
11 =	Offline/Online information	[0 = device connection interrupted 1 = device connection O.K.]
12 =	Device simulation switched on?	[0 = NO Simulation mode 1 = simulation mode]

13 =	Current device status information	[0 = Device status=OFF 1 = Device status=ON]
14 =	Communication channel defined?	[0 = NO communication channel 1 = Communication channel defined]
15 =	PLC components available ?	[0 = NO PLC component 1 = PLC component (DOS-Pcl) 2 = PLC component (WIN-Pcl)]
16 =	Monitor mode	[0 = NO monitor mode active 1 = Monitor mode active]

Example DSI2 Read the current device status information for the selected device.

FI command		**00_BR_DSI2**
Line	**Column**	**Answer**
1	1	00
	2	0
	3	0
	4	00000000
	5	0
	6	00000000
	7	00000000
	8	0
	9	0
	10	0
	11	1
	12	0
	13	1
	14	1
	15	2
	16	0

12.13 Device Type and Accompanying Components: DTY

MWYX device groups

Designation **DTY** **D**evice **TY**pe

Explanation The device type and the accompanying components of the selected device address are output.

FI command

BR_DTY1	**(Single Read)**
BC_DTY1	**(Cyclic Read)**

Response Structure The following table shows the general structure of the response to the "DTY1" FI command. A line with three columns is output for the device type, as well as the names of the first device component and the name of the second device component.

Line 1	Column 1	...	Column 3

Value Range/Meaning of Columns

1 =	Device Type	(see Chapter 6.1 "Elements of the FI Command" and "Identifier")
2 =	Component type1	IND_DEV.INI-Entry: Componenttype1=
3 =	Component type 2	IND_DEV.INI-Entry: Componenttype2=

Example DTY1 Output the device type and the accompanying components of device address 00.

FI command	**00_BR_DTY1**		
Answer			
Line	**Column 1**	**Column 2**	**Column 3**
1	ISP200-P	MTS-P	NONE

12.14 Diagnosis Window Data: DWD

MWYX device groups

Designation **DWD** **D**iagnosis **W**indow **D**ata

Explanation Diagnostic messages are output. The data is edited in such a way that they can be output directly in the diagnosis overview, i.e., where applicable, different types of diagnosis, such as ProVi and a process report, are returned simultaneously.

FI command Output all diagnostic messages.

For reasons of optimization, not all data is called up. Accordingly, the Diagnosis server must be informed that the data is required (see ADW).

BR_DWD1_(1){_(2)}	**(Single Read)**
BC_DWD1_(1){_(2)}	**(Cyclic Read)**
(1) = Type of diagnosis window	[1 = CNC error, 2 = sequence errors, 3 = general errors, 4 = messages, 10 = start preconditions, 11 = warnings, 12 = setup diagnosis]
(2) = Module number	[1...99] ! only for window type 1 -4 !

Output first diagnostic messages.

BR_DWD2_(1){_(2)}	**(Single Read)**
BC_DWD2_(1){_(2)}	**(Cyclic Read)**
(1) = Type of diagnosis window	[1 = CNC error, 2 = sequence errors, 3 = general errors, 4 = messages, 10 = start preconditions, 11 = warnings, 12 = setup diagnosis]
(2) = Module number	[1...99] ! only for window type 1 -4 !

Response Structure

The following table shows the general structure of the "DWD1" and "DWD2" FI commands. The number of lines depends on the number of messages pending. Different columns are valid according to the type of diagnosis.

If there are no messages, the number of lines is 0.

Line 1...n	Column 1	...	Column 14

Meaning of the Columns

1 =	Message text	[ASCII characters]
2 =	Time stamp day	[mm.dd.yyyy]
3 =	Time stamp hour	[hh:mm:ss]
4 =	Reference text available	[YES, NO]
5 =	Type of diagnosis	[1 = ProVi, 2 = SFC, 3 = MTC-NC, 4 = MTA-NC]
6 =	Message number	[ASCII characters]
7 =	Message ID	[ASCII characters] (DWORD, decimal) (ProVi)
8 =	Mechanism number	[0..31] (MTC-NC) [0] (MTA-NC)
9 =	2 byte additional information	[ASCII characters] (MTC NC)
10 =	Message group	[1...9999] (MTA-NC)
11 =	SFC entity name	[ASCII characters]
12 =	NC note	[ASCII characters] (MTC NC)
13 =	Analysis of criteria available	[YES, NO] (ProVi, SFC)
14 =	Message HTML file	[ASCII characters] (ProVi, MTC-NC)

Example DWD1

All diagnostic messages from module 3 in control unit 0.

There are two messages.

FI command		**00_BR_DWD1_4_3**
Line	**Column**	**Answer**
1	1	Guard not closed
	2	01.27.2000
	3	14:56:32
	4	YES
	5	1
	6	34
	7	43923028
	8	
	9	
	10	
	11	
	12	
	13	YES
	14	
2	1	Station waiting until tool-change command has ended.
	2	01.27.2000
	3	15:03:10
	4	YES
	5	3
	6	79
	7	
	8	1
	9	0
	10	
	11	
	12	
	13	NO
	14	

Example DWD2 First diagnostic message from module 3 in control unit 0.

There are two messages.

FI command		**00_BR_DWD2_4_3**
Line	**Column**	**Answer**
1	1	Guard not closed
	2	01.27.2000
	3	14:56:32
	4	YES
	5	1
	6	34
	7	43923028
	8	
	9	
	10	
	11	
	12	
	13	YES
	14	

Reference to Literature See chapter entitled "Literature" [13].

12.15 Component Information for a System Error: ECI

MWYX device group

Designation **ECI** **E**rror **C**omponent **I**nformation

Explanation When a system error is present, this command is used to define which controller component is causing the error. Here, PLC components are differentiated from general controller components (e.g. CNC, Synax, MTA, …).

FI command **BR_ECI1** **(Single Read)**

Response Structure The response to the "ECI1" FI command consists of one line with two columns.

Line 1	**Column 1**	**Column 2**

Value Range/Meaning of Columns

1 =	PLC component information	[0 = There is NO system error at the PLC) 1 = There is a system error at the PLC]
2 =	General information on controller components	[0 = There is NO system error at the general control component [0 = There is a system error at the general control component; the following applies: 2 = CNC component 3 = SYNAX component 4 = VISUAL-MOTION component 5 = MTA component

6 = TRANS 200 component]

Example ECI1 There is a system error present in device 0 (ISP200-P-G2)) which is caused by the PLC component.

FI command		**00_BR_ECI1**
Line	**Column**	**Response**
1	1	1
1	2	0

12.16 Existing Errors: EDE

MWYX device groups

Designation **EDE** **E**xisting **D**iagnosis **E**rror

Explanation Whether or not errors exist in a control unit or in a module is queried. These can be step chain errors, NC errors, MTA 200 errors or ProVi errors.

FI command Query whether there are errors in this control unit.

BR_EDE1 **(Single Read)**

BC_EDE1 **(Cyclic Read)**

Response Structure The following table shows the general structure of the "EDE1" FI command.

Line 1	**Column 1**

Meaning of the Columns 1 = Error exists [YES, NO]

Example EDE1 Do errors exist in control unit 0?

FI command		**00_BR_EDE1**
Line	**Column**	**Answer**
1	1	YES

FI command Query whether or not errors exist in a specific module.

BR_EDE2_(1) **(Single Read)**

BC_EDE2_(1) **(Cyclic Read)**

(1) = Module number [1...99]

Response Structure The following table shows the general structure of the "EDE2" FI command.

Line 1	**Column 1**

Meaning of the Columns 1 = Error exists [YES, NO]

Example EDE2 Do errors exist in module 1 on control unit 0?

FI command		**00_BR_EDE2_2**
Line	**Column**	**Answer**
1	1	NO

12.17 Existing Diagnosis Window: EDW

MWYX device groups

Designation **EDW** **E**xisting **D**iagnosis **W**indow

Explanation Which types of diagnosis window exist is queried.

FI command Output all types of diagnosis window.

BR_EDW1 **(Single Read)**

Response Structure The following table shows the general structure of the "EDW1" FI command. The number of lines depends on the number of types of window existing.

Line 0...n	Column 1	Column 2

Meaning of the Columns

1 = Type of diagnosis window	[1 = CNC error, 2 = sequence errors, 3 = general errors, 4 = messages, 10 = start requirements, 11 = warnings, 12 = setup diagnosis]
2 = Module number	[ASCII characters] 0 = Diagnosis window type does not belong to any module

Example EDW1 All types of diagnosis window in control unit 0.

There are three diagnosis windows.

FI command		**00_BR_EDW1**
Line	**Column**	**Answer**
1	1	10
	2	0
2	1	1
	2	3
3	1	2
	2	3

FI command Output all diagnosis window types for a module.

BR_EDW2_(1) **(Single Read)**

(1) = Module number [1...99]

Response Structure The following table shows the general structure of the "EDW2" FI command. The number of lines depends on the number of types of window existing.

Line 0...n	Column 1	Column 2

Meaning of the Columns

1 = Type of diagnosis window	[1 = CNC error, 2 = sequence errors, 3 = general errors, 4 = messages]
2 = Module number	[ASCII characters] 0 = Diagnosis window type does not belong to any module

Example EDW2

All types of diagnosis window in Module 3, Control unit 0.

There are two diagnosis windows.

FI command		00_BR_EDW2_3
Line	**Column**	**Answer**
1	1	1
	2	3
2	1	2
	2	3

FI command

Query a specific type of diagnosis window.

BR_EDW3_(1){_(2)}	**(Single Read)**
(1) = Type of diagnosis window	[1 = CNC error, 2 = sequence errors, 3 = general errors, 4 = messages, 10 = start requirements, 11 = warnings, 12 = setup diagnosis]
(2) = Module number	[1...99] ! only for window type 1 -4 !

Response Structure

The following table shows the general structure of the "EDW3" FI command.

Line 1	Column 1

Meaning of the Columns

1 = Type of diagnosis window exists	[YES, NO]

Example EDW3

Query whether or not an NC error window exists in module 3, control unit 0.

FI command		00_BR_EDW3_1_3
Line	**Column**	**Answer**
1	1	YES

Reference to Literature

See chapter literature [13].

12.18 Existing PLC Diagnoses: EPD

MWYX device groups

Designation

EPD **E**xisting **P**LC **D**iagnosis

Explanation

Which PLC diagnostic types exist is queried. Depending on the FI command, specific types are queried or else the diagnostic types for a device or a module are output together.

FI command

Query which PLC diagnostic types are available on a control unit.

BR_EPD1 **(Single Read)**

Response Structure

The following table shows the general structure of the "EPD1" FI command.

Line 1	Column 1-3

Meaning of the Columns

1 = Start requirement exists	[YES, NO]
2 = Warning exists	[YES, NO]
3 = Setup diagnosis exists	[YES, NO]

Example EPD1

Query PLC diagnostic types in control unit 0.

FI command		**00_BR_EPD1**
Line	**Column**	**Answer**
1	1	YES
	2	NO
	3	YES

FI command

Query which PLC diagnostic types are available in a module.

BR_EPD2_(1) **(Single Read)**

(1) = Module number [1...99]

Response Structure

The following table shows the general structure of the "EPD2" FI command.

Line 1	**Column 1-3**

Meaning of the Columns

1 = Messages exist [YES, NO]

2 = Errors exist [YES, NO]

3 = Step chains exist [YES, NO]

Example EPD2

Query the PLC diagnostic types in Module 2 on Control unit 0.

FI command		**00_BR_EPD2_2**
Line	**Column**	**Answer**
1	1	NO
	2	YES
	3	YES

FI command

Query a specific PLC diagnostic type.

BR_EPD3_(1){_(2)} **(Single Read)**

(1) = Message type [1 = error, 2 = messages, 3 = SFC,
10 = warnings, 11 = start requirements,
12 = setup diagnosis]

(2) = Module number [1...99] ! only for message type 1 -3!

Response Structure

The following table shows the general structure of the "EPD3" FI command.

Line 1	**Column 1**

Meaning of the Columns

1 = Diagnosis type exists [YES, NO]

Example EPD3

Are there any messages in module 4 in control unit 0?

FI command		**00_BR_EPD3_2_4**
Line	**Column**	**Answer**
1	1	YES

12.19 Existing ProVi Types: EPT

MWYX device groups

Designation **EPT** **E**xisting **P**roVi **T**ypes

Explanation Which ProVi types are programmed in the current PLC program is queried. The data is returned in a suitable form for the message texts of the small control panels. There is no need to define modules in Moduldef.ini.

FI command Output all ProVi types.

BR_EPT1 **(Single Read)**

Response Structure The following table shows the general structure of the "EPT1" FI command. The number of lines depends on the number of ProVi types existing.

Line 0...n	Column 1	Column 2

Meaning of the Columns

1 = Type [11 = error, 12 = messages,
20 = start requirements,
21 = warnings, 22 = setup diagnosis]

2 = Index [ASCII characters]

Example EPT1 All ProVi types in control unit 0.

There are three diagnosis windows.

FI command		**00_BR_EPT1**
Line	**Column**	**Answer**
1	1	20
	2	0
2	1	11
	2	3
3	1	12
	1	3

12.20 Error Status: EST

MWYX device groups

Designation **EST** Error **ST**ate

Explanation Queries the error state of a variable.

FI command Query the frozen error state of a variable.

BR_EST1!(1)!(2)	**(Single Read)**
BC_EST1!(1)!(2)	**(Cyclic Read)**
(1) = Error ID	[ASCII characters] (DWORD, decimal)
(2) = Variable name	[ASCII characters]

Note: The separator "!" is used in this command.

Response Structure

The following table shows the general structure of the "EXD1" FI command.

Line 1	Column 1

Meaning of the Columns

1 = Error state

WinPCL - Example EST

Read the value of WinPCL variable "IB_EXT24" in WinPCL program "Prog", at device address 00.

Exception:
The WinPCL variable "IB_EXT24" is declared in the WinPCL Program "Prog" as BOOL.

FI command		**00_BR_EST1!5892855!:Prog.IB_EXT24**
Line	**Column**	**Answer**
1	1	1

12.21 Execution Display: EXD

MWYX device groups

Designation

EXD **EX**ecution **D**isplay

Explanation

Information for displaying the execution of a movement is output.

FI command

Query the execution of a step or of an action.

BR_EXD1!(1)!(2)!(3)	**(Single Read)**
BC_EXD1!(1)!(2)!(3)	**(Cyclic Read)**
(1) = SFC entity name	[ASCII characters]
(2) = Step or action name	[ASCII - characters]
(3) = Behaviour of mode	[1 = all modes, 2 = manual mode]

Note: The separator "!" is used in this command.

Response Structure

The following table shows the general structure of the "EXD1" FI command.

Line 1	Column 1

Meaning of the Columns

1 = Execution [1 = can be executed, 0 = cannot be executed]

Example EXD1

Query the execution of the step "open" for the chain "clamp" in control unit 0 for all modes.

FI command		**00_BR_EXD1!Station03A.Clamp!Open!1**
Line	**Column**	**Answer**
1	1	1

FI command

Query whether the condition analysis (control image) of a step chain is enabled.

BR_EXD2!(1)	**(Single Read)**
(1) = SFC entity name	[ASCII characters]

Note: The separator "!" is used in this command.

Response Structure

The following table shows the general structure of the "EXD2" FI command.

Line 1	Column 1

Meaning of the Columns

1 = Enabled [1 = enabled, 0 = not enabled]

Example EXD2

Query whether the condition analysis of the "clamp" chain has been enabled.

FI command		00_BR_EXD2!Station03A.Clamp
Line	**Column**	**Answer**
1	1	1

12.22 Process Activated Language: LNG

MWYX device group

Designation

LNG Activated **L**a**NG**uage

Explanation

The country code of the language activated for the selected device address is output.

FI command

BR_LNG **(Single Read)**

Response Structure

The response to the FI command "LNG" consists of one line with one column for the country code of the activated language

Value Range of the Column

1 = Country code of the activated language [2 ASCII characters]

Country code	Language
EN	English (US/GB)
DE	German
ES	Spanish
IT	Italian
FR	French
HU	Hungarian
PT	Portuguese
SE	Swedish
CS	Czech

Example LNG

Read the country code of the language activated at device address 00.

FI command		00_BR_LNG
Line	**Column**	**Response**
1	1	SE

Explanation

This command is used to define the language settings of the selected device.

FI command

BW_LNG_(1) **(Single Read)**

(1) = Language to be set [see the windows language extension, e.g. DE,EN

Response Structure

The following table shows the general structure of the response to the FI command "BW_LNG". A line of 1 column is output.

Line 1	Column 1

Value Range/Meaning of Columns 1 = Status message (P_ACK) (P_ACK)

Example LNG Set the language setting of the device 0 to English.

FI command		**00_BW_LNG_EN**
Line	**Column**	**Response**
1	1	(P_ACK)

12.23 Read Reference Name of a PLC Variable: MAR

MWYX device groups

Designation **MAR** **M**ap **A**bsolute PCL-**R**eference

PLC Explanation The absolute reference name of a symbolic PLC variable is read out.

FI command Read the absolute reference name of a PLC variable.

BR_MAR_(1) **(Single Read)**

BC_MAR_(1) **(Cyclic Read)**

(1) = Identifier of the PLC variable

PLC – Example MAR Read the absolute reference name of the PLC variable with the identifier "abref" at device address 00.

<u>Assumption:</u>
The PLC variable with the identifier "abref" is of the type "INTEGER".

FI command		**00_BR_MAR_abref**
Line	**Column**	**Answer**
1	1	%M100.0

WinPlc Explanation The absolute reference name of a symbolic WinPlc PLC variable with program entity is read out.

FI command Read the absolute reference name of a WinPlc PLC variable.

BR_MAR1_(1) **(Single Read)**

BC_MAR_(1) **(Cyclic Read)**

(1) = Identifier of the PLC variable

Win PLC - Example MAR Read the absolute reference name of the Win PLC variable with the identifier "Prog.abref" at device address 00.

<u>Assumption:</u>
The Win PLC variable with the identifier "Prog.abref" is of the type "INTEGER" and is present in Win PLC program entity "Prog".

FI command		**00_BR_MAR1_:Prog.abref**
Line	**Column**	**Answer**
1	1	%M100.0

Reference to Literature See chapter entitled "Literature" [30].

12.24 Device Data of the Module Configuration: MCD

MWYX device groups

Designation **MCD** **M**odule **C**onfiguration: **D**evice Information

Explanation All device data for module configuration is read out from the "Moduldef.ini" file which is stored in the "[LW]:\Program Files\Indramat\MTGUI\CustomData\Resource" directory on standard installation. The device data is in the sections [DeviceAddrX], whereby "X" stands for the configured device addresses.

FI command Read out device data within the module configuration of the MWYX device group.

BR_MCD1	**(Single Read)**
BC_MCD1	**(Cyclic Read)**
BB_MCD1	**(Break Cyclic Read)**

Note: The "MCD1" FI command refers to all devices within the MWYX device group. Therefore, any valid device address can be indicated in the command line (see example MCD1).

Response Structure The following table shows the general structure of the response to the "MCD1" FI command. The number of lines depends on the number of configured devices. Each line consists of four columns for the device address as well as PLC-FB names for providing setup diagnostics, warning messages and start requirements.

Value Range of the Columns

1 = Device address	[0...15]
2 = PLC-FB name for the setup diagnostics	[max. 9 ASCII characters]
3 = PLC-FB name for the warning messages	[max. 9 ASCII characters]
4 = PLC-FB name for the start requirements	[max. 9 ASCII characters]

Example MCD1 Read all device data of the module configuration

<u>Assumption:</u>
The following devices have been configured in the MWYX device group:

- Device address 01 (ISP200-P)
- Device address 03 (ISP200-R)

FI command	**03_BR_MCD1**			
	Answer			
Line	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	01	PVSetup_1	PVWarn_1	PVStart_1
2	03	PVSetup_3	PVWarn_3	PVStart_3

Reference to Literature See chapter entitled "Literature" [36].

12.25 Module Data of the Module Configuration: MCM

MWYX device groups

Designation **MCM** **M**odule **C**onfiguration: **M**odul Information

Explanation All module data of a particular device is read out from the "Moduldef.ini'" file. This file is located in the directory "[LW]:\Program Files\Indramat\MTGUI\CustomData\Resource" and contains the data for all module configurations. The module data is located in sections [DeviceAddrX\ModulY], whereby "X" stands for the device addressed and "Y" for the configured module numbers.

FI command Read-out module data from the module configuration with respect to a device from the MWYX device group.

BR_MCM1 **(Single Read)**

BC_MCM1 **(Cyclic Read)**

BB_MCM1 **(Break Cyclic Read)**

Response Structure The following table shows the general structure of the response to the "MCM1" FI command. The number of lines depends on the number of configured modules of a device. Each line consists of four columns for the module number, module name and PLC-FB names for general module errors and module messages.

Line 1	Column 1	...	Column 4

Value Range of the Columns

1 = Module number	[0...99]
2 = Module name	[max. 28 ASCII characters]
3 = PLC-FB name for general module errors	[max. 9 ASCII characters]
4 = PLC-FB name for module messages	[max. 9 ASCII characters]

Example MCM1 Read the module data of device 03 from the module configuration:

Assumption:
The following modules have been defined:

- Module number 5
- Module number 7

FI command	**03_BR_MCM1**			
Answer				
Line	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	5	Module 5 - Milling	PVError_5	PVMsg_5
2	7	Module 7 - Drilling	PVError_7	PVMsg_7

Reference to Literature See chapter entitled "Literature" [36].

12.26 SFC Data of the Module Configuration: MCS

MWYX device groups

Designation **MCS** **M**odule **C**onfiguration: **S**FC Information

Explanation All SFC data of a particular module is read out from the "Moduldef.ini" file. This file is located in the directory "[LW]:\Program Files\Indramat\MTGUI\CustomData\Resource" and contains the data for all module configurations. The SFC data is located in sections [DeviceAddrX\ModulY\Sfc], whereby "X" stands for the device addressed and "Y" for the selected module number.

FI command Read out the SFC data with respect to the module of a device from the module configuration of the MWYX device group.

BR_MCS1_(1)	**(Single Read)**
BC_MCS1_(1)	**(Cyclic Read)**
BB_MCS1_(1)	**(Break Cyclic Read)**
(1) = Module number	[0...99]

Response Structure The number of lines depends on the number of configured Indrastep step chains for a device. Each line contains a column for the name of the Indrastep step chains.

Value Range of the Column 1 = Name of the Indrastep step chain [format W.X.Y.Z]

Format W.X.Y.Z	Value Range
W	Max. 9 ASCII characters
X	Max. 9 ASCII characters ! OPTIONAL !
Y	Max. 9 ASCII characters ! OPTIONAL !
Z	Max. 9 ASCII characters ! OPTIONAL !

Example MCS1 Read the name of the Indrastep step chain of module 5 from device 03 of the module configuration.

Assumption:
The following Indrastep step chains have been defined:

- ISFB_1
- FB_US.ISFB_3
- FB_US.ISFB_3.SW1
- FB_US.ISFB_3.SW1.ABBA

FI command		**03_BR_MCS1_5**
Line	**Column**	**Answer**
1	1	ISFB_1
2	1	FB_US.ISFB_3
3	1	FB_US.ISFB_3.SW1
4	1	FB_US.ISFB_3.SW1.ABBA

12.27 Downloading Message Texts: MFD

MWYX device groups

Designation **MFD** **M**essage **F**iles **D**ownload

FI command This is used to load the message texts into the device indicated. These message texts are required for small devices. The following message texts are transmitted, depending on the type of device:

- system error messages
- transmission error messages, and/or
- mechanism messages.

Note: This FI command is an FI job!

BW_MFD1 **(Single Write)**

Response Structure The response to the "MFD1" FI command consists of three lines, each with one column. The meaning of the elements is as follows:

Line 1 = Job ID	[01...20] (refer to chapter entitled "FI Commands for the MPCX Device Group", IFJ).
Line 2 = FI command	[string, in accordance to chapter entitled "Elements of the FI Command"]
Line 3 = FI job error code	(see chapter entitled "Error Codes")

Example MFD1 Load message texts into the device with device address 00.

FI command		**00_BW_MFD1**
Line	**Column**	**Answer**
1	1	01
2	1	00_BW_MFD1
3	1	0

12.28 Reading Machine Key Information: MKS

MWYX device groups

Designation **MKS** **M**achine **K**ey **S**tatus

Explanation Current machine key information can be read for the selected device.

FI command Read machine key information for selected device.

BR_MKS **(Single Read)**

BC_MKS **(Cyclic Read)**

BB_MKS **(Break Cyclic Read)**

Response Structure The following table shows the general structure of the response to the FI command "MKS".

Line 1	Column 1	Column 2

Value Range/Meaning of Columns

1 =	Information of machine key	[4 byte in HEX coding]
2 =	Information valid?	[0 = not valid, 1=valid]

Example MKS Read the current machine key information for device 0.

FI command		**00_BR_MKS**
Line	**Column**	**Answer**
1	1	00000000
	2	0

12.29 Writing the GUI-SK Block: MKT

MWYX device groups

Designation **MKT** **M**achine **K**ey **T**able

Explanation Writes the GUI-SK16 block in the PLC.

FI command Write GUI-SK16 block.

BW_MKT1_(1) **(Single Write)**

(1) = List of the 48 PLC variables for writing the GUI-SK16 block.

A distinction is made between the following cases:

1. Clear GUI-SK16 block.
2. Write the GUI-SK16 block with the 48 PLC variables, filling gaps with $SPACE.

Response Structure (P_ACK) is returned following successful transmission.

Line 1	Column 1

Value Range/Meaning of the Columns

1 =	Successfully completed	(P_ACK)

1. Example MKT1

Clear GUI-SK16 block:

FI command		**Value to be written: $EMPTY** **00_BW_MKT1**
Line	**Column**	**Answer**
1	1	(P_ACK)

2. Example MKT1

Write GUI-SK16 block:

FI command		**Value to be written: $EMPTY** **SPSVAR1,SPSVAR2,$SPACE,...** **00_BW_MKT1**
Line	**Column**	**Answer**
1	1	(P_ACK)

FI command

Write the GUI-SK16 block, writing only those PLC variables which are defined in the current PLC program. All undefined PLC variables are automatically replaced by $SPACE and returned as a partial result (column 2).

BW_MKT2_(1)	**(Single Write)**
(1) = List of the 48 PLC variables for writing the GUI-SK16 block.	A distinction is made between the following cases: 1. Clear GUI-SK16 block: BW_MKT2 $EMPTY 2. Write the GUI-SK16 block with the 48 PLC variables, filling gaps with $SPACE: BW_MKT1 SPSVAR1,SPSVAR2, $SPACE,$SPACE,....

Response Structure

After successful transmission, one line with two columns is returned.

Line 1	**Column 1**	**Column 2**

Value Range/Meaning of Columns

1 =	Status report	[0 = at least 1 PLC variable in the current PLC program is NOT defined 1 = ALL PLC variables could be written]
2 =	List of the NON-defined PLC variables in the current PLC program	[-- = ALL PLC variables could be written, or else list of the PLC variables that could not be written.] The individual PLC variables are separated by a comma.

Example MKT2

Write GUI-SK16 component with 48 PLC variables, while the PLC variables SPSVAR11 and SPSVAR12 are NOT defined in the current PLC program.

FI command		**Value to be written:** **SPSVAR1,SPSVAR2,...SPSVAR48** **00_BW_MKT2**
Line	**Column**	**Answer**
1	1	(P_ACK)
	2	--

Extended information

The variables are divided into 3 groups of 16 variables each and have the following meaning:

1.	Variables 1 - 16:	Machine function keys
2.	Variables 17 - 32:	Status pressed
3.	Variables 33 - 48:	Status shining

Note: When, for example, only the first 8 M keys are used, the telegram will contain only these 8 PLC variables. The other 40 variables need not be defined in the transmission parameter.

When certain areas, e.g. of M keys, are left unused, they must be filled up with '$SPACE' up to the next variable.

12.30 Read System Messages: MSG

MWYX device groups

Designation **MSG** **M**e**S**sa**G**e

Explanation Reading of system messages

FI command Message

CC_MSG_(1) **(Cyclic Read)**

(1) = SYS-Message number

Note: Exists only as a cyclic command

Response Structure The response of the FI command 'MSG' consists of the system message data.

Example MSG 00_CC_MSG_64 (64 = MSG_SYSERRGEN)

FI command		**00_CC_MSG_64/3**
Line	**Column**	**Answer**
1	1	00

Restriction The following system messages:

SYS Message	SYS Message number
MSG_PCLUPDBEG	52
MSG_PARUPDBEG	24
MSG_FWAUPDBEG	82

These commands cannot be used with the following programs:

- Bosch Rexroth OPC server
- Bosch Rexroth DDE server

12.31 Reading the Firmware/Monitor-Identification: MTC

MWYX device groups

Designation **MTC** **MT**-**C**NC Slot Software Version

FI command This command is used to read the firmware identification from the various control components (slot numbers).

Note: For the time this FI command is executed, the internal FI communication interlocks (fast timeout monitoring, offline operation, etc.) are switched off.

FI command

BR_MTC_(1) **(Single Read)**

BC_MTC_(1) **(Cyclic Read)**

(1) = Slot number [1=CNC, 2=SIO, 3=PLC, 4=APR1
5=APR2, 6=APR3, 7=APR4]

Response Structure The following table shows the general structure of the response to the FI command "MTC". A line of 1 column is output.

Line 1	Column 1

Value Range/Meaning of Columns 1 = Firmware identification string [max. 16 ASCII characters]

Example MTC Read the firmware identification of slot number 1 (CPU) of device 00.

FI command		**00_BR_MTC_1**
Line	**Column**	**Answer**
1	1	CPU01/0004-20V00

12.32 ProVi Diagnosis Data: PDD

MWYX device groups

Designation **PDD** Provi **D**iagnosis **D**ata

Explanation Data for ProVi criteria analysis is output.

FI command Output of files to indicate the detail in the editor.

BR_PDD1_(1)_(2){_(3)} **(Single Read)**

(1) = Message ID [ASCII characters]

(2) = Message type [1 = error, 2 = messages,
10 = warnings,
11 = start requirements,
12 = setup diagnosis]

(3) = Module number [1...99] ! only for message type 1 -2!

Response Structure The following table shows the general structure of the PDD1 FI command.

Line 1	Column 1	...	Column 5

Meaning of the Columns

1 = POU ID	[ASCII characters]
2 = Detail morpheme	[ASCII characters] (DWORD, decimal)
3 = Error ID	[ASCII characters] (DWORD, decimal)
4 = POE entity name	[ASCII characters]
5 = Nw ID (network ID)	[ASCII characters]

Example PDD1

Indication of data of a ProVi error with ID 43923028 from module 3 in control unit 0.

FI command		**00_BR_PDD1_43923028_1_1**
Line	**Column**	**Answer**
1	1	STATION_1_2
	2	98243823
	3	34985304
	4	Station2.Module3
	5	43493454

FI command

Output the I/O addresses to display a detail.

BR_PDD2_(1)_(2){_(3)}	**(Single Read)**
(1) = Message ID	[ASCII characters]
(2) = Message type	[1 = error, 2 = messages, 10 = warnings, 11 = start requirements, 12 = setup diagnosis]
(3) = Module number	[1...99] ! only for message type 1 -2!

Response Structure

The following table shows the general structure of the PDD2 FI command.

Line 1-n	**Column 1**	**Column 2**

Meaning of the Columns

1 = Variable morpheme	[ASCII characters] (DWORD, decimal)
2 = I/O address	[ASCII characters]

Example PDD2

Query of the I/O addresses of a ProVi error with ID 43923028 from module 3 in control unit 0.

Three variables have an I/O address.

FI command		**00_BR_PDD2_43923028_1_1**
Line	**Column**	**Answer**
1	1	98243823
	2	%I3.2.0
2	1	40923423
	2	%Q23.21.7
3	1	34985304
	2	%I100.3.5

FI command

Determine the multilingual comments for displaying a detail.

BR_PDD3_(1)_(2){_(3)}	**(Single Read)**
(1) = Message ID	[ASCII characters]
(2) = Message type	[1 = error, 2 = messages, 10 = warnings, 11 = start requirements, 12 = setup diagnosis]
(3) = Module number	[1...99] ! only for message type 1 -2!

Response Structure

The following table shows the general structure of the PDD3 FI command.

Line 1-n	Column 1	Column 2

Meaning of the Columns

1 = Comment morpheme	[ASCII characters] (DWORD, decimal)
2 = New comment	[ASCII characters]

Example PDD3

Query of the comments for indication of a ProVi error with ID 43923028 from module 3 in control unit 0.

Two comments are replaced by another text.

FI command		**00_BR_PDD3_43923028_1_1**
Line	**Column**	**Answer**
1	1	98243823
	2	Clamp open
2	1	40923423
	2	Clamp closed

FI command

Query of the status of a certain message

BR_PDD4_(1)_(2){_(3)}	**(Single Read)**
(1) = Message number	[ASCII characters]
(2) = Message type	[1 = error, 2 = messages, 10 = warnings, 11 = start requirements, 12 = setup diagnosis]
(3) = Module number	[1...99] ! only for message type 1 -2!

Response Structure

The following table shows the general structure of the PDD4 FI command.

Line 1-n	Column 1	Column 2

Meaning of the Columns

1 = Message is present	[YES, NO]
2 = Criteria analysis exists	[YES, NO]

Example PDD4

Query of the status of a ProVi error, number 1001 from module 3 in control 0.

This message is not present at the moment, and there is a criteria analysis.

FI command		**00_BR_PDD4_1001_1_1**
Line	**Column**	**Answer**
1	1	NO
	2	YES

FI command

Determination of the MessageID of a certain message

BR_PDD5!(1)!(2)!(3)!(4){!(5)}	**(Single Read)**
(1) = POU entity name	[ASCII characters]
(2) = Nw ID	[ASCII characters]
(3) = Message number	[ASCII characters]
(4) = Message type	[1 = error, 2 = messages, 10 = warnings, 11 = start requirements, 12 = setup diagnosis]
(5) = Module number	[1...99] ! only for message type 1 -2!

Note: The separator "!" is used in this command.

Response Structure

The following table shows the general structure of the PDD5 FI command.

Line 1-n	Column 1	...	Column 3

Meaning of the Columns

1 = Message ID	[ASCII characters] (DWORD, decimal)
2 = Message is present	[YES, NO]
3 = Criteria analysis exists	[YES, NO]

Example PDD5

Determination of the MessageID of a ProVi error, number 1001 from module 25.40 mm control 0.

Assumption:
This message is not present at the moment, and there is a criteria analysis.

FI command		**00_BR_PDD5!Station2.Modul3!43493454!1001!1!1**
Line	**Column**	**Answer**
1	1	240872342
	2	NO
	3	YES

12.33 Reading the Size of the PLC Memory: PMI

MWYX device group

Designation **PMI** **P**lc **M**emory **I**nformation

Explanation The current size of the PLC memory is read out.

FI command **CR_PMI** **(Single Read)**

Response Structure One line with two values in BYTE is output:

1. Total memory
2. Free memory available now.

Line 1	Column 1	Column 2

Example PMI Read the current size of the PLC memory at device address 00.

FI command		**00_CR_PMI**
Line	**Column**	**Answer**
1	1	123456
	2	3210

12.34 Issuing SYS Messages Specific to the PCL: PSM

MWYX device groups

Designation **PSM** **P**CL **S**ys **M**essage

Explanation Issues the most important SYS messages regarding the PCL programming interface – required for remote programming.

Note:
The appropriate device address is passed as the write value.

It allows the following SYS messages to be initiated:

- start of PCL download,
- end of PCL download,
- start of PLC online edit,
- end PLC online edit,
- start of PCL declaration change, and
- end of PCL declaration change.

FI command Issue the most important PCL SYS messages.

BW_PSM1_(1) **(Single Write)**

(1) = Requested SYS message [1= start of PCL download
2= end of PCL download
3= start of PCL online edit
4= end of PCL online edit
5= start of PCL declaration change
6= end of PCL declaration change]

Value to be written
device address

Response Structure

The following table shows the general structure of the response to the FI command "PSM1".

Line 1	Column 1	...	Column 8

Value Range/Meaning of Columns

1 =	Status report	[READY=SYS message has been correctly acknowledged by the WIN32 applications] [ERROR=SYS message has NOT been acknowledged by a WIN32 application within the pre-set time]
2 =	Task name (LogInIf name)	[Task name that has triggered the SYS message]
3 =	SYS message number	[contains the issued SYS message number]
4 =	Acknowledgement time	[contains the pre-set acknowledgement time]
5 =	Reference information	[contains, where applicable, the additional information transferred as a write value]
6 =	Length of reference information	[0 where NO reference information has been transferred]
7 =	Where applicable, LOG channel of the FI that has NOT acknowledged	[-- = acknowledgements have been completed in time or the LOG channel number of the WIN32 application that has NOT acknowledged in time]
8 =	Where applicable, task name that has NOT acknowledged in time	[-- = acknowledgements have been completed in time or the task name that has NOT acknowledged in time]

Example PSM1

Issue the SYS message Beginning PCL Download. The reference information, device address 00, is also transferred as a write value.

FI command		XX_BW_PSM1_1
Line	**Column**	**Answer**
1	1	READY
	2	WINPCL.EXE
	3	14
	4	30000
	5	00
	6	2
	7	--
	8	--

12.35 Edit PROVI Message Files: PVA

MWYX device groups

Designation **PVA** **PROVI**-Messages **A**ccess

Explanation This write command creates PROVI message files. With this write value, it is possible to decide whether the PROVI messages are to be generated according to the current PLC project, or selectively.

FI command **BW_PVA1** **(Single Write)**

Note: This command is an FI job command.

Value to be written

No write value exists	PROVI message files according to the current PLC project.
Write value exists	List of the requested PROVI message files (separated by a comma) according to the format: [PROVI-Diag-type: module number] Example: 01:01,01:02,02:02

Note: The value to be written is passed to the "acValue" parameter as an ASCII value in the "DataTransfer" routine.

Response Structure The response to the "BW_PVA1" FI command consists of three lines, each with one column. The meaning of the elements is as follows:

- Line 1 = Job ID [01...20]
  (see Chapter "FI Commands for the MPCX Device Group", IFJ).
- Line 2 = FI command
  [String, in accordance with Chapter "Elements of the FI Command"]
- Line 3 = FI Job Error Code (see Chapter "Error Codes")

...

Example PVA1 No write value is passed, i.e. the PROVI message files are generated according to the current PLC project.

FI command		**00_BW_PVA1**
Line	**Column**	**Answer**
1	1	01
2	1	00_BW_PVA1
3	1	0

Explanation The read command returns the most significant information on the created PROVI message files.

FI command **BR_PVA1** **(Single Read)**

....

Response Structure The following table shows the general construction of the answer of the FI command BR_PVA1. For each available PROVI message file, 1 line with 10 columns each is created.

Line 1...n	**Column 1**	**...**	**Column 10**

Value Range/Meaning of Columns

1 =	PROVI diagnosis type	[1..20]
2 =	PROVI diagnosis type designation	[The following designations can be returned: StartCondition, Error, Message, Warning, Setup]
3 =	Module number	[1..99]
4 =	PROVI diagnosis type and module number	[PROVI diagnosis type: module number, see write value for BW_PVA2]
5 =	Complete name of the PROVI message text file	[max. 200 ASCII characters]
6 =	Memory required for PROVI messages in the control	[figure in ASCII format]
7 =	Complete name of the PROVI index file	[max. 200 ASCII characters]
8 =	Memory required for PROVI index files in the control	[figure in ASCII format]
9 =	Total memory (text+index) required in the control	[figure in ASCII format]
10 =	Total memory for ALL PROVI files (text+index) required in the control	[figure in ASCII format]

Example PVA1

The most significant information of 2 available PROVI message files are returned.

FI command		**00_BR_PVA1_1**
Line	**Column**	**Answer**
1	1	1
	2	Error
	3	1
	4	01:01
	5	D:\Programs\indramat\Mtgui\Project_000\ Programmdata\Device_000\Diag\De\ERROR 1.TXT
	6	1345
	7	D:\Program Files\indramat\Mtgui\Project_000\ Programmdata\Device_000\Diag\De\ERROR 1.IDX
	8	234
	9	1579
	10	4491
2	1	2
	2	Message
	3	1
	4	02:01
	5	D:\Programs\indramat\Mtgui\Project_000\ Programmdata\Device_000\Diag\De\ERROR 1.TXT
	6	2456
	7	D:\Programs\indramat\Mtgui\Project_000\ Programmdata\Device_000\Diag\De\ERROR 1.TXT

	8	456
	9	2912
	10	4491

Explanation

This write command transmits PROVI message files into the selected device. Through the write value, it is possible to chose whether ALL or only the PROVI messages selected via the write value are to be transmitted.

FI command

BW_PVA2 **(Single Write)**

Note: This command is an FI job command.

Value to be written

No write value exists	All PROVI message files are transmitted into the selected device
Write value exists	List of the requested PROVI message files (separated by a comma) according to the format: [PROVI-Diag-type: module number] Example: 01:01,01:02,02:02

Note: The value to be written is passed to the "acValue" parameter as an ASCII value in the "DataTransfer" routine.

Response Structure

The response to the "BW_PVA2" FI command consists of three lines, each with one column. The meaning of the elements is as follows:

- Line 1 = Job ID [01...20]
  (see Chapter "FI Commands for the MPCX Device Group", IFJ).
- Line 2 = FI command
  [String, in accordance with Chapter "Elements of the FI Command"]
- Line 3 = FI Job Error Code (see Chapter "Error Codes")

....

Example PVA1

No write value is passed, i.e. all PROVI message files should be transmitted.

FI command		**00_BW_PVA2**
Line	**Column**	**Answer**
1	1	01
2	1	00_BW_PVA2
3	1	0

12.36 Formatted Input / Output of PLC Variables: PVF

MWYX device groups

Designation **PVF** **P**LC **V**ariable **F**ormatted

Explanation Formatted reading and writing of PLC variables, arrays and structures.

FI command Read PLC variables.

CR_PVF_(1)	**(Single Read)**
CC_PVF_(1)	**(Cyclic Read)**
CB_PVF_(1)	**(Break Cyclic Read)**
(1) = Identifier of the PLC variable	[acc. to declaration part of the PLC]

Response Structure One line with one column is output for single variables. For array and structure variables, one line per element is output, depending on the number of elements.

Line 1...n:	Column 1

n = number of elements.

Note: Only defined PLC variables can be read and written. Addressing a non-declared variable results in an error message. A PLC variable can only be read if its data length does not exceed 240 byte. (Refer also to chapter on "Programming" and "Guidelines").

Value Ranges ANSI / ASCII The value range of the response depends on the data type of the variable read. The following table indicates the range in which the results string is to be expected when reading out a single variable and into which C-data type this string can be converted without loss of information:

Data Type	Value Range	Can be converted to C-data type
BOOL	[0;1]	unsigned char
SINT	[-128...127]	char
INT	[-32768...32767]	short
DINT	[2147483648...2147483647]	long
USINT	[0...255]	unsigned char
UINT	[0...65535]	unsigned short
UDINT	[0...4294967295]	unsigned long
BYTE	[0x00...0xFF]	unsigned char
WORD	[0x0000....0xFFFF]	unsigned short
DWORD	[0x00000000...0xFFFFFFFF]	unsigned long
TIME	[0...4294967295]	unsigned long (msec)
CHAR	[$00...$20,!...~,$7F...$FF]	char
STRING	<String> whereby <String> string is a character string with a maximum of as many characters as are declared for the string in the PLC	Char[xx+1]] +1 i.e. room for the zero byte
REAL	[-3.402823567E+38...3.402823567E+38]	Float

Note: An empty string is identified by two single inverted commas: ' ' (do not confuse with the double inverted commas ")!

All single variables can be part of array and structure variables. The value ranges maintain their validity, even when within structured data types.

Binary Value Range

The value range of the response depends on the data type of the variable read. The following table indicates the value range in which to expect the binary value of a single variable and how many bytes are included in the binary byte sequence:

Data Type	**Value Range**	**Length (bytes)**
BOOL	[00$_H$...01$_H$]	1
SINT	[80 $_H$...7F $_H$] i.e. –128...127	1
INT	[8000 $_H$ (-32768)...7FFF $_H$ (32767)]	2
DINT	[80000000 $_H$ (-2147483648)... 7FFFFFFF $_H$ (2147483647)]	4
USINT	[00 $_H$ (0)...FF $_H$ (255)]	1
UINT	[00 $_H$ (0)...FFFF $_H$ (65535)]	2
UDINT	[0...4294967295]	4
BYTE	[0x00...0xFF]	1
WORD	[0x0000....0xFFFF]	2
DWORD;	[0x00000000...0xFFFFFFFF]	4
TIME	[0...4294967295]	4
CHAR	[$00...$20,!...~,$7F...$FF]	1
STRING	<String> whereby <String> string is a character string with a maximum of as many characters as are declared for the string in the PLC	XX+1
REAL	[-3.402823567E+38...3.402823567E+38]	4

Note: Binary array and structure elements are joined together without any spaces between (1-byte alignment).

PLC - Example 1 PVF

Read the value of the PLC variable "STK_TXT" in ASCII format from device address 00.

Assumption:
The "STK_TXT" variable is declared as STRING in the PLC program.

FI command		**00_CR_PVF_STK_TXT/1**
Line	**Column**	**Answer**
1	1	Repeat counter

WinPCL - Example 1 PVF

Read the value of WinPCL variable "STK_TXT" in ASCII format in WinPCL program "Prog" at device address 00.

Assumption:
The WinPCL variable "STK_TXT" is declared in WinPCL Program instance "Prog" as STRING.

FI command		**00_CR_PVF_:Prog.STK_TXT/1**
Line	**Column**	**Answer**
1	1	Repeat counter

PLC - Example 2 PVF

Read the value of the PLC array "BEG_END" in ANSI format from device address 00.

Assumption:
The "BEG_END" variable is declared as BYTE with 2 elements in the PLC program.

FI command		**00_CR_PVF_BEG_END/3**
Line	**Column**	**Answer**
1	1	0x00
2	1	0x1F

WinPCL - Example 2 PVF

Read the value of WinPCL array "BEG_END" in ANSI format in WinPCL program "Prog" at device address 00.

Assumption:
The WinPCL variable "BEG_END" is declared in WinPCL program entity s BYTE with two elements.

FI command		**00_CR_PVF_:Prog.BEG_END/3**
Line	**Column**	**Answer**
1	1	0x00
2	1	0x1F

PLC - Example 3 PVF

Read the value of the PLC structure "MSTRCT" in ASCII format from device address 00.

Assumption:
The "MSTRCT" variable is declared as a structure in the PLC program as follows:

```
TYP STRUCT
        T1      BOOL
        T2      CHAR
        T3      STRING[16]
        T4      TIME
END
```

FI command		**00_CR_PVF_MSTRCT/1**
Line	**Column**	**Answer**
1	1	0
2	1	A
3	1	ROBOT AXIS X
4	1	2000

WinPCL - Example 3 PVF

Read the value of WinPCL structure "MSTRCT" in ASCII format in WinPCL program "Prog" at device address 00.

Assumption:
The WinPCL variable "MSTRCT" is declared as a structure in WinPCL program entity "Prog" as follows:

```
TYP STRUCT
        T1      BOOL
        T2      CHAR
        T3      STRING[16]
        T4      TIME
END
```

FI command		**00_CR_PVF_:Prog.MSTRCT/1**
Line	**Column**	**Answer**
1	1	0
2	1	A
3	1	ROBOT AXIS X
4	1	2000

FI command

Write PLC variable.

CW_PVF_(1)	**(Single Write)**
(1) = Identifier of the PLC variable	[acc. to declaration part of the PLC]

Value to be written

Value of data element	[see value ranges]

Note: The value to be written is passed to the "acValue" parameter in the "DataTransfer" routine. The data code of the value is passed to the parameter "ValType".

Response Structure

One line is output with a column for acknowledgement of whether or not the FI command has been executed successfully.

(P_ACK) = **P**ositive **ACK**nowledge Data element has been set

Value Range of the value to be written in ANSI / ASCII Format

The value ranges agree for the most part with the ANSI / ASCII result-value ranges during read access. ANSI umlauts are thereby converted into ASCII umlauts. Only ASCII umlauts are stored in the control unit. For deviations to this, please refer to the following note:

Note: Strings are enclosed by two single inverted commas ' ' , e.g. 'drill'.

Special characters can be indicated in accordance with DIN-1131 by a $ sign.

The following are used:

- $' '
- $$ $
- $R \r (Carriage Return)
- $L \n (Linefeed)
- $P \f (Form feed)
- $T \t (Tab)
- $xx xx refers to a character written as a hexadecimal value, e.g. $20 (space)

Array and structure elements are separated by a space.

Value Range of the Value to be written in Binary Format

The value ranges agree with the binary result-value range during read access. For deviations to this, please refer to the following note.

PLC - Example 4 PVF

Write into the PLC variable "STK_TXT" at device address 00. The value is passed in ANSI format.

Assumption:
The "STK_TXT" variable is declared as STRING in the PLC program.

FI command		00_CW_PVF_STK_TXT/3
Line	**Column**	**Answer**
1	1	(P_ACK)

Value to be written:

Value of data element	'item counter'
Data code	/3

WinPCL - Example 4 PVF

Write into the WinPCL variable "STK_TXT" in WinPCL program "Prog" at device address 00. The value is passed in ANSI format.

Assumption:
The WinPCL variable "STK_TXT" is declared in WinPCL program entity "Prog" as STRING.

FI command		00_CW_PVF_:Prog.STK_TXT/3
Line	**Column**	**Answer**
1	1	(P_ACK)

Value to be written:

Value of data element	'item counter'
Data code	/3

PLC - Example 5 PVF

Write into the PLC byte array "BEG_END" at device address 00. The value is passed in ANSI format.

Assumption:
The "BEG_END" variable is declared as a BYTE array with 2 elements in the PLC program.

FI command		00_CR_PVF_BEG_END/3
Line	**Column**	**Answer**
1	1	(P_ACK)

Value to be written:

Value of data element	0x20 0x3f
Data code	/3

WinPCL - Example 5 PVF

Write into the WinPCL byte array "BEG_END" in WinPCL program "Prog" at device address 00. The value is passed in ANSI format.

Assumption:
The WinPCL variable "BEG_END" is declared in WinPCL program entity "Prog" as BYTE with two elements.

FI command		00_CW_PVF_:Prog.BEG_END/3
Line	**Column**	**Answer**
1	1	(P_ACK)

Value to be written:

Value of data element	0x20 0x3f
Data code	/3

PLC - Example 6 PVF

Write the value of element T3 of the PLC structure "MSTRCT" at device address 00. The string "COUNTER" is transferred in binary format.

Assumption:
The "MSTRCT" variable is declared as a structure in the PLC program as follows:

```
TYP STRUCT
          T1        BOOL
          T2        CHAR
          T3        STRING[16]
          T4        TIME
END
```

FI command		00_CW_PVF_MSTRCT.T3/2
Line	**Column**	**Answer**
1	1	(P_ACK)

Value to be written:

Value of data element	Binary sequence: 43 4F 55 4E 54 45 52 00
Data code	/2

WinPCL - Example 6 PVF

Write the value of element T3 of the WinPCL structure "MSTRCT" at device address 00. The string "COUNTER" is transferred in binary format.

Assumption:
The WinPCL variable "MSTRCT" is declared as a structure in WinPCL program entity "Prog" as follows:

```
TYP STRUCT
          T1        BOOL
          T2        CHAR
          T3        STRING[16]
          T4        TIME
END
```

FI command		00_CW_PVF_:Prog.MSTRCT.T3/2
Line	**Column**	**Answer**
1	1	(P_ACK)

Value to be written:

Value of data element	Binary sequence: 43 4F 55 4E 54 45 52 00
Data code	/2

PLC - Example 7 PVF

Write the value of the PLC structure "MSTRCT" from the structure "mstrct" previously stored in the C program at device address 00.

Assumption:
The "MSTRCT" variable is declared as a structure in the PLC program as follows:

```
TYP STRUCT
          T1        BOOL
          T2        CHAR
          T3        STRING[16]
          T4        TIME
END
```

To exchange binary data in a C program, the following "C" data type can be used:

```
#pragma pack(1)    //Write all elements
                   //without spaces next to each other.
typeder struct
{
      unsigned char T1;
      char         T2;
      char         T3[17]; //Space for zero byte
      unsigned long T4;
} Tymstrct;        // Declare structure
Tymstrct mstrct;   // Apply structure
```

FI command		**00_CW_PVF_MSTRCT/2**
Line	**Column**	**Answer**
1	1	(P_ACK)

Value to be written: address of the C structure.

Value of data element	&mstrct
Data code	/2

WinPCL - Example 7 PVF

Write the value of the WinPCL structure "MSTRCT" from the structure "mstrct" previously stored in the C program at device address 00.

Assumption:
The WinPCL variable "MSTRCT" is declared as a structure in WinPCL program entity "Prog" as follows:

```
TYP STRUCT
      T1      BOOL
      T2      CHAR
      T3      STRING[16]
      T4      TIME
END
```

To exchange binary data in a C program, the following "C" data type can be used:

```
#pragma pack(1)    //Write all elements
                   //without spaces next to each other.
typeder struct
{
      unsigned char T1;
      char         T2;
      char         T3[17]; //Space for zero byte
      unsigned long T4;
} Tymstrct;        // Declare structure
Tymstrct mstrct;   // Apply structure
```

FI command		**00_CW_PVF_:Prog.MSTRCT/2**
Line	**Column**	**Answer**
1	1	(P_ACK)

Value to be written: address of the C structure.

Value of data element	&mstrct
Data code	/2

12.37 ProVi Messages: PVM

MWYX device groups

Designation **PVM** **P**ro**V**i **M**essages

Explanation ProVi messages are output. These messages are assigned to a particular message type or module.

FI command Output all ProVi messages.

Not all data will be applied because of optimization reasons; therefore the diagnosis server must be informed that the data is required (see ADW).

BR_PVM1_(1){_(2)}	**(Single Read)**
BC_PVM1_(1){_(2)}	**(Cyclic Read)**
(1) = Message type	[1 = error, 2 = messages, 10 = warnings, 11 = start requirements, 12 = setup diagnosis]
(2) = Module number	[1...99] ! only for message type 1 -2!

Output first ProVi messages.

BR_PVM2_(1){_(2)}	**(Single Read)**
BC_PVM2_(1){_(2)}	**(Cyclic Read)**
(1) = Message type	[1 = error, 2 = messages, 10 = warnings, 11 = start requirements, 12 = setup diagnosis]
(2) = Module number	[1...99] ! only for message type 1 -2!

Response Structure The following table shows the general structure of the FI commands "PVM1" and "PVM2". The number of lines depends on the number of messages pending.

If there are no messages, the number of lines is 0.

Line 1...n	Column 1	...	Column 8

Meaning of the Columns

1 = Message text	[ASCII characters]
2 = Message number	[ASCII characters]
3 = Time stamp day	[mm.dd.yyyy]
4 = Time stamp time	[hh:mm:ss]
5 = Message ID	[ASCII characters] (DWORD, decimal)
6 = Reference text exists	[YES, NO]
7 = Criteria analysis exists	[YES, NO]
8 = Message HTML file	[ASCII characters]

Example PVM1

All ProVi errors from module 3 in control unit 0. There are two messages:

FI command		**00_BR_PVM1_1_3**
Line	**Column**	**Answer**
1	1	Guard not closed
	2	34
	3	01.27.2000
	4	14:56:32
	5	43923028
	6	YES
	7	NO
	8	
2	1	Oil pressure too low
	2	124
	3	01.27.2000
	4	15:03:10
	5	98234039
	6	NO
	7	YES
	8	

Example PVM2

The first ProVi error from module 3 in control unit 0.

There are two messages:

FI command		**00_BR_PVM2_1_3**
Line	**Column**	**Answer**
1	1	Guard not closed
	2	34
	3	01.27.2000
	4	14:56:32
	5	43923028
	6	YES
	7	NO
	8	

FI command

Output the reference information of a ProVi message.

BR_PVM3_(1)_(2){_(3)}	**(Single Read)**
(1) = Message ID	[ASCII characters]
(2) = Message type	[1 = error, 2 = messages, 10 = warnings, 11 = start requirements, 12 = setup diagnosis]
(3) = Module number	[1...99] ! only for message type 1 -2!

Response Structure

The following table shows the general structure of the "PVM3" FI command.

Line 1	**Column 1**	**...**	**Column 16**

Meaning of the Columns

1 =	Message text	[ASCII characters]
2 =	Message number	[ASCII characters]
3 =	Error category	[ASCII characters] (empty no category)
4 =	Time stamp day	[mm.dd.yyyy]
5 =	Time stamp hour	[hh:mm:ss]
6 =	Reference text available	[YES, NO]
7 =	Reference text	[ASCII characters]
8 =	Message ID	[ASCII characters] (DWORD, decimal)
9 =	Diagnosis source	[ASCII characters] (PLC, CNC)
10 =	POE name	[ASCII characters]
11 =	Detail name	[ASCII characters] (empty implementation)
12 =	Detail type	[1 = action block, 3 = transition, 4 = implementation]
13 =	Network number	[ASCII characters]
14 =	Variable name	[ASCII characters]
15 =	POU entity name	[ASCII characters]
16 =	POU type	[2 = program, 3 = function block]
17 =	Analysis of criteria available	[YES, NO]
18 =	Message HTML file	[ASCII characters]
19 =	Reference info HTML file	[ASCII characters]

Example PVM3

Reference text of a ProVi error with ID 43923028 from module 3 in control unit 0.

FI command		**00_BR_PVM3_43923028_1_3**
Line	**Column**	**Answer**
1	1	Guard not closed
	2	34
	3	1
	4	01.27.2000
	5	14:56:32
	6	YES
	7	Oil pressure too low Oil pipe leaking or insufficient oil.
	8	43923028
	9	PLC
	10	MODULE3
	11	
	12	4
	13	34
	14	EschutzT
	15	Station2.Module3
	16	3

	17	NO
	18	
	19	D:\Program Files\Indramat\MtGui\Project_000\ ProgramData\HMTL\DE\Error34.html

FI command

One after the other, all active ProVi messages are output. In the result, one line each is returned. After expiry of the set time, the next message is returned. The clock frequency can be set via the last parameter. This value can only be set once for the PC. The value transmitted last is always the valid value. Default setting is one second.

BR_PVM4_(1){_(2)_(3)}	**(Single Read)**
BC_PVM4_(1){_(2)_(3)}	**(Cyclic Read)**
(1) = Message type	[1 = error, 2 = messages, 10 = warnings, 11 = start requirements, 12 = setup diagnosis]
(2) = Module number	[1...99] ! only for message type 1 -2!
(3) = Clock frequency	[ASCII characters] Time in ms

Response Structure

The following table shows the general structure of the "PVM4" FI command.

If there are no messages, the number of lines is 0.

Line 1	Column 1	...	Column 8

Meaning of the Columns

1 = Message text	[ASCII characters]
2 = Message number	[ASCII characters]
3 = Time stamp day	[mm.dd.yyyy]
4 = Time stamp time	[hh:mm:ss]
5 = Message ID	[ASCII characters] (DWORD, decimal)
6 = Reference text available	[YES, NO]
7 = Criteria analysis exists	[YES, NO]
8 = Message index (1 = 1. message)	[ASCII characters]
9 = Message HTML file	[ASCII characters]

Example PVM4

ProVi errors from module 3 in control unit 0.

The 2nd message is being output. The clock frequency is to be 2 seconds.

FI command		00_BR_PVM4_1_3_2000
Line	**Column**	**Answer**
1	1	Oil pressure too low
	2	124
	3	01.27.2000
	4	15:03:10
	5	98234039
	6	NO
	7	YES
	8	2
	9	

12.38 Download of PLC Retain Variables: PVR

MWYX device groups

Designation **PVR** **P**LC **V**ariable **R**etain Backup

Explanation Download of PLC retain variables.

FI command **BW_PVR1!(1)** **(Single Write)**

(1) = Download file with path details.

Note: File and path details must be enclosed in inverted commas.

The separator "!" is used in this command.

Response Structure The response to the "PVR1" FI command consists of three lines, each with one column. The meaning of the elements is as follows:

- Line 1 = Job ID [01...20]
  (see Chapter "FI Commands for the MPCX Device Group: IFJ").
- Line 2 = FI command
  [String, in accordance with Chapter "Elements of the FI Command"]
- Line 3 = FI Job Error Code
  (see Chapter "Error Codes")

Example PVR1 00_BW_PVR1!"D:\Program Files\Indramat\Mtgui\Temp\download.ini"/3

FI command		**00_BW_PVR1!"D:\Program Files\Indramat\Mtgui\ Temp\download.ini"/3**
Line	**Column**	**Answer**
1	1	01
2	1	00_BW_PVR1!"D:\Program Files\Indramat\Mtgui\ Temp\download.ini" /3
3	1	0

Structure of Download File The structure of the "download.ini" file used in this example corresponds to an Ini file in Windows.

Note: Care must be taken in the use of upper and lower case letters.

12.39 Upload of PLC Retain Variables: PVR

MWYX device groups

Designation **PVR** **P**LC **V**ariable **R**etain Backup

Explanation PLC retain variables are uploaded via all active processes.

FI command **BR_PVR1!(1)** **(Single Read)**

(1) = Upload file with path details

Note: Enclose file and path details in inverted commas.
The separator "!" is used in this command.

Response Structure The response to the "PVR1" FI command consists of three lines, each with one column. The meaning of the elements is as follows:

- Line 1 = Job ID [01...20]
  (see Chapter "FI Commands for the MPCX Device Group", IFJ).
- Line 2 = FI command
  [String, in accordance with Chapter "Elements of the FI Command"]
- Line 3 = FI Job Error Code (see Chapter "Error Codes")

Example PVR 00_BR_PVR1_"D:\Program Files\Indramat\Mtgui\Temp\Upload.ini"/3

FI command		**00_BR_PVR1!"D:\Program Files\Indramat\Mtgui\Temp\upload.ini"/3**
Line	**Column**	**Answer**
1	1	01
2	1	00_BR_PVR1!"D:\Program Files\Indramat\Mtgui\Temp\upload.ini" /3
3	1	0

Structure of Upload File The upload file is structured in the Windows – "Ini" format structure.

Note: Care must be taken in the use of upper and lower case letters.

12.40 Reading the PLC Variable Declaration: PVT

MWYX device groups

Designation **PVT** **P**LC **V**ariable **T**ype

Explanation A PLC variable has a particular type. To evaluate complex variables such as structures and arrays, their components and types must be read out.

Refer also to PVF, Reading Structured PLC Variables.

FI command Read the PLC variable type.

BR_PVT_(1)	**(Single Read)**
(1) = Identifier of the PLC variable	[acc. to declaration part of the PLC]

Response Structure One line with 2 columns is output for each element of the variables.

Line 1...n:	Column 1	Column 2

n = number of elements.

Value Range/Meaning of Columns

1 = Identifier of the PLC variable	[acc. to declaration part of the PLC]
2 = Type	[see value range PVF]

Examples:
PLC: Reading of a variable

Assumption:
The "TEST" variable is declared as WORD in the PLC program.

FI command	00_BR_PVT_TEST	
Answer		
Line	**Column 1 (Name)**	**Name**
1	TEST	WORD

WinPCL: Reading a Variable

Assumption:
The WinPCL variable "TEST" is declared as WORD in WinPCL program "Prog".

FI command	00_BR_PVT_:Prog.TEST	
Answer		
Line	**Column 1 (Name)**	**Name**
1	TEST	WORD

PLC: Reading a Structure

Assumption:
The "TEST1" variable is declared as STRUCT in the PLC program.

```
STRUCT
        E1      BOOL
        E2      INT
        E3      SINT
END
```

FI command	00_BR_PVT_TEST1	
Answer		
Line	**Column 1**	**Column 2**
1	TEST1.E1	BOOL
2	TEST1.E2	INT
3	TEST1.E3	SINT

WinPCL: Reading a Structure

Assumption:

The WinPCL variable "TEST1" is declared as STRUCT in WinPCL program "Prog".

```
STRUCT
        E1      BOOL
        E2      INT
        E3      SINT
END
```

FI command	00_BR_PVT_:Prog.TEST1	
Answer		
Line	**Column 1**	**Column 2**
1	TEST1.E1	BOOL
2	TEST1.E2	INT
3	TEST1.E3	SINT

PLC: Reading an Array

Assumption:
The "TEST2" variable is declared as ARRAY in the PLC program.

```
ARRAY [
        0 .. 3
] OF    BOOL
```

FI command	00_BR_PVT_TEST2	
Answer		
Line	**Column 1**	**Column 2**
1	TEST2[0]	BOOL
2	TEST2[1]	BOOL
3	TEST2[2]	BOOL
4	TEST2[3]	BOOL

WinPCL: Reading an Array

Assumption:
The WinPCL variable "TEST2" is declared as ARRAY in WinPCL program "Prog".

```
ARRAY [
        0 .. 3
] OF    BOOL
```

FI command	00_BR_PVT_:Prog.TEST2	
Answer		
Line	**Column 1**	**Column 2**
1	TEST2[0]	BOOL
2	TEST2[1]	BOOL
3	TEST2[2]	BOOL
4	TEST2[3]	BOOL

PLC: Reading an Array of a Structure

Assumption:
The "TEST3" variable is declared as ARRAY in the PLC program.

```
ARRAY [
          0 .. 1
] OF      STRUCT1,
```

where STRUCT1 is declared as follows:

```
STRUCT
          E1        BOOL
          E2        INT
          E3        SINT
END
```

END

FI command	00_BR_PVT_TEST3	
Answer		
Line	Column 1	Column 2
1	TEST3[0].E1	BOOL
2	TEST3[0].E2	INT
3	TEST3[0].E3	SINT
1	TEST3[1].E1	BOOL
2	TEST3[1].E2	INT
3	TEST3[1].E3	SINT

WinPCL: Reading an Array of a Structure

Assumption:
The WinPCL variable "TEST3" is declared as ARRAY in WinPCL program "Prog".

```
ARRAY [
          0 .. 1
] OF      STRUCT1,
```

where STRUCT1 is declared as follows:

```
STRUCT
          E1        BOOL
          E2        INT
          E3        SINT
END
```

FI command	00_BR_PVT_:Prog.TEST3	
Answer		
Line	Column 1	Column 2
1	TEST3[0].E1	BOOL
2	TEST3[0].E2	INT
3	TEST3[0].E3	SINT
1	TEST3[1].E1	BOOL
2	TEST3[1].E2	INT
3	TEST3[1].E3	SINT

Assumption:
The data types are output according to IEC1131.

See also command PVF.

12.41 SFC Diagnosis Data: SDD

MWYX device group

Designation **SDD** **S**FC **D**iagnosis **D**ata

Explanation Data for step chain diagnosis is output. Depending on the FI command this data can concern disrupted steps, actions, transitions or a definite ID to display the action or transition.

FI command Output the disrupted step of a step chain.

BR_SDD1!(1)!(2)	**(Single Read)**
(1) = Module number	[1...99]
(2) = SFC entity name	[ASCII characters]

Note: The separator "!" is used in this command.

Response Structure The following table shows the general structure of the FI command "SDD1".

Line 1	**Column 1**	**...**	**Column 7**

Meaning of the Columns

1 = Step name	[ASCII characters]
2 = Detail type	[1 = action block, 2 =action network, 3 = transition]
3 = Detail name	[ASCII characters]
4 = POU ID	[ASCII characters]
5 = Detail morpheme	[ASCII characters] (DWORD, decimal)
6 = Error ID	[ASCII characters] (DWORD, decimal)
7 = POU entity name	[ASCII characters]

Example SDD1 Query disrupted step of the "clamp" chain in module 3 in control unit 0.

FI command		**00_BR_SDD1!3!Station03A.Clamp**
Line	**Column**	**Answer**
1	1	Open
	2	1
	3	Aopen
	4	SFC_1_2
	5	98243823
	6	34985304
	7	Station2.Module3

FI command Output the faulty action, monitor error or transition of a disrupted step.

BR_SDD2!(1)!(2)!(3)	**(Single Read)**
(1) = Module number	[1...99]
(2) = SFC entity name	[ASCII characters]
(3) = Step name	[ASCII characters]

Note: The separator "!" is used in this command.

Response Structure The following table shows the general structure of the FI command "SDD2".

Line 1	Column 1	...	Column 6

Meaning of the Columns

1 = Detail type	[1 = action block, 2 =action network, 3 = transition]
2 = Detail name	[ASCII characters]
3 = POU ID	[ASCII characters]
4 = Detail morpheme	[ASCII characters] (DWORD, decimal)
5 = Error ID	[ASCII characters] (DWORD, decimal)
6 = POU entity name	[ASCII characters]

Example SDD2

Query faulty action of the disrupted step "open" of the "clamp" chain in module 3 in control unit 0.

FI command		**00_BR_SDD1!3!Station03A.Clamp_Open**
Line	**Column**	**Answer**
1	1	1
	2	AOpen
	3	SFC_1_2
	4	98243823
	5	34985304
	6	Station2.Module3

FI command

Output the definite ID to display the action, monitor error or transition.

BR_SDD3!(1)!(2)!(3)!(4) **(Single Read)**

(1) = Module number	[1...99]
(2) = SFC entity name	[ASCII characters]
(3) = Detail type	[1 = action block, 2 = action network, 3 = transition]
(4) = Detail name	[ASCII characters]

Note: The separator "!" is used in this command.

Response Structure

The following table shows the general structure of the FI command "SDD3".

Line 1	Column 1	...	Column 4

Meaning of the Columns

1 = POU ID	[ASCII characters]
2 = Detail morpheme	[ASCII characters] (DWORD, decimal)
3 = Error ID	[ASCII characters] (DWORD, decimal)
4 = POU entity name	[ASCII characters]

Example SDD3

Query ID to display the action "aOpen" of the "clamp" chain in module 3 in control unit 0.

FI command		**00_BR_SDD3!3!Station03A.Clamp!1!aOpen**
Line	**Column**	**Answer**
1	1	SFC_1_2
	2	98243823
	3	34985304
	4	Station2.Module3

FI command

Output the I/O addresses to display a detail.

BR_SDD4!(1)!(2)!(3)!(4) **(Single Read)**

(1) = Module number [1...99]
(2) = SFC entity name [ASCII characters]
(3) = Detail type [1 = action block, 2 = action network, 3 = transition]
(4) = Detail name [ASCII characters]

Note: The separator "!" is used in this command.

Response Structure

The following table shows the general structure of the FI command "SDD4".

Line 1-n	Column 1	Column 2

Meaning of the Columns

1 = Variable morpheme	[ASCII characters] (DWORD, decimal)
2 = I/O address	[ASCII characters]

Example SDD4

Query I/O addresses to display the action "aOpen" of the "clamp" chain in module 3 in control unit 0.

Three variables have an I/O address.

FI command		**00_BR_SDD4!3!Station03A.Clamp!1!aOpen**
Line	**Column**	**Answer**
1	1	98243823
	2	%I3.2.0
2	1	40923423
	2	%Q23.21.7
3	1	34985304
	2	%I100.3.5

FI command

Determine the multilingual comments for displaying a detail.

BR_SDD5!(1)!(2)!(3)!(4) **(Single Read)**

(1) = Module number [1...99]
(2) = SFC entity name [ASCII characters]
(3) = Detail type [1 = action block, 2 =action network, 3 = transition]
(4) = Detail name [ASCII characters]

Note: The separator "!" is used in this command.

Response Structure

The following table shows the general structure of the FI command "SDD5".

Line 1-n	Column 1	Column 2

Meaning of the Columns

1 = Comment morpheme	[ASCII characters] (DWORD, decimal)
2 = New comment	[ASCII characters]

Example SDD5

Query comments to display the action "aOpen" of the "clamp" chain in module 3 in control unit 0.

Two comments are replaced by another text.

FI command		**00_BR_SDD5!3!Station03A.Clamp!1!aOpen**
Column	**Answer**	**Response**
1	1	98243823
	2	Clamp open
1	1	40923423
	2	Clamp closed

FI command

Output the action that has not been performed, or the transition of a step calculated based on the online status.

BR_SDD6!(1)!(2)!(3) **(Single Read)**

(1) = Module number [1...99]

(2) = SFC entity name [ASCII characters]

(3) = Step name [ASCII characters]

Note: The separator "!" is used in this command.

Response Structure

The following table shows the general structure of the FI command "SDD6".

Line 1	**Column 1**	**...**	**Column 6**

Meaning of the Columns

1 = Detail type	[1 = action block, 3 = transition]
2 = Detail name	[ASCII characters]
3 = POU ID	[ASCII characters]
4 = Detail morpheme	[ASCII characters] (DWORD, decimal)
5 = Error ID	[ASCII characters] (DWORD, decimal)
6 = POU entity name	[ASCII characters]

Example SDD6

Query the action that has not been performed for the step "open" of the "clamp" chain in module 3 in control unit 0.

FI command		**00_BR_SDD6!3!Station03A.Clamp_Open**
Line	**Column**	**Answer**
1	1	1
	2	AOpen
	3	SFC_1_2
	4	98243823
	5	34985304
	6	Station2.Module3

FI command

Determine the module number of a step chain.

BR_SDD7!(1) **(Single Read)**

(2) = SFC instances name [ASCII characters]

Note: The separator "!" is used in this command.

Response Structure

The following table shows the general structure of the FI command "SDD7".

Line 1	Column 1

Meaning of the Columns 1 = Module number [1...99]

Example SDD7 Inquiry of the module of the chain "clamp".

FI command		**00_BR_EXD1!Station03A.Clamp**
Line	**Column**	**Answer**
1	1	3

12.42 Set the Device Status Information: SDS

MWYX device groups

Designation **SDS** **S**et **D**evice **S**tatus

Explanation By this command, the device status information can be set; here, the configuration file IND_DEV.INI is adjusted as well.

Note: When this command is transmitted, the following system messages are generated:
MSG_DEVICEOFF or MSG_DEVICE_ON !

FI command With this command, the device status information of **ALL** defined devices can be set.

BW_SDS1_(1) **(Single Write)**

(1) = Device status information to be set

0 = Device status information OFF
1 = Device status information ON

Response Structure The following table shows the general structure of the response to the "SDS1" FI command.

Line 1	Column 1

Value Range/Meaning of Columns 1 = Status report [(P_ACK)]

Example SDS1 Set device status information to OFF for **ALL** defined devices.

FI command		**00_BW_SDS1_0**
Line	**Column**	**Answer**
1	1	(P_ACK)

FI command With this command, the device status information for a selected device can be set.

BW_SDS2_(1) **(Single Write)**

(1) = Device status information to be set

0 = Device status information OFF
1 = Device status information ON

Response Structure The following table shows the general structure of the response to the "SDS2" FI command.

Line 1	Column 1

Value Range/Meaning of Columns 1 = Status report [(P_ACK)]

Example: SDS2

Set device status information to OFF for the selected device 00.

FI command		**00_BW_SDS2_0**
Line	**Column**	**Answer**
1	1	(P_ACK)

12.43 Setting the FI Exclusive Mode: SEM

MWYX device group

Designation

SEM **S**et FI **E**xclusive **M**ode

Explanation

This command is used to activate FI Exclusive mode for the selected device address.

FI Exclusive mode: In this mode, **ALL** the processes logged in at the FI – **with the exception** of the process issuing the SEM command – are blocked from data communication with this device address. This mode is used for example for data communication which must NOT be interrupted by other data requests. However, it is **imperative** that this FI Exclusive mode is deleted once more through the DEM command.

FI command

BW_SEM1 **(Single Write)**

Response Structure

The following table shows the general structure of the response to the FI command "BW_SEM1". A line of 1 column is output.

Line 1	**Column 1**

Value Range/Meaning of Columns

1 = Status message (P_ACK) (P_ACK)

Example SEM1

Activate FI Exclusive mode for device address 0.

FI command		**00_BW_SEM1**
Line	**Column**	**Answer**
1	1	(P_ACK)

12.44 Sequencer Data: SFD

MWYX device groups

Designation **SFD** **SF**C **D**ata

Explanation Data for a step chain is outputted. Depending on the FI command this can concern a step chain comment, POE name, step comment, maximum time, action / transition / monitor error name (comment), qualifier and time value.

FI command Query the data for a step chain.

BR_SFD1!(1)!(2)	**(Single Read)**
(1) = Module number	[1...99]
(2) = SFC entity name	[ASCII characters]

Note: The separator "!" is used in this command.

Response Structure The following table shows the general structure of the "SFD1" FI command.

Line 1	Column 1	Column 2

Meaning of the Columns

1 = Step chain comment	[ASCII characters]
2 = POE name	[ASCII characters]

Example SFD1 Query data of the "clamp" chain in module 3 in control unit 0.

FI command		**00_BR_SFD1!3!Station03A.Clamp**
Line	**Column**	**Answer**
1	1	Clamping device
	2	CLAMP

FI command Query the data of a step.

BR_SFD2!(1)!(2)!(3)	**(Single Read)**
(1) = Module number	[1...99]
(2) = SFC entity name	[ASCII characters]
(3) = Step name	[ASCII characters]

Note: The separator "!" is used in this command.

Response Structure The following table shows the general structure of the "SFD2" FI command. The number of lines depends on the number of actions and transitions.

If there are no details the line number is 1.

Line 1	Column 1	...	Column 3
Line 2...n:	**Column 1**	**...**	**Column 6**

Meaning of the Columns

Line 1

1 = Step comment	[ASCII characters]
2 = Maximum time	[ASCII characters]
3 = Minimum time	[ASCII characters]

Line 2...n:

1 = Detail type [1 = action block, 3 = transition]
2 = Name [ASCII characters]
3 = Comment [ASCII characters]
4 = Boolean variable [YES, NO]
5 = Qualifier [ASCII characters]
6 = Time value [ASCII characters]

Example SFD2

Data for the step "Open" in the "clamp" chain in module 3 on control unit 0.

FI command		**00_BR_SFD2!3!Station03A.Clamp!Open**
Line	**Column**	**Answer**
1	1	Open clamping device
	2	T#5s
	3	
2	1	1
	2	aOpen
	3	Clamp open
	4	NO
	5	D
	6	T#3s
3	1	3
	2	tOpen
	3	Clamping device is open
	4	NO
	5	
	6	

FI command

Output the data for a detail.

BR_SFD3!(1)!(2)!(3)!(4) **(Single Read)**
(1) = Module number [1...99]
(2) = SFC entity name [ASCII characters]
(3) = Detail type [1 = action block, 2 = action network, 3 = transition]
(4) = Detail name [ASCII characters]

Note: The separator "!" is used in this command.

Response Structure

The following table shows the general structure of the "SFD3" FI command.

Line 1	**Column 1**	**Column 2**

Meaning of the Columns

1 = Comment [ASCII characters]
2 = Boolean variable [YES, NO]

Example SFD3

Data for the action "aOpen" in the "clamp" chain in module 3 in control unit 0.

FI command		**00_BR_SFD3!3!Station03A.Clamp!aOpen**
Line	**Column**	**Answer**
1	1	Clamp open
	2	NO

12.45 Sequencer Messages: SFE

MWYX device groups

Designation **SFE** **SF**C **E**rror

Explanation The sequencer messages of a module are output.

FI command Output all SFC messages.

For reasons of optimization, not all data is called up. Accordingly, the Diagnosis server must be informed that the data is required (see ADW).

BR_SFE1_(1)	**(Single Read)**
BC_SFE1_(1)	**(Cyclic Read)**
(1) = Module number	[1...99]

Output first SFC messages.

BR_SFE2_(1)	**(Single Read)**
BC_SFE2_(1)	**(Cyclic Read)**
(1) = Module number	[1...99]

Response Structure The following table shows the general structure of the FI commands "SFE1" and "SFE2". The number of lines depends on the number of messages pending.

If there are no messages, the number of lines is 0.

Line 1...n:	**Column 1**	**...**	**Column 7**

Meaning of the Columns

1 = Message text	[ASCII characters]
2 = SFC entity name	[ASCII characters]
3 = Step name	[ASCII characters]
4 = Time stamp day	[mm.dd.yyyy]
5 = Time stamp time	[hh:mm:ss]
6 = Type of error	[1 = time error, 2 = monitor error, 3 = monitor event]
7 = Is there condition analysis?	[YES, NO]

Example SFD1 All SFC messages from module 2 in control unit 0.

There are two messages:

FI command		**00_BR_SFE1_2**
Line	**Column**	**Answer**
1	1	TIME ERROR: Chain: chucking Step: up malfunction
	2	Station03A.Clamp
	3	Open
	4	01.27.2000
	5	11:56:32 AM
	6	1
	7	YES
2	1	ASSY ERROR: Chain: drilling Step: down malfunction
	2	Station02A.Drill
	3	Down
	4	01.27.200
	5	13:03:12
	6	2
	7	NO

Example SFE2 First SFC message from module 2 in control unit 0.

There are two messages.

FI command		**00_BR_SFE2_2**
Line	**Column**	**Answer**
1	1	TIME ERROR: Chain: chucking Step: up malfunction
	2	Station03A.Clamp
	3	Open
	4	01.27.2000
	5	14:56:32
	6	1
	7	YES

12.46 Sequencer Mode: SFM

MWYX device groups

Designation **SFM** **S**F**C** **M**ode

Explanation Queries step chain mode.

FI command Query the mode of a step chain.

BR_SFM1!(1)!(2)	**(Single Read)**
BC_SFM1!(1)!(2)	**(Cyclic Read)**
(1) = Module number	[1...99]
(2) = SFC entity name	[ASCII characters]

Note: The separator "!" is used in this command.

Response Structure The following table shows the general structure of the "SFM1" FI command.

Line 1	Column 1

Meaning of the Columns

1 = Mode	[1 = time error, 2 = monitor error, 3 = monitor event, 10 = stop, 11 = auto, 12 = manual, 13 = jog]

Example SFM1 Query mode of the "clamp" chain in module 3 in control unit 0.

FI command		**00_BR_SFM1!3!Station03A.Clamp**
Line	**Column**	**Answer**
1	1	1

12.47 Software Installation Data: SID

MWYX device groups

Designation **SID** **S**oftware **I**nstallation **D**ata

Explanation Information is returned regarding installation. This information includes installation paths, context information, the software versions used, DLL mode, plus service pack and release information.

FI command Read-in the installation data or the software version data.

BR_SID1 **(Single Read)**

Response Structure One line with 16 columns is output for the returned values.

Line 1	Column 1	...	Column 16

Meaning of the Columns

1 = Basic directory	[EXE files of the BOF]
2 = FI installation directory	[FI directory]
3 = Data directory	[in accordance with BOF]
4 = GBO version	[from INDRAMAT.ini]
5 = IF-DLL mode	[from INDRAMAT.ini]
6 = IF version	[from INDRAMAT.ini - from DLL mode 400]
7 = Service package info	[from INDRAMAT.ini - from DLL mode

	420]
8 = Release info	[from INDRAMAT.ini - from DLL mode 420]
9 = IF-Build-Info	[in accordance with Build process]
10 = Current context name	[in accordance with the installation]
11 = Physical installation path	[in accordance with the installation]
12 = Complete IF version indication string	
13 = WinPCL build number	[in accordance with WinPCL]
14 = Version number of the PLC compiler	[in accordance with WinPCL]
15 = Version number of the PLC linker	[in accordance with WinPCL]
16 = Version number of the PLC data basis	[in accordance with WinPCL]
17 = Platform version	

Example SID1 Return information on the current installation.

FI command		**00_BR_SID1**
Line	**Column**	**Answer**
1	1	--
	2	D:\Program Files\Indramat\MTGUI\Bin
	3	--
	4	005-22Vxx
	5	07.20
	6	07V00
	7	--
	8	--
	9	Build 3124 Mar 6 2003 08:53:55
	10	MTGUI_0-23T01 B3327
	11	D:\Program Files\Indramat\MTGUI\
	12	FI: 07V00 DLL-Mode: 07.20 Build 3124 Mar 6 2003 08:53:55
	13	347.15.4.11
	14	771
	15	515
	16	78
	17	Platform: 02V01 Build: 3214

Note: Refer to FI command "PHD" of the MPCX for working with absolute paths.

12.48 PLC Long Identification: SLI

MWYX device groups

Designation **SLI** **P**LC **L**ong **I**dentification

Explanation Returns the unit data from the PLC long identification.

FI command Read PLC long identification.

BR_SLI **(Single Read)**

Response Structure One line with 15 columns is output for the returned values.

Line 1	Column 1	Column...	Column 15

Value Range/Meaning of the Columns

1 =	device address	[00...15]
2 =	program number	[01...99]
3 =	Project name	[max. 8 ASCII characters]
4 =	Program name	[max. 8 ASCII characters]
5 =	User name	[acc. to password entry]
6 =	Program length	[bytes]
7 =	Compilation time	[LONG] (coded in long value)
8 =	Compilation date	[8 ASCII characters]
9 =	Compilation time	[8 ASCII characters]
10 =	Download time	[LONG] (coded in long value)
11 =	Download date	[8 ASCII characters]1
12 =	Download time	[8 ASCII characters]
13 =	Version of PLC long identification	[LONG]
14 =	RUN flags	[HEX value]
15 =	Compiler info	[LONG]

Example SLI Read the unit data from the PLC long identification.

FI command		**00_BR_SLI**
Line	**Column**	**Answer**
1	1	02
	2	01
	3	--
	4	MOT12
	5	TEST
	6	17672
	7	630163960
	8	15.12.99
	9	17:15:48
	10	630163961
	11	15.12.99
	12	17:15:50
	13	2
	14	0x0000
	15	13

Reference to Literature see chapter entitled "Literature" [30].

12.49 SERCOS Parameters: SPA

MSYX device group

Designation **SPA** **S**ERCOS **PA**rameter

Explanation A SERCOS drive parameter is output or written. Each parameter consists of 7 elements, whereby any combination of elements can be selected by element coding.

FI command

BR_SPA1_(1)_(2)_(3)	**(Single Read)**
BC_SPA1_(1)_(2)_(3)	**(Cyclic Read)**
BB_SPA1_(1)_(2)_(3)	**(Break Cyclic Read)**
BW_SPA1_(1)_(2)_(3)	**(Single Write)**
(1) = Drive address	[0...254]
(2) = Parameter No.	in format X-Y-ZZZZ
(3) = Element coding	[standard or advanced format]

Parameter No.

Format X-Y-ZZZZ	Value Range
X	S = standard data P = product data Y = SERCANS parameter
Y	[0..00.15] = parameter record
Z	[0...4095] = data block no.

Element Coding Element coding in standard format allows individual elements, such as the operating date, to be requested. If several elements are to be read out in one request, then the element coding can be OR'd in advanced format, e.g. operating date (0x40) and unit (0x08) produces OR'd (0x48) → 48

The advanced format 0x80 has priority over 0x40.

Element	Standard Format	Advanced Format	Format:	Example
Data status	S:	01H	Hexadecimal word	0x0000
Name	N	02H	String	NC cycle time (TNcyc)
Attribute	A	04H	Hexadecimal double word	0x60110001
Unit	U	08H	String	µs
Min. input value	L	10H	Decimal word	2000
Max. input value	H	20H	Decimal word	20000
Operating date	D	40H	(see Displaying the Operating Date	
Operating date, when no list		80H		

Displaying the Operating Date The display of the operating date depends on the parameter number requested.

Decimal Decimal values are given as floating points, e.g. 1.5. Leading spaces, zeros, plus and minus signs as well as trailing spaces are allowed.

Hexadecimal

Hexadecimal values are displayed by "0x...", e.g. 0x80. Up to a maximum of eight positions are allowed. Leading or trailing spaces are allowed. Leading additional zeros or plus and minus signs are not allowed.

Binary (max. 32 characters)

Leading or trailing spaces are allowed. The decimal point serves as separator:

e.g., 1111.0000.1010.1100.1111.0000.1010.1100

Note: Leading additional zeros or plus and minus signs are not allowed.

Lists of Variable Length

Lists always begin with two decimal numbers for the actual length and maximum length of the list. The length specification refers to the length of the list in the drive and therefore designates the number of bytes for storage (storage bytes). The number of elements in the list can be calculated using the attribute. The list elements are displayed according to the attribute. All parts of the list are separated from each other by a line feed ("\n").

Example:

Parameter S-0-0017, IDN list of all parameters

"400\n400\nS-0-0001\nS-0-0002\n..."

ASCII List

ASCII lists are a special form of variable length lists. The individual string characters are not separated by a line feed. When displaying the lists, a distinction is made between standard format and advanced format. In standard format, only the character string is returned, whereas in advanced format the actual length and the maximum length of the list (string) is also transmitted.

Example:

Parameter S-0-0030, operation date

Standard format:	"DKC2.1-SSE-01V09"
Advanced format:	"16\n16\nDKC2.1-SSE-01V09"

Note: When requesting SERCANS parameters the drive address can be anywhere within the range [0..254].

Response Structure

The following table shows the general structure of the response to the FI command "SPA1". Line 1 is output both when reading and when writing. Additional lines are only output when reading depending on the element coding.

Notes: If the element coding has been requested in standard format then the first line is not applicable.

Line 1 is a status line that either contains SERCOS / SERCANS errors or displays the successful processing of the FI command. If the command has been processed successfully, then columns 1 and 3 contain the value [0x0000].

In the first line, column 2 or column 4, the number of the drive is output that reports the SERCOS error or the global SERCANS error. Not all global SERCANS errors can be directly assigned to a drive. In this case, the single result "Drive No." is set to 0x0000.

Line	Column 1	Column 2	Column 3	Column 4
1	<SERCOS error>	<Drive no. SERCOS error>	<Global SERCANS error>	<Drive No. Global SERCANS error>
2	Read: Element corresponding to the element coding.			
...	...			
n	Reading: (n-1). Element corresponding to the element coding.			

Example SPA1 / read Read parameter S-0-0003 of the 3rd drive (element coding 0x48)

FI command	**00_BR_SPA1_3_S-0-0003_48**			
		Answer		
Line	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	0x0000	0x0000	0x0000	0x0000
2	µs			
3	2000			

Example SPA1 / write Write the ID number P-0-0037 in parameter S-0-0305 of the 3rd drive (element coding 0x40).

Technical background:

- Real time status bit 1 is to be assigned the trigger status word of the oscilloscope function of a DIAX04 drive.

FI command	**Value to be written: : P-0-0037** **00_BW_SPA1_3_S-0-0305_40**			
		Answer		
Line	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	0x0000	0x0003	0x0000	0x0000

Reference to Literature See chapter entitled "Literature" [41].

See chapter entitled "Literature" [46].

12.50 Active SERCOS Phase Switch-Over: SPH

MSYX device group

Designation **SPH** **S**ERCOS **PH**ase

Explanation All drives within a SERCOS ring are in the same communication phase. The phase status can be read-out or changed by this command.

FI command

BR_SPH	**(Single Read)**
BC_SPH	**(Single Write)**
BB_SPH	**(Break Cyclic Read)**
BW_SPH	**(Single Write)**

Value to be written/ Result The phase conditions allowed are shown by the numbers

Response Structure The following table shows the general structure of the response to the FI command "SPH". .In the first line, column 2 or column 4, the number of the drive is output that reports the SERCOS error or the current system error. Not all current system errors can be directly allocated to a drive. In this case, the single result "Drive No." is set to 0x0000.

Line	Column 1	Column 2	Column 3	Column 4
1	SERCOS error	<Drive No. SERCOS error>	<Current system error>	<Drive no. that has caused the current system error>
2	current phase Write: former phase			

Example BR_SPH Read SERCOS Phase Read the active phase of the synax control at device address 00.

<table>
<tr><td>FI command</td><td colspan="4">00_BR_SPH</td></tr>
<tr><td colspan="5">Answer</td></tr>
<tr><td>Line</td><td>Column 1</td><td></td><td>Column 1</td><td>Line</td></tr>
<tr><td>1</td><td></td><td>1</td><td></td><td>1</td></tr>
<tr><td>2</td><td>2</td><td>2</td><td>2</td><td>2</td></tr>
</table>

Example BW_SPH Write SERCOS Phase Switch-over the synax control (write) after phase 4; phase 2 is active.

<table>
<tr><td>FI command</td><td colspan="4">Value to be written:4
00_BW_SPH</td></tr>
<tr><td colspan="5">Answer</td></tr>
<tr><td>Line</td><td>Column 1</td><td>Column 2</td><td>Column 3</td><td>Column 4</td></tr>
<tr><td>1</td><td>0x0000</td><td>0x0000</td><td>0x0000</td><td>0x0000</td></tr>
<tr><td>2</td><td>2</td><td></td><td></td><td></td></tr>
</table>

Reference to Literature See chapter entitled "Literature" [42]

12.51 Requesting Watch List Allocations: WLA

MWYX device groups

Designation **WLA** **W**atch **L**ist **A**llocation

Explanation Requests free watch list allocations. A maximum of ten free watch list allocations can be requested with one FI command.

BR_WLA1_(1)	**(Single Read)**
(1) =Number of requested free watch list allocations	The required number of free watch list allocations is identified here. The allowed value range: 1..10.

Response Structure The following table shows the general structure of the response to the FI command "WLA1".

Line 1	**Column 1**	**...**	**Column n**

Value Range/Meaning of Columns

1 =	1. free watch list allocation	Value range: 1..15
2 =	2. free watch list allocation	Value range: 1..15
3 =	3. free watch list allocation	Value range: 1..15
n =	nth free watch list allocation	Value range: 1..15

Example WLA1 Request four free watch list allocations.

Assumption:
Watch list allocations 3 and 5 are already assigned!

FI command		**00_BR_WLA1_4**
Line	**Column**	**Answer**
1	1	0
	2	1
	3	2
	4	4

12.52 Freeing Watch List Allocations: WLF

MWYX device groups

Designation **WLF** **W**atch **L**ist **F**ree

Explanation Previously requested watch list allocations are freed again.

FI command Free ALL assigned watch list allocations for the selected device.

BR_WLF1 **(Single Read)**

Note: The FI command "WLF1" frees ALL assigned watch list allocations, including those of other WIN32 applications.

Response Structure The following table shows the general structure of the response to the FI command "WLF1".

Line 1	**Column 1**	**...**	**Column n**

Value Range/Meaning of Columns

1 = 1. freed watch list allocation Value range: 1..15
2 = 2. freed watch list allocation Value range: 1..15
3 = 3. freed watch list allocation Value range: 1..15
n = nth freed watch list allocation Value range: 1..15

Example WLF1

Free ALL assigned watch list allocations.

Assumption:
The following watch list numbers have been allocated: 0,1,2,3.

FI command		**00_BR_WLF1**
Line	**Column**	**Answer**
1	1	0
	2	1
	3	2
	4	3

FI command

Free the required watch list allocations for a selected device.

BR_WLF2_(1)_{(2)..(10)} **(Single Read)**

(1)..(10) = List of watch list allocations to be released

A maximum of 10 watch list allocations can be transferred here to be freed again.

Response Structure

The following table shows the general structure of the response to the FI command "WLF2".

Line 1	**Column 1**	**...**	**Column n**

Value Range/Meaning of Columns

1 = 1. freed watch list allocation Value range: 1..15
2 = 2. freed watch list allocation Value range: 1..15
3 = 3. freed watch list allocation Value range: 1..15
n = nth freed watch list allocation Value range: 1..15

Example WLF2

Free required watch list allocations:
Assumption: Watch list allocations 0,3,4, and 8 have first been requested using the FI command "WLA1".

FI command		**00_BR_WLF2_0_3_4_8**
Line	**Column**	**Answer**
1	1	0
	2	3
	3	4
	4	8

13 Reference to Literature

13.1 Information in Bosch Rexroth Literature

[1]

More detailed information regarding acceleration value and value range is contained in the Bosch Rexroth documentation:

NC Programming Instructions, chapter entitled "Interpolation Requirements/ Programmable Acceleration ACC", DOK-MTC200-NC**PRO*Vxx-AW0x-EN-P.

[2]

More detailed information regarding the arguments of the trigonometric functions is contained in the Bosch Rexroth documentation:

NC Programming Instructions, chapter "Angle Dimension for Trigonometrical Functions RAD, DEG", DOK-MTC200-NC**PRO*Vxx-AW0x-EN-P.

[3]

More detailed information regarding the axis speeds is contained in the Bosch Rexroth documentation:

NC Programming Instructions, chapter entitled "Interpolation Functions/ Linear Interpolation, Rapid Traverse Rate G00", DOK-MTC200-NC**PRO*Vxx-AW0x-EN-P

Parameter Description, chapter entitled "Maximum Track Speed", DOK-MT*CNC-PAR*DES*Vxx-AW0x-EN-P.

[4]

More detailed information regarding the structure of an NC block is contained in the Bosch Rexroth documentation:

NC Programming Instructions, chapter entitled "Elements of an NC Block",
DOK-MTC200-NC**PRO*Vxx-AW0x-EN-P.

[5]

Additional information regarding the reference spindle as well as NC programming of the cutting speed is contained in the Bosch Rexroth documentation:

NC Programming Instructions, chapter entitled "Spindle Speed, Constant Cutting Speed G96 / Selection of Reference Spindle SPF", DOK-MTC200-NC**PRO*Vxx-AW0x-EN-P.

[6]

More detailed information regarding D-corrections is contained in the Bosch Rexroth documentation:

"NC Programming Instructions Vxx", chapter entitled "D-Corrections", DOK-MTC200-NC**PRO*Vxx-AW0x-EN-P.

[7]

More detailed information regarding events and their treatment is contained in the Bosch Rexroth documentation:

"NC Programming Instructions Vxx", chapter entitled "Events", DOK-MTC200-NC**PRO*Vxx-AW0x-EN-P.

[8] More detailed information regarding tool management is contained in the Bosch Rexroth documentation:

"NC Programming Instructions Vxx", chapter "Commands for Tool Management / Cutter Selection E",
DOK-MTC200-NC**PRO*Vxx-AW0x-EN-P.

[9] More detailed information regarding feedrate override is contained in the Bosch Rexroth documentation:

"CNC/PLC Interface Description xxVRS",
chapter entitled "Feedrate and Spindle Override PxxCSOVRD",
DOK-MTC200-SPS*GWY*Vxx-AW0x-EN-P.

[10] More detailed information regarding the feedrate is contained in the Bosch Rexroth documentation:

"CNC NC Programming Instructions Vxx", chapter entitled "Feedrate",
DOK-MTC200-NC**PRO*Vxx-AW0x-EN-P.

[11] More detailed information regarding the mode of operation of the G functions, as well as classification of the G-code groups, is contained in the Bosch Rexroth documentation:

"NC Programming Instructions Vxx",
chapter entitled "Table of G-Code Groups",
DOK-MTC200-NC**PRO*Vxx-AW0x-EN-P.

[12] More detailed information regarding the mode of operation of the M functions, as well as classification of the M function groups, is contained in the Bosch Rexroth documentation:

"NC Programming Instrucitons Vxx",
chapter entitled "Table of M Function Groups",
DOK-MTC200-NC**PRO*Vxx-AW01x-EN-P.

[13] More detailed information regarding the diagnostics system and the accompanying types of message is contained in the Bosch Rexroth documentation:

"xxVRS GUI", Application Description, Chapter 3 "Diagnostics", DOK-MTC200-GBO*GEN*Vxx-AW0x-EN-P.

[14] More detailed information regarding the machine parameters and their classification within the system, process, axis and APR-SERCOS parameters can be found in the Bosch Rexroth documentation:

"Rexroth MTC 200 MCI Operating Instructions xxVRS",
chapter entitled "Machine Parameters",
DOK-MTC200-GBO*MCI*Vxx-AW0x-EN-P

"Parameter Description",
DOK-MT*CNC-PAR*DES*Vxx-AW0x-EN-P.

[15] More detailed information regarding the elements of an NC record and the note is contained in the Bosch Rexroth documentation:

"CNC NC Programming Instructions Vxx", chapter entitled "NC Word", DOK-MTC200-NC**PRO*Vxx-AW0x-EN-P.

[16] More detailed information regarding the display possibilities within user interfaces, as well as the definition of axis data, is contained in the Bosch Rexroth documentation:

"Rexroth MTC 200 xxVRS GUI", chapter "Survey of Axis Data", DOK-MTC200-GBO*GEN*Vxx-AW0x-EN-P.

[17] More detailed information regarding the NC data structure is contained in the Bosch Rexroth documentation:

"NC Programming Instructions Vxx", chapter entitled "Program and Data Organization", DOK-MTC200-NC**PRO*Vxx-AW0x-EN-P.

[18] More detailed information regarding the rapid override is contained in the Bosch Rexroth documentation:

"CNC/PLC Interface Description xxVRS",
chapter entitled "Feedrate and Spindle Override"; "Rapid Override PxxCSOVRD",
DOK-MTC200-SPS*GWY*Vxx-AW0x-EN-P.

[19] Additional information regarding the selection of the reference spindle in the NC program is contained in the Bosch Rexroth documentation:

"NC NC Programming Instructions Vxx", Application Description,
chapter entitled "Spindle Speed", "Selecting the Reference Spindle SPF"
DOK-MTC200-NC**PRO*Vxx-AW0x-EN-P.

[20] Additional information regarding the selection of the spindle speed in the NC program is contained in the Bosch Rexroth documentation:

"NC Programming Instructions Vxx",
chapter entitled "Additional Functions M" / "Switching Gear",
DOK-MTC200-NC**PRO*Vxx-AW0x-EN-P.

[21] More detailed information regarding the spindle override is contained in the Bosch Rexroth documentation:

"CNC/PLC Interface Description xxVRS",
chapter entitled "Feedrate and Spindle Override PxxCSOVRD",
DOK-MTC200-SPS*GWY*Vxx-AW0x-EN-P.

[22] More detailed information regarding the axis speeds is contained in the Bosch Rexroth documentation:

"MT-CNC Numeric Control for Multiple Axis, Multiple Process Applications", chapter entitled "Maximum Track Acceleration",
DOK-MT*CNC-PAR*DES*Vxx-AW0x-EN-P.

[23] More detailed information regarding the structure and elements of the tool data is contained in the Bosch Rexroth documentation:

"CNC NC Programming Instructions Vxx, Application Description",
chapter entitled "Access to Tool Data by NC Program TLD",
DOK-MTC200-NC**PRO*Vxx-AW0x-EN-P.

[24] More detailed information regarding the use of zero-point offsets and zero offset tables is contained in the Bosch Rexroth documentation:

"CNC NC Programming Instructions Vxx", Application Description,
chapter "Zero-Point Offsets, Zero Offest Tables O",
DOK-MTC200-NC**PRO*Vxx-AW0x-EN-P.

[25] Additional information regarding the display of the axis position in the GBO is contained in the Bosch Rexroth documentation:

"Rexroth MTC 200 xxVRS GUI",
DOK-MTC200-GBO*GEN*Vxx-AW0x-EN-P.

[26] More detailed information on resetting the device is contained in the Bosch Rexroth documentation:

"PLC Programming Instructions xxVRS", Chapter "Putting into Operation / Start", DOK-CONTRL-SPS*PRO*Vxx-AW0x-EN-P.

[27] More detailed information regarding the configuration of the device axes is contained in the Bosch Rexroth documentation:

"Parameter Description", chapter "System Parameters" as well as chapter "Axis Parameters", DOK-MT*CNC-PAR*DES*Vxx-AW0x-EN-P.

[28] Additional information regarding process parameters and their functions as well as value ranges is contained in the Bosch Rexroth documentation:

"CNC/PLC Interface Description xxVRS Application Description,
chapter "External Mechanisms",
DOK-MTC200-SPS*GWY*Vxx-AW0x-EN-P.

[29] Additional information regarding the function of the NC parameters and the structure of the NC parameter records is contained in the Bosch Rexroth documentation:

"Rexroth MTC 200 Parameter Description xxVRS",
DOK-MTC200-PAR*DES*Vxx-AW0x-EN-P.

[30] More detailed information concerning the PLC Programming System is contained in the Bosch Rexroth documentation:

"PLC Programming Instructions xxVRS Application Description" DOK-CONTRL-SPS*PRO*Vxx-AW0x-EN-P.

[31] More detailed information regarding the structure of NC packages is contained in the Bosch Rexroth documentation:

"Rexroth MTC 200 NC Programming Instructions xxVRS", chapter "Sub-Programs",
DOK-MTC200-NC**PRO*Vxx-AW0x-EN-P.

[32] More detailed information regarding the structure of tool lists is contained in the Bosch Rexroth documentation:

"Rexroth MTC 200 xxVRS GUI", chapter "Tool Data Handling BOF" and chapter "Tool Data Handling GBO", DOK-MTC200-GBO*GEN*Vxx-AW0x-EN-P.

[33] More detailed information regarding the use of machine data is contained in the Bosch Rexroth documentation:

"CNC Machine Data xxVRS Application Description"
DOK-MT*CNC-MAS*DAT*Vxx-AW0x-EN-P.

[34] Additional information regarding process parameters and their functions as well as value ranges is contained in the Bosch Rexroth documentation:

"Parameter Description" chapter "Process Parameters"
DOK-MT*CNC-PAR*DES*Vxx-AW0x-EN-P.

[35] Additional information regarding process parameters and their functions as well as value ranges is contained in the Bosch Rexroth documentation:

"MT-CNC Numeric Control for Multiple Axes, Multiple Process Applications", Chapter 2 "Process Parameters",
DOK-MT*CNC-PAR*DES*V15-ANW1-EN-P.

[36] Additional information regarding module configuration and the structure of the "Moduldef.ini" file is contained in the following Bosch Rexroth documentation:

"Diagnostics and Message System for HMI System ProVi", chapter "Configure Moduldef.ini", DOK-MTC200-DIAG*PROVI*-AW0x-EN-P.

[37] More detailed information on selecting the NC program and the NC memory is contained in the Bosch Rexroth documentation:

"Rexroth MTC 200 xxVRS GUI",
chapter entitled "Operation Survey of the Administration of NC Programs",
DOK-MTC200-GBO*GEN*Vxx-AW0x-EN-P.

[38] More detailed information regarding the contents of parameter records is contained in the Bosch Rexroth documentation:

"Rexroth MTC 200 Parameter Description xxVRS",
chapter entitled "Processing / Displaying Contents of Parameter Records",
DOK-MTC200-PAR*DES*Vxx-AW0x-EN-P.

[39] More detailed information regarding NC variables is contained in the Bosch Rexroth documentation:

"NC Programming Instructions Vxx",
chapter entitled "Assigning Variables and Mathematical Functions",
DOK-MTC200-NC**PRO*Vxx-AW0x-EN-P.

[40] More detailed information regarding servo lag is contained in the Bosch Rexroth documentation:

"NC Programming Instructions Vxx",
chapter entitled "Movement Records and Interpolation Requirements",
DOK-MTC200-NC**PRO*Vxx-AW0x-EN-P.

[41] Additional information regarding the function of the standard and product-specific SERCOS parameters (S and P) is contained in the Bosch Rexroth Documentation:

"DIAX04 Drive with Servo Functions", Appendix A Description of Parameters, DOK-DIAX04-SSE-02VRS**-FKB1-EN-P.

[42] More detailed information regarding the communication phases is contained in the Bosch Rexroth documentation:

"DIAX04 Drive with Servo Functions", General Instructions on Putting into Operation, DOK-DIAX04-SSE-xxVRS**-FKBx-EN-P.

[43] More detailed information regarding tool management is contained in the Bosch Rexroth documentation:

"MT-CNC Numeric Control for Multiple Axes, Multi-Process Applications, Tool Data Handling",
DOK-MT*CNC-BOF*WZH*Vxx-AW0x-EN-P.

[44] More detailed information regarding zero offsets is contained in the Bosch Rexroth documentation:

Rexroth MTC 200 NC Programming Instructions xxVRS",

chapter entitled "Zero Offset", and chapter entitled "Reading and Writing of the Zero Offset Data from the NC Program OTD",

DOK-MTC200-NC**PRO*Vxx-AW0x-EN-P.

[45] More detailed information regarding SERCANS errors is contained in the Bosch Rexroth documentation:

"SERCANS /SERCVME SERCOS Interface Assemblies with Universal µP Interface or VMEbus", Application Description, System Structure and Axis Structure.

[46] Additional information regarding the function of the SERCANS System Parameters (Y) is contained in the Bosch Rexroth Documentation:

"SERCANS SERCOS Interface Assemblies", Chapter 10 "Description of Parameters",
DOK-SERCAN-SER-VxxVRS**-AW0x-EN-P.

14 Index

15 Service & Support

15.1 Helpdesk

Unser Kundendienst-Helpdesk im Hauptwerk Lohr am Main steht Ihnen mit Rat und Tat zur Seite. Sie erreichen uns

Our service helpdesk at our headquarters in Lohr am Main, Germany can assist you in all kinds of inquiries. Contact us

- telefonisch - by phone: **49 (0) 9352 40 50 60**
  über Service Call Entry Center Mo-Fr 07:00-18:00
  - via Service Call Entry Center Mo-Fr 7:00 am - 6:00 pm
- per Fax - by fax: **+49 (0) 9352 40 49 41**
- per e-Mail - by e-mail: **service.svc@boschrexroth.de**

15.2 Service-Hotline

Außerhalb der Helpdesk-Zeiten ist der Service direkt ansprechbar unter

After helpdesk hours, contact our service department directly at

+49 (0) 171 333 88 26

oder - or **+49 (0) 172 660 04 06**

15.3 Internet

Unter **www.boschrexroth.com** finden Sie ergänzende Hinweise zu Service, Reparatur und Training sowie die **aktuellen** Adressen *) unserer auf den folgenden Seiten aufgeführten Vertriebs- und Servicebüros.

☐ Verkaufsniederlassungen
▨ Niederlassungen mit Kundendienst

Außerhalb Deutschlands nehmen Sie bitte zuerst Kontakt mit unserem für Sie nächstgelegenen Ansprechpartner auf.

*) Die Angaben in der vorliegenden Dokumentation können seit Drucklegung überholt sein.

At **www.boschrexroth.com** you may find additional notes about service, repairs and training in the Internet, as well as the **actual** addresses *) of our sales- and service facilities figuring on the following pages.

☐ sales agencies
▨ offices providing service

Please contact our sales / service office in your area first.

*) Data in the present documentation may have become obsolete since printing.

15.4 Vor der Kontaktaufnahme... - Before contacting us...

Wir können Ihnen schnell und effizient helfen wenn Sie folgende Informationen bereithalten:

1. detaillierte Beschreibung der Störung und der Umstände.
2. Angaben auf dem Typenschild der betreffenden Produkte, insbesondere Typenschlüssel und Seriennummern.
3. Tel.-/Faxnummern und e-Mail-Adresse, unter denen Sie für Rückfragen zu erreichen sind.

For quick and efficient help, please have the following information ready:

1. Detailed description of the failure and circumstances.
2. Information on the type plate of the affected products, especially type codes and serial numbers.
3. Your phone/fax numbers and e-mail address, so we can contact you in case of questions.

15.5 Kundenbetreuungsstellen - Sales & Service Facilities

Deutschland - Germany

vom Ausland:	(0) nach Landeskennziffer weglassen!
from abroad:	don't dial (0) after country code!

Vertriebsgebiet Mitte Germany Centre Rexroth Indramat GmbH Bgm.-Dr.-Nebel-Str. 2 / Postf. 1357 97816 Lohr am Main / 97803 Lohr **Kompetenz-Zentrum Europa** Tel.: +49 (0)9352 40-0 Fax: +49 (0)9352 40-4885	**SERVICE** **CALL ENTRY CENTER** **MO – FR** **von 07:00 - 18:00 Uhr** **from 7 am – 6 pm** **Tel. +49 (0) 9352 40 50 60** service.svc@boschrexroth.de	**SERVICE** **HOTLINE** **MO – FR** **von 17:00 - 07:00 Uhr** **from 5 pm - 7 am** **+ SA / SO** **Tel.: +49 (0)172 660 04 06** **oder / or** **Tel.: +49 (0)171 333 88 26**	**SERVICE** **ERSATZTEILE / SPARES** verlängerte Ansprechzeit - extended office time - ♦ nur an Werktagen - only on working days - ♦ von 07:00 - 18:00 Uhr - from 7 am - 6 pm - **Tel. +49 (0) 9352 40 42 22**
Vertriebsgebiet Süd Germany South Bosch Rexroth AG Landshuter Allee 8-10 80637 München Tel.: +49 (0)89 127 14-0 Fax: +49 (0)89 127 14-490	Vertriebsgebiet West Germany West Bosch Rexroth AG Regionalzentrum West Borsigstrasse 15 40880 Ratingen Tel.: +49 (0)2102 409-0 Fax: +49 (0)2102 409-406 +49 (0)2102 409-430	Gebiet Südwest Germany South-West Bosch Rexroth AG Service-Regionalzentrum Süd-West Siemensstr.1 70736 Fellbach Tel.: +49 (0)711 51046–0 Fax: +49 (0)711 51046–248	
Vertriebsgebiet Nord Germany North Bosch Rexroth AG Walsroder Str. 93 30853 Langenhagen Tel.: +49 (0) 511 72 66 57-0 Service: +49 (0) 511 72 66 57-256 Fax: +49 (0) 511 72 66 57-93 Service: +49 (0) 511 72 66 57-783	Vertriebsgebiet Mitte Germany Centre Bosch Rexroth AG Regionalzentrum Mitte Waldecker Straße 13 64546 Mörfelden-Walldorf Tel.: +49 (0) 61 05 702-3 Fax: +49 (0) 61 05 702-444	Vertriebsgebiet Ost Germany East Bosch Rexroth AG Beckerstraße 31 09120 Chemnitz Tel.: +49 (0)371 35 55-0 Fax: +49 (0)371 35 55-333	Vertriebsgebiet Ost Germany East Bosch Rexroth AG Regionalzentrum Ost Walter-Köhn-Str. 4d 04356 Leipzig Tel.: +49 (0)341 25 61-0 Fax: +49 (0)341 25 61-111

Europa (West) - Europe (West)

vom Ausland: (0) nach Landeskennziffer weglassen,
from abroad: don't dial (0) after country code,

Italien: 0 nach Landeskennziffer mitwählen
Italy: dial 0 after country code

Austria - Österreich

Bosch Rexroth GmbH
Electric Drives & Controls
Stachegasse 13
1120 Wien
Tel.: +43 (0)1 985 25 40
Fax: +43 (0)1 985 25 40-93

Austria – Österreich

Bosch Rexroth GmbH
Electric Drives & Controls
Industriepark 18
4061 Pasching
Tel.: +43 (0)7221 605-0
Fax: +43 (0)7221 605-21

Belgium - Belgien

Bosch Rexroth NV/SA
Henri Genessestraat 1
1070 Bruxelles

Tel: +32 (0) 2 582 31 80
Fax: +32 (0) 2 582 43 10
info@boschrexroth.be
service@boschrexroth.be

Denmark - Dänemark

BEC A/S
Zinkvej 6
8900 Randers

Tel.: +45 (0)87 11 90 60
Fax: +45 (0)87 11 90 61

Great Britain – Großbritannien

Bosch Rexroth Ltd.
Electric Drives & Controls
Broadway Lane, South Cerney
Cirencester, Glos GL7 5UH

Tel.: +44 (0)1285 863000
Fax: +44 (0)1285 863030
sales@boschrexroth.co.uk
service@boschrexroth.co.uk

Finland - Finnland

Bosch Rexroth Oy
Electric Drives & Controls
Ansatie 6
017 40 Vantaa

Tel.: +358 (0)9 84 91-11
Fax: +358 (0)9 84 91-13 60

France - Frankreich

Bosch Rexroth SAS
Electric Drives & Controls
Avenue de la Trentaine
(BP. 74)
77503 Chelles Cedex
Tel.: +33 (0)164 72-70 00
Fax: +33 (0)164 72-63 00
Hotline: +33 (0)608 33 43 28

France - Frankreich

Bosch Rexroth SAS
Electric Drives & Controls
ZI de Thibaud, 20 bd. Thibaud
(BP. 1751)
31084 Toulouse
Tel.: +33 (0)5 61 43 61 87
Fax: +33 (0)5 61 43 94 12

France – Frankreich

Bosch Rexroth SAS
Electric Drives & Controls
91, Bd. Irène Joliot-Curie
69634 Vénissieux – Cedex
Tel.: +33 (0)4 78 78 53 65
Fax: +33 (0)4 78 78 53 62

Italy - Italien

Bosch Rexroth S.p.A.
Via G. Di Vittorio, 1
20063 Cernusco S/N.MI
Hotline: +39 02 92 365 563
Tel.: +39 02 92 365 1
Service: +39 02 92 365 326
Fax: +39 02 92 365 500
Service: +39 02 92 365 503

Italy - Italien

Bosch Rexroth S.p.A.
Via Paolo Veronesi, 250
10148 Torino

Tel.: +39 011 224 88 11
Fax: +39 011 224 88 30

Italy - Italien

Bosch Rexroth S.p.A.
Via Mascia, 1
80053 Castellamare di Stabia NA

Tel.: +39 081 8 71 57 00
Fax: +39 081 8 71 68 85

Italy - Italien

Bosch Rexroth S.p.A.
Via del Progresso, 16 (Zona Ind.)
35020 Padova

Tel.: +39 049 8 70 13 70
Fax: +39 049 8 70 13 77

Italy - Italien

Bosch Rexroth S.p.A.
Via Isonzo, 61
40033 Casalecchio di Reno (Bo)

Tel.: +39 051 29 86 430
Fax: +39 051 29 86 490

Netherlands - Niederlande/Holland

Bosch Rexroth Services B.V.
Technical Services
Kruisbroeksestraat 1
(P.O. Box 32)
5281 RV Boxtel
Tel.: +31 (0) 411 65 16 40
+31 (0) 411 65 17 27
Fax: +31 (0) 411 67 78 14
+31 (0) 411 68 28 60
services@boschrexroth.nl

Netherlands – Niederlande/Holland

Bosch Rexroth B.V.
Kruisbroeksestraat 1
(P.O. Box 32)
5281 RV Boxtel

Tel.: +31 (0) 411 65 19 51
Fax: +31 (0) 411 65 14 83
www.boschrexroth.nl

Norway - Norwegen

Bosch Rexroth AS
Electric Drives & Controls
Berghagan 1 or: Box 3007
1405 Ski-Langhus 1402 Ski

Tel.: +47 (0) 64 86 41 00

Fax: +47 (0) 64 86 90 62

Hotline: +47 (0)64 86 94 82
jul.ruud@rexroth.no

Spain - Spanien

Bosch Rexroth S.A.
Electric Drives & Controls
Centro Industrial Santiga
Obradors s/n
08130 Santa Perpetua de Mogoda
Barcelona
Tel.: +34 9 37 47 94 00
Fax: +34 9 37 47 94 01

Spain – Spanien

Goimendi S.A.
Electric Drives & Controls
Parque Empresarial Zuatzu
C/ Francisco Grandmontagne no.2
20018 San Sebastian

Tel.: +34 9 43 31 84 21
- service: +34 9 43 31 84 56
Fax: +34 9 43 31 84 27
- service: +34 9 43 31 84 60
sat.indramat@goimendi.es

Sweden - Schweden

Bosch Rexroth AB
Electric Drives & Controls
- Varuvägen 7
(Service: Konsumentvägen 4, Älfsjö)
125 81 Stockholm

Tel.: +46 (0)8 727 92 00
Fax: +46 (0)8 647 32 77

Sweden - Schweden

Bosch Rexroth AB
Electric Drives & Controls
Ekvändan 7
254 67 Helsingborg
Tel.: +46 (0) 42 38 88 -50
Fax: +46 (0) 42 38 88 -74

Switzerland East - Schweiz Ost

Bosch Rexroth Schweiz AG
Electric Drives & Controls
Hemrietstrasse 2
8863 Buttikon
Tel. +41 (0) 55 46 46 111
Fax +41 (0) 55 46 46 222

Switzerland West - Schweiz West

Bosch Rexroth Suisse SA
Av. Général Guisan 26
1800 Vevey 1

Tel.: +41 (0)21 632 84 20
Fax: +41 (0)21 632 84 21

Europa (Ost) - Europe (East)

vom Ausland: (0) nach Landeskennziffer weglassen
from abroad: don't dial (0) after country code

Czech Republic - Tschechien
Bosch -Rexroth, spol.s.r.o.
Hviezdoslavova 5
627 00 Brno
Tel.: +420 (0)5 48 126 358
Fax: +420 (0)5 48 126 112

Czech Republic - Tschechien
DEL a.s.
Strojírenská 38
591 01 Zdar nad Sázavou
Tel.: +420 566 64 3144
Fax: +420 566 62 1657

Hungary - Ungarn
Bosch Rexroth Kft.
Angol utca 34
1149 Budapest
Tel.: +36 (1) 422 3200
Fax: +36 (1) 422 3201

Poland – Polen
Bosch Rexroth Sp.zo.o.
ul. Staszica 1
05-800 Pruszków
Tel.: +48 22 738 18 00
– service: +48 22 738 18 46
Fax: +48 22 758 87 35
– service: +48 22 738 18 42

Poland – Polen
Bosch Rexroth Sp.zo.o.
Biuro Poznan
ul. Dabrowskiego 81/85
60-529 Poznan
Tel.: +48 061 847 64 62 /-63
Fax: +48 061 847 64 02

Romania - Rumänien
East Electric S.R.L.
Bdul Basarabia no.250, sector 3
73429 Bucuresti
Tel./Fax:: +40 (0)21 255 35 07
+40 (0)21 255 77 13
Fax: +40 (0)21 725 61 21
eastel@rdsnet.ro

Romania - Rumänien
Bosch Rexroth Sp.zo.o.
Str. Drobety nr. 4-10, app. 14
70258 Bucuresti, Sector 2
Tel.: +40 (0)1 210 48 25
+40 (0)1 210 29 50
Fax: +40 (0)1 210 29 52

Russia - Russland
Bosch Rexroth OOO
Wjatskaja ul. 27/15
127015 Moskau
Tel.: +7-095-785 74 78
+7-095 785 74 79
Fax: +7 095 785 74 77
laura.kanina@boschrexroth.ru

Russia - Russland
ELMIS
10, Internationalnaya
246640 Gomel, Belarus
Tel.: +375/ 232 53 42 70
+375/ 232 53 21 69
Fax: +375/ 232 53 37 69
elmis_ltd@yahoo.com

Turkey - Türkei
Bosch Rexroth Otomasyon
San & Tic. A..S.
Fevzi Cakmak Cad No. 3
34630 Sefaköy Istanbul
Tel.: +90 212 413 34 00
Fax: +90 212 413 34 17
www.boschrexroth.com.tr

Turkey - Türkei
Servo Kontrol Ltd. Sti.
Perpa Ticaret Merkezi B Blok
Kat: 11 No: 1609
80270 Okmeydani-Istanbul
Tel: +90 212 320 30 80
Fax: +90 212 320 30 81
remzi.sali@servokontrol.com
www.servokontrol.com

Slowenia - Slowenien
DOMEL
Otoki 21
64 228 Zelezniki

Tel.: +386 5 5117 152
Fax: +386 5 5117 225
brane.ozebek@domel.si

Africa, Asia, Australia – incl. Pacific Rim

Australia - Australien
AIMS - Australian Industrial
Machinery Services Pty. Ltd.
28 Westside Drive
Laverton North Vic 3026
Melbourne
Tel.: +61 3 93 14 3321
Fax: +61 3 93 14 3329
Hotlines: +61 3 93 14 3321
+61 4 19 369 195
enquires@aimservices.com.au

Australia - Australien
Bosch Rexroth Pty. Ltd.
No. 7, Endeavour Way
Braeside Victoria, 31 95
Melbourne
Tel.: +61 3 95 80 39 33
Fax: +61 3 95 80 17 33
mel@rexroth.com.au

China
Shanghai Bosch Rexroth
Hydraulics & Automation Ltd.
Waigaoqiao, Free Trade Zone
No.122, Fu Te Dong Yi Road
Shanghai 200131 - P.R.China
Tel.: +86 21 58 66 30 30
Fax: +86 21 58 66 55 23
richard.yang_sh@boschrexroth.com.cn
gf.zhu_sh@boschrexroth.com.cn

China
Shanghai Bosch Rexroth
Hydraulics & Automation Ltd.
4/f, Marine Tower
No.1, Pudong Avenue
Shanghai 200120 - P.R.China
Tel: +86 21 68 86 15 88
Fax: +86 21 58 40 65 77

China
Bosch Rexroth China Ltd.
15/F China World Trade Center
1, Jianguomenwai Avenue
Beijing 100004, P.R.China
Tel.: +86 10 65 05 03 80
Fax: +86 10 65 05 03 79

China
Bosch Rexroth China Ltd.
Guangzhou Repres. Office
Room 1014-1016, Metro Plaza,
Tian He District, 183 Tian He Bei Rd
Guangzhou 510075, P.R.China
Tel.: +86 20 8755-0030
+86 20 8755-0011
Fax: +86 20 8755-2387

China
Bosch Rexroth (China) Ltd.
A-5F., 123 Lian Shan Street
Sha He Kou District
Dalian 116 023, P.R.China
Tel.: +86 411 46 78 930
Fax: +86 411 46 78 932

China
Melchers GmbH
BRC-SE, Tightening & Press-fit
13 Floor Est Ocean Centre
No.588 Yanan Rd. East
65 Yanan Rd. West
Shanghai 200001
Tel.: +86 21 6352 8848
Fax: +86 21 6351 3138

Hongkong
Bosch Rexroth (China) Ltd.
6th Floor,
Yeung Yiu Chung No.6 Ind Bldg.
19 Cheung Shun Street
Cheung Sha Wan,
Kowloon, Hongkong
Tel.: +852 22 62 51 00
Fax: +852 27 41 33 44
alexis.siu@boschrexroth.com.hk

India - Indien
Bosch Rexroth (India) Ltd.
Electric Drives & Controls
Plot. No.96, Phase III
Peenya Industrial Area
Bangalore – 560058
Tel.: +91 80 51 17 0-211...-218
Fax: +91 80 83 94 345
+91 80 83 97 374
mohanvelu.t@boschrexroth.co.in

India - Indien
Bosch Rexroth (India) Ltd.
Electric Drives & Controls
Advance House, II Floor
Ark Industrial Compound
Narol Naka, Makwana Road
Andheri (East), Mumbai - 400 059
Tel.: +91 22 28 56 32 90
+91 22 28 56 33 18
Fax: +91 22 28 56 32 93
singh.op@boschrexroth.co.in

India - Indien
Bosch Rexroth (India) Ltd.
S-10, Green Park Extension
New Delhi – 110016
Tel.: +91 11 26 56 65 25
+91 11 26 56 65 27
Fax: +91 11 26 56 68 87
koul.rp@boschrexroth.co.in

Indonesia - Indonesien
PT. Bosch Rexroth
Building # 202, Cilandak
Commercial Estate
Jl. Cilandak KKO, Jakarta 12560
Tel.: +62 21 7891169 (5 lines)
Fax: +62 21 7891170 - 71
rudy.karimun@boschrexroth.co.id

Japan
Bosch Rexroth Automation Corp.
Service Center Japan
Yutakagaoka 1810, Meito-ku,
NAGOYA 465-0035, Japan
Tel.: +81 52 777 88 41
+81 52 777 88 53
+81 52 777 88 79
Fax: +81 52 777 89 01

Japan
Bosch Rexroth Automation Corp.
Electric Drives & Controls
2F, I.R. Building
Nakamachidai 4-26-44, Tsuzuki-ku
YOKOHAMA 224-0041, Japan
Tel.: +81 45 942 72 10
Fax: +81 45 942 03 41

Korea
Bosch Rexroth-Korea Ltd.
Electric Drives and Controls
Bongwoo Bldg. 7FL, 31-7, 1Ga
Jangchoong-dong, Jung-gu
Seoul, 100-391
Tel.: +82 234 061 813
Fax: +82 222 641 295

Korea
Bosch Rexroth-Korea Ltd.
1515-14 Dadae-Dong, Saha-gu
Electric Drives & Controls
Pusan Metropolitan City, 604-050
Tel.: +82 51 26 00 741
Fax: +82 51 26 00 747
eunkyong.kim@boschrexroth.co.kr

Malaysia
Bosch Rexroth Sdn.Bhd.
11, Jalan U8/82, Seksyen U8
40150 Shah Alam
Selangor, Malaysia
Tel.: +60 3 78 44 80 00
Fax: +60 3 78 45 48 00
hockhwa@hotmail.com
rexroth1@tm.net.my

Singapore - Singapur
Bosch Rexroth Pte Ltd
15D Tuas Road
Singapore 638520
Tel.: +65 68 61 87 33
Fax: +65 68 61 18 25
sanjay.nemade
@boschrexroth.com.sg

South Africa - Südafrika
TECTRA Automation (Pty) Ltd.
71 Watt Street, Meadowdale
Edenvale 1609
Tel.: +27 11 971 94 00
Fax: +27 11 971 94 40
Hotline: +27 82 903 29 23
georgv@tectra.co.za

Taiwan
Bosch Rexroth Co., Ltd.
Taichung Branch
1F., No. 29, Fu-Ann 5th Street,
Xi-Tun Area, Taichung City
Taiwan, R.O.C.
Tel : +886 - 4 -23580400
Fax: +886 - 4 -23580402
jim.lin@boschrexroth.com.tw
david.lai@boschrexroth.com.tw

Taiwan
Bosch Rexroth Co., Ltd.
Tainan Branch
No. 17, Alley 24, Lane 737
Chung Cheng N.Rd. Yungkang
Tainan Hsien, Taiwan, R.O.C.
Tel : +886 - 6 –253 6565
Fax: +886 - 6 –253 4754
charlie.chen@boschrexroth.com.tw

Thailand
NC Advance Technology Co. Ltd.
59/76 Moo 9
Ramintra road 34
Tharang, Bangkhen,
Bangkok 10230
Tel.: +66 2 943 70 62
+66 2 943 71 21
Fax: +66 2 509 23 62
Hotline +66 1 984 61 52
sonkawin@hotmail.com

Nordamerika – North America

USA
Headquarters - Hauptniederlassung

Bosch Rexroth Corporation
Electric Drives & Controls
5150 Prairie Stone Parkway
Hoffman Estates, IL 60192-3707

Tel.: +1 847 6 45 36 00
Fax: +1 847 6 45 62 01
servicebrc@boschrexroth-us.com
repairbrc@boschrexroth-us.com

USA Central Region - Mitte

Bosch Rexroth Corporation
Electric Drives & Controls
Central Region Technical Center
1701 Harmon Road
Auburn Hills, MI 48326

Tel.: +1 248 3 93 33 30
Fax: +1 248 3 93 29 06

USA Southeast Region - Südwest

Bosch Rexroth Corporation
Electric Drives & Controls
Southeastern Technical Center
3625 Swiftwater Park Drive
Suwanee, Georgia 30124

Tel.: +1 770 9 32 32 00
Fax: +1 770 9 32 19 03

USA SERVICE-HOTLINE

- 7 days x 24hrs -

+1-800-REX-ROTH
+1 800 739 7684

USA East Region – Ost

Bosch Rexroth Corporation
Electric Drives & Controls
Charlotte Regional Sales Office
14001 South Lakes Drive
Charlotte, North Carolina 28273

Tel.: +1 704 5 83 97 62
+1 704 5 83 14 86

USA Northeast Region – Nordost

Bosch Rexroth Corporation
Electric Drives & Controls
Northeastern Technical Center
99 Rainbow Road
East Granby, Connecticut 06026

Tel.: +1 860 8 44 83 77
Fax: +1 860 8 44 85 95

USA West Region – West

Bosch Rexroth Corporation
7901 Stoneridge Drive, Suite 220
Pleasant Hill, California 94588

Tel.: +1 925 227 10 84
Fax: +1 925 227 10 81

Canada East - Kanada Ost

Bosch Rexroth Canada Corporation
Burlington Division
3426 Mainway Drive
Burlington, Ontario
Canada L7M 1A8

Tel.: +1 905 335 5511
Fax: +1 905 335 4184
Hotline: +1 905 335 5511
michael.moro@boschrexroth.ca

Canada West - Kanada West

Bosch Rexroth Canada Corporation
5345 Goring St.
Burnaby, British Columbia
Canada V7J 1R1

Tel. +1 604 205 5777
Fax +1 604 205 6944
Hotline: +1 604 205 5777
david.gunby@boschrexroth.ca

Mexico

Bosch Rexroth Mexico S.A. de C.V.
Calle Neptuno 72
Unidad Ind. Vallejo
07700 Mexico, D.F.

Tel.: +52 55 57 54 17 11
Fax: +52 55 57 54 50 73
mariofelipe.hernandez@boschrexroth.com.mx

Mexico

Bosch Rexroth S.A. de C.V.
Calle Argentina No 3913
Fracc. las Torres
64930 Monterrey, N.L.

Tel.: +52 81 83 65 22 53
+52 81 83 65 89 11
+52 81 83 49 80 91
Fax: +52 81 83 65 52 80
mario.quiroga@boschrexroth.com.mx

Südamerika – South America

Argentina - Argentinien

Bosch Rexroth S.A.I.C.
"The Drive & Control Company"
Rosario 2302
B1606DLD Carapachay
Provincia de Buenos Aires

Tel.: +54 11 4756 01 40
+54 11 4756 02 40
+54 11 4756 03 40
+54 11 4756 04 40
Fax: +54 11 4756 01 36
+54 11 4721 91 53
victor.jabif@boschrexroth.com.ar

Argentina - Argentinien

NAKASE
Servicio Tecnico CNC
Calle 49, No. 5764/66
B1653AOX Villa Balester
Provincia de Buenos Aires

Tel.: +54 11 4768 36 43
Fax: +54 11 4768 24 13
Hotline: +54 11 155 307 6781
nakase@usa.net
nakase@nakase.com
gerencia@nakase.com (Service)

Brazil - Brasilien

Bosch Rexroth Ltda.
Av. Tégula, 888
Ponte Alta, Atibaia SP
CEP 12942-440

Tel.: +55 11 4414 56 92
+55 11 4414 56 84
Fax sales: +55 11 4414 57 07
Fax serv.: +55 11 4414 56 86
alexandre.wittwer@rexroth.com.br

Brazil - Brasilien

Bosch Rexroth Ltda.
R. Dr.Humberto Pinheiro Vieira, 100
Distrito Industrial [Caixa Postal 1273]
89220-390 Joinville - SC

Tel./Fax: +55 47 473 58 33
Mobil: +55 47 9974 6645
prochnow@zaz.com.br

Columbia - Kolumbien

Reflutec de Colombia Ltda.
Calle 37 No. 22-31
Santafé de Bogotá, D.C.
Colombia

Tel.: +57 1 368 82 67
+57 1 368 02 59
Fax: +57 1 268 97 37
reflutec@neutel.com.co
reflutec@007mundo.com

Notes

Rexroth
Bosch Group

Bosch Rexroth AG
Electric Drives and Controls
P.O. Box 13 57
97803 Lohr, Germany
Bgm.-Dr.-Nebel-Str. 2
97816 Lohr, Germany
Phone +49 93 52-40-50 60
Fax +49 93 52-40-49 41
service.svc@boschrexroth.de
www.boschrexroth.com

R911299217

DOK-CONTRL-FUN*COM*V08-AW01-EN-P