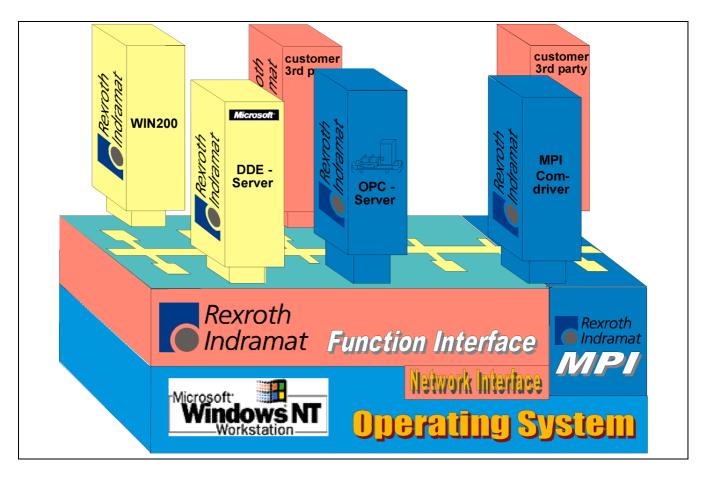
Rexroth



MTC200/ISP200/MTA200/TRANS200 Function Interface 07VRS

Application Manual

SYSTEM200



DOK-CONTRL-FUN*INT*V07-AW01-EN-P

Title	MTC200/ISP200/MTA200/TRANS200 Function Interface
Turne of Deserves (offer	07VRS
Type of Documentation	Application Manual
Document Typecode	DOK-CONTRL-FUN*INT*V07-AW01-EN-P
Internal File Reference	Document Number 120-0400-B375-01/EN
Purpose of Documentation	This documentation is used:
	 to give an overview of the function interface functionalities
	 to define the application possibilities and
	 for projecting and development of the user-defined user interfaces in C/C ++ and Visual Basic.

evisions

Description	Release Date	Notes
120-0400-B375-01/EN	10.02	Valid from version 22

Copyright	© 2002 Rexroth Indramat GmbH Copying this document, giving it to others and the use or communication of the contents thereof without express authority, are forbidden. Offenders are liable for the payment of damages. All rights are reserved in the event of the grant of a patent or the registration of a utility model or design (DIN 34-1).
Validity	The specified data is for product description purposes only and may not be deemed to be guaranteed unless expressly confirmed in the contract. All rights are reserved with respect to the content of this documentation and the availability of the product.
Published by	Rexroth Indramat GmbH BgmDrNebel-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.de/ Dept. BRC/ESS1 (FW) Dept. BRC/ESM6 (DiHa)
Note	This document has been printed on chlorine-free bleached paper.

Contents

1	Nev	w in Versions 07VRS	1-1
	1.1	Version 07V00	1-1
	1.2	Version 06V00	1-4
	1.3	Version 05V00	1-7
	1.4	Version 04V03	1-8
	1.5	Version 04V02	1-10
	1.6	Version 04V01	1-10
	1.7	Version 04V00	1-11
	1.8	The Data Interface Newsletter	1-12
2	Gei	neral	2-1
	2.1	Introduction	2-1
	2.2	The Function Interface from the User's Point of View	2-1
	2.3	Protection against dangerous movements	2-2
3	Str	ucture and Configuration Examples	3-1
	3.1	The Structure of the Function Interface	
		Logic process	
		Communication process	
		Management process	
	3.2	Configuration Examples and Connection Options	3-3
		MPI Connection with Profibus FMS	3-3
		Rexroth Indramat GUI and Server	3-4
		Connection to the function interface	
		Communication between a Client and Rexroth Indramat Devices	3-5
		Communication between several clients and Rexroth Indramat devices	3-6
4	Pro	gramming	4-1
	4.1	Guidelines	
		Software for Developing User Programs (Clients) (PRO VERSION)	
		Settings for the C++ Development Environment	
	4.2	Routines for Logging in and Logging Out	
		"LogInIf" Login Routine	
		"LogOutIf" Log out Routine	
	4.3	Data Transfer and Result Evaluation Routines	
		"DataTransfer" Routine	
		"ReadGroupItem" Routine	4-12
		"GetNumberOfGroups" Routine	4-13

		"GetNumberOfRows" Routine	
		"GetNumberOfItems" Routine	
	4.4	Routine for Cyclical Reading via Pipes	
		"StartCyclicPipe" Routine	
		"ReadCyclicPipe" Routine	
		"StopCyclicPipe" Routine	
		"SuspendCyclicPipe" Routine	
		"ResumeCyclicPipe" Routine	
	4.5	System messages in the Network	
		Data Types	
		Programming interface	
		Example of Programming	
		What SYS Messages are there and how should I react to them?	
	4.6	COM - Automation Interface	
		IFIObject	
		IFIData	
		Example for the Total Script	
	4.7	Tips and Tricks when Working with the Interface	
		Clearing the Memory using the "KILLTASK.EXE" Tool	
		Issuing FI Commands using the "VBDemo" Application	
		Outputting Diagnostic Messages	
		Windows NT Key Combinations	
_	_		
5	Inst	alling Windows NT and the Function Interface	5-1
	5.1	The Windows NT Operating System	
		Multitasking and Windows NT	
		Windows NT Task Manager	5-2
	5.2	Setting the Windows NT System Properties	
		Performance Features	
		Date/Time properties	
	5.3	Installing the Function Interface	
	5.4	Directory and File Structure of the Function Interface	
		Contents of the "INDRAMAT.INI" File	
		Example Entries in the "INDRAMAT.INI" File	5-6
		Contents of the "IND_DEV.INI" File	
		Example Entries in the "IND_DEV.INI" file	
		Contents of the "[LW]:\Winnt\System32" System Directory	5-10
		Contents of the "[LW]:\Winnt\System32\Drivers" Driver Directory	
		Contents of the "[LW]:\\MTGUI\BasicData\Help" Directory	
		Contents of the "[LW]:\\MTGUI\BasicData\Resource" Directory	5-11
		Contents of the "[LW]:\\MTGUI\BasicData\Resource" Directory Example Entries in the "VERSION.DAT" File	5-11 5-11
		Contents of the "[LW]:\\MTGUI\BasicData\Resource" Directory	5-11 5-11
6	Cor	Contents of the "[LW]:\\MTGUI\BasicData\Resource" Directory Example Entries in the "VERSION.DAT" File	5-11 5-11
6	Cor 6.1	Contents of the "[LW]:\\MTGUI\BasicData\Resource" Directory Example Entries in the "VERSION.DAT" File Contents of the "[LW]:\\MTGUI\Bin" Directory	5-11 5-11 5-12 6-1
6		Contents of the "[LW]:\\MTGUI\BasicData\Resource" Directory Example Entries in the "VERSION.DAT" File Contents of the "[LW]:\\MTGUI\Bin" Directory	5-11 5-11 5-12 6-1 6-1

		Selector	6-3
		Data code	6-3
	6.2	Data Tables	6-4
		General Parameters for the MWCX Device Groups	6-4
		Meanings of the Axes for the MWCX Device Group	6-5
		Axis Types for the MWCX Device Group	6-5
		Base Units	6-5
	6.3	Overview of FI Commands	6-6
		Overview of the MPCX Device Group	6-6
		Overview of the MTCX Device Group	6-7
		Overview of the MWCX Device Group	6-12
		Overview of the MSCX Device Group	6-18
		Overview of the MWMX Device Group	6-18
		Overview of the MWSX Device Group	6-20
		Overview of the MWAX Device Group	6-22
		Overview of the MSYX Device Group	6-24
		Overview of the MWYX Device Group	6-25
		Logical Connection between FI Commands	6-27
	6.4	Command Execution Times	6-30
		Command execution times for the MPCX device group	6-31
		Command execution times for the MWCX device group	6-32
		Command execution times for the MSCX device group	6-35
		Command execution times for the MWSX device group	6-35
		Command execution times for the MWAX device group	6-36
		Command execution times for the MSYX device group	6-36
7	Fur	nction Interface Commands	7-1
	7.1	FI Commands for the MPCX Device Group	7-1
		Interrupting Function Interface Jobs: BFJ	
		Outputting the Device Configuration: CCP	
		Reading the FI communication error counts: CEI	
		Commands for Executing WIN32 Applications: CPR	
		Removing Function Interface Jobs: DFJ	
		Deleting of the FI Command Stack Administration: DFS	
		Command for Terminating WIN32 Applications: DPR	
		Error Information: ERI	
		Far Configuration Parameter: FCP	
		Far Device Configuration Parameter: FDC	
		Further Info Text: FIT	
		Far PC Configuration Parameters: FPC	
		Writing/Reading the General FI Data Buffer: GDB	
		Initialization of a V24 Communication Address: ICA	
		Information regarding Function Interface Jobs: IFJ	
		Reading of the FI Command Stack Administration: IFS	
		Reading of PC Date and PC Time: LDT	
		Activated Language of the Rexroth Indramat GUI: LNG	



	Read System Messages: MSG	7-26
	NT Shutdown Functions: NST	7-27
	Formatting a Parameter Download Data File: PAF	7-28
	Generating Physical Directory Names: PHD	7-29
	Writing and Reading of a PC Port Address (Byte Access): POB	7-30
	Writing and Reading of a PC Port Address (Word Access): POB	7-31
	Ready Message for a WIN32 Application: RP	7-32
	Triggering an FI Device Polling Cycle: SDP	7-32
	Focusing Commands: SFW	7-33
	Issuing a SYS Message: SSM	7-34
7.2	FI Commands for the MTCX Device Group	7-37
7.3	FI Commands for the MWCX Device Group	7-39
	Active Acceleration Value: AAC	7-39
	Active Angle Dimension (RAD/DEG): AAD	7-40
	Actual (Current) Axis Speed (Spindle Speed): AAS	7-40
	Active NC Block: ABI	7-42
	Active Cutting Speed of the Reference Spindle: ACS	7-43
	Active D-Correction Number: ADN	7-43
	Active Event Monitoring: AEM	7-44
	Active Edge Number: AEN	7-45
	Active Feedrate Override: AFO	7-45
	Actual (Current) Feedrate: AFR	7-46
	Active G Functions: AGF	7-47
	Active M Functions: AMF	7-48
	Active Mechanism Messages: AMM	7-48
	Active NC Memory Size: ANM	7-56
	Active Machine Parameter Index: API	7-57
	Active Note in NC Program (Note and NC Record Number): APM	7-59
	Active Note in NC Program (only NC Record Number): APN	7-60
	Actual (Current) Position Value of an Axis: APO	7-61
	Active NC Program Number: APP	7-62
	Axis Reference Flags: ARF	7-63
	Actual (Current) Rapid Override: APO	7-64
	Axis Reference Table: ART	7-65
	Actual (Current) Spindle Data: ASD	7-66
	Active Spindle for Process: ASF	7-67
	Actual (Current) Spindle Gear Level: ASG	7-68
	Active System Error Messages: ASM	7-69
	Actual (Current) NC Sequence Number: ASN	7-74
	Actual (Current) Spindle Override: ASO	7-75
	Actual (Current) Spindle Speed: ASS	7-76
	Active Tool Number: ATN	7-77
	Reading Actual (Current) Tool Place Information: ATP	7-77
	Access to Actual (Current) Tool Data Record: ATR	7-79
	Accepting the Data Record for the Current Tool: ATU	7-81
	Active Zero-Offset Bank: AZB	7-81

NC Cycle Download: CCA	7-82
NC Cycle Upload: CCA	7-85
Position Set point of an Axis: CPO	7-88
Trigger Control Reset: CRT	7-90
Device Axis Configuration Parameter: DAC	7-90
NC D-Correction Download: DCA	7-93
NC D-Correction Upload: DCA	7-95
Reading D-Correction Data: DCD	7-96
Device Configuration Parameter: DCP	7-97
C Correction Register: DCR	7-99
Setting the Communication Timeout Time DCTMWCX device group	7-101
Long Identification of NC/PLC Data Records: DIS	7-102
Delete NC Program: DPN	7-107
Delete NC Program Package: DPP	7-108
Reading the Device Status Information: DSI	7-108
Tool Management Configuration Data: DTC	7-111
Distance to Go of Axis Movement: DTG	7-115
Device Type and Accompanying Components: DTY	7-117
Diagnosis Window Data: DWD	7-117
Existing errors: EDE	7-120
Existing Diagnosis Window: EDW	7-121
Existing NC Diagnoses: END	7-122
Existing PLC Diagnoses: EPD	7-123
End Point of an Axis Movement: EPO	7-124
Existing ProVi Types: EPT	7-126
Error Status: EST	7-127
Execution Display: EXD	7-127
Global Process Parameter Configuration: GPC	7-128
Global Process Parameters: GPP	7-131
Insert NC Program Package: IPP	7-133
Module Assignment of a Process: MAP	7-134
Read Reference Name of a PLC Variable: MAR	7-135
Device Data of the Module Configuration: MCD	7-136
Module Data of the Module Configuration: MCM	7-137
Process Data of the Module Configuration: MCP	7-138
SFC Data of the Module Configuration: MCS	7-138
Machine Data Download: MDA	7-139
Machine Data Upload: MDA	7-142
Inputting an NC Record: MDI	7-144
Reading and Writing Machine Data: MDS	7-146
Downloading Message Texts: MFD	7-149
Maximum Feedrate Override: MFO	7-150
Maximum Feedrate: MFR	7-151
Reading Machine Key Information : MKS	7-152
Writing the GUI-SK Block: MKT	7-152
Maxim Rapid Override: MRO	



Read System Messages: MSG	.7-154
Maximum Spindle Override: MSP	7-155
Maximum Spindle Speed: MSS	.7-156
Reading the Firmware Identification: MTC	7-156
User Machine Data: MTD	7-157
NC Program Compile and Download: NCA	7-158
NC Program Upload: NCA	.7-162
NC Messages: NCM	.7-166
NC Events Download: NEA	.7-168
NC Events Upload: NEA	.7-170
Status of NC Events: NEV	.7-172
Selection of NC Memory: NMM	.7-173
Reading NC Parameters: NPA	.7-173
Activate NC Compiler: NPC	.7-177
Activate NC Download: NPD	.7-177
Read NC Package Directory: NPI	.7-180
Selection of the NC Program in the Active NC Memory: NPS	.7-181
Next Tool Number: NTN	.7-182
NC Zero Point Download: NUA	.7-182
NC Zero Point Upload: NUA	.7-185
NC Variables Download: NVA	
NC Variables Download: NVA	.7-189
Reading and Writing NC Variables: NVS	.7-191
Optimum Position Distance from Axes: POD	
Parameter Download: PAA	
Parameter Upload: PAA	.7-196
Process Axis Configuration Data: PAC	
Deactivate Parameters for an Offline Device PAD	
Setting Parameters Active for an Offline Device: PAS	7-200
ProVi Diagnosis Data: PDD	
Reading the Parameter Definition Table: PDT	.7-203
Programmed Feed Velocity: PFR	
Generating Physical Directory Names: PHD	
Active NC Program Information: PPA	
Read NC Program Directory: PPD	
Expert or Import NC Program: PPN	
Renaming of an NC Part Program: PPN	
Renaming of an NC Program Package: PPP	
Reading an NC Record: PPS	
Issuing SYS Messages Specific to the PCL: PSM	
Programmed Spindle Speed: PSS	
Process Tool Management Configuration: PTC	
Edit PROVI Message Files: PVA	
Formatted Input / Output of PLC Variables: PVF	
ProVi Messages: PVM	
Download of PLC Retain Variables: PVR	

Upload of PLC Retain Variables: PVR	7-229
Reading the PLC Variable Declaration: PVT	
Repositioning Data: REP	7-233
SFC Diagnosis Data: SDD	
Set the Device Status Information: SDS	
Sequencer Data: SFD	7-241
Sequencer Messages: SFE	
Sequencer Mode: SFM	7-245
Software Installation Data: SID	7-245
Servo Lag of an Axis: SLA	
PLC Long Identification: SLI	7-248
SERCOS Parameters: SPA	7-249
Active SERCOS Phase Switch-Over: SPH	7-251
Selected NC Program: SPP	7-252
Reading or Writing Tool Data Record: TDA	7-253
Loading Tool Data into the Control Unit: TDD	7-258
Access to Tool Data Record: TDR	7-259
Tool Insert Finish: TIF	
Tool Insert Initiate: TII	7-263
Tool Basic Data List: TLB	7-264
Tool Data Record Elements: TLD	7-267
Tool Edge Data List: TLE	7-273
Tool Move : TMV	7-275
Information on Grippers/Spindles/Tool Magazine Locations: TPI	7-276
Torque: TQE	7-278
Removing Tool Data Record: TRM	7-279
Resetting Remaining Tool Life of a Tool: TRS	
Requesting Watch List Allocations: WLA	7-281
Freeing Watch List Allocations: WFL	
Data of the Zero Offset Table: ZOD	7-282
Value Ranges	7-287
Flow Diagram for Command Groups	7-291
Using the Tool Command in Practice	
FI Commands for the MSCX Device Group	
Determining the Actual (Current) System Error: ASE	
Deleting the Actual (Current) System Error: CSE	
Setting the Communication Timeout Time DCT	
Reading the Device Status Information: DSI	7-297
Device Type and Accompanying Components: DTY	7-300
Read System Messages: MSG	7-301
Generating Physical Directory Names: PHD	7-301
Set the Device Status Information: SDS	
Software Installation Data: SID	
SERCOS Parameters: SPA	
Active SERCOS Phase Switch-Over: SPH	
FI Commands for the MWMX and MWSX Device Group	7-309

7.4

7.5



	Active System Error Messages: ASM	.7-309
	Trigger Control Reset: CRT	
	Setting the Communication Timeout Time DCT	.7-315
	Long ID of PLC Data Block: DIS	
	Reading the Device Status Information: DSI	
	Device Type and Accompanying Components: DTY	
	Diagnosis Window Data: DWD	
	Existing errors: EDE	
	Existing Diagnosis Window: EDW	
	Existing SPS Diagnoses: EPD	
	Existing ProVi Types: EPT	
	Error Status: EST	
	Execution Display: EXD	
	Read Reference Name of a PLC Variable: MAR	
	Device Data of the Module Configuration: MCD	
	Module Data of the Module Configuration: MCM	
	SFC Data of the Module Configuration: MCS	
	Downloading Message Texts: MFD	
	Reading Machine Key Information : MKS	
	Writing the GUI-SK Block: MKT	
	Read System Messages: MSG	
	Reading the Firmware Identification: MTC	
	ProVi Diagnosis Data: PDD	
	Generating Physical Directory Names: PHD	
	Issuing SYS Messages Specific to the PCL: PSM Edit PROVI Message Files: PVA	
	-	
	Formatted Input / Output of PLC Variables: PVF	
	ProVi Messages: PVM	
	Download of PLC Retain Variables: PVR	
	Upload of PLC Retain Variables: PVR	
	Reading the PLC Variable Declaration: PVT	
	Set the Device Status Information: SDS	
	Sequencer Data: SFD	
	Sequencer Messages: SFE	
	Sequencer Mode: SFM	
	Software Installation Data: SID	
	PLC Long Identification: SLI	
	Requesting Watch List Allocations: WLA	
	Freeing Watch List Allocations: WFL	
FI	Commands for the MWAX Device Group	
	MTA200 Messages: ADM	
	Active Mechanism Messages: AMM	
	Actual (Current) Position of an Axis: APO	
	Active System Error Messages: ASM	
	Reading and Writing CMOS RAM ASCII Parameters: CMA	
	Reading and Writing CMOS RAM Floating Point Parameters: CMF	.7-381

7.6

Read and Write CMOS RAM Integer Parameters: CMI	. 7-381
Trigger Control Reset: CRT	. 7-382
Device Configuration Parameter: DCP	. 7-383
Setting the Communication Timeout Time DCT	. 7-385
Long ID of PLC Data Block: DIS	. 7-386
Reading the Device Status Information: DSI	. 7-388
Distance to Go of Axis Movement: DTG	. 7-391
Device Type and Accompanying Components: DTY	. 7-392
Diagnosis Window Data: DWD	
Existing MTA200 Diagnoses: EAD	
Existing errors: EDE	
Existing Diagnosis Window: EDW	
Existing ProVi Types: EPT	
Error Status: EST	
Execution Display: EXD	
Module Assignment of a Process: MAP	
Read Reference Name of a PLC Variable: MAR	
Device Data of the Module Configuration: MCD	
Module Data of the Module Configuration: MCM	
Process Data of the Module Configuration: MCP	
SFC Data of the Module Configuration: MCS	
Downloading Message Texts: MFD	
Reading Machine Key Information : MKS	
Writing the GUI-SK Block: MKT	
Read System Messages: MSG	
Reading the Firmware Identification: MTC	
Parameter Download: PAA	
ProVi Diagnosis Data: PDD	
-	
Generating Physical Directory Names: PHD	
Actual (Current) Information on Position of All Axes: POI	
Issuing SYS Messages Specific to the PCL: PSM	
Edit PROVI Message Files: PVA	
Formatted Input / Output of PLC Variables: PVF	
ProVi Messages: PVM	
	7 400
Download of PLC Retain Variables: PVR	
Download of PLC Retain Variables: PVR Upload of PLC Retain Variables: PVR	. 7-433
Download of PLC Retain Variables: PVR Upload of PLC Retain Variables: PVR Reading the PLC Variable Declaration: PVT	. 7-433 . 7-433
Download of PLC Retain Variables: PVR Upload of PLC Retain Variables: PVR Reading the PLC Variable Declaration: PVT SFC Diagnosis Data: SDD	. 7-433 . 7-433 . 7-436
Download of PLC Retain Variables: PVR Upload of PLC Retain Variables: PVR Reading the PLC Variable Declaration: PVT SFC Diagnosis Data: SDD Setting of Device Status Information: SDS	. 7-433 . 7-433 . 7-436 . 7-440
Download of PLC Retain Variables: PVR Upload of PLC Retain Variables: PVR Reading the PLC Variable Declaration: PVT SFC Diagnosis Data: SDD Setting of Device Status Information: SDS Sequencer Data: SFC	. 7-433 . 7-433 . 7-436 . 7-440 . 7-441
Download of PLC Retain Variables: PVR Upload of PLC Retain Variables: PVR Reading the PLC Variable Declaration: PVT SFC Diagnosis Data: SDD Setting of Device Status Information: SDS Sequencer Data: SFC Sequencer Messages: SFE	. 7-433 . 7-433 . 7-436 . 7-440 . 7-441 . 7-443
Download of PLC Retain Variables: PVR Upload of PLC Retain Variables: PVR Reading the PLC Variable Declaration: PVT SFC Diagnosis Data: SDD Setting of Device Status Information: SDS. Sequencer Data: SFC Sequencer Messages: SFE Sequencer Mode: SFM	. 7-433 . 7-433 . 7-436 . 7-440 . 7-441 . 7-443 . 7-444
Download of PLC Retain Variables: PVR Upload of PLC Retain Variables: PVR Reading the PLC Variable Declaration: PVT SFC Diagnosis Data: SDD Setting of Device Status Information: SDS. Sequencer Data: SFC Sequencer Messages: SFE. Sequencer Mode: SFM Software Installation Data: SID	. 7-433 . 7-433 . 7-436 . 7-440 . 7-441 . 7-443 . 7-444 . 7-445
Download of PLC Retain Variables: PVR Upload of PLC Retain Variables: PVR Reading the PLC Variable Declaration: PVT SFC Diagnosis Data: SDD Setting of Device Status Information: SDS Sequencer Data: SFC Sequencer Messages: SFE Sequencer Mode: SFM Software Installation Data: SID PLC Long Identification: SLI	. 7-433 . 7-433 . 7-436 . 7-440 . 7-441 . 7-443 . 7-444 . 7-445 . 7-446
Download of PLC Retain Variables: PVR Upload of PLC Retain Variables: PVR Reading the PLC Variable Declaration: PVT SFC Diagnosis Data: SDD Setting of Device Status Information: SDS. Sequencer Data: SFC Sequencer Messages: SFE. Sequencer Mode: SFM Software Installation Data: SID	. 7-433 . 7-436 . 7-440 . 7-441 . 7-443 . 7-444 . 7-445 . 7-446 . 7-447



	7.7	FI Commands for the MSYX Device Group		
		Determining the Actual (Current) System Error: ASE		
		Deleting the Actual (Current) System Error: CSE		
		Setting the Communication Timeout Time DCT	7-450	
		Device Type and Accompanying Components: DTY	7-451	
		Read System Messages: MSG	7-452	
		Generating Physical Directory Names: PHD	7-452	
		Setting of Device Status Information: SDS	7-453	
		Software Installation Data: SID	7-454	
		SERCOS Parameters: SPA	7-455	
		Active SERCOS Phase Switch-Over: SPH	7-458	
8	8 Error codes			
	8.1	General Error Result Line	8-1	
	8.2	Error Codes 200 to 999	8-1	
	8.3	Error Codes 1000 to 1999		
	8.4	Error Codes 2000 to 2999		
	8.5	Error Codes 4000 to 4999		
	8.6	Error Codes 5000 to 5999		
	8.7	Error Codes 6000 to 6999		
	8.8	Error Codes 7000 to 7999		
	8.9	Error Codes 8000 to 8999		
	8.10	Error Codes 10000 and above		
	8.11	Error Codes 35000 and above		
	8.12	Error Codes 100000 and above		
	8.13	Error Codes 110000 and above	8-33	
	8.14	Error Codes 210000 and above		
	8.15	SERCOS Error		
	8.16	Global SERCANS Error		
8.17 Structure of Error File after Download				
	Examples for the MWCX Device Group8			
9	Ans	swers to Frequently Asked Questions	9-1	
	9.1	Function Interface FAQs	9-1	
	9.2	Windows NT FAQs	9-2	
10	Ref	erence to Literature	10-1	
	10.1	Information in Rexroth Indramat Literature		
11	Glo	ssary	11-1	
12	List	t of Figures	12-1	
		-		
13	Ind	ex	13-1	
14	Ser	vice & Support	14-1	

14.1	Helpdesk	14-1
14.2	Service-Hotline	14-1
14.3	Internet	14-1
14.4	Vor der Kontaktaufnahme Before contacting us	14-1
14.5	Kundenbetreuungsstellen - Sales & Service Facilities	14-2





1 New in Versions 07VRS

1.1 Version 07V00

General Information • The Visual Motion device group MWMX was included. For the commands, see the MWSX device group, as the WinPcl part is identical with this group. The Visual Motion component has been realized under SCP (Scalable Communication Platform).

- The MWYX device group has been included. The commands are a combination of the MSYX and MWSX device group, but united in a chapter of their own.
- The item "COM Automation Interface" was included in the chapter "Programming".
- The FI command "PVS" was removed from the description as the existing FI command "PVF" possesses the efficiency of "PVS", plus the array handling. Accordingly, do not use "PVS" any more.
- From IF Version 07, the FI commands "DPN", "DPP" "IPP", "NPC", "NPD", "NPI", "PPD" and PPN" are re-activated and have been included in the description (MWCX device group).
- **New FI Commands** FI command "ANM" supplies the size of the current NC magazine (MWCX device group).
 - FI command "ASD" supplies the current spindle data (MWCX device group).
 - FI command "CCA" causes an Upload of NC cycles by an Upload file (MWCX device group).
 - FI command "CEI" displays the counts of the logged communication errors (MPCX device group).
 - FI command "DCA" causes an Upload of NC D corrections by an Upload file (MWCX device group).
 - FI command "DCT" sets the Timeout for a device or sets the time back to the default value (MWCX-, MWSX-, MWMX-, MWAX-, MSYX-, MWYX- and MSCX device group).
 - FI command "DSF" deletes the FI command Stack management (MPCX device group).
 - FI command "ICA" initializes a communication address with new parameters (MPCX device group).
 - FI command "IFS" supplies the current occupancy state of FI command Stack management (MPCX device group).
 - FI command "LDT" reads and writes the local PC date and the local PC time of the day (MPCX device group).
 - FI command "MDA" has been extended by the command MDA4 with which <u>all</u> Machine Data Page definitions can be deleted in the selected device (MWCX device group).
 - FI command "MSG" is used to read System Messages (MPCX-, MSCX-, MWCX-, MWSX-, MWMX-, MWAX- device group).
 - FI command "MTC" is used to read the Firmware identification from the different control components (MWCX-, MWSX-, MWMX- and MWAX- device group).



- FI command "NCA" causes an Upload of NC programs by an Upload file (MWCX device group).
- FI command "NEA" causes an Upload of NC events by an Upload file (MWCX device group).
- FI command "NUA" causes an Upload of NC zero points by an Upload file (MWCX device group).
- FI command "NVA" causes an Upload of NC variables by an Upload file (MWCX device group).
- FI command "PAD" sets a parameter set inactive if the device is in Offline mode (MWCX device group).
- FI command "PAF" sorts a parameter download file (MPCX device group).
- FI command "PAS" sets a parameter set active if the device is in Offline mode (MWCX device group).
- FI command "PDD" supplies data for the ProVi criteria analysis (MWCX-, MWSX-, MWMX- and MWAX device group).
- FI command "POB" is used to write and read a PC port address (BYTE access) (MPCX device group).
- FI command "POI" supplies the current position information of all axes (MWAX device group).
- FI command "POW" is used to write and read a PC port address (WORD access) (MPCX device group).
- FI command "PVA" is used to write and read Provi Message files (MWCX-, MWSX-, MWYX- and MWAX device group).
- FI command "PVM" was extended by the command "PVM4" (MWCX-, MWSX-, MWMX- and MWAX device group).
- FI command "PVR" executes an Upload or Download of PLC retain variables (MWAX-, MWCX-, MWMX-, MWSX- and MWYX device group).
- FI command "REP" supplies data to return to the contour (MWCX device group).
- FI command "SDP" starts a FI device interrogation cycle (MPCX device group).
- FI command "SDS" sets the device status (ON/OFF) which is also entered in IND_DEV.ini (MSCX-, MWCX-, MWSX-, MWMX-, MSYX-, and MWAX device group).
- FI command "TPI" supplies information about grippers, spindles and tool magazine locations (MWCX device group).
- Modifications of FI Commands
 - FI command "AMM" reports the active mechanism errors and was increased by a file name for additional information for the message text or the extended text in the return value (MWCX-, and MWAX device group).
 - FI command "API" supplies always 0 as index of the active parameter set (MWCX device group).
 - FI command "APP" supplies the active NC block number and was extended in the reply by an output in the setup mode.
 - FI command "ART" was extended by "<u>binary</u> reading of the current axis reference table for an Offline device" (DeviceStatus=OFF) (MWCX device group).
 - FI command "ASM" reports the active system errors and was increased by a file name for additional information for the message text or the extended text in the return value (MWCX-, and MWAX device group).



- FI command "CCP" supplies the configuration settings from IND_DEV.ini and was extended by columns 14 (device protocol) and 15 (device simulation) in the reply (MPCX device group).
- FI command "DCD" supplies the value of D correction index and was extended by measuring unit [mm/inch] in the return (MWCX device group).
- FI command "DCR" reads and writes the values of a D correction set and was extended by measuring unit [mm/inch] in the return (MWCX device group).
- FI command "DIS" was revised and realized as B command (MWCX-, MWSX-, MWMX-, MWAX device group).
- FI command "DSI" supplies the most essential information about device status and was extended by columns 12 (device simulation switched on) and 13 (device status information) in the reply (MSCX-, MWCX-, MWSX-, MWMX-, and MWAX device group).
- FI command "DTY" was revised and realized as B command (MWCX-, MWSX-, MWMX-, MWAX-, MSCX-, MWSYX- and MWYX device group).
- FI command "DWD" supplies diagnosis messages and was extended in return value by the criteria analysis and a file name for additional information (MWCX-, MWSX-, MWMX-, and MWAX device group).
- FI command "ERI" now also supplies an error text with a WIN NT error code (MPCX device group).
- FI command "MKT" is used to write the GUI-SK 16 block in the PLC and was extended by command MKT2 (MWCX-, MWSX-, MWMX- and MWAX device group).
- FI command "NCM" supplies NC messages and was extended by a file name for additional information for the message text in the return value (MWCX device group).
- FI command "NVS" was set during writing NC variables to data type "LONG" or "Doublereal" (MWCX device group).
- FI command "PVM" supplies ProVi messages and was enlarged in return value by the criteria analysis and a file name for additional information for the message type or the extended text (MWCX-, MWSX-, MWMX-, and MWAX device group).
- FI command "ZOD" was extended by measuring unit [mm/inch] during reading an offset page (MWCX device group).
- Authorized Errors
 FI command "DIS1" supplies "--" in all result columns if no valid parameter set is in the selected device (MWCX device group).



1.2 Version 06V00

General Information

- Chapter 1 has been extended to provide information on safety under the heading "Protection against dangerous movements".
 - The section in the chapter entitled "Programming" concerning SYS messages has been revised.
 - A separate chapter entitled "Literature" has been appended.
 - Documentation of previously undocumented and new commands for the software standard 05-21V00 WIN-HMI.
 - Box 19 in the table on basic value range data is used to classify tools. The user can no longer edit (MWCX device group).
 - The MTCX device group has been almost entirely converted to the MWCX device group. This has created a designation for the newly introduced WinPcl.
 - Only the MTVNC remains in the MTCX device group. Its instruction set is a subset of the MWCX device group. Individual commands are listed only in table form with detailed reference to explanations in the MWCX device group.
 - The MISX device group has been converted to the MWSX device group. This has created a designation for the newly introduced WinPcl.
 - The MTAX device group has been converted to the MWAX device group. This has created a designation for the newly introduced WinPcl.
 - From IF Version 06 the FI commands "IPP", "NPC", "NPD", "NPI", "PPD" and PPN" are no longer valid and have been removed from the description (MWCX device group).
 - The list of error codes has been extended.

New FI Commands

- The FI command "ADM" supplies all messages from the Andron NC (MWAX device group).
- The FI command "ARF" indicates the reference flags of an axis for a process (MWCX device group).
- The FI command "ART" returns the complete axis reference tables of a system (MWCX device group).
- The FI command "ATR" returns the complete basic data and cutter data of the current processing tool (MWCX device group).
- The FI command "ATU" causes the NC to accept the data record of the current tool changed after editing (MWCX device group).
- The FI command "BFJ" interrupts FI jobs (MPCX device group).
- The FI command "CCA" triggers the downloading of NC cycles by means of the download file (MWCX device group).
- The FI command "CPR" starts a Win32 application (MPCX device group).
- FI command "DCA" triggers the downloading of NC D corrections by means of the download file (MWCX device group).
- The FI command "DPR" exits a Win32 application (MPCX device group).
- The FI command "DSI" returns the most important information on the status of the device (MSCX, MWCX, MWSX, and MWAX device groups).
- The FI command "DTC2" returns tool management data such as basic user data and cutter user data (MWCX device group).

- The FI command "DWD" outputs all diagnostic messages (MWCX, MWSX, and MWAX device groups).
- The FI command "EAD" returns which Andron diagnostic types are available in a module (MWAX device group).
- The FI command "EDE" returns whether or not there are errors present (MWCX, MWSX, and MWAX device groups).
- The FI command "EDW" returns all diagnostic window types (MWCX, MWSX, and MWAX device groups).
- The FI command "END" returns which NC diagnostic types are available in a module (MWCX device group).
- The FI command "EPD" returns which PLC diagnostic types are available on a control unit (MWCX and MWSX device groups).
- The FI command "EPT" returns all ProVi types (MWCX, MWSX, and MWAX device groups).
- The FI command "EST" queries the error state of a variable (MWCX, MWSX and MWAX device groups).
- The FI command "EXD" shows the extent to which a step or action can be executed (MWCX, MWSX and MWAX device groups).
- The FI command "GDB" writes/reads the data for the general FI data buffer (MPCX device group).
- The FI command "MDA" uploads or downloads machine data records (MWCX device group).
- The FI command "MDS" is for writing and reading machine data (MWCX device group).
- The FI command "MFD" loads the message text into the device indicated (MWCX, MWSX, and MWAX device groups).
- The FI command "MKS" returns information on the machine buttons (MWCX, MWSX, and MWAX device groups).
- The FI command "MKT" writes the GUI-SK16 block in the PLC (MWCX, MWSX and MWAX device groups).
- The FI command "NCA" triggers the downloading of NC programs by means of the download file (MWCX device group).
- The FI command "NCM" returns all messages from the Rexroth Indramat NC (MWCX device group).
- The FI command "NEA" triggers the downloading of NC events by means of the download file (MWCX device group).
- The FI command "NPA5" returns a list of a maximum of 10 parameters of the same parameter type (MWCX device groups).
- The FI command "NST" brings the operating system to a stop (MPCX device group).
- The FI command "NUA" triggers the downloading of NC zero points by means of the download file (MWCX device group).
- The FI command "NVA" triggers the downloading of NC variables by means of the download file (MWCX device group).
- The FI command "PAA" uploads or downloads parameter records (MWCX and MTAX device groups).
- The FI command "PDT" returns parameters from the definition table for the selected device in binary form (MWCX device group).
- The FI command "PHD" generates a physical directory with the assistance of the BDI (MPCX, MSCX, MWAX, MWSX, MWCX, and MSYX device groups).



- The FI command "PSM" initiates the most important SYS messages with regard to the PCL programming interface (MWCX, MWSX and MWAX device groups).
- The FI command "PVM" returns all ProVi messages (MWCX, MWSX, and MWAX device groups).
- The FI command "RPR" informs the caller that the program is now active (MPCX device group).
- The FI command "SDD" returns data for the step chain diagnosis (MWCX, MWSX, and MWAX device groups).
- The FI command "SFD" returns data for a step chain (MWCX, MWSX, and MWAX device groups).
- The FI command "SFE" returns all the step chain messages for a module (MWCX, MWSX, and MWAX device groups).
- The FI command "SFM" returns the operating mode of a step chain (MWCX, MWSX, and MWAX device groups).
- The FI command "SFW" focuses the screen (MPCX device group).
- The FI command "SSM" is for issuing SYS messages (MPCX device group).
- The FI command "WLA" requests free watch list allocations (MWCXdevice group).
- The FI command "WLF" frees up requested watch list allocations (MWCX-device group).
- Modifications of FI Commands
 The FI command "AAS" returns information about the current axis speed and, in return, has been extended by an additional unit of measurement [inch/min] (MWCX device group).
 - The FI command "AFR" returns information about the current feed velocity and, in return, has been extended by an additional unit of measurement [inch/min] (MWCX device group).
 - The FI command "APO" returns the current axis position value and, in return, has been extended by an additional unit of measurement [inch] (MWCX device group).
 - The FI command "CPO" returns the current axis position command value and, in return, has been extended by an additional unit of measurement [inch] (MWCX device group).
 - The FI command "DTG" returns the distance to go of an axis and, in return, has been extended by an additional unit of measurement [inch]" (MWCX device group).
 - The FI command "EPO" returns the end point of an axis movement and, in return, has been extended by an additional unit of measurement [inch]" (MWCX device group).
 - The FI command "MFR" returns the maximum feedrate and, in return, has been extended by an additional unit of measurement [inch/min]" (MWCX device group).
 - The FI command "OPD" returns the optimum position distance of an axis and, in return, has been extended by an additional unit of measurement [inch]" (MWCX device group).
 - The FI command "PFR" returns the value for the programmed feedrate and, in return, has been extended by an additional unit of measurement [inch/min]" (MWCX device group).
 - The FI command "SLA" returns the current servo lag of an axis and, in return, has been extended by an additional unit of measurement [inch]" (MWCX device group).

- The FI command "TDR" returns the complete basic data and tool edge data of a tool and has been extended by information concerning optional data elements (MTCX device group).
- The FI command "TLD" returns elements of the basic data and cutter data of a tool and has been extended by information concerning optional data elements (MTCX device group).

1.3 Version 05V00

General Information

- Documentation of previously undocumented and new commands for the software standard 05-19V00 WIN-HMI.
 - The chapter entitled "Practical Use of Tool Commands" has been included to better explain the tool commands (MTCX device group).

FI Commands Changes / Additions

- The FI command "ATP" returns information on the current tool location (MTCX device group).
- The FI command "ERI1" returns the error text and the additional text of an FI error code or a NACK error number (MPCX device group).
- The FI command "MAR" reads the reference names of a PLC variable (MTCX device group):
- FI command "MTD1" for reading and writing machine user data has been expanded (MTCX device group). The FI command "MTD" is no longer used for reading!
- The FI command "PVF" is for the formatted reading and writing of PLC variables, arrays and structures (MTCX device group).
- The FI command "PVS" is for reading and writing PLC variables, arrays and structures (MTCX device group).
- The FI command "PVT" reads the declaration of PLC variables, including structures and arrays (MTCX device group).
- The FI command "TDR" returns the complete basic data and tool edge data of a tool (MTCX device group).
- The FI command "TLB" returns the basic data of the tool list (MTCX device group).
- The FI command "TLD" returns elements of the basic data or cutter data of a tool in the tool memory (MTCX device group).
- The FI command "TLE" returns the cutter data of the tool list (MTCX device group).
- The FI command "TII" initiates a tool replacement (MTCX device group).
- The FI command "TMV" moves an entire tool data record comprising the basic data and defined cutter data (MTCX device group).
- The FI command "TRS" resets the percentage life time of a tool to 100% (MTCX device group).
- The FI command "TIF" terminates a tool replacement (MTCX device group).



- The result [*X] of an FI command with invalid parameters has been changed into [--] (2 hyphens). This concerns the following commands of the MTCX device group: AAS1, AAS2, APO1, APO2, ASO1, ASS, CPO1, CPO2, DTG1, DTG2, EPO1, EPO2, MSS, OPD1, OPD2, PSS, SLA1, SLA2, TQE1, TQE2.
- Chapter 05 contains the table "Logical Linking of FI Commands" with direct links to the individual commands in the help file.
- The table "Availability of FI Commands" in Chapter 05 has been split into separate sections for each device group. It now contains direct links to the individual commands in the help file.
- The new device group MSYX (SYNAX200-P, SYNAX200-R) has been incorporated into the documentation.

1.4 Version 04V03

General Information	 Documents previously undocumented and new commands for the software standard 05-18V06 WIN-HMI.
	 Includes a table with logical links for the FI commands in Chapter 5.
	 Inclusion of a table above the command times in chapter 05.
	 Includes the component types for the NC and PLC hardware in the file "IND_DEV.INI". Expansion of the FI command "DTY" by the output of the component types "Componenttype1=" and "Componenttype2=".
Identification of versions,	Entries in the "C:\IND_BASE\INDRAMAT.INI" file:
Rexroth Indramat	• IfDIIMode = 04.20
software components	• IfVersion = 04V03
	Software components contained normally within the function interface :
	all Rexroth Indramat System 200 MT-CNC user interfaces for Version 18V06.
FI Commands Changes / Additions	 The FI command "AMM7" for outputting active mechanism messages and errors (MTAX device group).
	• FI command "DCD1": The values of a D-correction register are read out (MTCX device group).
	 The FI command "DTC1" returns the most important system parameter data of the tool management (MTCX device group).
	• The FI command "FIT1" returns the additional text of an FI error code or a NACK error number (MPCX device group).
	 The FI command "PTC1" returns the tool management data of all defined NC processes. The FI command "PTC2" returns the tool management data of a defined NC process. Only for the MTCX device group.
	 The FI command "PPD" reads an NC program directory (MTCX device group).
	 The FI command "PPN" converts an NC program from the NC program directory into an ASCII file and vice versa (MTCX device group).
	 The FI command "PPP" changes the designation of an NC program package (MTCX device group).
	 The FI command "PVF" handles the formatted reading and writing of PLC variables, arrays and structures (MTCX, MISX and MTAX device groups).

- The FI command "PVT" reads the type of PLC variables, including structures and arrays with their elements (MTCX, MISX and MTAX device groups).
- The FI command "SID1" returns information regarding the installation. This information includes installation paths, the software version used, DLL mode, plus service pack and release information. Valid for all device groups.
- The FI command "SLI" returns the single data from the SPS long ID such as the number, name and length of the program, the compiling date and more (MTCX, MISX and MTAX device groups).
- The FI command "TII" initiates a tool replacement (MTCX device group).
- The FI command "TIF" initiates the end of a tool replacement (MTCX device group).
- The FI command "TLB1", or "TLB2" returns basic data of the tool list such as tool memory, designation, correction type and more (MTCX device group).
- The FI command "TLE1", or "TLE2" returns cutter data of the tool list such as tool memory, location number, tool status and more (MTCX device group).
- The FI command "DCR1" has been implemented for reading and writing the D-correction register with the newly formatted output. (MTCX device group). The FI command "DCR" is no longer used!
- The FI command "DTY1" for outputting the device type has been expanded by the corresponding components. (MTCX, MSCX, MISX, and MTAX device groups). The FI command "DTY" is no longer used!
- The FI command "ZOD" for reading and writing data from the zero offset table has been expanded to include the FI commands "ZOD1" and "ZOD2" (MTCX and device group). The FI command "ZOD" is no longer used!
- New, speed-optimized FI command "GPP" for reading out the global process parameters. (MTCX device group).
- New, speed-optimized FI command "NPD3 and "NPD4" for the NC download of small NC part programs. (MTCX device group).
- "NPD1" and "NPD2" commands for the NC download expanded by the value to be written "Initialization" (MTCX device group).
- Expansion of the FI command "CCP" by the output of the component types "Componenttype1=" and "Componenttype2=" (MPCX device group).
- New FI command "CCP5" for outputting the configuration data of the device that is addressed via the indicated device address (MPCX device group).
- New FI commands "CMA, CMF and CMI" for reading and writing of CMOS RAM, ASCII, floating point and integer parameters. (MTAX Device Group)
- New FI command "CRT" for triggering a control reset for the selected device (MTAX device group).
- New FI command "NPS" for preselecting the NC program located in the NC memory for processing (MTCX device group).
- New FI command "NMM" for selecting the NC memory for the NC program processing (MTCX device group).
- Addition to the documentation of the FI command "SPA" of the Ident. Number formats. The error return in the event of a form error in the value to be written has also been improved (MTCX, and MSCX device group).



• New FI command "TDAx, TMV and TRM" for editing complete tool data records (MTCX device group).

1.5 Version 04V02

General Information	• New chapter in the documentation: "Answers to Frequently Asked Questions (FAQ)".
	 Most resource wastage has been eliminated in Service Pack 2 of the MT-CNC user interface GUI 18V05.
Identification of versions,	Entries in the "C:\IND_BASE\INDRAMAT.INI" file:
Rexroth Indramat software components	• IfDIIMode = 04.10
	• If Version = 04V02
	Software components contained normally within the function interface :
	 all Rexroth Indramat System 200 MT-CNC user interfaces for Version 18V05 with Service Pack 3.
FI Commands Changes / Additions	• Module commands MCD1, MCM1 and MCS1 enabled for the MISX device group.
	• Module commands MCD1, MCM1, MCP1, MCS1, MAP1 enabled for the MTAX device group.
	• Addition of CR_APO2, CR_DTG2, CR_CMA, CW_CMA, CR_CMI, CW_CMI, CR_CMF and CW_CMF for the MTAX device group.
Basic processes	Waste of resources in logic process resolved.
Changes / Additions	• Expansion from 15 to a max. of 255 group requests during cyclic requests (see "Routines for Cyclic Reading via Pipes").

1.6 Version 04V01

General Information	 Inclusion of the PRO-VERSION as a software option in the installation program.
Identification of versions, Rexroth Indramat software components	 Entries in the "C:\IND_BASE\INDRAMAT.INI" file: IfDIIMode = 04.10 IfVersion = 04V01
	 Software components contained normally within the function interface : all Rexroth Indramat System 200 MT-CNC user interfaces for Version 18V05.
FI Commands Changes / Additions	 Expansions to the device-independent access functions. New FI command "CRT" for triggering a control reset. (MTCX and MISX device groups).
Basic processes Changes / Additions	 Error correction of the telegram optimizer (correction of timeout recognition). New SYS message "MSG_PCALIVE" in PC network.



1.7 Version 04V00

		st to the previous 03VRS versions, fundamental changes have le in this version:
General Information		ery of a Visual Basic example connection to the function ace (application including source codes).
	 Delive formation 	ery of printed documentation as online help in Windows NT/95 help t.
	Provis	sion of an installation program for the function interface.
	• New!	FI commands for an NC download. (MTCX device group)
Identification of versions,	Entries in	the "C:\IND_BASE\INDRAMAT.INI" file:
Rexroth Indramat	• IfDIIM	lode = 04.00
software components	• IfVers	sion = 04V00
	Software	components contained normally within the function interface :
		exroth Indramat System 200 MT-CNC user interfaces for Version
		4 with Service Pack 2.
FI Commands Changes / Additions	outpu	mmand "XYZ" are implemented as "XYZ1" with re-formatted t: AAC1, AAS1, ADN1, AFO1, APO1, ARO1, ASO1, AZB1, 1, MRO1, MSO1. The FI command "XYZ" should no longer be
		FI command "ABN" has been replaced by the FI commands " and "AMM".
	New f	unctions for the BOF/GBO for calling up WIN32 applications.
		functions for a WIN32 application at the function interface for g up the BOF/GBO.
		nsion of the data structures for the BOF/GBO. Support for oth Indramat TRANS200 device types.
	Note:	The Rexroth Indramat devices TRANS200-P and TRANS200- R are still at the development stage and therefore cannot yet be ordered.
	 Expai 	nded function calls for the device configuration.
	Mess	age for activating or deactivating a PC in the PC network.
	 Expansion 	nsions to the device-independent access functions.
Basic processes Changes / Additions		correction of the telegram optimizer (correction of timeout nition).
	 Expai 	nsion in the PLC data optimizer.
	• Enlar	ged input buffer for the telegram optimizer.
	Rewo	rking of the internal interface.
	 Error routin 	correction in data provision by means of the "ReadGroupItem" e.
	• Error	correction of the communication channel.
	• Error	corrections in the internal DLL interfaces.
		ction of the INDIF200.DLL (correction of the binary result for le data).

• Changes in LogOutIf(), with regard to the selective KILLTASK



- Reworking of the COMVIEW interface for WIN200.
- Moving of the new SYS-Message interface into the file "INDIF000.H".

1.8 The Data Interface Newsletter

We will be informing you by email of new developments and updates to the Rexroth Indramat Products MPI and Function Interface.

Please send an email request with the message subscribe to:

owner-ml_datainterface@proxy.indramat.de

To unsubscribe, please proceed identically, but instead write Message: **unsubscribe.**

Note: Your email address will be kept confidential and not passed on to third parties.



2 General

2.1 Introduction

Δ

The Rexroth Indramat Function Interface is a unified data interface produced by Indramat for application programs (sometimes referred to as clients) based on the Windows NT platform.

- **Requirements** To obtain free access to data on existing NC and PLC data, it is necessary to provide a data interface that is as open, reasonably priced and as simple to handle as possible. The main requirements are to be able to access CNC/PLC data with a large range of functions and rapid access and reaction speeds. Several clients can access the data.
 - **Objective** The Rexroth Indramat Function Interface aims to do exactly that, i.e. it allows access to all required control data via a compact, functional interface. This therefore allows the customer to completely create his own user interface in the programming languages Visual C++ or Visual Basic. The user is thus provided with a powerful interface with which he can communicate with Indramat devices and user interfaces using mnemonic function calls. The Function Interface is therefore a universal solution for data communication.

Availability	y This description is valid for the following versions		
	WinHMI:	22Vxx	
	Function Interface:	07Vxx	
	Windows NT Workstation/Server:	4.0	
	Visual C++:	5.0 + 6.0	
	Visual Basic:	5.0 + 6.0	

2.2 The Function Interface from the User's Point of View

The Function Interface is a client (service requester) – server (service provider) interface and provides the user with a library (DLL) for communication services. The services, i.e. the functions of the DLL, fulfill the communication tasks that are required for reading individual data, the cyclic reading of data, the cyclic reading of data groups, the writing of data and for processor communication with Rexroth Indramat user interfaces and devices.

The Function Interface can communicate with a maximum of ten independent user programs (clients). This means a user program can, for example, be a customized user interface, a Rexroth Indramat OPC-Server or a communication driver to another data interface.

Up to ten parallel communication channels, one for each client, are supported between the Function Interface and the device. One communication channel can connect with a maximum of 64 data terminal devices.

The physical communication address in this case can be a serial interface (RS232/RS485), a Dual-Port-RAM or a Shared Memory area.



2.3 Protection against dangerous movements

Dangerous movements can be caused by the faulty control of connected motors. The reasons can be extremely varied:

- careless or faulty wiring or cabling,
- errors in operating the components,
- faults in the measured-value and signal transmitters,
- · defective components, and/or
- errors in the software.

These faults can occur immediately after switching on or at any time during operation.

The Rexroth Indramat Function Interface is communication software which can be used to change the values of variables in the control unit.

As far as possible, monitoring in the drive components precludes faults in the connected drives. Where personnel safety is concerned, particularly where there is a risk of physical injury and/or damage to property, this fact should not be relied on exclusively. Until the built-in monitoring systems become active faulty drive movement is always to be expected; the degree of movement depends on the type of control unit and the operating status.





Dangerous movements! Risk of death, injury, severe physical injury or damage to property!

⇒ For the reasons given above, protection of personnel is to be guaranteed by means of monitoring or other higher-ranking measures within the system.

For this purpose risk and fault analysis are to be provided for by the system designer according to the specific conditions within the system. The safety regulations applicable for the system are also to be taken into consideration. Arbitrary movements in the machine or other erratic functions can occur if safety devices are switched off, bypassed or activated wrongly.

To avoid accidents, physical injury and/or damage to property:

- ⇒ Do not stay within the motional range of the machine or machine parts. Possible measures to prevent personnel accidentally accessing the machine:
 - protective fencing
 - protective grid
 - protective cover
 - light barrier
- ⇒ Fencing and covers must be adequately secured against the maximum possible force of movement.
- ⇒ Position emergency stop switches within the immediate vicinity and so that they are easily accessible. Check that the emergency stop equipment is functioning before start-up. Do not operate the device if the emergency stop switch is not functioning correctly.
- ⇒ Protect against the device starting unintentionally by providing safety isolation for the drive's power connection by means of an emergency stop circuit or by using a safe starting lockout function.
- \Rightarrow Before accessing or entering the danger area bring the drives safely to a standstill.
- ⇒ Secure vertical axes against falling or slipping after switching off the motor power by, for example:
 - mechanically locking the vertical axis,
 - providing external brake/catching/clamping mechanisms or
 - adequately counterbalancing the axis.

The standard motor holding brake provided or an external motor holding brake controlled directly by the drive controller are not sufficient on their own to guarantee the safety of personnel!

- ⇒ De-energize electrical equipment by means of the main switch and secure against reconnection during:
 - maintenance and repair work
 - cleaning work
 - lengthy breaks in operation
- ⇒ Avoid operating high-frequency, remote controlled and radio devices in the vicinity of the device electronics and their power supply cables. If the use of these devices cannot be avoided, check the system and installation for possible faults in all working areas before switching on the system. If necessary, the system will require a



special EMC test.

3 Structure and Configuration Examples

3.1 The Structure of the Function Interface

Viewed as a complete component, the function interface consists of the following three basic processes:

- Logic process
- Communication process and
- Management process

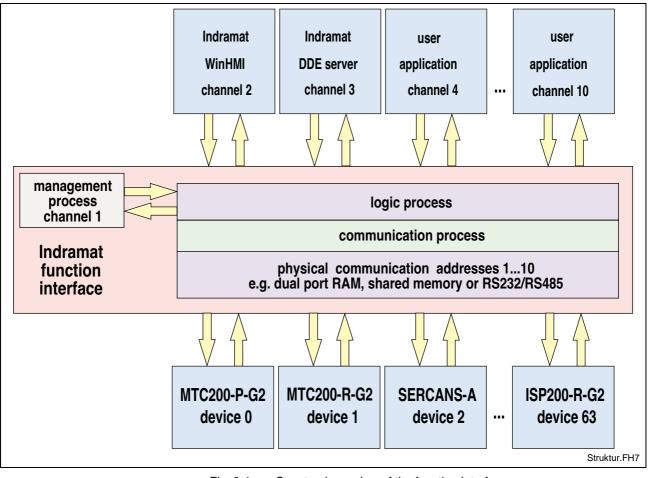


Fig. 3-1: Structural overview of the function interface

Logic process

The logic process provides the user program (client) with the actual data interface along with the services described in the previous chapter. To do this, it opens a logic channel (LOG channel) for every connected client. The number of active LOG channels therefore directly depends on the number of the connected clients. Furthermore, the logic process is a data interface to all defined devices and to the management and status terminal data that are monitored by the management process. As far as the user program (client) is concerned, the logic process is the server. On the other hand, the logic process provides the connection to the communication process via a shared memory. Data is distributed to the individual logic channels via this connection.



Note: The maximum number of LOG channels available to function interface applications is administered dynamically. If a function interface application exceeds this limit then an error message is issued. The chapter entitled "Programming" describes how the data interface to the logic process is to be handled and how to allow data access from the client to the function interface.

Communication process

The communication process executes the requirements of the various logic channels, generates communication to the devices together with the time allocations and initializes all devices configured on starting. The communication process and the logic process thereby allow data access to the respective Rexroth Indramat devices (MTC200-P-G2, ISP200-P-G2, SERCANS etc.).

On the one hand, it exchanges Rexroth Indramat telegrams with the logic process via the shared memory. On the other hand, it exchanges internal telegrams with the configured Rexroth Indramat devices via the dual port RAM or via a serial interface. The communication process opens a communication channel (thread) for each of these configured devices. It thereby allows simultaneous communication via various communication methods and via several parallel interfaces.

Note: The configuration of the communication addresses as well as the processing options of the individual Rexroth Indramat devices are carried out by the Rexroth Indramat system configurator and stored in the "IND_DEV.INI" file (see Chapter "Directory and File Structure of the Function Interface").

Management process

The management process is designed as an internal user program and uses the first LOG channel for communication with the logic process. It provides static and dynamic configuration data, delivers the more valuable functions and creates the corresponding data structure for each configured device. The management process thereby collects, for example, MTC200-P-G2 control data together with data from the PC hard drive which a client can then access. The management process thereby fulfills administrative tasks.

Note: The "Function Interface Commands" chapter describes how to access data from the Rexroth Indramat devices and the PC hard drive.



3.2 Configuration Examples and Connection Options

MPI Connection with Profibus FMS

The following figure shows the connection of the Rexroth Indramat MPI (Multi-Protocol-Interface) with Profibus FMS design-type and additional clients to the function interface.

The first LOG channel (logic channel 1) of the function interface is used by a user program (client), e.g. a customized user interface. The Rexroth Indramat MT-CNC user interface (**WinHMI = Win**dows **H**uman **M**achine Interface) runs under Windows NT. The MPI connection to the function interface is made via the second LOG channel (logic channel 2).

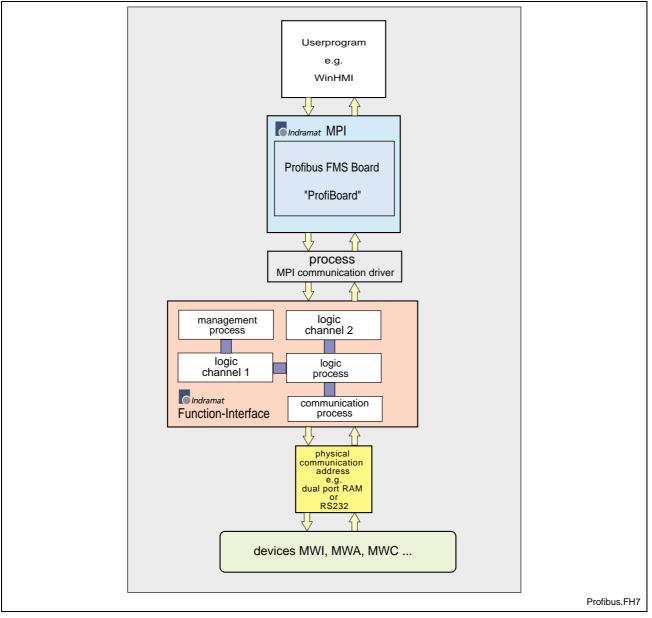


Fig. 3-2: MPI with Profibus FMS connection



Rexroth Indramat GUI and Server

The following figure shows the software structure with the 21VRS Rexroth Indramat GUI (WinHMI) as well as when using the Rexroth Indramat DDE server. It also shows the connection of an OPC server.

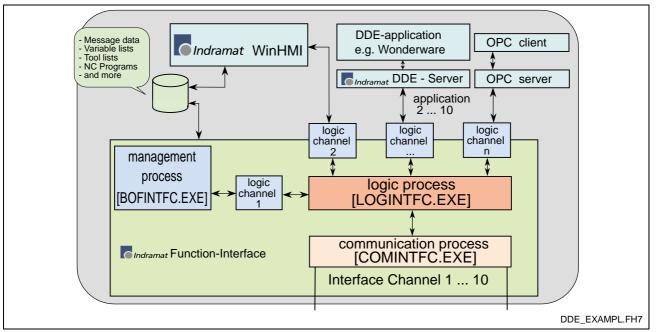
The components "WinMTC" and "WinHMI" are component parts of the Rexroth Indramat GUI WIN200. The DDE server allows connection via standard communication mechanisms to external program packages such as WONDERWARE "InTouch". Furthermore, using the NetDDE option, the DDE server allows a connection to be made via a network.

 OPC^{TM} stands for OLE for Process Control. OLE (Object Linking and Embedding) was originally introduced by Microsoft for communication between software components. Today, we refer to the terms COM (Component Object Model) or DCOM.

The goal of OPC is to create a unified communication interface for process data from any sources such as PLC and NC controls.

The user (developer of OPC client programs) therefore has the following advantages:

- Only minimum knowledge of the controls is required in order to communicate with the control software.
- No adjustment has to be made if an application has to communicate with different makes of control.







Connection to the function interface

The following illustration shows the various options for connecting an application to the function interface.

Direct connection can be achieved via:

- Rexroth Indramat's own GUI, WinHMI.
- programs written by the user in Visual C++ or Visual Basic (customer 3rd party).

The following are examples of indirect connection:

- DDE server,
- OPC server, and
- MPI Com Driver.

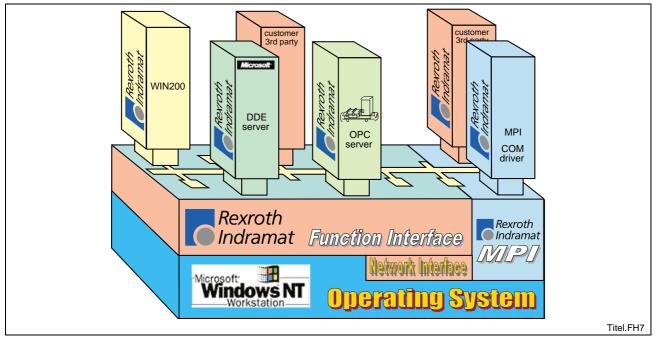


Fig. 3-4: Overview of the connection options

Communication between a Client and Rexroth Indramat Devices

The following figure shows the process of communication of a client on a Rexroth Indramat PC (BTV30) with two Rexroth Indramat devices (MTC200-R-G2 and MTC200-P-G2). On the one hand, the device 00 (MTC200-R-G2) communicates with the communication process via the serial interface (COM1), while device 01 (MTC200-P) communicates via a dual port RAM. The communication process opens a thread for each communication channel that has been configured. The client shown can access data from both devices. To do this, the appropriate device address is specified in the function interface command (FI command) (see Chapter "Design and Availability of the FI Command").

Note: Several cyclic requests (FI commands) can easily be combined at both devices. (See chapter entitled "Data Transfer and Result Evaluation Routines".)

During the initialization phase of the function interface, the configuration data of Rexroth Indramat devices is compared to the actual status. FI commands that have been requested are thereby checked as to their validity for the configured device group. Any errors in command mnemonics can then already be intercepted at the top level.



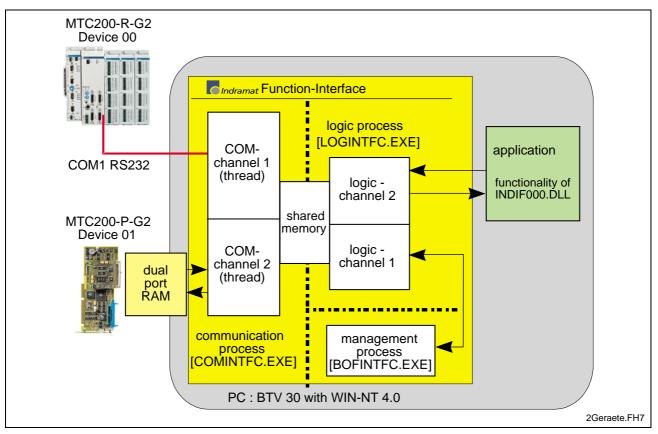


Fig. 3-5: Communication between a Client and Rexroth Indramat Devices

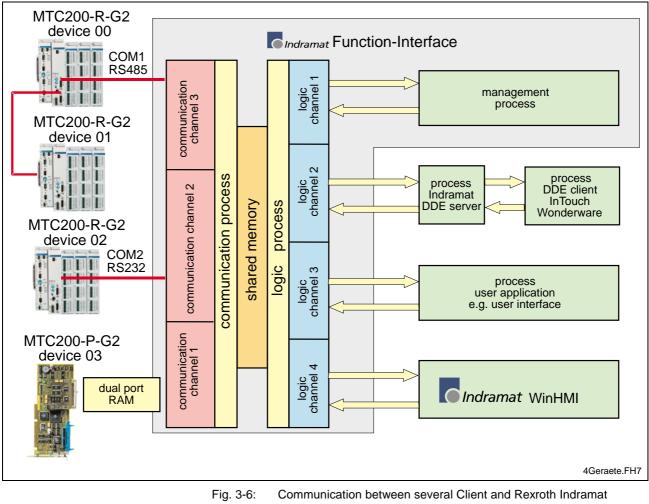
Communication between several clients and Rexroth Indramat devices

The following figure shows the software structure of the function interface when communicating with several devices during the operation of several clients.

Note: Combining the decentralized MTC200-R-G2 with the integrated MTC200-P-G2 is a practical configuration, for example, for a rotary transfer machine.

Here, the function interface allows parallel communication via various interfaces. In the following example, four programs are connected to the function interface in the direction of the clients. Every client can communicate with every device, independently of the other clients. When operating with several devices and several clients, the function interface works like a two-stage, buffered multiplexer. The communication process comprises a multiplexer in the direction of the device and the logic process comprises a multiplexer in the direction of the clients.





Devices





4 Programming

4.1 Guidelines

All user software (clients) that wants to access the function interface must be created in one of the following program languages:

- Visual C/C++ (32 bit version), or
- Visual Basic Version 5.0 and above.

The following should be observed when programming:

• the computer should be a Pentium Processor running at a min. of 200 MHz and with a RAM of at least 64 MB.

Note: Parts of the Rexroth Indramat function interface require the highest priority as a Windows NT process.

• Absolute paths should be avoided in the application as any later changes in the drive path (e.g. from C:\ to D:\) or in the directory structure are not supported.

Note: The system directory as well as the Windows NT disk drive can also be freely selected.

The following conditions and statuses of the controls or devices must be considered when programming:

- During a PLC program and/or parameter download from the Rexroth Indramat GUI, other applications must not read or write control data. The system messages (SYS-MSGs) from the call interface are used in evaluating this status. The system messages for the PLC program and/or parameter download are to be considered in the logic of the client.
- Reading and writing of PLC data is limited. Using the FI command "PVS" (see Chapter "Function Interface Commands"), PLC variables with a maximum length of 240 bytes can be read and written. PLC structures and arrays can have a dynamic length. Extremely precise planning is required for communication with the PLC.
- In principle, any PLC variable can be written using the function interface. However, only those PLC variables that are also found in the PLC program should be written in the application.

Note. Write-access to non-declared PLC variables should be avoided.

• Signals from the process/axis interface should never be directly changed by the application. Use a read/write buffer in the PLC.





The control system can only be operated safely and correctly with the function interface when the guidelines are observed.

If the guidelines are not observed then all claims against Rexroth Indramat are excluded.

Software for Developing User Programs (Clients) (PRO VERSION)

Note: In Version 07, the function interface cannot be installed separately, but only within the context of the relevant Rexroth Indramat GUI.

Settings for the C++ Development Environment

In order to make the functions of the "INDIF000.DLL" library of the function interface globally available, the following header files:

- INDIF000.H,
- INDIFX00.H and
- INDRAMAT.H

are to be included in the client with the syntax "#include".

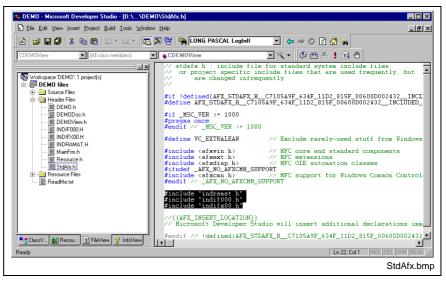


Fig. 4-1: Including the Rexroth Indramat Header Files in the Client

For Visual C++ 5.0, the entry "Multithreaded DLL" should be selected in the "For Win32 Release" project settings in the "C/C++" tab page under the category "Code-Generation" in the "Use run-time library" combo box.



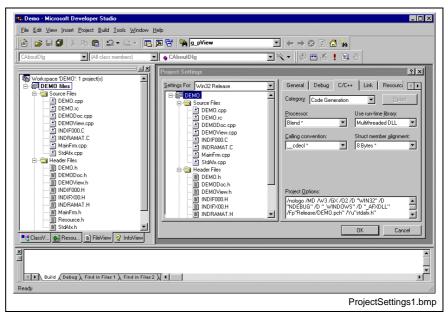


Fig. 4-2: Project settings "For Win32 Release": Multithreaded DLL

In project settings "For Win32 Debug", select the "Debug Multithreaded DLL" entry in the "Use run-time library" combo box under the category "Code-Generation" in the "C/C++" tab page.

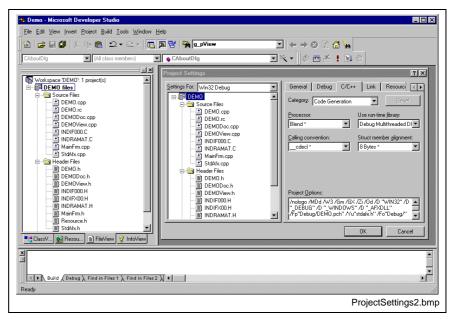


Fig. 4-3: Project settings "For Win32 Debug": Debug Multithreaded DLL

In addition, select the setting "Not using precompiled Headers" in the "For All Configurations" project settings in the "C/C++" tab page under the category "Precompiled Headers" for the following C sources:

- INDIF000.C and
- INDRAMAT.C.



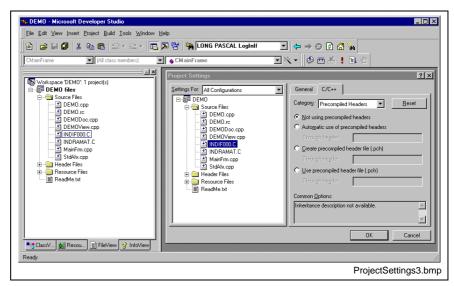


Fig. 4-4: "For All Configurations" project settings

4.2 Routines for Logging in and Logging Out

Before being able to use the access functions described in the following chapter, the login routine "LogInIf" must always be called up first. Once work with the function interface has been completed, then the logout routine "LogOutIf" should be called.

"LogInlf" Login Routine

- **Explanation** A client connects to the management structure of the function interface via the "LogInIf" routine.
 - Syntax LONG PASCAL LogInlf (CHAR *IpcTaskName, CHAR *IpcCommandLine, CHAR *IpcParentWinName, HANDLE *IhTerminateEvent, UCHAR luclfChannel,

UCHAR luclfChannelGrp,

HANDLE *IhSysMsgEvent,

UCHAR *lucTaskld,

DWORD *IdwIFChannelId);

Pass Parameters	Parameters	Explanation
	[IN] lpcTaskName	Pointer to the name of the client
	[IN] lpcCommandLine	Pointer to the command row for the management and logic process. As a rule, the pass parameters are passed on to the client here. By this means the function interface can be switched to diagnostics mode via the command row of the client.
	[IN] lpcParentWinName	Pointer to the name of the parent window of the process. Max. length = MAX_PARENT_WIN_NAME_LEN. (See file "INDIF000.H" or "INDIF000.BAS") NULL = no parent window (normal case)



Parameters	Explanation
[OUT] IhTerminateEvent	HANDLE to the termination event of a process.
[IN] luclfChannel	Decides whether or not the process requests a LOG channel 0= no LOG channel request >0 = LOG channel is requested (normal case).
[IN] luclfChannelGrp	Maximum number of function calls within a group request [1MAXGRP]. Default 10 (refer to entries in the "INDIFX00.H" file)
[OUT] IhSysMsgEvent	HANDLE on the SYS-Msg-Event.
[OUT] lucTaskld	TaskID, that is assigned to a client on logging in for management reasons [1MAX_TASK_ANZAHL] (see entries in the "INDIFX00.H" file).
[OUT] IdwIFChannelId	Assigned ID of the Communication Channel [2 to 8]

Return Values

0: Request successful.

1 ...n: Request unsuccessful (see chapter "Error Codes").

Note: Additionally, an error can be queried with the "ReadGroupItem" routine in the form of a general error result line. For additional information, refer to the chapter "General Error Result Lines".

LogInlf - Example (Visual Basic: VBDEMO.FRM)

Private Sub Form Load() 'INPUT-Values of the LogInIf-routine 'Application's name Dim TaskName As String Dim CommandLine As String 'Command for starting conditions, e.g. "/C=t /B=w" 'Titlebar's (Window's)name Dim ParentWinName As String Dim IfChannel As Byte 'Function Interface Channel Dim IfChannelGrp As Byte 'Value for group request 'Resultbuffer Dim ResBuf As String * 32768 'Return-Values of the LogInIf-routine Dim TaskId As Byte Dim IfChannelId As Long 'General declarations Dim lRet As Long 'Routine's returnvalue 'Error message string Dim ErrMsg As String Dim nHookList (0 To 4) As Integer 'Number of FI-System Messages (FI-SYS-MSGs) Dim lpThreadId As Long 'Timer interval initialisation TimerInterval.Caption = CyclicOutputTimer.Interval CycleTime.Value = CyclicOutputTimer.Interval TaskName = "VBDemo.exe" 'Application's name CommandLine = Command 'Command for starting conditions, e.g. /C=t /B=w 'Titlebar's (Window's)name ParentWinName = "VBDemo" 'Function-Interface Channel 1 requested IfChannel = 1 If Channel Grp = 10'Max. value for group request 'Default Returnvalue = 1 for error handling lRet = 1'Call LogInIf-Routine (Start Interface)



```
lRet = LogInIf(TaskName, CommandLine, ParentWinName, SysThread.hTerminateEvent,
IfChannel, IfChannelGrp, SysThread.hSysMsqEvent, TaskId, IfChannelId)
'Error handling & Function-interface channel identification output
If lRet Then 'error handling
   VBDemoStatus.BackColor = QBColor(12) 'set BackgroundColor to bright red
   ErrMsg = "LogIn Error code: " + CStr(lRet)
   VBDemoStatus.Caption = ErrMsg
     'Function-interface channel identification output
Else
   VBDemoStatus.BackColor = QBColor(10) 'set BackgroundColor to bright green
   VBDemoStatus.Caption = "Login succeeded on FI-Channel " & IfChannelId
End If
'Creating Funktion-Interface-System-Message-List (FI-SysMsg)
nHookList(0) = 4
                           'Number of FI-SYS-MSGs
nHookList(1) = MSG_PCLUPDBEG
                            'PLC Download Begin
                          'PLC Download End
'PLC Download End
'Parameter Download Begin
nHookList(2) = MSG_PCLUPDEND
nHookList(3) = MSG PARUPDBEG
nHookList(4) = MSG_PARUPDEND
                          'Parameter Download End
                                   'Ptr-Handed over in Basic is equal to C
lRet = HookIfMsgList(nHookList(0))
            'error handling
If lRet Then
   SYS Messages.BackColor = QBColor(12) 'set BackgroundColor to bright red
   ErrMsg = "HookIfMsgList terminated with error code: " + CStr(lRet)
   SYS_Messages.Caption = ErrMsg
End If
'Starting FI-SYS-Msg Thread
        *******
hThread = CreateThread(0,
                 Ο,
                 AddressOf SysThread.SysMsgThreadProc,
                 ο, _
                 Ο.
                 lpThreadId)
If hThread = 0 Then 'error handling
   SYS Messages.BackColor = QBColor(12) 'set BackgroundColor to bright red
   ErrMsg = "Thread couldn't be created" & Err.LastDllError
   SYS Messages.Caption = ErrMsg
End If
'Process verification for the Function-Interface
lRet = DataTransfer("XX BW RPR1", 0, 0, 1, ResBuf, 32768, 1)
End Sub
                         LogInlf - Example (Visual C++)
// General Declarations
LONG lRet;
CHAR acErrMsg[80];
// Input parameters of the LogInIf routine
```

```
HANDLE ghTerminateEv = ZERO;
HANDLE ghSysMsgEv = ZERO;
UCHAR gucTaskld = 0;
DWORD gdwlFChannelld
                        = 0;
// LogInIf routine (Start Interface)
IRet = LogInIf("VCDemo.exe", // Name of user program,
                        // Command, e.g. "/C=t",
m lpCmdLine,
                         // Window's Name,
"Demo",
                         // HANDLE on TerminateEvent,
&ghTerminateEv,
1,
                         // Interface channel requested,
                         // Max. number of function requests in group,
// HANDLE on SYS-Msg-Event,
10,
&ghSysMsgEv,
                        // Task-ID,
&gucTaskld,
```

Rexroth 🔵 Indramat

"LogOutlf" Log out Routine

Explanation	A client logs out from the management structure of the function interface via the "LogOutIf" routine.		
Syntax	LONG PASCAL LogOutIf ();		
Pass Parameters	None		
Return Values	0: Request successful.1 …n: Request unsuccessful (see chapter "Error Codes").		
	Note: Additionally, an error can be queried with the "ReadGroupItem" routine in the form of a general error result line. For additional information. refer to the chapter "General		

LogOutlf - Example (Visual Basic: VBDEMO.FRM)

```
Public Sub Form Terminate()
'IN-/Output Values
! * * * * * * * * * * * * * * * * *
Dim lRet As Long
                           'Routine's returnvalue
Dim ErrMsg As String
                           'Error message string
'Closing Function-Interface Channel
lRet = LogOutIf()
                          'Stop Function-Interface
If lRet Then 'error handling
   VBDemoStatus.BackColor = QBColor(12) 'set BackgroundColor to bright red
   ErrMsg = "LogOut termination with error code: " + CStr(lRet)
   VBDemoStatus.Caption = ErrMsg
End If
CloseHandle (hThread)
                          'Thread clearance
```

End Sub

LogOutlf - Example (Visual C++)

Error Result Lines".



4.3 Data Transfer and Result Evaluation Routines

The "DataTransfer" routine is used for reading and writing data to and from Rexroth Indramat devices. The data delivered in the result buffer is structured as follows:

						1
Single Request	Group E	Element 1	Line 1	Column 1		Column j
			:	:	:	:
			Line m	Column 1		Column j
Notes: In case of an error, (return value contains a general error result line evaluated in a separate routine (Codes").			that may	have to be		
	As only one command row and no group of command rows (also called group request) can be processed via the "DataTransfer" routine, it only has one results group. The data in the result buffer can be evaluated using the "ReadGroupItem" routine.					
	The "GetN for a group	lumberOfRo	ws" routine GetNumberC	as the number determines the offtems" routine	number o	f lines (rows)
Group request (ONLY for cyclic requests)						
	Group E	Element 1	Line 1	Column 1		Column j
			:	:	:	:
			Line i	Column 1		Column j
		:	:	:	:	:
	Group E	Element n	Line 1	Column 1		Column j
			:	:	:	:
			Line i	Column 1		Column j
Example of a group request	group ele elements i	ments can s as follows:	be accesse	S BR_ABN d with <i>[bGrou</i> µ	o] The me	
		Element (b)		BR_NPS.		
	•	Element (b)	• •	BR_ABN.		
	3. Group	Element (bo	Group = 3):	BR_AGF.		
	Note:		m of 256 co as a group r	ommand rows equest.	(FI comma	inds) can be



"DataTransfer" Routine

Explanation Data is read or written in accordance to the configured functions using the "DataTransfer" routine (see chapter "Function Interface Commands").

Syntax LONG PASCAL SetIfMsgConf (CH

CHAR *pszFunction, CHAR acValue[], LONG ValLen, LONG ValType, CHAR acResBuf[], LONG IMaxResLen, LONG IResBufType);

ass Parameters	Parameters	Explanation			
	[IN] pszFunction	Command row			
	[IN] acValue	Value to be written			
	[IN] ValLen	Length of value to be written			
	[IN] ValType	Data code of the value to be written (see chapter "Design and Availability of the F1 Command", Data Code)			
	[OUT] acResBuf	Result buffer			
	[IN] IMaxResLen	Length of the result buffer depending on the requested data. The RESULT_BUF_SIZE constant from the INDIF000.h file can be taken as the guide value.			
	[IN] IResBufType	Data code of results data (see Chapter "Design and Availability of the FI Command", Data code)			
	Note: The data delivered in the result buffer is coded. To access the single elements, the content of the result buffer must be processed using the "ReadGroupItem" routine:				
Return Values	0: Request successful. 1n: Request unsuccessful (see chapter "Error Codes").				
	an error can be queried in more detail with the em" routine in the form of a general error result tional information. refer to the chapter "General nes".				

DataTransfer - Example (Visual Basic: VBDEMO.FRM)

Private Sub DataTransferFunc()

'Read/Write Data from/to the various devices via the function-interface ************* Dim ResBuf As String * 32768 'Resultbuffer Dim lRet As Long 'Routine's returnvalue 'Value's length Dim lLen As Long Dim pszFunction As String 'FI-command 'Error message string Dim ErrMsg As String Dim ErrMsg As String 'Error message string Dim szBuf As String * 32768 'Buffer for controller data 'Flag for data validation Dim DataValidation As Boolean Dim szVal As String 'Writevalue pszFunction = SingleRequest.Text 'Hand over FI-command from Editbox szVal = WriteValue.Text 'Hand over WriteValue from Editbox 'DataTransfer to function-interface lRet = DataTransfer(pszFunction, szVal, Len(szVal), 1, ResBuf, 32768, 1) 'error handling If lRet Then ErrMsg = "DataTransfer terminated with error code: " + CStr(lRet) SingleRequestStatus.Caption = ErrMsg SingleRequestStatus.BackColor = QBColor(12) 'set BackgroundColor to bright red lRet = ReadGroupItem(ResBuf, 1, -1, -1, szBuf, 32768, lLen, DataValidation) Output.Text = szBuf Else 'Valid reply SingleRequestStatus.BackColor = QBColor(10) 'set BackgroundColor to bright green SingleRequestStatus.Caption = "DataTransfer command was succesfully completed" lRet = ReadGroupItem(ResBuf, 1, -1, -1, szBuf, 32768, lLen, DataValidation) Output.Text = szBuf End If End Sub DataTransfer - Example (Visual C++) // General Declarations //*************** ******** LONG lRet; CHAR acErrMsg[80]; int i,j; // Starting Parameters of the DataTransfer-Routine CHAR * szValue = ""; CHAR * szFunction = "02 CR CCP4"; CHAR acResultbuf [RESULT BUF SIZE]; // Starting Parameters of the GetNumberOfRows-Routine LONG lNumOfRows; // Starting Parameters of the GetNumberOfItems-Routine LONG lNumOfItems: // Starting Parameters of the ReadGroupItem-Routine LONG lltemLen; CHAR acItembuf[50]; BOOL boItemValid; // Access to Function Interface lRet = DataTransfer (szFunction, // Command row, szValue, // Value, strlen(szValue), // Length of value, // Data code of value, 1, // Result buffer, acResultbuf, // Length of result buffer,
// Data code of result data RESULT BUF SIZE, 1); // Error Handling

```
if (lRet)
     {
   sprintf(acErrMsg,"Function-Interface DataTransfer ErrorCode:%ld ",lRet);
     MessageBox (GetFocus(),acErrMsg,"Function Interface Error", MB OK);
     }
// Get number of rowa
//**************************
// Result data,
&lNumOfRows);
                                // Number of rows
 // Error Handling
 if (lRet)
     {
   sprintf(acErrMsg,"Function-Interface GetNumberOfRows ErrorCode:%ld ",lRet);
     MessageBox (GetFocus(),acErrMsg,"Function Interface Error", MB OK);
     }
// Result Evaluation
//***********************
for (i=1; i<=lNumOfRows; i++)</pre>
{
     // Determine Number of Elements
     lRet = GetNumberOfItems(acResultbuf,
                                      // Group element,
     1.
     i,
                                      // Row,
     &lNumOfItems);
                                      // Number of elements ?
     // Error Handling
     if (lRet)
     {
   sprintf(acErrMsg,"Function-Interface GetNumberOfItems ErrorCode:%ld ",lRet);
     MessageBox (GetFocus(),acErrMsg,"Function Interface Error", MB_OK);
     }
     // Evaluate all Lines
     //********
     for (j=1; j<=lNumOfItems; j++)</pre>
     {
       // Evaluate all Results of a Line
       lRet=ReadGroupItem(acResultbuf, // Result data,
                                      // Group element,
       1,
       i,
                                      // Row,
                                      // Column,
// Individual result,
       j,
       acItembuf,
                                      // Length of individual result buffer,
       50,
                                      // Length of result,
       &lItemLen,
       &boItemValid);
                                      // Valid value ?
       // Error Handling
         if (lRet)
        {
        sprintf(acErrMsg,"Function-Interface ReadGroupItem ErrorCode:%ld ",lRet);
       MessageBox (GetFocus(),acErrMsg,"Function Interface Error", MB_OK);
       }
     }
}
```



"ReadGroupItem" Routine

Pass P

Explanation This routine allows a single result, an entire row or a table of a single or group request to be read out. All results data must be evaluated using this routine.

Syntax LONG PASCAL ReadGroupItem (

CHAR acResBuf[] BYTE bGroup, LONG IRow, LONG IItem, CHAR acItemBuf[], LONG IItemBufLen, LONG *pIItemLen, BOOL *pboItemValid);

Parameters	Parameters	Explanation and Value Range
	[IN] CHAR acResBuf[]	Buffer for the entire result
	[IN] BYTE bGroup	Details of group element [1 to n]
	[IN] LONG IRow	-1: Output of a complete table, i.e. all rows of a request
		[1 to n]: the respective result line
	[IN] LONG IItem	 Output of a row Output of the requested command with management information
		[1n]: Individual result (Element of a row)
	[OUT] CHAR acItemBuf[]	Buffer for requested partial result
	[IN] LONG IItemBufLen	Length of buffer for partial result
	[OUT] LONG *plltemLen	Length of partial result
	[OUT] BOOL *pboltemValid	TRUE: with valid value of the partial result

Return Values

0: Request successful.

1 ...n: Request unsuccessful (see chapter "Error Codes").

Example of "ReadGroupItem" Routine

The following example assumes that a single request (bGroup = 1) has been requested:

Line	Column 1	Column 2	Column 3	Column 4
1	E1	E2	E3	
2	E4	E5		
3	E6	E7	E8	E9



Example of Syntax	Result
ReadGroupItem(acResBuf, 1, 1, 1, acItemBuf, ILenBuf, &IItemLen, &boItemValid)	E1
ReadGroupItem(acResBuf, 1, 2, 1, acItemBuf, ILenBuf, &IItemLen, &boItemValid)	E2
ReadGroupItem(acResBuf, 2, 2, 1, acItemBuf, ILenBuf, &IItemLen, &boItemValid)	E5
ReadGroupItem(acResBuf, 3, 4, 1, acItemBuf, ILenBuf, &IItemLen, &boItemValid)	E9
ReadGroupItem(acResBuf, 2, 3, 1, acItemBuf, ILenBuf, &IItemLen, &boItemValid)	Error code
ReadGroupItem(acResBuf, 1, -1, 1, acItemBuf, ILenBuf, &IItemLen, &boItemValid)	E1 E2 E3
ReadGroupItem(acResBuf, 2, -1, 1, acItemBuf, ILenBuf, &IItemLen, &boItemValid)	E4 E5
ReadGroupItem(acResBuf, 3, -1, 1, acItemBuf, ILenBuf, &IItemLen, &boItemValid)	E6 E7 E8 E9
ReadGroupItem(acResBuf, -1, -1, 1, acItemBuf, ILenBuf, &IItemLen, &boItemValid)	E1 E2 E3 E4 E5 E6 E7 E8 E9
ReadGroupItem(acResBuf, 1, 0, 1, acItemBuf, ILenBuf, &IItemLen, &boItemValid)	supplies e.g.: 001234567800_CC

Example of Visual Basic/ C++ (see "DataTransfer" routine)

"GetNumberOfGroups" Routine

Explanation	The "GetNumberOfGroups" elements.	routine	returns	the	number	of	group
Syntax	LONG PASCAL ReadGroup	ltem (СН	AR *p	szValBuf	,	

LONG *plGroupSize);

Pass Parameters	Parameters	Explanation	
	[IN] CHAR *pszValBuf	Buffer for the entire result	
	[OUT] LONG *plGroupSize	Number of group elements	
Return Values	 0: Request successful. 1n: Request unsuccessful (see chapter "Error Codes"). 		
	Note: Additionally, an error can be queried in more detail with th "ReadGroupItem" routine in the form of a general error resuline. For more detailed information, please refer to the chapt "Error Code".		
Example of Visual Basic/ C++	+ (see "DataTransfer" routine)		

"GetNumberOfRows" Routine

Explanation The "GetNumberOfRows" routine determines the number of rows of the indicated group element.

Syntax LONG PASCAL GetNumberOfRows (CHAR *pszValBuf, BYTE bGroupIndex, LONG *plNumberOfRow);

Pass Parameters	Parameters	Explanation		
	[IN] CHAR *pszValBuf	Buffer for the entire result		
	[IN] BYTE bGroupIndex	Number of group elements		
	[OUT] LONG *plNumberOfRow	Number of rows of a group element		
Return Values	0: Request successful. 1 …n: Request unsuccessful (see chapter "Error Codes").			
	Note: Additionally, an error can be queried in more detail with the "ReadGroupItem" routine in the form of a general error result line. For additional information. refer to the chapter "General Error Result Lines".			
Example of Visual Basic/ C++	(see "DataTransfer" routine)			

"GetNumberOfItems" Routine

Explanation	The "GetNumberOfItems" routine determines the number of partial
	results, depending on the row number as well as the number of the group element.

Syntax	LONG PASCAL GetNumberOfItems (CHAR *pszValBuf,
		BYTE bGroupIndex,
		BYTE bRowIndex,
		LONG *plNumberOfItems);
s Parameters	Parameters	Explanation

Pass Parameters	Parameters	Explanation
	[IN] CHAR *pszValBuf	Buffer for the entire result
	[IN] BYTE bGroupIndex	Number of group elements
	[IN] BYTE bRowIndex	Row index 0: number of all partial results
	[OUT] LONG plNumberOfItems	Number of partial results for a particular row.
Return Values	es 0: Request successful. 1n: Request unsuccessful (see chapter "Error Codes").	
Note: Additionally, an error can be queried in "ReadGroupItem" routine in the form of a line. For additional information. refer to the Error Result Lines".		ne form of a general error result
Example of Visual Basic/ C++	(see "DataTransfer" routine)	



4.4 Routine for Cyclical Reading via Pipes

The pipe access functions are used for cyclical reading of device data via the function interface. Several command rows can be passed simultaneously via a group request. The command rows of a group request are separated by a space (refer here also to the "ReadGroupItem" routine).

Note: A maximum of 256 command rows (FI commands) can be abstracted as a group request.

A pipe is started by the "StartCyclicPipe" routine and then provides itself continually with updated data. Asynchronous to this, access to this data is now made via the "ReadCyclicPipe" routine. The cyclical request is stopped by the "StopCyclicPipe" routine.

"StartCyclicPipe" Routine

Explanation The "StartCyclicPipe" routine starts a pipe for cyclical reading of the data.

Syntax LONG PASCAL StartCyclicPipe (WORD wPipe, CHAR *pszFunktion, LONG IBufSize, LONG IGroupSize, DWORD dwSleep);

Pass Parameters	Parameters	Explanation
	[IN] wPipe	Pipe number [11000]
	[IN] *pszFunktion	Group of strings according to the defined function requests.
	[IN] IBufSize	Size of result buffer [Byte]
	[IN] IGroupSize	Number of group elements [1 to n]
	[IN] dwSleep	Read delay time [ms]
"ReadGroupItem" routine in the form of a		
		onally, an error can be queried with the GroupItem" routine in the form of a general error result for additional information. refer to the chapter "General

Error Result Lines".



StartCyclicPipe - Example (Visual Basic: VBDEMO.FRM)

Public Sub StartCyclicFunc() 'Start of a cyclic request ********* Dim lRet As Long 'Routine's returnvalue 'Error message string Dim ErrMsg As String Dim pszFunction As String 'FI-command pszFunction = CyclicRequest.Text 'Hand over FI-Command from Editbox If Not CyclicRun Then 'in case of a cyclic request has NOT been started lRet = StartCyclicPipe(1, pszFunction, 32768, 2, 250) If lRet Then 'error handling CyclicRequestStatus.BackColor = QBColor(12) 'set BackgroundColor to bright red ErrMsg = "StartCyclicPipe terminated with error code:" + CStr(lRet) CyclicRequestStatus.Caption = ErrMsg Exit Sub 'in case of an error has occured End If 'Flag for a cyclic request is started CyclicRun = True CyclicOutputTimer.Enabled = True 'Timer output is started End If End Sub

StartCyclicPipe - Example (Visual C++)

// General Declarations

```
LONG lRet;
CHAR acErrMsg[80];
// Starting Parameters of the StartCyclicPipe - Routine
//*****
CHAR * szGroupFunction = "00_CC_AGF_0 00_CC_PVS_ErrorFlg";
// Open Pipe
//*******
lRet = StartCyclicPipe(wPipeNo,
                                   // Pipe - number,
                              // Function call group,
szGroupFunction.
RESULT BUF SIZE,
                              // Size result buffer,
                              // Number group elements,
2,
                              // Reading delay time [ms]
500);
// Error Handling
//*****
if (lRet)
    {
  sprintf(acErrMsg,"Function-Interface LogInIf ErrorCode:%ld ",lRet);
     MessageBox (GetFocus(),acErrMsg,"Function Interface Error", MB_OK);
    }
```



"ReadCyclicPipe" Routine

The "ReadCyclicPipe" routine reads the data of a pipe that has been Explanation started by "StartCyclicPipe".

Syntax LONG PASCAL ReadCyclicPipe (

WORD wPipe, CHAR acResult[], LONG IBufSize, BYTE *pbGroupFault, LONG *plAttr);

Pass Parameters	Parameters	Explanation
	[IN] WORD wPipe	Number of the pipe
	[OUT] CHAR acResult[]	Buffer for the entire result
	[IN] LONG IBufSize	Buffer size of the entire result
	[OUT] BYTE *pbGroupFault	Number of the group element in case of error
[OUT] LONG *plAttr		Result attribute
Return Values	s 0: Request successful.	

1 ...n: Request unsuccessful (see chapter "Error Codes").

Note: The pass parameter [OUT] BYTE *pbGroupFault contains the number of the faulty group element. Further information on the meaning of the "ReadCyclicPipe" routine error code can be requested in the form of a general error result line using the "ReadGroupItem" routine. For additional information. refer to the chapter "General Error Result Lines".

ReadCyclicPipe - Example (Visual Basic: VBDEMO.FRM)

```
Private Sub CyclicOutputTimer Timer()
```

```
'IN-/Output Values
! * * * * * * * * * * * * * * * * *
Dim lRet As Long
                                     'Routine's returnvalue
Dim ErrMsg As String
                                     'Error message string
Dim ResultBuffer As String * 32768
Dim lNumberOfRows As Long
                                     'Number of Rows ->
Dim i As Long
                                     'Index for the number of rows
Dim szBuf As String * 256
                                     'Buffer for controller data
Dim lLen As Long
                                     'Value's lenght
Dim DataValidation As Boolean
                                     'Flag for data validation
Dim bGroup As Byte
Dim lAttr As Long
lRet = ReadCyclicPipe(1, ResultBuffer, 32768, bGroup, lAttr)
If lRet Then
               'error handling
    CyclicRequestStatus.BackColor = QBColor(12) 'set BackgroundColor to bright red
    ErrMsg = "ReadCyclicPipe terminated with error code: " + CStr(lRet)
    CyclicRequestStatus.Caption = ErrMsg
    Exit Sub
End If
OutputList.Clear
If lRet = 0 Then
            lRet = GetNumberOfRows(ResultBuffer, 1, lNumberOfRows)
            Rows.Text = lNumberOfRows
                          'error handling
            If lRet Then
                CyclicRequestStatus.BackColor = QBColor(12) 'set BackgroundColor to
bright red
                ErrMsg = "GetNumberOfRows terminated with error code: " + CStr(lRet)
                CyclicRequestStatus.Caption = ErrMsg
```



```
End If
           For i = 1 To lNumberOfRows
               lRet = ReadGroupItem(ResultBuffer, 1, i, -1, szBuf, 32768, lLen,
DataValidation)
               If lRet Then
                             'error handling
                  CyclicRequestStatus.BackColor = QBColor(12) 'set BackgroundColor to
bright red
                  ErrMsg = "ReadGroupItem terminated with error code: " + CStr(lRet)
                  CyclicRequestStatus.Caption = ErrMsg
               End If
               OutputList.AddItem (szBuf)
           Next
           CyclicRequestStatus.BackColor = QBColor(10) 'set BackgroundColor to bright
green
           CyclicRequestStatus.Caption = "ReadCyclicPipe command was succesfully
completed"
End If
End Sub
                           ReadCyclicPipe - Example (Visual C++)
// General Declarations
//*********************************
LONG lRet;
CHAR acErrMsg[80];
int i;
// Input parameters of the ReadCyclicPipe routine
CHAR acResultbuf [RESULT_BUF_SIZE];
UCHAR bIndexItemFault;
LONG lAttr;
// Input parameters of the GetNumberOfGroups routine
//*****
                ***********
LONG lNumOfGroups;
// Read pipe
//*********
lRet = ReadCyclicPipe(wPipeNo,
                                      // Pipe number,
                                      // Result buffer,
acResultbuf,
RESULT_BUF_SIZE,
                                      // Length result buffer,
&bIndexItemFault,
                                      // Index of the group
                                      // element with error,
// result attribute
&lAttr);
// Error handling
if (lRet)
{
     sprintf(acErrMsg,"Function-Interface ReadCyclicPipe ErrorCode: %ld ",lRet);
   MessageBox (GetFocus(),acErrMsg,"Function Interface Error", MB_OK);
// Determine number of groups
// Result buffer,
lRet = GetNumberOfGroups(acResultbuf,
      &lNumOfGroups); // Number of groups,
// Error handling
if (lRet)
sprintf(acErrMsg,"Function interface GetNumberOfGroups ErrorCode: %ld",lRet);
  MessageBox (GetFocus(),acErrMsg,"Function Interface Error", MB OK);
// Evaluation of results
//*******************
for (i=1; i<=lNumOfGroups; i++)</pre>
{
     // Results evaluation for each group result
     // e.g. LONG lItemLen;
            CHAR acItembuf[50];
     11
     11
            int iItemValid;
     11
                                     // Result buffer,
     lRet=ReadGroupItem(acResultbuf,
                               // Group element,
// Line,
     i,
    1,
     1,
                                // Element,
                               // Individual result buffer,
// Length of the individual result buffer,
     acItembuf,
     50,
                               // Length of the individual result,
     &lItemLen,
     &iItemValid);
                                // Individual result valid?
```

```
// Error handling
if (lRet)
{
    sprintf(acErrMsg,"Function-Interface ReadGroupItem ErrorCode:%ld ",lRet);
    MessageBox (GetFocus(),acErrMsg,"Function Interface Error", MB_OK);
    }
```

"StopCyclicPipe" Routine

Explanation	The "StopCyclicPipe" routine stops the data request of a pipe that has been started by "StartCyclicPipe". LONG PASCAL StopCyclicPipe (WORD wPipe);		
Syntax			
Pass Parameters	Parameters	Explanation	
	[IN] WORD wPipe	Pipe number	
Return Values	0: Request successful. 1n: Request unsuccessful (see chapter "Error Codes").		
	•	can be queried with the n the form of a general error result tion. refer to the chapter "General	

StopCyclicPipe - Example (Visual Basic: VBDEMO.FRM)

```
Public Sub StopCyclicFunc()
'Stop of a cyclic request
'IN-/Output Values
Dim lRet As Long
                          'Routine's returnvalue
Dim ErrMsg As String
                          'Error message string
'Cyclic request termination
   *******
If CyclicRun Then 'in case of a cyclic request has been started
   CyclicOutputTimer.Enabled = False
                                         'Timer output is stopped
   lRet = StopCyclicPipe(1)
   If lRet Then
                 'error handling
       CyclicRequestStatus.BackColor = QBColor(12) 'set BackgroundColor to bright red
       ErrMsg = "StopCyclicPipe terminated with error code:" + CStr(lRet)
       CyclicRequestStatus.Caption = ErrMsg
   End If
   CyclicRun = False 'Flag for a cyclic request is stopped
End If
End Sub
```

Error Result Lines".

StopCyclicPipe - Example (Visual C++)

"SuspendCyclicPipe" Routine

Explanation	The "SuspendCyclicPipe" routine sets the data request of a pipe that has been started by "StartCyclicPipe" into standby mode. It is used to stop communication while at the same time maintaining the management structure
	of the function interface established by the "StartCyclicPipe" routine (see "ResumeCyclicPipe" routine).

Syntax	LONG	PASCAL SuspendCyclicPipe(WORD wPipe);
Pass Parameters	Parame	ters	Explanation
	[IN] WO	RD wPipe	Number of the pipe
Return Values	es 0: Request successful. 1n: Request unsuccessful (see chapter "Error Codes").		r "Error Codes").
	Note:	Additionally, an error car "ReadGroupItem" routine in the line. For additional information. Error Result Lines".	form of a general error result

SuspendCyclicPipe - Example (Visual Basic: VBDEMO.FRM)

```
Public Sub SuspendCyclicFunc()
'Stand-by-Modus for a cyclic request
'IN-/Output Values
! * * * * * * * * * * * * * * * * *
Dim lRet As Long
                           'Routine's returnvalue
Dim ErrMsg As String
                           'Error message string
 If CyclicRun Then 'in case of a cyclic request has been started
   CyclicOutputTimer.Enabled = False
                                           'Timer output is stopped
   lRet = SuspendCyclicPipe(1)
                   'error handling
     If lRet Then
         CyclicRequestStatus.BackColor = QBColor(12) 'set BackgroundColor to bright red
         ErrMsg = "SuspendCyclicPipe terminated with error code: " + CStr(lRet)
         CyclicRequestStatus.Caption = ErrMsg
    End If
 End If
End Sub
```

SuspendCyclicPipe - Example (Visual C++)

"ResumeCyclicPipe" Routine

Explanation	The "ResumeCyclicPipe" routine reactivates the data request of a pipe that has been set to standby mode by the "SuspendCyclicPipe" routine.	
Syntax	LONG PASCAL SuspendCyclicPipe (WORD wPipe);
Pass Parameters	Parameters	Explanation
	[IN] WORD wPipe	Number of the pipe
Return Values	 0: Request successful. 1n: Request unsuccessful (see chapter "Error Codes"). 	
	"ReadGroupItem" routine in the	n be queried with the e form of a general error result refer to the chapter "General

ResumeCyclicPipe - Example (Visual Basic: VBDEMO.FRM)

```
Public Sub ResumeCyclicFunc()
'Activates a suspended cyclic Pipe
**********************
'IN-/Output Values
! * * * * * * * * * * * * * * * * *
Dim lRet As Long
                           'Routine's returnvalue
                           'Error message string
Dim ErrMsg As String
'Cyclic request termination
If CyclicRun Then 'in case of a cyclic request has been started
   CyclicOutputTimer.Enabled = True 'Timer output is started
   lRet = ResumeCyclicPipe(1)
   If lRet Then
                 'error handling
       CyclicRequestStatus.BackColor = QBColor(12) 'set BackgroundColor to bright red
       ErrMsg = "ResumeCyclicPipe terminated with error code: " + CStr(lRet)
       CyclicRequestStatus.Caption = ErrMsg
   End If
End If
End Sub
```



ResumeCyclicPipe - Example (Visual C++)

4.5 System messages in the Network

SYS-Messages are used to report particular events in the system to the applications. Generally speaking the application is synchronized to the changed control system data.

Examples of this are PLC program download, parameter download, system error.

Example

PLC program download

Before the PLC GUI downloads a PLC program the SysMessage MSG_PCLUPDBEG is reported.

The PLC GUI reports the end of the download with the SYS-Message MSG_PCLUPDEND.

After that, the applications will re-synchronize access to the PLC data.

PLC user interface	SYS Message	Application status
Translate program		PLC available (1)
	\rightarrow MSG_PCLUPDBEG \rightarrow	PLC available
		PLC blocked
	← MSG_PCLUPDBEG_Q ←	PLC blocked
Download		PLC blocked
	\rightarrow MSG_PCLUPDEND \rightarrow	PLC blocked
		PLC blocked
	\leftarrow MSG_PCLUPDEND_Q \leftarrow	PLC available
!! Download ended!!		PLC available

Fig. 4-5: Synchronization via SYS-Messages

- Transparent across the network.
- Messages logged in to a particular device.
- Multiple logins in one application to support different process packages running in a process.
- Dynamically expandable message type for messages relating to the application such as for updating global network data.
- Failure of the network connection: The failure of the connection to a device is acknowledged for each requested message.

Data Types

	TyVMsgCallback The following callback has been defined for receiving network messages:		
Declaration	LONG (TyVMsgCall back*)(VOID LONG CONST CHAR CONST CHAR LONG	*pParam, lMsgId, *psczDevice *pcData lDataLen);
	Parameters		
lMsgld	Numerical identification of the SYS-Message.		
	Gives the system written.	area in which the co	ontrol component messages are
	Gives the user are IDs.	ations can be assigned message	
psczDevice	The device for which	ch a message is gene	rated or received.
pcData	User data of a message.		

Programming interface

Constants

SYSMSG_ALLPC	Login for all PCs in the network
SYSMSG_ALLDEVICE	Login for all devices
SYSMSG_LOCALPC	Login for all devices of the local PC
Flags for SysMsgHookCreate	
SYSMSG_MANUAL_ACK	Manual acknowledgement
This flag must be set if the ac	knowledgement of the system messa

This flag must be set if the acknowledgement of the system message is not to be carried out automatically after the callback function.

 $\ref{eq:linear}$ In this case, the application must call up

SYSMSG_LOGOUT_CALLBACK even when logging out.

TySysMsgCallbackInfo Data Type

This data type is used to provide the callback function with data from the system message.

	•
typedef	struct {
VOID	*pUserParam;
LONG	<pre>lDevice;</pre>
CONST	CHAR *pcData;
LONG	<pre>lDataLen;</pre>
LONG	lStatus;
LONG	lHookId;
LONG	lMsgId;
LONG	lPc;
LONG	lFarDevice;
)	a 111 1 7 6

} TySysMsgCallbackInfo;



Name	Description
pUserParam	User-defined parameter. Refer also to SysMsgHookCreate
IDevice	The local device address of the system message, or SYSMSG_ALLDEVICE, if the system message is not issued specifically for the device.
pcData	Message data
IdataLen	Length of data
IStatus	Callback status 0: OK Otherwise, logout or error
lHookld	ID of the connected hook
lMsgld	The message ID
IPc	The PC from which the system message originates. SYSMSG_LOCALPC, if the system message has been requested from the local PC.
IFarDevice	The device address in the network. If the network has not been activated then the local device address is accepted. SYSMSG_ALLDEVICE, if the system message has not been issued for a specific device.

TySysMsgCallback Data Type

typedef VOID (__stdcall

*TySysMsgCallback) (TySysMsgCallbackInfo *pCbInfo);

The function pointer data type for the callback.

SysMsgHookCreate

The callback function is called up in a try catch block.

The callback is activated in a separate thread.

Note: No further system messages are handled while the function is active.

The transmitted data is generated on the stack and becomes invalid at the end of the return of the function.

Declaration	LONG SysMsgHookCreate(LONG LONG LONG LONG TySysMsgCallback VOID LONG	<pre>*plHookId, lDevice, lPC, lMsgId, CallbackFunc, *pParam, lFlags);</pre>
Parameters:	plHookId	Output Parameter: Handle on the hook	
This value is needed to deactivate the callback acknowledgement. Refer also to SysMsgHookD SysMsgHookAcknowledge()			
	IDevice	Device number for which this callback i or Far Device number.	s to be activated. A valid local
		To log on the system message for all de SYSMSG_ALLDEVICE can also be tran the PC for which the callback is activated	sferred. In this case IPC defines
IPC Defines		Defines the PC for which the callback is	s to be activated.
		!! This parameter is only effective if IDevice is transferred with SYSMSG_ALLDEVICE.	
		SysMsgHookCreateAll() must be used to devices of all PCs in a network.	activate the callback for all



Declaration	LONG SysMsgHookCreate(LONG LONG LONG TySysMsgCallback VOID LONG	<pre>*plHookId, lDevice, lPC, lMsgId, CallbackFunc, *pParam, lFlags);</pre>
	lMsgld	The message ID	
	CallbackFunc	Callback user function. !! This function is activated in a separate thread.	
		11 No further system messages are h active.The transmitted data is generated or	
		the end of the return of the function.	
	pParam	User Parameter for the callback func	ction.
	IFlags	The value of this parameter switches on certain hook options. The options can be abstracted by means of a logical OR operation ' '.	
		The system message is not acknowl callback function is exited. The appli of the system message by means of function.	cation must acknowledge receipt
		SYSMSG_LOGOUT_CALLBACK: On LogoutIF(), the callback is activation	ted one last time.
		If necessary, the application can dele hook.	ete user data allocated to the
Return value:	0	OK	
	NET_ETIMEOUT:	Time for making a connection to a re	emote device has been exceeded.
	NET_EINVPARAM	Invalid parameter	
	NET_EFALSE	Loginlf must be called up before sys	tem messages can be logged in.

SysMsgHookAcknowledge

Acknowledgement of a system message in the network.

Declaration	LONG SysMsgHookAcknowledge	(LONG lHookId);
Parameters:	lHookId	HookId of the system message that is to be acknowledged.
Return value:	0	ОК
	NET_EINVPARAM	Invalid parameter
	NET_EFALSE	LoginIf must be called up before this function can be called up.

SysMsgHookDelete

Deletion of a system message in the network.

Declaration	LONG SysMsgHookDelete	(LONG	lHookId);
Parameters:	lHookId	HookId of the sys	tem message that is to be deleted.
Return0value:	0	OK	
	NET_EINVPARAM	Invalid parameter	
	NET_EFALSE	Loginlf must be c	alled up before this function can be called up.



SysMsgHookCreateAll

Generates multiple hooks for system messages. A hook is generated for each PC in the network and is activated for each device.

A hook ID is returned for each PC. If plHookId is 0, the hook IDs are not returned.

Declaration	LONG SysMsgHookCreateAll	(LONG LONG LONG LONG TySysMsgCallback VOID LONG	<pre>*plHookId, lHooktabMax, *plHooktabCnt, lMsgId, CallbackFunc, *pParam, lFlags);</pre>
Parameters:	plHookld	Output parameter: handles on the hool	ts to the various PCs
		These values are needed to deactivate the callback or to make a manual acknowledgement. Refer also to SysMsgHookDelete() a SysMsgHookAcknowledge()	
	IHookTabMax	Size of the hook table the user is administering. → The function generates as many hooks as there are PCs declared in the network.	
	plHooktabCnt	Transfer of number of hooks generated	I.
	IMsgld	The message ID	
	CallbackFunc	Callback user function.	
		!! This function is activated in a separa	te thread.
		I No further system messages are han active.	dled while the function is
		The transmitted data is generated on the invalid at the end of the return of the fu	
	pParam	User Parameter for the callback function.	
	IFlags	The value of this parameter switches o options can be OR operated ' '.	n certain hook options. The
		The system message is not acknowled callback function is exited. The applicar receipt of the system message by mea SysMsgHookAcknowledge() function.	tion must acknowledge
		SYSMSG_LOGOUT_CALLBACK: On LogoutIF(), the callback is activated	l one last time.
		If necessary, the application can delete hook.	e user data allocated to the
Return value:	0	ОК	
	NET_ETIMEOUT:	Time for making a connection to a remote device has been exceeded.	
	NET_EINVPARAM	Invalid parameter	
	NET_EFALSE	LoginIf must be called up before syster in.	n messages can be logged

Example of Programming

The following example describes an application of the system message mechanism. A class is declared that works with the functions described:

Declaration

```
#include "indif000.h" // Declaration of the FI-Routines
//trigger up to maximum of 100 PCs in the PC network(more
than //100 are currently not permitted)
#define D nSYSMSG MAX PC
                        100
//e.g., react to 28 messages
#define D_nSysMSG_COUNT
                        28
// Data structure to management (s.u.)
typedef struct _TyHookTable
{
 LONG alHookId[D nSYSMSG MAX PC];
 LONG lCount;
} TyHookTable;
// Class declaration
_____
 Description:
Sample Class for FI System message Handling
        _ _ _ _ _ _ _ _ _ _ _ _
History:
class KSysMsgSample
{
 //*
 //* Construction
 //*
 public:
     KSysMsgSample();
     ~KSysMsgSample();
// ...
 //*
 //* Attributes
 //*
 private:
     // Callback Routine
     static void stdcall
     SysMsgCallback(TySysMsgCallbackInfo* pCbInfo);
```



```
//Help Routines
       LONG HookSysMsg (void);
      LONG UnhookSysMsg (void);
       // Data structures to SysMsg-Handling
       static LONG s_alMsgList[D_nSysMSG_COUNT];
       static LONG s_lMsgCount;
TyHookTable c_aoHookTable[D_nSysMSG_COUNT];
};
Implementation
// List of SYS messages to be handled (example)
LONG KSysMsgSample::s_alMsgList[] =
{
  MSG PARUPDBEG,
  MSG_PCLUPDBEG,
  MSG_LAGCHABEG,
  MSG_MECERRGEN,
  MSG SYSERRGEN,
  MSG FWAUPDBEG,
  MSG MEMUPDBEG,
  MSG_ACTERRBEG,
  MSG DEVERRBEG,
  MSG_MDLERRBEG,
  MSG_PARUPDEND,
  MSG_PCLUPDEND,
  MSG_LAGCHAEND,
  MSG_MECERRDEL,
  MSG_SYSERRDEL,
  MSG_FWAUPDEND,
  MSG MEMUPDEND,
  MSG ACTERREND,
  MSG_DEVERREND,
  MSG_MDLERREND,
  MSG_STRTUPCHG,
  MSG_WARNINCHG,
  MSG_SETUP_CHG,
  MSG_MESSAGCHG,
  MSG_ERROR_CHG,
  MSG_SFCERRCHG,
  MSG_SFCMODCHG,
  MSG DMPSELCHG
};
// Login of system messages
LONG KSysMsgSample::HookSysMsg (void)
{
  LONG lResult = 0;
```



```
for (INT i=0; i< D_nSysMSG_COUNT; ++i)</pre>
  {
    // Login SysMessages for all PCs in network.
    lResult = :: SysMsgHookCreateAll(
       c_aoHookTable[i].alHookId,
       D_nSYSMSG_MAX_PC,
       &(c_aoHookTable[i].lCount),
      s_alMsgList[i],
      SysMsgCallback,
      static_cast<VOID*>(this),
      0);
  }
  return lResult;
}
// Logout of system messages
LONG KSysMsgSample::UnhookSysMsg (void)
{
  for (INT i=0; i< D_nSysMSG_COUNT; ++i)</pre>
  {
    for (INT j=0; j<c_aoHookTable[i].lCount; j++)</pre>
    {
      ::SysMsgHookDelete(
      c aoHookTable[i].alHookId[j]);
    }
  }
  return 0;
}
// Callback function
void KSysMsgSample::SysMsgCallback(
  TySysMsgCallbackInfo* pCbInfo)
{
  KSysMsgSample * pInstance =
  static cast<KSysMsgSample*>(pCbInfo->pUserParam);
  switch (pCbInfo->lMsgId)
  {
      case MSG PARUPDBEG:
       case MSG_PCLUPDBEG:
       case MSG_FWAUPDBEG:
       // handle begin download
         break;
      case MSG PCLUPDEND:
       case MSG_PARUPDEND:
       case MSG FWAUPDEND:
         // handle end download
         break;
```



//etc. } }

What SYS Messages are there and how should I react to them?

The SYS messages that are most often used for a client are listed in the following table. All system messages can be found in the configuration files "INDIF000.H" and "INDIF000.BAS". The system messages always consist of a SYS-MSG and a SYS-MSG acknowledgement.

Note: The device address that has set the system message is returned as an ASCII character in the buffer "CHAR *pcSysMsgBuffer" of "GetIfMsg". The buffer may possibly contain additional SYS-MSG information such as the parameter identification string.

SYS Message	What happens?	Reaction from the Client
MSG_FWAUPDBEG	The firmware download commences, i.e. the firmware is loaded into the System200 components by the GUI.	Communication between all configured Indramat devices is interrupted.
MSG_FWAUPDEND	End of firmware download.	Communication recommences. PLEASE NOTE: Process and axis configuration data etc., may have changed.
MSG_PARUPDBEG	The parameter download commences, i.e. the parameter set is loaded into the System200 components by the user interface.	Communication between all configured Indramat devices is interrupted.
MSG_PARUPDEND	End of parameter download	Resumption of communicationPLEASE NOTE: Process and axis configuration data etc., may have changed.
MSG_PCLUPDBEG	The PLC program download commences, i.e. the PLC program is loaded into the System200 components by the GUI.	Communication between all configured Rexroth Indramat devices is interrupted.
MSG_PCLUPDEND	End of PLC program download	Communication recommences.
MSG_MEMUPDBEG	GUI begins to delete the data memory in the System200 components.	Communication between all configured Indramat devices is interrupted.
MSG_MEMUPDEND	GUI has deleted the data memory in the System200 components.	Communication recommences. PLEASE NOTE:Configuration data has been deleted.
MSG_SYSERRGEN	If there is a system error, this SYS-MSG is issued, i.e. the PLC cannot be accessed at the moment.	No interruption of communication to the Rexroth Indramat devices is necessary. (Used for presenting a system error from a particular Rexroth Indramat device in graphic form).
MSG_SYSERRDEL	A system error is deleted.	No interruption of communication to the Rexroth Indramat devices is necessary. (Information that the system error is no longer present at a particular Rexroth Indramat device.)

MSG_MECERRGEN	This SYS-MSG is issued if there is a fault in the mechanism.	No interruption of communication to the Rexroth Indramat devices is necessary. (Is used for presenting a system error from a particular Rexroth Indramat device in graphic form).
MSG_MECERRDEL	A mechanism error is deleted.	No interruption of communication to the Rexroth Indramat devices is necessary. (Information that the system error is no longer present at a particular Rexroth Indramat device.)
MSG_LAGCHABEG	A language switch has been initialized at the Rexroth Indramat GUI.	No interruption of communication to the Rexroth Indramat devices is necessary. (Information that the user interface language is being switched.)
MSG_LAGCHAEND	A language switch has been completed at the Rexroth Indramat GUI.	No interruption of communication to the Rexroth Indramat devices is necessary. (Information, that the user interface language has been switched.)
MSG_PCALIVE	A PC/device logs in/out of the PC network. The device address/FarDevice address that has set the system message is returned as an ASCII character in the buffer "CHAR *pcSysMsgBuffer" of the "GetIfMsg". The buffer also contains the information "0" for PC logged out or a "1" for PC logged in.	Depending on the application, this system message is used on the one hand purely as information regarding the logging in/logging out of PCs. On the other hand, communication to the PC/device or the re-establishment of communication with subsequent synchronization is required.

SYS-MSG Example (Visual Basic: SYSTHREAD.BAS)

```
Public Sub SysMsgThreadProc()
'This subroutine is processing FI-SYS-Messages
******
                  Dim lWait As Long
Dim hEvList(0 To 1) As Long
'Transmission of events whose the thread is reacting
hEvList(0) = hTerminateEvent
hEvList(1) = hSysMsgEvent
Demo.SYS Messages.BackColor = QBColor(10) 'set BackgroundColor to bright green
Demo.SYS_Messages.Caption = "Thread is runnning"
Do 'Threadloop
   lWait = WaitForMultipleObjects(2, hEvList(0), 0, INFINITE
   If lWait = 0 Then 'TerminateEvent from another FI-application has occured
       Demo.SYS Messages.BackColor = QBColor(12) 'set BackgroundColor to bright red
       Demo.SYS_Messages.Caption = "Terminate Event has occured"
       Demo.TerminateEvent = True
                 'End of the threadloop
       Exit Do
   ElseIf lWait = 1 Then
   'SysMsqs which are received from the FI
    Dim lRet As Long
                                      'Routine's returnvalue
       Dim nMsgNr As Integer
       Dim strMsgBuf As String * 256
       Dim strTaskName As String * 256
       Dim szMsg As String
       'Getting SYS-MSG-number
       lRet = GetIfMsg(nMsgNr, strMsgBuf, 256, strTaskName)
       If lRet Then
                     'error handling
           szMsg = "GetIfMsg terminated with error code: " + CStr(lRet)
          Demo.SYS_Messages.Caption = szMsg
       Else
          Select Case nMsqNr
              Case MSG PCLUPDBEG
                  Demo.SuspendCyclicFunc
                                         'Termination of a cyclic request
                  lRet = SetIfMsgConf(MSG_PCLUPDBEG_Q) 'verification of the SYS-Message
                              'error handling
                  If lRet Then
                     szMsg = "SetIfMsgConf terminated with error code: " +
CStr(lRet)
                     Demo.SYS Messages.Caption = szMsg
```



End If Demo.SYS Messages.BackColor = QBColor(11) 'set BackgroundColor to bright blue Demo.SYS_Messages.Caption = "PLC Download BEGIN" Case MSG_PCLUPDEND lRet = SetIfMsgConf(MSG PCLUPDEND Q) 'verification of the SYS-Message Demo.ResumeCyclicFunc 'Start of a cyclic request Demo.SYS_Messages.BackColor = QBColor(6) 'set BackgroundColor to brown Demo.SYS Messages.Caption = "PLC Download END" If lRet Then 'error handling szMsg = "SetIfMsgConf terminated with error code: " + CStr(lRet) Demo.SYS_Messages.Caption = szMsg End If Sleep (2000) 'Wait 2 sec. Demo.SYS Messages.BackColor = QBColor(10) 'set BackgroundColor to bright green Demo.SYS_Messages.Caption = "Thread is still runnning" Case MSG_PARUPDBEG Demo.SuspendCyclicFunc 'Termination of a cyclic request lRet = SetIfMsgConf(MSG_PARUPDBEG_Q) 'verification of the SYS-Message If lRet Then 'error handling szMsg = "SetIfMsgConf terminated with error code: " + CStr(lRet) Demo.SYS_Messages.Caption = szMsg End If Demo.SYS Messages.BackColor = QBColor(11) 'set BackgroundColor to bright blue Demo.SYS_Messages.Caption = "Parameter Download BEGIN" Case MSG_PARUPDEND lRet = SetIfMsgConf(MSG_PARUPDEND_Q) 'verification of the SYS-Message 'Start of a cyclic request Demo.ResumeCyclicFunc Demo.SYS Messages.BackColor = QBColor(6) 'set BackgroundColor to brown Demo.SYS_Messages.Caption = "Parameter Download END" If lRet Then 'error handling szMsg = "SetIfMsgConf terminated with error code: " + CStr(lRet) Demo.SYS Messages.Caption = szMsg End If Sleep (2000) 'Wait 2 sec. Demo.SYS_Messages.BackColor = QBColor(10) Demo.SYS_Messages.Caption = "Thread is still runnning" End Select End If ElseIf lWait = 2 Then 'End of Threadloop Exit Do End If Loop End Sub

IDL

4.6 COM - Automation Interface

The function interface supports two simple COM automation interfaces: IFIObject, and IFIData.

IFIObject

Explanation	IFIObject represents the following functions of a function interface (FI) LogInIf, LogOutIf, and DataTransfer.		
	IFIObject::LogInIf		
L description	HRESULT LogInIf([in] BSTR bstrTaskName)		
Explanation	Compared tot he FI function, this method has been greatly simplified. For a more detailed description, refer to the "LogInIf" FI function.		
Parameters	bstrTaskName: Name of the user program		
	Example		
	Visual Basic Script		
	Dim oFIObject		
	On Error Resume Next		
	<pre>Set oFIObject = CreateObject("Indif000.FIObject")</pre>		
	<pre>if Not oFIObject is Nothing then oFIObject.LogInIf("fi.vbs")</pre>		
	end if		
	Set oFIObject = Nothing		
	Example		
	JavaScript		
	var oFIObject;		
	<pre>try{ oFIObject = new ActiveXObject("Indif000.FIObject");</pre>		
	<pre>if (oFIObject == null) {</pre>		
	{ return;		
	}		
	oFIObject.LogInIf("fi.js");		
	oFIObject = null;		
	}		



catch(e)

IFIObject::LogOutIf

IDL description	HRESULT LogOutIf();	
Explanation	Here, refer to the "LogOutIf" FI function description.	
	IFIObject::DataTransfer	
IDL description	HRESULT DataTransfer([in] BSTR bstrFunction, [in, defaultvalue(32000)] long IResSize, [in, defaultvalue(3)] long IResType, [in, defaultvalue("")] BSTR bstrValue, [in, defaultvalue(3)] long IValType, [out, retval] IFIData **ppoData);	
Explanation	Compared to the "DataTransfer" FI function, the IFIObject DataTransfer was made easier to operate in view of the parameters. The sequence of the parameters has been changed, and some parameters have been pre-assigned default values.	
Parameters	bstrFunction:Function interface commandIResSize:Length of the result buffer, default 32000 bytesIResType:Data code of result data, default 3 (ANSI)bstrValue:Writing value, default empty stringIValType:Data code of the value to be written, default 3(ANSI)ppoData:A DataTransfer request results in an IFIData object. In case of error, a zero object is returned.	
	Example	
	<pre>Visual Basic Script Dim oFIData Set oFIData = oFIObject.DataTransfer("00_CR_PVF_bool0") if Not oFIData is Nothing then end if</pre>	
	<pre>The following source code corresponds exactly to the above example Dim oFIData Set oFIData=oFIObject.DataTransfer("00_CR_PVF_bool0",32000,3,"", 3) if Not oFIData is Nothing then end if</pre>	

Example

JavaScript

```
var oFIData;
oFIData = oFIObject.DataTransfer("00_CR_PVF_bool0");
if (oFIData == null)
{
    return;
}
```



The following source code corresponds exactly to the above example
var oFIData;
oFIData = oFIObject.DataTransfer("00_CR_PVF_bool0", 32000,
3, "", 3);
if (oFIData == null)
{
 return;
}

IFIData

 Explanation
 IFIObject represents the following functions of a function interface (FI)

 GetNumberOfRows,
 GetNumberOfItems, and

 ReadGroupItem.
 ReadGroupItem.

IFIData::GetNumberOfRows

- **IDL description** HRESULT GetNumberOfRows([out, retval] long *plRows)
 - **Explanation** Here, refer to the "GetNumberOfRows" FI function description.
 - **Parameters** pRows: Number of rows of an FI result.

Example

Visual Basic Script

Dim IRow

IRow = oFIData.GetNumberOfRows()

Example

JavaScript

var IRow;

IRow = oFIData.GetNumberOfRows()

IFIData::GetNumberOfItems

- **IDL description** HRESULT GetNumberOfItems([in] long IRow, [out, retval] long *plItem)
 - **Explanation** Here, refer to the "GetNumberOltems" FI function description.

Parameters IRow: [1, 256] plltem: Number of partial results

Example

Visual Basic Script

Dim IColumn IColumn = oFIData.GetNumberOfItems(1)



Example

JavaScript

var IColumn;

IColumn = oFIData.GetNumberOfItems(1);

IFIData::ReadGroupItem

IDL description			roupItem([in] long IRow, [in] long IItem, 2000)] long lBufSize, [out, retval] BSTR *pbstrVal)
Explanation			e "ReadGroupItem" FI function description. The only efault value assignment of the IBufSize parameter.
Parameters	IRow:	-1:	Output of a complete table, i.e. all rows of a request
		[1256]:	the respective result line
	lltem:	-1: 0:	Output of a row Output of the requested command with management information
		[1n]:	Individual result (Element of a row)
	lBufSiz	e:	Length of the buffer for partial result, default 32000 bytes.
	pbstrV	al:	Requested partial result
	Examp	le	
	Visual	Basic Scri	pt

Dim strText strText = oFIData.ReadGroupItem(-1, -1)

Example

JavaScript

var strText;

strText = oFIData.ReadGroupItem(-1, -1);

Example for the Total Script

Visual Basic Script	Dim oFIObject
	Dim oFIData
	Dim lRow
	Dim lColumn
	Dim strText
	On Error Resume Next
	<pre>Set oFIObject = CreateObject("Indif000.FIObject")</pre>
	if Not oFIObject is Nothing then
	oFIObject.LogInIf("fi.vbs")
	if Err = 0 then
	Set oFIData =
	oFIObject.DataTransfer("00_CR_PVF_bool0")
	if Not oFIData is Nothing then



```
lRow = oFIData.GetNumberOfRows()
                       MsgBox "Rows: " & CStr(lRow)
                       lColumn = oFIData.GetNumberOfItems(1)
                       MsgBox "Columns: " & CStr(lColumn)
                       MsgBox oFIData.ReadGroupItem(-1, -1)
                   end if
                   call oFIObject.LogOutIf()
               end if
           end if
           Set oFIObject = Nothing
           var oFIObject;
JavaScript
           var oFIData;
           var lRow;
           var lColumn;
           var strText;
           function FI()
           {
             try{
               oFIObject = new ActiveXObject("Indif000.FIObject");
               if (oFIObject == null)
               {
                 return;
               }
               oFIObject.LogInIf("fi.js");
               oFIData = oFIObject.DataTransfer("00 CR PVF bool0");
               if (oFIData == null)
           {
             return;
           }
               lRow = oFIData.GetNumberOfRows();
               strText = "Rows: " + lRow.toString();
               WScript.Echo ( strText );
               lColumn = oFIData.GetNumberOfItems(1);
               strText = "Columns: " + lColumn.toString();
               WScript.Echo ( strText );
               WScript.Echo (oFIData.ReadGroupItem(-1, -1));
               oFIData = null;
               oFIObject.LogOutIf();
               oFIObject = null;
             }
             catch(e)
           }
           FI();
```



4.7 Tips and Tricks when Working with the Interface

This chapter provides you with tips and tricks that are designed to help you to proceed faster when developing your user program (client) (see also Chapter "Installing Windows NT and the Function Interface").

Furthermore, we have discovered that using Windows NT without a mouse always tends to present difficulties and we have therefore listed the most important Windows NT key combinations in a table.

Problem	Remedy
In your application, you issue an FI command and receive:	Frequent cause:Device address has not been given or been incorrectly given!
 no response - or - an unexpected response - or - an error code (see Chapter "Error Codes") 	Check the correct details of the FI command (see Chapter "Installing Windows NT and the Function Interface" and Chapter "Function Interface Commands").
	Issue the FI command that is causing problems using the VBDemo program (see Issuing FI Commands with the VBDemo Application).
Your client no longer reacts	See clearing the Memory using the "KILLTASK.EXE" Tool
Your client terminates "DR. WATSON" with a	See clearing the Memory using the "KILLTASK.EXE" Tool
Windows access violation.	Correct the programming error and re-start your application.
The entire system (Windows NT, client and Rexroth Indramat GUI) is reacting slowly.	Check the Windows NT settings for improved performance, idling activity, swapping the core-mode driver and idling activity in accordance with Chapter "Installing Windows NT and the Function Interface".
Clearing the memory using the "KILLTASK.EXE" tool doesn't work, i.e. neither the three base processes of the function interface nor the client is	Start Task manager, for example, using the key combination <ctrl>+<shift>+<esc> (see chapter "Windows NT Task Manager").</esc></shift></ctrl>
removed from memory.	Click on the "Processes" tab page.
	Terminate the three basic processes of the function interface and your user program, if applicable: LOGINTFC.EXE (logic process) COMINTFC.EXE (communication process) BOFINTFC.EXE (management process) via the <terminate process=""> button.</terminate>
Your application terminates because: - required files are missing - or -	Check to make sure the required files are located in their respective directories.
- path entries do not exist or are incorrect.	Check the path entries.
	Note! Absolute paths should be avoided in the application as any later changes in the drive path (e.g. from C:\ to D:\) or in the directory structure are not supported. The system directory as well as the Windows NT disk drive can also be freely selected.

Clearing the Memory using the "KILLTASK.EXE" Tool

This tool can be used when creating software for clearing the memory. After a standard installation (see chapter "Installing Windows NT and the Function Interface") it is located in the default directory "C:\Programme\Indramat\MTGUI\Bin\".

The tool provides you with the following two options for clearing the memory:

- complete reinitialization and
- selective reinitialization of the function interface.

complete reinitialization On starting the "KILLTASK.EXE" application, the following three basic processes of the function interface:

- logic process (LOGINTFC.EXE),
- communication process (COMINTFC.EXE),
- management process (BOFINTFC.EXE)

are removed from the memory, as well as all applications connected to the function interface.



You should first terminate all other (stable) function interface applications correctly.

Only run Killtask after doing this!

If this stipulation is not observed then all claims against Rexroth Indramat are excluded.

To completely reinitialize, proceed as follows:

- \Rightarrow Click on Start and then on the "Run" option.
- \Rightarrow Click on the <Find> button to search for the "KILLTASK.EXE" tool.
- Note: After a standard installation (see Chapter "Installing Windows NT and the Function Interface") the "KILLTASK.EXE" application is located in the default directory "C:\Programme\Indramat\MTGUI\Bin".
- \Rightarrow Click on the <OK> button.

All applications connected to the function interface – and the basic processes of the function interface itself – are removed from the memory.

Run	? 🗙
2	Type the name of a program, folder, or document, and Windows will open it for you.
<u>O</u> pen:	C:\MT-CNC\ind_drv\Killtask.exe
	Run in Separate Memory Space
	OK Cancel <u>B</u> rowse
	Killtask00.b

Fig. 4-6: "Run" dialog box of Windows NT: Complete re-initialization



Selective reinitialization Only those applications that are connected to the function interface are removed from the memory and from the function interface management structure.

To selectively reinitialize, proceed as follows:

- \Rightarrow Click on Start and then on the "Run" option.
- Note: You can search for the "KILLTASK.EXE" application by clicking on the "Find..." button. After a standard installation (see chapter "Installing Windows NT and the Function Interface") this file is located in the default directory "C:\Programme\Indramat\MTGUI\Bin\".
- \Rightarrow In the text box, enter the name of the application that is to be removed from the memory and from the management structure of the function interface (here VBDemo.exe).
- \Rightarrow Then click on the <OK> button.

The client (here "VBDemo.exe") is removed from the memory and from the management structure of the function interface.

Run	? ×
<u>;</u>	Type the name of a program, folder, or document, and Windows will open it for you.
<u>O</u> pen:	C:\MT-CNC\ind_drv\Killtask.exe VBDemo.exe
	Run in Separate Memory Space
	OK Cancel <u>B</u> rowse
	Killtask01.bmp

Fig. 4-7: "Run" dialog box of Windows NT: Selective re-initialization



Issuing FI Commands using the "VBDemo" Application

Single FI commands and cyclical requests can be issued by the "VBDemo" application.

To start the application, proceed as follows:

- $\Rightarrow\,$ Click on start, point to Programs, then to Rexroth Indramat and finally to FI.
- \Rightarrow Click on VBDemo.

VBDemo		
- Single Requests Data <u>T</u> ransfer	FI-Command 00_CR_AGF_0	Value to write to device
Response to Sir	igle Request	Groups Rows Items
Single Request		
⊢ Cyclic Requests St <u>a</u> rtCyclic St <u>o</u> pCyclic	S <u>u</u> spendCyclic <u>R</u> esumeCyclic	Cycle Time 500 [ms] FI-Command Counter 00_BC_ASM1/3 Groups Rows Items
Response to Cy		
VBDemo Connect		SYS-Message Thread is runnning
		VBDEMO.bmp

Fig. 4-8: The "VBDemo" user program

"Single Requests" Dialog Box	This dialog box allows single requests to be issued that both read and write using the "Data Transfer" routine.
	To do this, enter the FI command in the "FI-Command" entry field. If a write request is made, then also enter the value that you wish to write in the "Value to write to device" box (see Chapter "Function Interface Commands").
	Then issue the FI command you have entered to the function interface by clicking on the <data<u>Transfer> button.</data<u>
	The response from the function interface is displayed in the "Response to Single Request" text box.
"Cyclic Requests" Dialog Box	This dialog box allows cyclic requests that write to be issued using the "StartCyclicPipe" routine.
	To do this, enter the FI command in the entry field "FI-Command" (see chapter Function Interface Commands).
	Then issue the FI command entered cyclically to the function interface by clicking in the $$ button.
	The response from the function interface is displayed in the "Response to Cyclic Request" text box.
	Note: You can change the request time from between 10 to 100 ms using the "Cyclic Time" thumb switch.

To stop the cyclic request, click on the <StopCyclic> button. This will cause the "StopCyclicPipe" routine to be executed.



...

	Note:	Clicking on the <suspendcyclic> button processes the "suspendCyclicPipe" routine and sets the cyclic request to standby mode. To reactivate the cyclic request, click on the <<u>R</u>esumeCyclic> button. This will cause the "ResumeCyclicPipe" routine to be executed.</suspendcyclic>
"VBDemo Connection Status" Dialog Box	Displays are two s	the login status of the application at the function interface. There statuses:
		dialog box is shaded green and shows the function interface anel (LOG channel) that has been assigned to the application.
		dialog box is shaded in red and shows the error code that has a caused by logging in via the "LogInIf" login routine.
Starting	To start	the "VBDemo" program in diagnostics mode, proceed as follows:
VBDemo" in Diagnostics Mode		n the Windows NT Explorer. To do this, click on Start, point to rams and then click on Windows NT Explorer.
		Winnt, go to Profiles and into the User Profile by which the tion interface was installed.
		on the Start Menu, point to Programs, then to Rexroth Indramat finally click on "FI".
		on VBDemo and open the Properties dialog box via the Explorer u file.
		t on the tab page link and enter the start parameter "/c=t /b=w" in Target" text field.
		on the < <u>C</u> lose> button and VBDemo will be started in nostics mode the next time it is called.

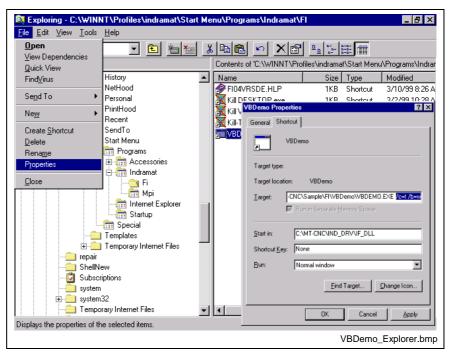


Fig. 4-9: Starting VBDemo in the diagnostics mode of the function interface



Outputting Diagnostic Messages

By passing on the start parameters when starting the management process "BOFINTFC.EXE", various function interface diagnostic messages can be outputted to the screen.

To start the function interface in diagnostics mode, proceed as follows:

- \Rightarrow Click on Start and then on the "Run" option.
- Note: You can search for the management process "BOFINTFC.EXE" by clicking on the "Find" button. After a standard installation (see chapter "Installing Windows NT and the Function Interface Commands") this file is located in the default directory "C:\Programme\Indramat\MTGUI\Bin\".
- \Rightarrow Enter the start parameter "/c=t /b=w" in lower case letters in the text box (observe spaces between entries).
- \Rightarrow Then click on the <OK> button. The function interface is now started in diagnostics mode.

Run	? ×
2	Type the name of a program, folder, or document, and Windows will open it for you.
<u>O</u> pen:	C:\MT-CNC\ind_drv\Bofintfc.exe /c=t /b=w ▼ ■ Run in Separate Memory Space
	OK Cancel <u>B</u> rowse
	AusfuehrenDiagnose.br

Fig. 4-10: "Run" dialog box of Windows NT: BOFINTFC.EXE\$

COMINTFC /c=t +G10 03	LOGINTFC /c=t +G10 03.61 Dec 14 1998	×
Channels: 1 2	INDIF200.DLL 03.60 Dec 14 1998 G=10	INDIF200.DLL 03.60 Dec 14 1998 G=10
PC:0 SI:0 XZ:0 WZ:0 TZ:0 TIMER idle	L5: LogChannel opened	
CNCTASK1 init.ready		
	LOGTASK1 UsrAdmEv: 1 S>0:0_5610 ready	
H BOFINT 05.13 27.11.98		
L1: BOFINTFC.EXE + 0098 L2: VBDEMO.EXE + 0095		
L3:		
L4:		
L5: —		
L6: —		
L7: —		
L8: —		
S1: 0000 QL:00000000 0:0		
S2: 0000 QL:00000000 0:0 S3: 0000 QL:00000000 0:0		
S4: 0000 QL:00000000 0:0		
S5: 0000 QL:0000000 0:0		
SE: 0000:0000 ME: 0000:0000		
MK: 0000 MI: 0000 SI: 0000		
		Diagnose.bmp

Fig. 4-11: Diagnose mode of the function interface



- **Meaning of the Counters** Five counters are shown in the 3rd line of the diagnostics window of the communication process (COMINTFC.EXE). The individual counters mean the following:
 - **PC** Number of communication errors that have occurred in the direction of transmission from device \rightarrow PC.
 - SI Number of communication errors that have occurred in the direction of transmission from PC \rightarrow device.
 - **XZ** Number of communication repetitions that were required to transfer a valid telegram to the device.
 - WZ The counter is increased if, in spite of five repetitions, it has not been possible to transmit a valid telegram to the device. The counter is increased by one if the "XZ" counter has been previously increased by five. In this case, the timeout counter is also increased by one.
 - **TZ** Timeout counter. The number of timeouts that occur are counted in this counter. A timeout is generated if, in spite of five repetitions, it has not been possible to transmit a valid telegram to the device.

The active control channels are displayed in the lower lines (CNC/DMA-Task).

Data accesses made by the individually connected applications are displayed in the diagnostics window of the LOG channel on the left side of the screen.

In the control window of the management process (BOFINT), the applications are shown that are known in the management mechanism of the BOF process.

Windows NT Key Combinations

The most important key combinations for using Windows NT without a mouse are displayed in the following table.

Action	Key combination
Open start bar	<ctrl>+<esc></esc></ctrl>
Navigate within the opened start bar and in the opened submenus in the start bar	<arrow key="" left,="" right="">, or <arrow down="" key="" up,=""></arrow></arrow>
Select (start) the applications in the opened submenus in the start bar	<enter></enter>
Start Windows NT Task Manager	<ctrl>+<shift>+<esc></esc></shift></ctrl>
Move within the Windows NT menu	<tab></tab>
"Right mouse click" on button moved to	<shift>+<f10></f10></shift>
Switch within a menu to other tab pages	<ctrl>+<tab></tab></ctrl>
Switch between opened applications	<alt>+<tab></tab></alt>



5 Installing Windows NT and the Function Interface

5.1 The Windows NT Operating System

Using the Windows NT operating system and the possibility of running various applications parallel to one another requires a powerful computer.

The hardware requirements depend directly on the number and nature of the applications running concurrently on the PC. This should be taken into account during the project planning phase. The network cards used and their drivers require a great deal of computing power which might then not be available for the rest of the system. Hardware must therefore be selected with great care and utmost precision.

Multitasking and Windows NT

Whereas under Windows 3.1x what is known as "cooperative" or "nonpre-emptive" multitasking controlled several applications running concurrently, genuine "pre-emptive" multitasking is integrated into Windows NT.

- **Non-pre-emptive Multitasking** Here, it is not the operating system that decides how much computing time is to be allocated, but the application itself; and the application decides when to surrender time back for a short while to the operating system. The disadvantage of this is that when several applications are running simultaneously, working with them in parallel is only possible to a limited degree.
 - **Pre-emptive Multitasking** The operating system itself decides how much computing time is to be allocated to the individual applications. Switching between individual applications is now much more fluid a process as the operating system is able to distribute computing time faster and at shorter intervals, creating the impression that several instructions really can work "simultaneously and unrestrictedly".



Note: For the Windows NT Operating System, we recommend a PC with a Pentium processor and at least 32 MB RAM, as well as at least 500 MB available space on the hard drive.

Windows NT Task Manager

The applications running can be monitored and controlled by the Task Manager i.e., applications that have been started can be overlaid on the desktop or can be terminated. Furthermore, it is possible to start applications or switch to other applications.

Calling the Task Manager

 \Rightarrow using the key combination <Ctrl>+<Shift>+<Esc>

 \Rightarrow clicking with the right mouse button on the taskbar

olications Processes	Performa	ance		
Image Name	PID	CPU	CPU Time	Mem Usage
System Idle Process	0	98	1:59:57	16 K
System	2	00	0:00:39	200 K
smss.exe	20	00	0:00:00	200 K
csrss.exe	24	00	0:02:35	912 K
WINLOGON.EXE	34	00	0:00:01	132 K
SERVICES.EXE	40	00	0:00:03	980 K
LSASS.EXE	43	00	0:00:00	376 K
EXPLORER.EXE	62	01	0:00:04	2052 K
LEXBCES.EXE	67	00	0:00:00	20 K
RPCSS.EXE	72	00	0:00:02	740 K
NDDEAGNT.EXE	77	00	0:00:00	88 K
SPOOLSS.EXE	86	00	0:00:00	204 K
internat.exe	103	00	0:00:00	148 K
FINDFAST.EXE	116	00	0:01:08	1692 K
TASKMGR.EXE	145	01	0:00:00	1460 K
LOGINTFC.exe	168	00	0:00:00	2880 K
Mtvnc40v.exe	213	00	0:00:00	3992 K
Bofintfc.exe	297	00	0:00:03	6280 K
Comintfc.exe	302	00	0:00:00	2276 K
				End Process

Fig. 5-1: Windows NT Task Manager

Note: You can bring applications consecutively up to the front of the screen (overlay them) using the key combination <Alt>+<Tab> without having to make your selection using Task Manager.



5.2 Setting the Windows NT System Properties

Performance Features

To guarantee an optimal reaction time for the function interface, the performance boost for the application in the foreground should be set to "none".

Note: Safe and error-free operation of the function interface is only ensured when the performance boost for the application in the foreground is set to "none".

To make this setting, proceed as follows:

- $\Rightarrow\,$ Click on start, point to Settings, then to System Control and finally to System.
- \Rightarrow Click on the "System Properties" tab page and set the thumb to "none".
- \Rightarrow Then click on the <OK> button.

em Properties Startup/Shutd		Hardware Pro	പ്പം	User Profiles
General	Pe	rformance		invironment
Application Per	formance			
	formance boost	for the foregr	ound applicati	on.
Boost:	None	- 		Maximum
Virtual Memory Total paging fil	le size for all dis	k volumes:	75 MB	<u>C</u> hange
		OK	Cancel	Apply

Fig. 5-2: Setting the WindowsNT system property "Performance

Note: The setting for "Virtual Memory" may differ from the setting of your system.





Date/Time properties

It is required for the exchange of Rexroth Indramat files between two PCs, to have an identical time zone setting. Furthermore, the automatic clock adjustment for daylight saving (switching between summer and winter times) must be deactivated.

To make this setting, proceed as follows:

- \Rightarrow Click on Start, point to Settings, then to System Control and finally to Date/Time Properties.
- \Rightarrow Click on the Time Zone tab page and deactivate the "Automatically adjust clock for <u>d</u>aylight saving changes" toggle button.
- \Rightarrow Then click on the <OK> button.

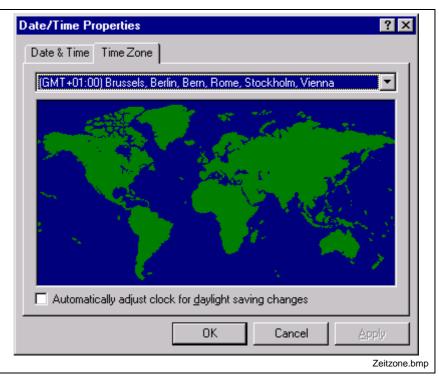


Fig. 5-3: Date and time settings

5.3 Installing the Function Interface

Note: In Version 07, the function interface cannot be installed separately, but only within the context of the relevant Rexroth Indramat GUI.



5.4 Directory and File Structure of the Function Interface

Contents of the "INDRAMAT.INI" File

The global settings for the function interface are stored in the "Indramat.ini" file. The function interface looks for the file in the "C:\Programme\Indramat\MTGUI\BasicData\Resource" directory.

However, the default directories as well as the drive [LW] can be freely selected. The "INDRAMAT.INI" file corresponds to the Microsoft Windows INI standard and is constructed as follows:

Identifier	Values	Explanation		
[lfConfig]	This contains the configuration settings for the function interface			
lfInstDir=	z.B.: C:\Programme\ Indramat\MTGUI\Bin	Directory in which the three basic processes of the function interface are installed. This entry is set by the installation program.		
AndInstDir=	z. B.: C:\MTA200	!Optional! Directory for MTA200 control software. Details refer to the "MTA200.EXE" application.		
IfDIIMode=	z.B.: 04.10 03.xx [0070], 04.xx [00,10]	Here the mode is specified that is to be supported by the function interface. The IfDIIMode of a more recent version of the function interface can, for example, be operated in the same mode as the previous version for troubleshooting.		
IfVersion=	z.B.: 06V00	Current version of the function interface.		
GBOVERSION=	z.B.: 005-21Vxx	Current version of the Rexroth Indramat GUI.		
INDRAMAT_x=	x=1 to 9 Name of file	Reference to directory C:\Programme\Indramat\MTGUI\BasicData\Resource The existence of the files named here is checked on starting the function interface. The following applies: File identifier without an extension is a DLL. e.g. Indramat = INDRAMT.DLL Several file identifiers are separated by a "comma".		
IND_DLL_x=	x=1 to 9 Name of file	Reference to directory [LW]:\\MTGUI\Bin. The existence of the files named here is checked on starting the function interface. The following applies: A file identifier without an extension is a DLL, e.g. NDFS100 = INDFS100.DLL. Several file identifiers are separated by a "comma".		
IF_DLL_x=	x=1 to 9 Name of file	Reference to directory [LW]:\\MTGUI\Bin. The existence of the files named here is checked on starting the function interface. The following applies: File identifier without an extension is a DLL. e.g. INDIF000 = INDIF000.DLL Several file identifiers are separated by a "comma".		
[Install]	This contains entries	regarding the installed System200 software components.		
HMIVersion=	z.B.: 01V06	Version ID of the System200 software component WIN-HMI		
TYP=	z.B.: HMI	System200 software component WIN-HMI		
ServicePack=	z.B.: 2 [1,2,]	Service Pack ID of the installed System200 software components		
SP_Release=	[1, to F]	State of the Service Pack release ID (F = Final Version)		



Example Entries in the "INDRAMAT.INI" File

[IfConfig] IfInstDir=C:\Program Files\Indramat\MTGUI\Bin AndInstDir=C:\MTA200 IfDIIMode=04.10 IFVERSION=04V02 GBOVERSION=005-21V09 INDRAMAT_1=indramat IND_DLL_1=indfs100,indma110,indma900,indut140,indof160 IF_DLL_1=indif000,indif120,indif130,Indif150 IF_DLL_2=indif200,indif210,indif220,indif300,indif310,indif320 IF_DLL_3=indif330,indif340,indif350,indif360,indif400 IF_DLL_4=indif500,indif510,indif520,indif530,indif540,indif550 IF_DLL_5=indif600,indif610,indif700,indif810,indif820,indif840 IF_DLL_6=indifA00 [Install] HMIVersion=01V06

[Install] HMIVersion=01V06 TYP=HMI SP_Release=F

The DLL entries (If_DLL_1,...) can be expanded up to the ninth entry (If_DLL_9). A check for the existence of the DLLs is only made when the DLLs have been previously entered at the corresponding parameters. If the file name is given without an extension then the extension is automatically assumed to be "DLL". If the existence of another file is to be checked then the file extension of this file must also be entered, e.g., "Userprogram.dat".

Contents of the "IND_DEV.INI" File

The configuration of the individual communication addresses and the settings of the various Rexroth Indramat devices are determined in this file. The "IND_DEV.INI" file is edited by the system configurator and is located in the "[LW]\...\CONFIG\" directory.

Rexroth Indramat system configurator The Rexroth Indramat System Configurator is an editor that sets and lists the devices connected to the control PC. The device addresses, the device type and the description of the communication path to the device are used for this. The goal is to create a 1:1 copy of the device structure connected to the control PC; this structure is termed the system configuration.

The system configuration is stored on the control PC. Furthermore, the devices can also be assigned basic properties, e.g. a Type MTVNC device (virtual MTC for the function "Offline Simulation") can be assigned to a device of Type MTC200-P or MTCNC, etc., in order to form a simulation pair. Here, the parameter records of the real device can be used by the virtual device allowing a simulation of NC programs to be started.

Note: An online help is also included in the system configurator. It can be called up by pressing the <F1> function key while the program is running.



System Configuration		
<u>File Edit Options H</u> elp		
Defined Devices	Devid	ce Data
B- 🔜 System Configuration	Device Address:	
00 VDF-315 DR-4 0209-1	Device Type:	
мтс200-Р	Comm. Address :	
01 01 MTCNC Olaf_COM	Comm. Type:	
MTCNC		
		E <u>x</u> it
		Systemkonfigurator.bmp

Fig. 5-4: Rexroth Indramat system configurator

The "IND_DEV.INI" file corresponds to the Microsoft Windows INI standard and is structured as follows:

Identifier	Values	Explanation	
[CommAddrX]	X = 18	Assignment of the communication channel (thread) of the function interface.	
CommStr=	V24, Port [COM14], baud rate, parity, type of interface, packet counter	Communication via RS232 serial interface, e.g. V24,COM1,19200,NONE,RS232,TCON. Communication via RS485 serial interface, e.g. V24,COM2,19200,NONE,R485H,TCON	
	- or - DMA, address, offset, length	For communication via a dual port RAM, a DMA channel is also required for the MTC200-P, e.g., DMA,\$D000,\$0000,\$2000.	
	- or - SHM, Channel No. [115]	Communication channel to the MTVNC via a shared memory, e.g. SHM, 1.	
Timeout=	>= 1000 [msec] Preset = 3500 [msec]	! OPTIONAL ! Time in which a response must be received from the device.	
	- or – only for dual	port RAM (e.g. for MTC200-P)	
CommStr=	DPR, address [\$C000,\$0000 \$FE00,\$0000], Length, RAM0, Packet-Counter	Communication via dual port-RAM, e.g., MTC200-P DPR, \$D000,\$0000,\$2000,RAM0,TCOFF.	
PortAddr=	\$200, \$204, \$31C e.g., \$31C	Address of the MTC/MTS card according to the settings on the respective card.	
PortVal=	\$20, \$21, \$3F e.g., \$28 at address [\$D000,\$0000]	Configuration byte for setting the physical memory address of the MTC/MTS card.	
Timeout=	>= 1000 [msec] Preset = 3500 [msec]	! OPTIONAL ! Time in which a response must be received from the device.	
[DeviceAddrX]	X = 00.15	device address	
Component type1=	e.g., MTS-P01.2 NONE, MTS-P, MTS-P01.2, MTS-P02.2, MTS-R-M1, PPC-R	Name of the PLC component type	
Component type2=	e.g., MTC-P-G2 NONE, MTC-P, MTC-R, PPC-R	Name of the NC component type	
DeviceName=	Max. 32 ASCII characters	Device name; e.g., Processing Center 12T34	



Identifier	Values	Explanation
DeviceType=	e.g. MTC200-P-G2	Device type: MTC with PLC PC variant
	MTVNC	Virtual MTC
	MTC200-P-G2 MTC200-R-G2	MTC with PLC PC variant MTC with PLC RECO variant
	ISP200-P-G2 ISP200-R-G2	Standalone PLC PC variant Standalone PLC RECO variant
	TRA200-R	TRANS200 RECO variant
	ECODRIVE03	Ecodrive03
	MTA200-P	MTA200 control
	SERCANS-A SERCANS-P SYNAX-P SYNAX-R	SERCANS-A card (via serial interface) SERCANS-P card (via serial interface) SYNAX PC variant SYNAX RECO variant
DeviceAssign=	015, NO	Assignment of a simulation pair. The MTVNC is, for example, hereby assigned to a real MTCNC.
DeviceStatus=	ON, OFF	Assignment of whether or not the device is incorporated into the management structure of the function interface.
MtvncMode=	OFF, RUN, STANDBY	! Only for virtual MTC (MTVNC) ! Status of the MTVNC with inactive utilization
MtvncMemory=	256, 257 16383 [KB] Preset = 512 [KB]	! Only for virtual MTC (MTVNC) ! Size of the PC memory used by the MTVNC.
CommAddr=	18	Assignment of the communication address. Corresponds to the [CommAddr18] parameter.
PLC=	YES, NO	PLC support for the device. E.g. one MTVNC, TRANS200-R has no PLC, therefore the parameter PLC=NO is set.
[DeviceOrder]	This contains the configuration settings for the system configurator.	
Order=	0,1,2,15	Order in which the configured devices are displayed.
[NetManager]	This contains the configuration settings for the network driver "NETINTFC.EXE"	
NetManagerMode=	OFF, RUN	Starts the network device driver.

Example Entries in the "IND_DEV.INI" file

Entry	Explanation
[CommAddr1]	Communication address 1
CommStr=DPR,\$D000,\$0000,\$2000,RAM0,TCON	Settings for communication via dual port RAM
PortAddr=\$31C	Port address of the MTC/MTS card
PortVal=\$28	Physical memory address of the MTC/MTS card.
[CommAddr2]	Communication address 2
CommStr=DMA,\$D000,\$0000,\$2000	Assignment of the DMA channel.
[CommAddr3]	Communication address 3
CommStr=V24,COM1,19200,NONE,RS232,TCON	Settings for communication via RS232.
[CommAddr4]	Communication address 4
CommStr=DPR,\$D200,\$0000,\$2000,RAM0,TCON	Settings for communication via dual port RAM
PortAddr=\$318	Port address of the MTC/MTS card
PortVal=\$29	Physical memory address of the MTC/MTS card.
[CommAddr5]	Communication address 5
CommStr=SHM,1	Settings for communication via shared memory.



Entry	Explanation
[DeviceAddr0]	Device address 00
CommAddr=1	Assigned communication channel
Componenttype1= MTS-P-G2	PLC components MTS-P-G2
Componenttype2= MTC-P-G2	CNC components MTC-P-G2
DeviceAssign=NO	No MTVNC assigned
DeviceName=VDF-315 DR-4 0209-15	Device name
DeviceStatus=ON	Device is available and ready for operation
DeviceType=MTC200-P	Device type
PLC=YES	PLC support
[DeviceAddr1]	Device address 01
CommAddr=4	Assigned communication channel
Componenttype1= MTS-P02.02	PLC components MTS-P02.02
Componenttype2= MTC-P-G2	CNC components MTC-P-G2
DeviceAssign=NO	No MTVNC assigned
DeviceName= Processing center 12T35	Device name
DeviceStatus=ON	Device is available and ready for operation
DeviceType=MTC200-P	Device type
PLC=YES	PLC support
[DeviceAddr2]	Device address 02
CommAddr=5	Assigned communication channel
Componenttype1= NONE	PLC component not available
Componenttype2= NONE	CNC component not available
DeviceAssign=1	Assigned to device address 01 (simulation pair)
DeviceName= V-Processing center 12T34	Device name
DeviceStatus=ON	Device is available and ready for operation
DeviceType=MTVNC	Device type
MtvncMemory=512	Size of the PC memory
MtvncMode=RUN	Status during inactive use
PLC=NO	No PLC support
[DeviceAddr3]	Device address 03
CommAddr=3	Assigned communication channel
Componenttype1= NONE	PLC component not available
Componenttype2= PPC-R	CNC component PPC-R
DeviceAssign=NO	No MTVNC assigned
DeviceName= TRANS200	Device name
DeviceStatus=ON	Device is ready for operation
DeviceType=TRANS200-R	Device type
PLC=NO	No PLC support
[DeviceOrder] Order=3,0,1,2	Order in which the configured devices are displayed in the system configurator
[NetManager] NetManagerMode=RUN	Network driver is started
[BofManager] PollDeviceStatus=OFF PollDeviceStatusRate=4000 PollDeviceStatusCheckFactor=4	On switching on (ON), the device status of the devices is requested cyclically. The cycle time of a device request is controlled by this value. If a device can not be addressed then a request is no longer made until a time has passed that is the product of PollDeviceStatusRate multiplied by the PollDeviceStatusCheckFactor.

Contents of the "[LW]:\Winnt\System32\" System Directory

The following Microsoft class libraries are stored in the system directory of Windows NT "[LW]:\Winnt\System32\":

File	Explanation	
MFC30.DLL	Microsoft class libraries	
MSVCRT20.DLL	Microsoft class libraries	
MFC40.DLL	Microsoft class libraries	
MFC42.DLL	Microsoft class libraries	
MSVCRT40.DLL	Microsoft class libraries	
MSVCRT.DLL	Microsoft class libraries	
MSVCP50.DLL	Microsoft class libraries	
MSVBVM50.DLL	Microsoft class libraries	
COMCTL32.OCX	Dialog elements for Visual Basic applications	
COMDLG32.OCX	Dialog elements for Visual Basic applications	
REGSVR32.EXE	Application for registering the OCX files	

Contents of the "[LW]:\Winnt\System32\Drivers\" Driver Directory

The following files of the core-mode driver are stored in the driver directory of Windows NT "[LW]:\Winnt\System32\Drivers\":

File	Explanation
MTCNC00I.SYS	Windows NT core-mode driver
MTCNC00I.INI	Configuration file of the core-mode driver
REGINI.EXE	Application for registering the core-mode driver

Contents of the "[LW]:\...\MTGUI\BasicData\Help\" Directory

The drive as well as the path "[LW]:\...\" are pre-set during the standard installation routine to "C:\Programme\Indramat\MTGUI\". The following Windows 95/NT help files for the printed English and German manuals are stored in the "C:\...\MTGUI\BasicData\Help\[Language]\" directory:

File	Explanation	
FIVRS_DE.HLP	Windows 95/NT help file in German	
FIVRS_DE.CNT	Definition file of the Windows 95/NT help file	
FIVRS_EN.HLP	Windows 95/NT help file in English (in preparation)	
FIVRS_EN.CNT	Definition file of the Windows 95/NT help file	



Contents of the "[LW]:\...\MTGUI\BasicData\Resource" Directory

The drive as well as the path "[LW]:\...\" are pre-set during the standard installation routine to "C:\Programme\Indramat\MTGUI\Bin". The following files are contained in the "C:\...\MTGUI\BasicData\Resource" directory:

File	Explanation
BOFINTFC.DAT	BOF process definition file
INDRAMAT.INI	File with global function interface settings
LOGINTFC.DAT	Definition file of the logic process
MECX.DAT	Definition file for the MECX device group
MISX.DAT	Definition file for the MWSX device group
MPCX.DAT	Definition file for the MPCX device group
MSCX.DAT	Definition file for the MSCX device group
MSYX.DAT	Definition file for the MSYX device group
MTAX.DAT	Definition file for the MWAX device group
MTCX.DAT	Definition file for the MWCX device group
VERSION.DAT	Definition file for the version ID

Example Entries in the "VERSION.DAT" File

The version ID of the individual parts of the program as well as the version of the function interface are entered in the "VERSION.DAT" file. This applies to all program parts (EXE, DLL) of the function interface. The following example shows the entries in this file:

Name Duild Version Data Start Decemeter				
Name	Build	Version	Date	Start Parameter
IFVERSION	113	04V00	Feb 22	
INDRAMAT.DLL	113	04.01	Feb 15	
INDFS100.DLL	113	03.14	Feb 22	
INDIF300.DLL	113	03.63	Feb 16	
INDUT140.DLL	113	03.09	Feb 22	
INDIF310.DLL	113	03.32	Feb 16	
INDOF160.DLL	113	03.15	Feb 22	
INDIF200.DLL	113	03.71	Feb 22	
BOFINTFC.EXE	113	05.16	Feb 16	/b=w/c=t
INDIF360.DLL	113	03.07	Feb 03	
LOGINTFC.EXE	113	04.00	Feb 22	/c=t +G10
COMINTFC.EXE	113	04.00	Feb 22	/c=t +G10
INDIF210.DLL	113	04.00	Feb 22	
INDIF330.DLL	113	03.30	Feb 16	
INDIF540	113	03.01	Feb 22	
INDIF130.DLL	113	03.16	Feb 22	
INDIF810.DLL	113	04.00	Feb 03	
INDIF350.DLL	113	03.35	Feb 03	
INDIF320.DLL	113	03.28	Feb 03	
INDIF340.DLL	113	03.31	Feb 03	



Contents of the "[LW]:\...\MTGUI\Bin" Directory

The drive as well as the path "[LW]:\...\" are pre-set during the standard installation routine to "C:\Programme\Indramat\MTGUI\". The following function libraries of the function interface are contained in the C:\...\MTGUI\Bin directory:

File	Explanation
BOFINTFC.EXE	BOF process
COMINTFC.EXE	Communication process
INDFS100.DLL	Processing the file ID
INDIF000.DLL	General functions for the user.
INDIF120.DLL	Functions for outputting the trace file.
INDIF130.DLL	Functions for the BOF process.
INDIF150.DLL	Functions for the logic and communication process.
INDIF200.DLL	Functions for the logic and communication process.
INDIF210.DLL	Functions for the logic and communication process.
INDIF220.DLL	Functions for the logic and communication process.
INDIF300.DLL	Functions for the BOF process.
INDIF310.DLL	Functions for the BOF process.
INDIF320.DLL	Functions for the DOS - Windows NT connections.
INDIF330.DLL	Functions for the BOF process.
INDIF340.DLL	Functions for the BOF process.
INDIF350.DLL	Functions for the DOS - Windows NT connections.
INDIF360.DLL	Functions for file access.
INDIF400.DLL	BOF process access to parameters.
INDIF500.DLL	Functions for access optimization.
INDIF510.DLL	Functions for access optimization.
INDIF520.DLL	Functions for access optimization.
INDIF530.DLL	Functions for access optimization.
INDIF540.DLL	Functions for access optimization.
INDIF550.DLL	Functions for access optimization.
INDIF560.DLL	Support for network connections and PLC and NC optimizer
INDIF570.DLL	Server for fast M-K communication
INDIF600.DLL	Functions for access to NC programs.
INDIF610.DLL	Functions for access to NC programs.
INDIF700.DLL	Functions for access to NC compiler.
INDIF800.DLL	Higher functions of all device groups.
INDIF810.DLL	Functions for access to the MWCX device groups.
INDIF820.DLL	Functions for access to the MSCX device groups.
INDIF830.DLL	Functions for access to the MECX device groups.
INDIF840.DLL	Functions for access to the MWAX device groups.
INDIF860.DLL	Functions for access to the MSYX device groups.
INDIF870.DLL	Functions for access to the MWSX device groups.



File	Explanation
INDIFA00.DLL	Functions for HMI support of the MWCX device groups.
INDIFZ00.DLL	Functions for access to the MWAX device groups.
INDMA900.DLL	Processing the MAP file
INDMA110.DLL	Connecting the MAP file
INDOF160.DLL	Using various system utilities
INDRAMAT.DLL	Access to global settings (GetInstPath, etc.)
INDUT140.DLL	Using various system utilities
KILLTASK.EXE	Application for terminating function interface applications (see Chapter "Programming")
LOGINTFC.EXE	Logic process
NETINTFC.EXE	Application for connection of client/server
VBDemo.exe	Test program in Visual Basic
IFDemo.exe	Test program in Visual C++





6 Construction and Availability of the FI Command

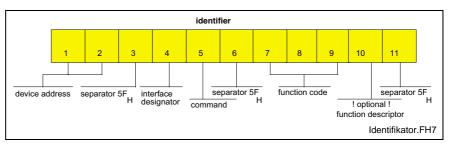
6.1 Elements of the FI Command

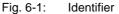
The function interface commands are subdivided into the following elements:

- 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.





device address The device address corresponds to the system address within the Rexroth Indramat GUI. This means, for example, that device 00 corresponds to system 0. Please observe, however, that the Rexroth Indramat 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
[0063]	MWCX	MTC200-P-G2, MTC200-R-G2, MTVNC
[00]	MSCX	SERCANS-A, SERCANS-P
[0063]	MWMX	VMISP200-P-G2, VMISP200-R-G2
[0063]	MWSX	ISP200-P-G2, ISP200-R-G2
[0063]	MWAX	MTA200-P (MTA200-controller)
[0063]	MSYX	SYNAX200-P, SYNAX200-R
[0063]	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 <u>M</u>anagement process

<u>Controller</u> (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 <u>W</u> rite	(Writing)
	R = Single <u>R</u> ead	(Reading)
	$C = \underline{C}$ yclic Read	(Cyclic reading)
	B = <u>B</u> 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 (*ILen). 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 (5 F_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.

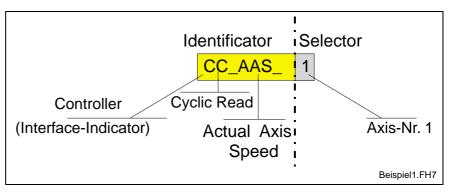


Fig. 6-2: Example 1: Cyclic reading in ASCII code

Example 2 (cyclic reading)

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

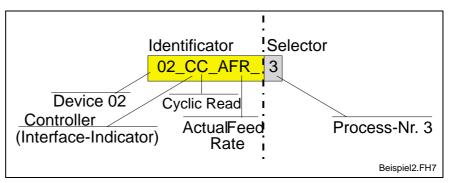


Fig. 6-3: Example 2: Cyclic reading in ASCII code

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.

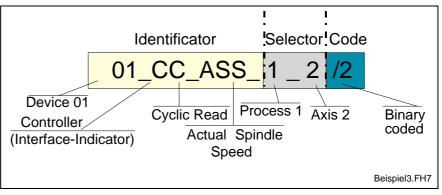


Fig. 6-4: Example 3: Cyclic reading in binary code

6.2 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	132
NC memory	1=A, 2=B
NC block No.	09999
NC program number	099
NC packet	199
Zero point database	09
Spindle number	S1, S2, S3
NC process number	06
Mechanism number	031
Drive address	0254
Tool number	09999999
Duplo No.	19999
Data block	0 = basic tool data 19 = tool edge data
Data element	128 for basic tool data 140 for tool edge data
Memory	M = magazine/turret S = spindle G = gripper X = index data
Location	1999 for M 14 for S,G 016770215 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
0н	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

	Measurement System					
Base unit	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			



6.3 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	Process	Read	Write	Cyclic
BFJ1	Break Function Interface Jobs	В	R		
BFJ2	Break Function Interface Jobs	В	R		
CCP1	Cell Configuration Parameter	В	R		
CCP2	Cell Configuration Parameter	В	R		
CCP3	Cell Configuration Parameter	В	R		
CCP4	Cell Configuration Parameter	В	R		
CCP5	Cell Configuration Parameter	В	R		
CEI1	Communication Error Info	В	R		
CPR1	Create PRocess	В		W	
CPR2	Create PRocess	В		W	
DFJ1	Delete Function Interface Job	В	R		
DFJ2	Delete Function Interface Job	В	R		
DFS1	Delete IF Command Stack	В		W	
DPR1	Delete PRocess	В		W	
ERI1	ERror Information	В	R		
FCP1	Far Device Configuration Parameter	В	R		С
FCP2	Far Device Configuration Parameter	В	R		С
FCP3	Far Device Configuration Parameter	В	R		С
FDC1	Far Device Configuration	В	R		С
FIT1	Further Info Text	В	R		
FPC1	Far PC Configuration	В	R		С
GDB1	Global Data Buffer	В	R	W	
ICA1	Initialisation Communication Address	В		W	С
IFJ1	Information about Function Interface Jobs	В	R		С
IFJ2	Information about Function Interface Jobs	В	R		С
IFS1	IF Command Stack Info	В	R		
LDT1	PC Local Date Time	В	R		С
LNG	Active LaNGuage	В	R	W	
MSG	MeSsaGe	С			С
NST1	NT-ShuT-Down	В		W	
NST2	NT-ShuT-Down	В		W	
PAF1	PArameter File Converted	В		W	
PHD1	Physical Directory	В	R		



Com.	Description	Process	Read	Write	Cyclic
POB1	POrt Byte Access	В	R	W	
POW1	POrt Word Access	В	R	W	
RPR1	Ready PRocess	В		W	
SDP1	Start Device Polling	В		W	
SFW1	Set Focus to Window	В		W	
SFW2	Set Focus to Window	В		W	
SSM1	Set Sys Message	В		W	
SSM2	Set Sys Message	В		W	

Fig. 6-5: Overview of the MPCX device group

Overview of the MTCX Device Group

Com.	Description	Process	Read	Write	Cyclic
AAC1	Actual ACceleration	С	R		С
AAD	Active Angle Dimension	С	R		С
AAS1	Actual Axis Speed	С	R		С
AAS2	Actual Axis Speed	С	R		С
ABI	Actual NC-Block Information	В	R		С
ACS	Actual Cutting Speed	С	R		С
ADN1	Active D-Correction Number	С	R		С
AEM	Active Event Monitoring	С	R		С
AEN	Active Edge-Number	С	R		С
AFO1	Active Feedrate Override	С	R		С
AFR	Active FeedRate	С	R		С
AGF	Active G-Function	С	R		С
AMF	Active M-Function	С	R		С
AMM1	Active Mechanism Message	В	R		С
AMM2	Active Mechanism Message	В	R		С
AMM3	Active Mechanism Message	В	R		С
AMM4	Active Mechanism Message	В	R		С
AMM5	Active Mechanism Message	В	R		С
ANM	Active NC Memory Size	С	R		
API1	Actual Parameter Index	В	R		С
API2	Actual Parameter Index	В	R		С
APM	Active Part-Program Message	С	R		С
APN	Active Part-Program Message Number	С	R		С
APO1	Actual Machine POsition	С	R		С
APO2	Actual Machine POsition	С	R		С
APP	Active Part-Program number	С	R		С
ARF	Axis Reference Flags	С	R		С
ARO1	Actual Rapid Override	С	R		С



Com.	Description	Process	Read	Write	Cyclic
ART	Axis Reference Table	С	R		С
ART	Axis Reference Table	В	R		
ASD	Actual Spindel Data	С	R		С
ASF	Actual Spindle For Process	С	R		С
ASG	Actual Spindle Gear	С	R		С
ASM1	Active System-Fault Message	В	R		С
ASM2	Active System-Fault Message	В	R		С
ASM3	Active System-Fault Message	В	R		С
ASM4	Active System-Fault Message	В	R		С
ASM5	Active System-Fault Message	В	R		С
ASN	Actual Sequence Number	С	R		С
ASO1	Actual Spindle Override	С	R		С
ASS	Actual Spindle Speed	С	R		С
ATN	Active Tool-Number	С	R		С
ATP1	Actual Tool Place Information	С	R		С
ATP2	Actual Tool Place Information	С	R		С
ATP3	Actual Tool Place Information	С	R		С
ATR	Actual Tool Data Record	С	R		С
ATU	Actual Tool Data Update	С	R		
AZB1	Active Zero Offset Bank	С	R		С
CCA1	NC Cycle Access	В	R	W	
CPO1	Command POsition (COMMAND)	С	R		С
CPO2	Command POsition by log. AxisNo	С	R		С
CRT	Control ReseT	С		W	
DAC1	Device Axis Configuration Parameter	В	R		С
DAC2	Device Axis Configuration Parameter	В	R		С
DCA1	NC D-Correction Access	В	R	W	
DCD1	D-Correction Data	С	R		С
DCP1	Device Configuration Parameter	В	R		С
DCP2	Device Configuration Parameter	В	R		С
DCR1	D-Correction Record	С	R	W	С
DCT1	Device Communication Timeout	В		W	
DCT2	Device Communication Timeout	В		W	
DIS1	Data Identification String Parameter	С	R		
DIS3	Data Identification String NC Packet	С	R		
DIS4	Data Identification String Tool List	С	R		
DIS5	Data Identification String Machine	С	R		
DIS6	Data Identification String NC Program	С	R		
DPN	Delete NC Program	В		W	
DPP	Delete Program Package	В		W	

Com.	Description	Process	Read	Write	Cyclic
DSI1	Device Status Information	В	R		С
DSI2	Device Status Information	В	R		С
DTC1	Device Tool Management Configuration	В	R		С
DTC2	Device Tool Management Configuration	В	R		С
DTG1	Distance To Go	С	R		С
DTG2	Distance To Go by log. AxisNo.	С	R		С
DTY1	Device TYpe	В	R		
DWD1	Diagnosis Window Data	В	R		С
DWD2	Diagnosis Window Data	В	R		С
EDE1	Existing Diagnosis Error	В	R		С
EDE2	Existing Diagnosis Error	В	R		С
EDW1	Existing Diagnosis Window	В	R		
EDW2	Existing Diagnosis Window	В	R		
EDW3	Existing Diagnosis Window	В	R		
END1	Existing NC Diagnosis	В	R		
END2	Existing NC Diagnosis	В	R		
EPD1	Existing PLC Diagnosis	В	R		
EPD2	Existing PLC Diagnosis	В	R		
EPD3	Existing PLC Diagnosis	В	R		
EPO1	ProgrammEd PO sition (END)	С	R		С
EPO2	ProgrammEd PO sition (END)	С	R		С
EPT1	Existing ProVi Types	В	R		
EST1	Error STate	В	R		С
EXD1	EXecution Display	В	R		С
EXD2	EXecution Display	В	R		
GPC1	Global Process Configuration	В	R		С
GPC2	Global Process Configuration	В	R		С
GPP1	Global Process Parameter	В	R		С
GPP2	Global Process Parameter	В	R		С
IPP	Insert NC-Program Package	В		W	
MAP1	Module Assign of Process	В	R		С
MCD1	Module Configuration: Device Information	В	R		С
MCM1	Module Configuration: Module Information	В	R		С
MCP1	Module Configuration: Process Information	В	R		С
MCS1	Module Configuration: SFC- Information	В	R		С
MDA1	Machine Data Access	В	R	W	
MDA2	Machine Data Access	В	R	W	
MDA4	Machine Data Access	В		W	
MDI	Manual Data Input	С		W	
MDS1	Machine Data Single	В	R	W	



Com.	Description	Process	Read	Write	Cyclic
MFD1	Message Files Download	В		W	
MFO1	Maximal Feedrate Override	С	R		С
MFR	Maximal FeedRate	С	R		С
MKS	Machine Key Status	В	R		С
MKT1	Machine Key Table	В		W	
MRO1	Maximal Rapid Override	С	R		С
MSG	MeSsaGe	С			С
MSO1	Maximal Spindle Override	С	R		С
MSS	Maximal Spindle Speed	С	R		С
MTC	MT-CNC Slot Software Version	В	R		
MTD	Machine Table Data	С	R	W	С
NCA1	NC Program Access	В	R	W	
NCA3	NC Program Access	В		W	
NCM1	NC Messages	В	R		С
NCM2	NC Messages	В	R		С
NEA1	NC Event Access	В	R	W	
NEV	NC Event	С	R	W	С
NMM	NC MeMory selection	С		W	
NPA1	NC PArameter	В	R		С
NPA2	NC PArameter	В	R		С
NPA3	NC PArameter	В	R		С
NPA4	NC PArameter	В	R		С
NPA5	NC PArameter	В	R		
NPC1	NC-Package Compiling	В	R		
NPD1	NC-Package Download	В		W	
NPI	NC-Package Directory	В	R		
NPS	NC Program Selection	С		W	
NTN	Next Tool-Number	С	R		С
NUA1	NC Offset Data Access	В	R	W	
NVA1	NC Variable Access	В	R	W	
NVS	NC Variable Single	С	R	W	С
OPD1	Optimal Position Distance by Axis sign.	С	R		С
OPD2	Optimal Position Distance by phys. AxisNo	С	R		С
PAA1	PArameter Access	В	R	W	
PAA2	PArameter Access	В	R	W	
PAC1	Process Axis Configuration Parameter	В	R		С
PAC2	Process Axis Configuration Parameter	В	R		С
PAD1	PArameter Deactivate	В		W	
PAS1	PArameter Set Active	В		W	
PDT	Parameter Definition Table	В	R		



Com.	Description	Process	Read	Write	Cyclic
PFR	Programmed FeedRate	С	R		С
PHD1	Physical Directory	В	R		
PPA	Part Program Active	В	R		С
PPD	Part Program Directory	В	R		
PPN	Part.Program NC	В	R	W	А
PPP	Part Program Package	В			С
PPS	Part Program-Sequence	С	R		
PSS	Programmed Spindle Speed	С	R		С
PTC1	Process Tool Management Configuration	В	R		С
PTC2	Process Tool Management Configuration	В	R		С
PVM1	ProVi Messages	В	R		С
PVM2	ProVi Messages	В	R		С
PVM3	ProVi Messages	В	R		С
PVM4	ProVi Messages	В	R		С
REP1	REPositioning Data	С	R		
REP2	REPositioning Data	С	R		С
SDD1	Sfc Diagnosis Data	В	R		
SDD2	Sfc Diagnosis Data	В	R		
SDD3	Sfc Diagnosis Data	В	R		
SDD4	Sfc Diagnosis Data	В	R		
SDD5	Sfc Diagnosis Data	В	R		
SDD6	Sfc Diagnosis Data	В	R		
SDS1	Set Device Status	В		W	
SDS2	Set Device Status	В		W	
SFD1	SFc Data	В	R		
SFD2	SFc Data	В	R		
SFD3	SFc Data	В	R		
SFE1	SFc Error	В	R		С
SFE2	SFc Error	В	R		С
SFM1	SFc Mode	В	R		С
SID1	Software Installation Data	В	R		С
SLA1	Actual Servo LAg	С	R		С
SLA2	Actual Servo LAg	С	R		С
SLI	SPS (PLC) Long Identification	В	R		С
SPP	Selected Part Program Number	С	R		С
TDA1	Tool DAta	В	R	W	
TDA2	Tool DAta	В	R		
TDD	Tool Data Download	С		W	
TDR1	Tool Data Record of Place	С	R		С
TDR2	Tool Data Record	С	R		С



Com.	Description	Process	Read	Write	Cyclic
TIF	Tool Insert Finish	С	R		
TII	Tool Insert Initiated	С	R		
TLB1	TooL Basicdata List	В	R		С
TLB2	TooL Basicdata List	В	R		С
TLD1	TooL Data of Place	С	R	W	С
TLD2	TooL Data of Tool	С	R	W	С
TLD3	TooL Data of Place	С	R	W	С
TLD4	TooL Data of Tool	С	R	W	С
TLE1	TooL Edgedata List	В	R		С
TLE2	TooL Edgedata List	В	R		С
TMV	Tool MoVe	С	R		
TPI1	Tool Position Information	В	R		
TPI2	Tool Position Information	В	R		
TQE1	Actual TorQuE	С	R		С
TQE2	Actual TorQuE	С	R		С
TRM	Tool ReMove	С	R		
TRS	Tool ReSet	С	R		
ZOD	Zero Offset Data	С		W	С
ZOD1	Zero Offset Data	С	R		С
ZOD2	Zero Offset Data	С	R		С

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

Overview of the MWCX Device Group

Com.	Description	Process	Read	Write	Cyclic
AAC1	Actual ACceleration	С	R		С
AAD	Active Angle Dimension	С	R		С
AAS1	Actual Axis Speed	С	R		С
AAS2	Actual Axis Speed	С	R		С
ABI	Actual NC-Block Information	В	R		С
ACS	Actual Cutting Speed	С	R		С
ADN1	Active D-Correction Number	С	R		С
AEM	Active Event Monitoring	С	R		С
AEN	Active Edge-Number	С	R		С
AFO1	Active Feedrate Override	С	R		С
AFR	Active FeedRate	С	R		С
AGF	Active G-Function	С	R		С
AMF	Active M-Function	С	R		С
AMM1	Active Mechanism Message	В	R		С
AMM2	Active Mechanism Message	В	R		С
AMM3	Active Mechanism Message	В	R		С
AMM4	Active Mechanism Message	В	R		С



Com.	Description	Process	Read	Write	Cyclic
AMM5	Active Mechanism Message	В	R		С
ANM	Active NC Memory Size	С	R		
API1	Actual Parameter Index	В	R		С
API2	Actual Parameter Index	В	R		С
APM	Active Part-Program Message	С	R		С
APN	Active Part-Program Message Number	С	R		С
APO1	Actual Machine POsition	С	R		С
APO2	Actual Machine POsition	С	R		С
APP	Active Part-Program number	С	R		С
ARF	Axis Reference Flags	С	R		С
ARO1	Actual Rapid Override	С	R		С
ART	Axis Reference Table	С	R		С
ART	Axis Reference Table	В	R		
ASD	Actual Spindel Data	С	R		С
ASF	Actual Spindle For Process	С	R		С
ASG	Actual Spindle Gear	С	R		С
ASM1	Active System-Fault Message	В	R		С
ASM2	Active System-Fault Message	В	R		С
ASM3	Active System-Fault Message	В	R		С
ASM4	Active System-Fault Message	В	R		С
ASM5	Active System-Fault Message	В	R		С
ASN	Actual Sequence Number	С	R		С
ASO1	Actual Spindle Override	С	R		С
ASS	Actual Spindle Speed	С	R		С
ATN	Active Tool-Number	С	R		С
ATP1	Actual Tool Place Information	С	R		С
ATP2	Actual Tool Place Information	С	R		С
ATP3	Actual Tool Place Information	С	R		С
ATR	Actual Tool Data Record	С	R		С
ATU	Actual Tool Data Update	С	R		
AZB1	Active Zero Offset Bank	С	R		С
CCA1	NC Cycle Access	В	R	W	
CPO1	Command POsition (COMMAND)	С	R		С
CPO2	Command POsition by log.Axis No	С	R		С
CRT	Control ReseT	С		W	
DAC1	Device Axis Configuration Parameter	В	R		С
DAC2	Device Axis Configuration Parameter	В	R		С
DCA1	NC D-Correction Access	В	R	W	
DCD1	D-Correction Data	С	R		С
DCP1	Device Configuration Parameter	В	R		С

Com.	Description	Process	Read	Write	Cyclic
DCP2	Device Configuration Parameter	В	R		С
DCR1	D-Correction Record	С	R	W	С
DCT1	Device Communication Timeout	В		W	
DCT2	Device Communication Timeout	В		W	
DIS1	Data Identification String Parameter	С	R		
DIS2	Data Identification String PLC Program	С	R		
DIS3	Data Identification String NC Packet	С	R		
DIS4	Data Identification String Tool List	С	R		
DIS5	Data Identification String Machine	С	R		
DIS6	Data Identification String NC Program	С	R		
DPN	Delete Programm NC	В		W	
DPP	Delete Program Package	В		W	
DSI1	Device Status Information	В	R		С
DSI2	Device Status Information	В	R		С
DTC1	Device Tool Management Configuration	В	R		С
DTC2	Device Tool Management Configuration	В	R		С
DTG1	Distance To Go	С	R		С
DTG2	Distance To Go by log. AxisNo.	С	R		С
DTY1	Device TYpe	В	R		
DWD1	Diagnosis Window Data	В	R		С
DWD2	Diagnosis Window Data	В	R		С
EDE1	Existing Diagnosis Error	В	R		С
EDE2	Existing Diagnosis Error	В	R		С
EDW1	Existing Diagnosis Window	В	R		
EDW2	Existing Diagnosis Window	В	R		
EDW3	Existing Diagnosis Window	В	R		
END1	Existing NC Diagnosis	В	R		
END2	Existing NC Diagnosis	В	R		
EPD1	Existing PLC Diagnosis	В	R		
EPD2	Existing PLC Diagnosis	В	R		
EPD3	Existing PLC Diagnosis	В	R		
EPO1	ProgrammEd POsition (END)	С	R		С
EPO2	ProgrammEd POsition (END)	С	R		С
EPT1	Existing ProVi Types	В	R		
EST1	Error STate	В	R		С
EXD1	EXecution Display	В	R		С
EXD2	EXecution Display	В	R		
GPC1	Global Process Configuration	В	R		С
GPC2	Global Process Configuration	В	R		С
GPP1	Global Process Parameter	В	R		С

Com.	Description	Process	Read	Write	Cyclic
GPP2	Global Process Parameter	В	R		С
IPP	Insert NC-Program Package	В		W	
MAP1	Module Assign of Process	В	R		С
MAR	Map Absolute PCL Reference	В	R		
MCD1	Module Configuration: Device Information	В	R		С
MCM1	Module Configuration: Module Information	В	R		С
MCP1	Module Configuration: Process Information	В	R		С
MCS1	Module Configuration: SFC- Information	В	R		С
MDA1	Machine Data Access	В	R	W	
MDA2	Machine Data Access	В	R	W	
MDA4	Machine Data Access	В		W	
MDI	Manual Data Input	С		W	
MDS1	Machine Data Single	В	R	W	
MFD1	Message Files Download	В		W	
MFO1	Maximal Feedrate Override	С	R		С
MFR	Maximal FeedRate	С	R		С
MKS	Machine Key Status	В	R		С
MKT1	Machine Key Table	В		W	
MRO1	Maximal Rapid Override	С	R		С
MSG	MeSsaGe	С			С
MSO1	Maximal Spindle Override	С	R		С
MSS	Maximal Spindle Speed	С	R		С
MTC	MT-CNC Slot Software Version	В	R		
MTD	Machine Table Data	С	R	W	С
NCA1	NC Program Access	В	R	W	
NCA3	NC Program Access	В		W	
NCM1	NC Messages	В	R		С
NCM2	NC Messages	В	R		С
NEA1	NC Event Access	В	R	W	
NEV	NC Event	С	R	W	С
NMM	NC MeMory selection	С		W	
NPA1	NC PArameter	В	R		С
NPA2	NC PArameter	В	R		С
NPA3	NC PArameter	В	R		С
NPA4	NC PArameter	В	R		С
NPA5	NC PArameter	В	R		
NPC1	NC-Package Compiling	В	R		
NPD1	NC-Package Download	В		W	
NPI	NC-Package Directory	В	R		
NPS	NC Program Selection	С		W	



Com.	Description	Process	Read	Write	Cyclic
NTN	Next Tool-Number	С	R		С
NUA1	NC Offset Data Access	В	R	W	
NVA1	NC Variable Access	В	R	W	
NVS	NC Variable Single	С	R	W	С
OPD1	Optimal Position Distance by Axis sign.	С	R		С
OPD2	Optimal Position Distance by phys. AxisNo	С	R		С
PAA1	PArameter Access	В	R	W	
PAA2	PArameter Access	В	R	W	
PAC1	Process Axis Configuration Parameter	В	R		С
PAC2	Process Axis Configuration Parameter	В	R		С
PAD1	PArameter Deactivate	В		W	
PAS1	PArameter Set Active	В		W	
PDD1	Provi Diagnosis Data	В	R		
PDD2	Provi Diagnosis Data	В	R		
PDD3	Provi Diagnosis Data	В	R		
PDD4	Provi Diagnosis Data	В	R		
PDD5	Provi Diagnosis Data	В	R		
PDT	Parameter Definition Table	В	R		
PFR	Programmed FeedRate	С	R		С
PHD1	Physical Directory	В	R		
PPA	Part Program Active	В	R		С
PPD	Part Program Directory	В	R		
PPN	Part.Program NC	В	R	W	A
PPP	Part Program Package	В			A
PPS	Part Program Sequence	С	R		
PSM	PLC Sys Message	В		W	
PSS	Programmed Spindle Speed	С	R		С
PTC1	Process Tool Management Configuration	В	R		С
PTC2	Process Tool Management Configuration	В	R		С
PVA1	PROVI Messages Access	В	R	W	
PVA2	PROVI Messages Access	В		W	
PVF	PLC Variable Formatted	С		W	С
PVM1	ProVi Messages	В	R		С
PVM2	ProVi Messages	В	R		С
PVM3	ProVi Messages	В	R		С
PVM4	ProVi Messages	В	R		С
PVR1	PLC Variable Retain Backup	В	R	W	
PVT	PLC Variable Type	В	R		
REP1	REPositioning Data	С	R		
REP2	REPositioning Data	С	R		С

Com.	Description	Process	Read	Write	Cyclic
SDD1	Sfc Diagnosis Data	В	R		
SDD2	Sfc Diagnosis Data	В	R		
SDD3	Sfc Diagnosis Data	В	R		
SDD4	Sfc Diagnosis Data	В	R		
SDD5	Sfc Diagnosis Data	В	R		
SDD6	Sfc Diagnosis Data	В	R		
SDS1	Set Device Status	В		W	
SDS2	Set Device Status	В		W	
SFD1	SFc Data	В	R		
SFD2	SFc Data	В	R		
SFD3	SFc Data	В	R		
SFE1	SFc Error	В	R		С
SFE2	SFc Error	В	R		С
SFM1	SFc Mode	В	R		С
SID1	Software Installation Data	В	R		С
SLA1	Actual Servo LAg	С	R		С
SLA2	Actual Servo LAg	С	R		С
SLI	SPS (PLC) Long Identification	В	R		С
SPA1	Sercos PArameter	В	R	W	С
SPH	Sercos PHase	С	R	W	
SPP	Selected Part Program Number	С	R		С
TDA1	Tool DAta	В	R	W	
TDA2	Tool DAta	В	R		
TDD	Tool Data Download	С		W	
TDR1	Tool Data Record of Place	С	R		С
TDR2	Tool Data Record	С	R		С
TIF	Tool Insert Finish	С	R		
TII	Tool Insert Initiated	С	R		
TLB1	TooL Basicdata List	В	R		С
TLB2	TooL Basicdata List	В	R		С
TLD1	TooL Data of Place	С	R	W	С
TLD2	TooL Data of Tool	С	R	W	С
TLD3	TooL Data of Place	С	R	W	С
TLD4	TooL Data of Tool	С	R	W	С
TLE1	TooL Edgedata List	В	R		С
TLE2	TooL Edgedata List	В	R		С
TMV	Tool MoVe	С	R		
TPI1	Tool Position Information	В	R		
TPI2	Tool Position Information	В	R		
TQE1	Actual TorQuE	С	R		С



Com.	Description	Process	Read	Write	Cyclic
TQE2	Actual TorQuE	С	R		С
TRM	Tool ReMove	С	R		
TRS	Tool ReSet	С	R		
WLA1	Watch List Allocation	В	R		
WLF1	Watch List Free	В	R		
WLF2	Watch List Free	В	R		
ZOD	Zero Offset Data	С		W	С
ZOD1	Zero Offset Data	С	R		С
ZOD2	Zero Offset Data	С	R		С

Fig. 6-7:	Overview of the MWCX device group
-----------	-----------------------------------

Overview of the MSCX Device Group

Com.	Description	Process	Read	Write	Cyclic
ASE	Actual System Error	С	R		С
CSE	Clear System Error	С		W	
DCT1	Device Communication Timeout	В		W	
DCT2	Device Communication Timeout	В		W	
DSI1	Device Status Information	В	R		С
DSI2	Device Status Information	В	R		С
DTY1	Device TYpe	В	R		
MSG	MeSsaGe	С			С
PHD1	Physical Directory	В	R		
SDS1	Set Device Status	В		W	
SDS2	Set Device Status	В		W	
SID1	Software Installation Data	В	R		С
SPA1	Sercos PArameter	В	R	W	С
SPH	Sercos PHase	С	R	W	С

Fig. 6-8: Overview of the MSCX device group

Overview of the MWMX Device Group

Com.	Description	Process	Read	Write	Cyclic
ASM1	Active System Fault Message	В	R		С
ASM2	Active System Fault Message	В	R		С
ASM3	Active System Fault Message	В	R		С
ASM4	Active System Fault Message	В	R		С
ASM5	Active System Fault Message	В	R		С
CRT	Control ReseT	С		W	
DCT1	Device Communication Timeout	В		W	
DCT2	Device Communication Timeout	В		W	
DIS2	Data Identification String PLC Program	С	R		
DSI1	Device Status Information	В	R		С



Com.	Description	Process	Read	Write	Cyclic
DSI2	Device Status Information	В	R		С
DTY1	Device Type	В	R		
DWD1	Diagnosis Window Data	В	R		С
DWD2	Diagnosis Window Data	В	R		С
EDE1	Existing Diagnosis Error	В	R		С
EDE2	Existing Diagnosis Error	В	R		С
EDW1	Existing Diagnosis Window	В	R		
EDW2	Existing Diagnosis Window	В	R		
EDW3	Existing Diagnosis Window	В	R		
EPD1	Existing PLC Diagnosis	В	R		
EPD2	Existing PLC Diagnosis	В	R		
EPD3	Existing PLC Diagnosis	В	R		
EPT1	Existing ProVi Types	В	R		
EST1	Error STate	В	R		С
EXD1	EXecution Display	В	R		С
EXD2	EXecution Display	В	R		
MAR	Map Absolute PCL Reference	В	R		
MCD1	Module Configuration: Device Information	В	R		С
MCM1	Module Configuration: Module Information	В	R		С
MCS1	Module Configuration: SFC- Information	В	R		С
MFD1	Message Files Download	В		W	
MKS	Machine Key Status	В	R		С
MKT1	Machine Key Table	В		W	
MSG	MeSsaGe	С			С
MTC	MT-CNC Slot Software Version	В	R		
PDD1	Provi Diagnosis Data	В	R		
PDD2	Provi Diagnosis Data	В	R		
PDD3	Provi Diagnosis Data	В	R		
PDD4	Provi Diagnosis Data	В	R		
PDD5	Provi Diagnosis Data	В	R		
PHD1	Physical Directory	В	R		
PSM	PLC Sys Message	В		W	
PVA1	PROVI Messages Access	В	R	W	
PVA2	PROVI Messages Access	В		W	
PVF	PLC Variable Formatted	С		W	С
PVM1	ProVi Messages	В	R		С
PVM2	ProVi Messages	В	R		С
PVM3	ProVi Messages	В	R		С
PVM4	ProVi Messages	В	R		С
PVR1	PLC Variable Retain Backup	В	R	W	



Com.	Description	Process	Read	Write	Cyclic
PVT	PLC Variable Type	В	R		
SDD1	Sfc Diagnosis Data	В	R		
SDD2	Sfc Diagnosis Data	В	R		
SDD3	Sfc Diagnosis Data	В	R		
SDD4	Sfc Diagnosis Data	В	R		
SDD5	Sfc Diagnosis Data	В	R		
SDD6	Sfc Diagnosis Data	В	R		
SDS1	Set Device Status	В		W	
SDS2	Set Device Status	В		W	
SFD1	SFc Data	В	R		
SFD2	SFc Data	В	R		
SFD3	SFc Data	В	R		
SFE1	SFc Error	В	R		С
SFE2	SFc Error	В	R		С
SFM1	SFc Mode	В	R		С
SID1	Software Installation Data	В	R		С
SLI	SPS (PLC) Long Identification	В	R		С
WLA1	Watch List Allocation	В	R		
WLF1	Watch List Free	В	R		
WLF2	Watch List Free	В	R		

Fig. 6-9:	Overview of the MWMX device group
-----------	-----------------------------------

Overview of the MWSX Device Group

Com.	Description	Process	Read	Write	Cyclic
ASM1	Active System Fault Message	В	R		С
ASM2	Active System Fault Message	В	R		С
ASM3	Active System Fault Message	В	R		С
ASM4	Active System Fault Message	В	R		С
ASM5	Active System Fault Message	В	R		С
CRT	Control ReseT	С		W	
DCT1	Device Communication Timeout	В		W	
DCT2	Device Communication Timeout	В		W	
DIS2	Data Identification String PLC Program	С	R		
DSI1	Device Status Information	В	R		С
DSI2	Device Status Information	В	R		С
DTY1	Device Type	В	R		
DWD1	Diagnosis Window Data	В	R		С
DWD2	Diagnosis Window Data	В	R		С
EDE1	Existing Diagnosis Error	В	R		С
EDE2	Existing Diagnosis Error	В	R		С



Com.	Description	Process	Read	Write	Cyclic
EDW1	Existing Diagnosis Window	В	R		
EDW2	Existing Diagnosis Window	В	R		
EDW3	Existing Diagnosis Window	В	R		
EPD1	Existing PLC Diagnosis	В	R		
EPD2	Existing PLC Diagnosis	В	R		
EPD3	Existing PLC Diagnosis	В	R		
EPT1	Existing ProVi Types	В	R		
EST1	Error STate	В	R		С
EXD1	EXecution Display	В	R		С
EXD2	EXecution Display	В	R		
MAR	Map Absolute PCL Reference	В	R		
MCD1	Module Configuration: Device Information	В	R		С
MCM1	Module Configuration: Module Information	В	R		С
MCS1	Module Configuration: SFC- Information	В	R		С
MFD1	Message Files Download	В		W	
MKS	Machine Key Status	В	R		С
MKT1	Machine Key Table	В		W	
MSG	MeSsaGe	С			С
MTC	MT-CNC Slot Software Version	В	R		
PDD1	Provi Diagnosis Data	В	R		
PDD2	Provi Diagnosis Data	В	R		
PDD3	Provi Diagnosis Data	В	R		
PDD4	Provi Diagnosis Data	В	R		
PDD5	Provi Diagnosis Data	В	R		
PHD1	Physical Directory	В	R		
PSM	PLC Sys Message	В		W	
PVA1	PROVI Messages Access	В	R	W	
PVA2	PROVI Messages Access	В		W	
PVF	PLC Variable Formatted	С		W	С
PVM1	ProVi Messages	В	R		С
PVM2	ProVi Messages	В	R		С
PVM3	ProVi Messages	В	R		С
PVM4	ProVi Messages	В	R		С
PVR1	PLC Variable Retain Backup	В	R	W	
PVT	PLC Variable Type	В	R		
SDD1	Sfc Diagnosis Data	В	R		
SDD2	Sfc Diagnosis Data	В	R		
SDD3	Sfc Diagnosis Data	В	R		
SDD4	Sfc Diagnosis Data	В	R		
SDD5	Sfc Diagnosis Data	В	R		



Com.	Description	Process	Read	Write	Cyclic
SDD6	Sfc Diagnosis Data	В	R		
SDS1	Set Device Status	В		W	
SDS2	Set Device Status	В		W	
SFD1	SFc Data	В	R		
SFD2	SFc Data	В	R		
SFD3	SFc Data	В	R		
SFE1	SFc Error	В	R		С
SFE2	SFc Error	В	R		С
SFM1	SFc Mode	В	R		С
SID1	Software Installation Data	В	R		С
SLI	SPS (PLC) Long Identification	В	R		С
WLA1	Watch List Allocation	В	R		
WLF1	Watch List Free	В	R		
WLF2	Watch List Free	В	R		

Fig. 6-10: Overview of MWSX device group

Overview of the MWAX Device Group

Com.	Description	Process	Read	Write	Cyclic
ADM1	MTA200 Messages	В	R		С
ADM2	MTA200 Messages	В	R		С
ADM3	MTA200 Messages	В	R		
AMM7	Active Mechanism Message	В	R		С
APO2	Actual Machine POsition	С	R		С
ASM1	Active System Fault Message	В	R		С
ASM2	Active System Fault Message	В	R		С
ASM3	Active System Fault Message	В	R		С
ASM4	Active System Fault Message	В	R		С
ASM5	Active System Fault Message	В	R		С
CMA	CMOS RAM ASCII Parameter	С	R	W	
CMF	CMOS RAM Floating point Parameter	С	R	W	
CMI	CMOS RAM Integer Parameter	С	R	W	
CRT	Control ReseT	С		W	
DCP1	Device Configuration Parameter	В	R		С
DCP2	Device Configuration Parameter	В	R		С
DCT1	Device Communication Timeout	В		W	
DCT2	Device Communication Timeout	В		W	
DIS1	Data Identification String Parameter	С	R		
DIS2	Data Identification String PLC Program	С	R		
DSI1	Device Status Information	В	R		С
DSI2	Device Status Information	В	R		С
DTG2	Distance To Go by log. AxisNo.	С	R		С



Com.	Description	Process	Read	Write	Cyclic
DTY1	Device Type	В	R		
DWD1	Diagnosis Window Data	В	R		С
DWD2	Diagnosis Window Data	В	R		С
EAD1	Existing MTA200 Diagnosis	В	R		
EAD2	Existing MTA200 Diagnosis	В	R		
EDE1	Existing Diagnosis Error	В	R		С
EDE2	Existing Diagnosis Error	В	R		С
EDW1	Existing Diagnosis Window	В	R		
EDW2	Existing Diagnosis Window	В	R		
EDW3	Existing Diagnosis Window	В	R		
EPT1	Existing ProVi Types	В	R		
EST1	Error STate	В	R		С
EXD1	EXecution Display	В	R		С
EXD2	EXecution Display	В	R		
MAP1	Module Assign of Process	В	R		С
MAR	Map Absolute PCL Reference	В	R		
MCD1	Module Configuration: Device Information	В	R		С
MCM1	Module Configuration: Module Information	В	R		С
MCP1	Module Configuration: Process Information	В	R		С
MCS1	Module Configuration: SFC- Information	В	R		С
MFD1	Message Files Download	В		W	
MKS	Machine Key Status	В	R		С
MKT1	Machine Key Table	В		W	
MSG	MeSsaGe	С			С
MTC	MT-CNC Slot Software Version	В	R		
PAA2	PArameter Access	В	R	W	
PDD1	Provi Diagnosis Data	В	R		
PDD2	Provi Diagnosis Data	В	R		
PDD3	Provi Diagnosis Data	В	R		
PDD4	Provi Diagnosis Data	В	R		
PDD5	Provi Diagnosis Data	В	R		
PHD1	Physical Directory	В	R		
POI	POsition Information	В	R		С
PSM	PLC Sys Message	В		W	
PVA1	PROVI Messages Access	В	R	W	
PVA2	PROVI Messages Access	В		W	
PVF	PLC Variable Formatted	С		W	С
PVM1	ProVi Messages	В	R		С
PVM2	ProVi Messages	В	R		С
PVM3	ProVi Messages	В	R		С



Com.	Description	Process	Read	Write	Cyclic
PVM4	ProVi Messages	В	R		С
PVR1	PLC Variable Retain Backup	В	R	W	
PVT	PLC Variable Type	В	R		
SDD1	Sfc Diagnosis Data	В	R		
SDD2	Sfc Diagnosis Data	В	R		
SDD3	Sfc Diagnosis Data	В	R		
SDD4	Sfc Diagnosis Data	В	R		
SDD5	Sfc Diagnosis Data	В	R		
SDD6	Sfc Diagnosis Data	В	R		
SDS1	Set Device Status	В		W	
SDS2	Set Device Status	В		W	
SFD1	SFc Data	В	R		
SFD2	SFc Data	В	R		
SFD3	SFc Data	В	R		
SFE1	SFc Error	В	R		С
SFE2	SFc Error	В	R		С
SFM1	SFc Mode	В	R		С
SID1	Software Installation Data	В	R		С
SLI	SPS (PLC) Long Identification	В	R		С
WLA1	Watch List Allocation	В	R		
WLF1	Watch List Free	В	R		
WLF2	Watch List Free	В	R		

Overview of the MSYX Device Group

Com.	Description	Process	Read	Write	Cyclic
ASE	Actual System Error	С	R		С
CSE	Clear System Error	С		W	
DCT1	Device Communication Timeout	В		W	
DCT2	Device Communication Timeout	В		W	
DTY	Device TYpe	В	R		
MSG	MeSsaGe	С			С
PHD1	Physical Directory	В	R		
SDS1	Set Device Status	В		W	
SDS2	Set Device Status	В		W	
SID1	Software Installation Data	В	R		С
SPA1	Sercos PArameter	В	R	W	С
SPH	Sercos PHase	С	R	W	С

Fig. 6-12:	Overview of the MSYX device group
------------	-----------------------------------



Overview of the MWYX Device Group

Com.	Description	Process	Read	Write	Cyclic
ASE	Actual System Error	С	R		С
ASM1	Active System Fault Message	В	R		С
ASM2	Active System Fault Message	В	R		С
ASM3	Active System Fault Message	В	R		С
ASM4	Active System Fault Message	В	R		С
ASM5	Active System Fault Message	В	R		С
CRT	Control ReseT	С		W	
CSE	Clear System Error	С		W	
DCT1	Device Communication Timeout	В		W	
DCT2	Device Communication Timeout	В		W	
DIS2	Data Identification String PLC Program	С	R		
DSI1	Device Status Information	В	R		С
DSI2	Device Status Information	В	R		С
DTY1	Device Type	В	R		
DWD1	Diagnosis Window Data	В	R		С
DWD2	Diagnosis Window Data	В	R		С
EDE1	Existing Diagnosis Error	В	R		С
EDE2	Existing Diagnosis Error	В	R		С
EDW1	Existing Diagnosis Window	В	R		
EDW2	Existing Diagnosis Window	В	R		
EDW3	Existing Diagnosis Window	В	R		
EPD1	Existing PLC Diagnosis	В	R		
EPD2	Existing PLC Diagnosis	В	R		
EPD3	Existing PLC Diagnosis	В	R		
EPT1	Existing ProVi Types	В	R		
EST1	Error STate	В	R		С
EXD1	EXecution Display	В	R		С
EXD2	EXecution Display	В	R		
MAR	Map Absolute PCL Reference	В	R		
MCD1	Module Configuration: Device Information	В	R		С
MCM1	Module Configuration: Module Information	В	R		С
MCS1	Module Configuration: SFC- Information	В	R		С
MFD1	Message Files Download	В		W	
MKS	Machine Key Status	В	R		С
MKT1	Machine Key Table	В		W	
MSG	MeSsaGe	С			С
MTC	MT-CNC Slot Software Version	В	R		

Com.	Description	Process	Read	Write	Cyclic
PDD1	Provi Diagnosis Data	В	R		
PDD2	Provi Diagnosis Data	В	R		
PDD3	Provi Diagnosis Data	В	R		
PDD4	Provi Diagnosis Data	В	R		
PDD5	Provi Diagnosis Data	В	R		
PHD1	Physical Directory	В	R		
PSM	PLC Sys Message	В		W	
PVA1	PROVI Messages Access	В	R	W	
PVA2	PROVI Messages Access	В		W	
PVF	PLC Variable Formatted	С		W	С
PVM1	ProVi Messages	В	R		С
PVM2	ProVi Messages	В	R		С
PVM3	ProVi Messages	В	R		С
PVM4	ProVi Messages	В	R		С
PVR1	PLC Variable Retain Backup	В	R	W	
PVT	PLC Variable Type	В	R		
SDD1	Sfc Diagnosis Data	В	R		
SDD2	Sfc Diagnosis Data	В	R		
SDD3	Sfc Diagnosis Data	В	R		
SDD4	Sfc Diagnosis Data	В	R		
SDD5	Sfc Diagnosis Data	В	R		
SDD6	Sfc Diagnosis Data	В	R		
SDS1	Set Device Status	В		W	
SDS2	Set Device Status	В		W	
SFD1	SFc Data	В	R		
SFD2	SFc Data	В	R		
SFD3	SFc Data	В	R		
SFE1	SFc Error	В	R		С
SFE2	SFc Error	В	R		С
SFM1	SFc Mode	В	R		С
SID1	Software Installation Data	В	R		С
SLI	SPS (PLC) Long Identification	В	R		С
SPA1	Sercos PArameter	В	R	W	С
SPH	Sercos PHase	С	R	W	С
WLA1	Watch List Allocation	В	R		
WLF1	Watch List Free	В	R		
WLF2	Watch List Free	В	R		

Fig. 6-13:	Overview of the MWYX device group
------------	-----------------------------------

Logical Connection between FI Commands

Group	Device group	FI Commands
Axes	MTCX	AAD, AAS1, AAS2, ARF, ART, CPO1, CPO2, DTG1, DTG2, EPO1, EPO2, OPD1, OPD2, PAC1, PAC2, REP1, REP2, SLA1, SLA2, TQE1, TQE2
	MWCX	AAD, AAS1, AAS2, ARF, ART, CPO1, CPO2, DTG1, DTG2, EPO1, EPO2, OPD1, OPD2, PAC1, PAC2, REP1, REP2, SLA1, SLA2, TQE1, TQE2
	MWAX	DTG2
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	DWD1, DWD2, EDE1, EDE2, EDW1, EDW2, EDW3, END1, END2
	MWCX	DWD1, DWD2, EDE1, EDE2, EDW1, EDW2, EDW3, END1, END2, EPD1, EPD2, EPD3, SDD1, SDD2, SDD3, SDD4, SDD5, SDD6
	MWMX	DWD1, DWD2, EDE1, EDE2, EDW1, EDW2, EDW3, EPD1, EPD2, EPD3, SDD1, SDD2, SDD3, SDD4, SDD5, SDD6
	MWSX	DWD1, DWD2, EDE1, EDE2, EDW1, EDW2, EDW3, EPD1, EPD2, EPD3, SDD1, SDD2, SDD3, SDD4, SDD5, SDD6
	MWAX	DWD1, DWD2, EAD1, EAD2, EDE1, EDE2, EDW1, EDW2, EDW3, SDD1, SDD2, SDD3, SDD4, SDD5, SDD6
	MWYX	DWD1, DWD2, EDE1, EDE2, EDW1, EDW2, EDW3, EPD1, EPD2, EPD3, SDD1, SDD2, SDD3, SDD4, SDD5, SDD6
Download/ Upload	MTCX	CCA1, DCA1, MDA1, MDA2, MDA4, MFD1, NCA1, NEA1, NUA1, NVA1, PAA1, PAA2
	MWCX	CCA1, DCA1, MDA1, MDA2, MFD1, NCA1, NEA1, NUA1, NVA1, PAA1, PAA2, PVR1
	MWMX	MFD1, PVR1
	MWSX	MFD1, PVR1
	MWAX	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	DCP1, DCP2, DSI1, DSI2, DTC1, DTY1, MCD1, SDS1, SDS2
	MSCX	DSI1, DSI2, DTY1, SDS1, SDS2
	MWMX	DSI1, DSI2, DTY1, MCD1, SDS1, SDS2
	MWSX	DSI1, DSI2, DTY1, MCD1, SDS1, SDS2
	MWAX	DCP1, DCP2, DSI1, DSI2, DTY1, MCD1, SDS1, SDS2
	MWYX	DSI1, DSI2, DTY1, MCD1, SDS1, SDS2

Group	Device group	FI Commands			
Configuration	MPCX	CCP1, CCP2, CCP3, CCP4, CCP5, FCP1, FCP2, FCP3, FDC1			
	MTCX	DAC1, DAC2, DCP1, DCP2, DTY1, GPC1, GPC2, PAC1, PAC2, PTC1, PTC2			
	MWCX	DAC1, DAC2, DCP1, DCP2, DTY1, GPC1, GPC2, PAC1, PAC2, PTC1, PTC2			
	MSCX	DTY1			
	MWMX	DTY1			
	MWSX	DTY1			
	MWAX	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	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	ASM1, ASM2, ASM3, ASM4, 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			
	MWYX	ASM1, ASM2, ASM3, ASM4, ASM5, 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	МТСХ	ABI, AGF, AMF, ANM, APM, APN, APP, ASN, CCA1, DCA1, DIS1, DIS2, DIS3, DIS4, DIS5, 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, DIS2, DIS3, DIS4, DIS5, 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			
Override	MTCX	AFO1, ARO1, ASO1, MFO1, MRO1, MSO1			
	MWCX	AFO1, ARO1, ASO1, MFO1, MRO1, MSO1			



Group	Device group	FI Commands
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
Process	MPCX	CPR1, CPR2, DPR1, NST1, NST2, RPR1
	MTCX	GPC1, GPC2, GPP1, GPP2, MAP1, MCP1, PAC1, PAC2, PTC1, PTC2
	MWCX	GPC1, GPC2, GPP1, GPP2, MAP1, MCP1, PAC1, PAC2, PTC1, PTC2
	MWAX	MAP1, MCP1
Cutters	MTCX	AEN, TLE1, TLE2
	MWCX	AEN, TLE1, TLE2
Sercos	MWCX	SPA1, SPH
	MSCX	SPA1, SPH
	MSYX	SPA1, SPH
	MWYX	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
PLC	MWCX	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	DIS2, EDE1, EDE2, EPD1, EPD2, EPD3, EPT1, EST1, EXD1, EXD2, MAR, MKT1, PVA1, PVA2, PVF, PVT, SDD1, SDD2, SDD3, SDD4, SDD5, SDD6, SFD1, SFD2, SFE1, SFE2, SFM1, SLI
	MWSX	DIS2, EDE1, EDE2, EPD1, EPD2, EPD3, EPT1, EST1, EXD1, EXD2, MAR, MKT1, PVA1, PVA2, PVF, PVT, SDD1, SDD2, SDD3, SDD4, SDD5, SDD6, SFD1, SFD2, SFE1, SFE2, SFM1, SLI
	MWAX	DIS2, EDE1, EDE2, EST1, EPT1, EXD1, EXD2, MAR, MKT1, PVA1, PVA2, PVF, PVT, SDD1, SDD2, SDD3, SDD4, SDD5, SDD6, SFD1, SFD2, SFE1, SFE2, SFM1, SLI
	MWYX	DIS2, EDE1, EDE2, EPD1, EPD2, EPD3, EPT1, EST1, EXD1, EXD2, MAR, MKT1, PVA1, PVA2, PVF, PVT, SDD1, SDD2, SDD3, SDD4, SDD5, SDD6, SFD1, SFD2, SFE1, SFE2, SFM1, SLI
Feed	МТСХ	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	DTG2
Tool	МТСХ	AEN, ATN, ATP1, ATP2, ATP3, ATR, ATU, DIS4, DTC1, DTC2, NTN, PTC1, PTC2, TDA1, TDA2, TDR1, TDR2, TIF, TII, TLB1, TLB2, TLD1, TLD2, TLD3, TLD4, TLE1, TLE2, TMV, TPI1, TPI2, TRM, TRS
	MWCX	AEN, ATN, ATP1, ATP2, ATP3, ATR, ATU, DIS4, DTC1, DTC2, NTN, PTC1, PTC2, TDA1, TDA2, TDR1, TDR2, TIF, TII, TLB1, TLB2, TLD1, TLD2, TLD3, TLD4, TLE1, TLE2, TMV, TPI1, TPI2, TRM, TRS
		TRM, TRS

Fig. 6-14: Logical conjunctions of FI commands



6.4 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 <u>cannot</u> 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. 6-15: 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 Components	NC Components	Communication configuration
MPCX	PC	None	None	None
MWCS	MTC200-P-G2 MTC200-R-G2	MTS-P	MTC-P	DPR, TCON
MSCX	SERCANS-A, SERCANS-P	None	None	V24 19200 Baud TCON
MWMX	VMISP200-P-G2, VMISP200-R-G2	MTS-P02.2	None	DPR, TCON
MWSX	ISP200-P-G2, ISP200-R-G2	MTS-P02.2	None	DPR, TCON
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. 6-16: 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. 6-17: Command execution times of the MPCX 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
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



Com.	Description	Example	[ms]
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
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



Com.	Description	Example	[ms]	
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	
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	SPS (PLC) 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		
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	



Com.	Description	Example	[ms]
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. 6-18: 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. 6-19: 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	SPS (PLC) Long Identification	00_BR_SLI	10

Fig. 6-20: Command execution times of the MWSX 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
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

Command execution times for the MWAX device group

Fig. 6-21: 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. 6-22: Command execution times of the MSYX device group



7 Function Interface Commands

7.1 FI Commands for the MPCX Device Group

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").

Interrupting Function Interface Jobs: BFJ

Designation	BFJ	Break-F	unction-Inte	rface 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	e interrupted with the	BFJ command!
FI command	Interrupt all FI jobs that are running.				
	XX_BR_E	3FJ1		(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 1n		Column 1	Column 2
Value Range/Meaning of Columns	1 =ID of job to be interrupted[0120]2 =FI command string.				
Example BFJ1	Interrupts	all FI jobs	that are run	ning.	
	<u>Note:</u> 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 comm	and	XX_BR_B	J1	
	Line	Column	Answer		
	1	1	01		
		2	02_BW_PA	A2_C:\DOWNLOAD1.	PDL /3
	2	1	02		
		2	01_BW_PA	A2_C:\DOWNLOAD2.	PDL/3V3
FI command	d Interrupt the selected FI job.				
	XX_BR_	BFJ2_(1)		(Single Rea	ld)
	(1) _ ID /	of ioh to he	interrupted	[0120]	



Response Structure		e response to the FI two columns.				
		Line 1		Column 1	Column 2	
Value Range/Meaning	1 = ID of job to be interrupted			[0120]		
of Columns	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

Outputting the Device Configuration: CCP

Designation	ССР	Cell Configura	ation P arameter			
Explanation	The configuration settings are read in from the "IND_DEV.INI" file. The configuration of the individual communication addresses and the settings of the various Rexroth Indramat devices are determined in this file (see Chapter 5 "Installation").					
FI command		the configuratic EV.INI" file.	on settings of	all devices c	lefined in the	
	XX_BR	_CCP1	(Single Read)			
Response Structure	comma	owing table shows nd "CCP1". The r onfigurable device	esponse consists	of a maximur		
		Line 1n:	Column 1		Column 15	
Value Range/Meaning	1 =	Device address	IND_DE\	/.INI entry: [Dev	viceAddrX]	
of Columns	2 =	Device name	IND_DE\	/.INI entry: Dev	iceName=	
	3 =	Device Type	IND_DE\	/.INI entry: Dev	iceTyp=	
	4 =	PLC support	IND_DE\	/.INI entry: PLC)=	
	5 =	Device status	IND_DE\	/.INI entry: Dev	iceStatus=	
	6 =	Assignment of a simulation pair	IND_DE\	/.INI entry: Dev	iceAssign=	
	7 =	Device mode	IND_DE\	/.INI entry: Mtv	ncMode=	
	8 =	Communication channel	IND_DE\	/.INI entry: : [Co	ommAddrX]	
	9 =	Description of the communication channel	e IND_DE\	/.INI entry: Con	nmStr=	
	10 =	Timeout value	IND_DE\	/.INI entry: Tim	eout=	
	11 =	Device group	(see Cha	pter 6.1 "Identil	fier")	
	12 =	PLC component		INI entry: Com	ponent type1=	
	13 =	PLC component		INI entry: Com		
	14 =	Device log	IND_DEV	INI entry: Devic	ceProtocol=	

- 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:

- Device address 00: SERCANS-A
- Device address 15: MTC200-P

FI comm	and	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	МТСХ
	12	MTS-P01.02
	13	MTC-P
	14	CNC
	15	OFF

FI command

Output the configuration settings of the selected device type. (Single Read)

(1)= Device type

BR_CCP2

[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 1n:		Column 1		Column 15
Value Range/Meaning	1 =	Device address	IND_DE	V.INI entry: [De	viceAddrX]
of Columns	2 =	Device name	IND_DE	V.INI entry: [De	viceName=
	3 =	Device Type	IND_DE	V.INI entry: [De	viceTyp=
	4 =	PLC support	IND_DE	V.INI entry: PLC	C=
	5 =	Device status	IND_DE	V.INI entry: Dev	viceStatus=
	6 =	Assignment of a simulation pair	IND_DE'	V.INI entry: Dev	viceAssign=
	7 =	Device mode	IND_DE	V.INI entry: Mtv	ncMode=
	8 =	Communication channel	IND_DE	V.INI entry: : [C	ommAddrX]
	9 =	Description of the communication char		V.INI entry: Cor	nmStr=
	10 =	Timeout value	IND_DE	V.INI entry: Tim	eout=
	11 =	Device group	(see Cha	apter 6.1 "Identi	fier")
	12 =	PLC component type	e IND_DE	/.INI entry: Com	ponent type1=
	13 =	CNC component typ	be IND_DE	/.INI entry: Com	ponent type2=
	14 =	Device log	IND_DE	/.INI entry: Devi	ceProtocol=
	15 =	Device simulation	IND_DE	/.INI entry: Devi	ceSimulation=

Example CCP2 Read the configuration settings of the defined devices of type SERCANS-A.

Assumption:

The following device types have been defined:

- Device address 00: SERCANS-A
- Device address 03: MTA200-P
- Device address 15: MTC200-P

FI comm	and	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



	FI comm	and	XX_E	BR_CC	P2_SERCA	NS-A	
	Line	Column	Ansv	ver			
		15	OFF				
FI command	Output the configuration data of the devices that are addressed via the stipulated communication channel.					dressed via the	
	BR_CCP	3_(1)			(Single F	Read)	
	(1) = Con	nmunication	chan	nel	IND_DE	/.INI entry: : [Co	ommAddrX]
Response Structure	comman		The re	espons	e consists	of a maximum	ponse to the FI n of n=16 lines
	L	ine 1n:		Co	umn 1		Column 15
Value Range/Meaning	1 =	Device addr	ess		IND_DEV	INI entry: [Devi	ceAddrX]
of Columns	2 =	Device nam	е		IND_DEV	INI entry: [Devi	ceName=
	3 =	Device Type	e		IND_DEV	INI entry: [Devi	ceTyp=
	4 =	PLC suppor	t		IND_DEV.INI entry: PLC=		
	5 =	Device statu	IS		IND_DEV.INI entry: DeviceStatus=		
		Assignment simulation p			IND_DEV.INI entry: DeviceAssign=		
	7 =	Device mode			IND_DEV.INI entry: MtvncMode=		
		Communica channel	ition		IND_DEV.INI entry: : [CommAddrX]		
		Description (communicat channel			IND_DEV	INI entry: Com	mStr=
	10 =	Timeout val	ue		IND_DEV	INI entry: Time	out=
	11 =	Device grou	p		(see Chap	oter 6.1 "Identifie	ər")
	12 =	PLC compo	nent t	type	IND_DEV.	INI entry: Comp	onent type1=
	13 =	CNC compo	onent	type	IND_DEV.	INI entry: Comp	onent type2=
	14 =	Device log			IND_DEV.	INI entry: Device	Protocol=
	15 =	Device simu	ulation	n	IND_DEV.	INI entry: Device	Simulation=
Example CCP3		e configura ication chan			f the dev	ices that are	addressed via
	Assumpt						
		wing device	•••				
	• Com	munication	chanr	nel 4: S	SERCANS-	A	

- Communication channel 5: MTA200-P
- Communication channel 1: MTC200-P

	FI comm	nand	XX_	BR_CCI	P3_1]
	Line	Column	Ans	wer			
	1	1	15				
		2	Trar	nsport un	it		
		3	MTC	200-P			
		4	YES	;			
		5	ON				
		6	NO				
		7	OFF	:			
		8	1				
		9	DPF	R,\$D000,	\$0000,\$2	000,RAM0,	TCON
		10	3500	0	· · · · ·		
		11	МТС	X			
		12	MTS	S-P01.2			
		13	МТС)-P			
		14	CNC)			
		15	OFF	:			
			<u> </u>				
FI command		ne configura d communic				ices that a	are addressed via the
	BR_CCI			(Single			
		vice group			-	MISX, MT	RX, MTAX]
				(see Cł	napter 6.	1 "Identifie	r")
Response Structure	comman		The r	espons	e consis	ts of a ma	he response to the FI aximum of n=16 lines
	Line 1	-		Colum			Column 15
Value Range/Meaning of Columns		Device add					y: [DeviceAddrX]
		Device nam	-				y: [DeviceName= y: [DeviceTyp=
		Device Type PLC support				EV.INI entr	
		Device state					y: DeviceStatus=
	6 =	Assignment simulation p	t of a				y: DeviceAssign=
		Device mod			IND_DE	EV.INI entr	y: MtvncMode=
		Communica channel	ation		IND_DE	EV.INI entr	y: : [CommAddrX]
		Description communica channel		9	IND_DE	EV.INI entr	y: CommStr=
	10 =	Timeout val	lue		IND_DE	EV.INI entr	y: Timeout=
	11 =	Device grou	цр		(see Ch	apter "Ider	ntifier")
	12 =	PLC compo	onent	type	IND_DE	V.INI entry	: Component type1=
	13 =	CNC compo	onent	type	IND_DE	V.INI entry	: Component type2=
	14 =	Device log				-	: DeviceProtocol=
	4 -	Devile e star	1-1-	-		1/1011	, Device Circulation

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:

- Device address 00: MSCX
- Device address 03: MTCX

	• Dev	ice address	\$ 03: MTCX				
	FI com	nand	XX_E	BR_CCP4_MSCX			
	Line	Column	Answ	/er			
	1	1	00				
		2	Press	sure barrel drive			
		3	SERG	CANS-A			
		4	NO				
		5	ON				
		6	NO				
		7	OFF				
		8	4				
		9	V24,0	V24,COM2,19200,EVEN,RS232,TCOFF			
		10	3500				
		11	MSC	MSCX			
		12	NON	E			
		13	NON				
		14	CNC				
		15	OFF				
			0.1				
Response Structure	The follo	vice addres	shows	(Single Read) [0063] the general struct sponse consists of			
	1		The rea	-		1	
		Line 1n:		Column 1	•••	Column 15	
/alue Range/Meaning	1 =	Device add	dress	IND_DEV	.INI entry: [Dev	iceAddrX]	
of Columns	2 =	Device nan	ne	IND_DEV.INI entry: [DeviceName=			
	3 =	Device Typ		IND_DEV	.INI entry: [Dev	iceTyp=	
	4 =	PLC suppo	ort	IND_DEV	IND_DEV.INI entry: PLC=		
	5 =	Device stat	tus	IND_DEV	IND_DEV.INI entry: DeviceStatus=		
	6 =	Assignmen simulation		IND_DEV	IND_DEV.INI entry: DeviceAssign=		
	7 =	Device mode		IND_DEV	IND_DEV.INI entry: MtvncMode=		
	8 =	Communic channel	ation	IND_DEV.INI entry: : [CommAddrX]			
	9 =	Description communica channel		IND_DEV	.INI entry: Com	mStr=	
	10 =	Timeout va	alue	IND DEV	.INI entry: Time	eout=	
	-	- ·		·			

12 = PLC component type IND_DEV.INI entry: Component type1=

11 = Device group

DOK-CONTRL-FUN*INT*V07-AW01-EN-P



- 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 device addresses have been defined:

Device address 00: MSCX

Device address 03: MTCX

FI comma	and	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

Reading the FI communication error counts: CEI

Designation	CEI	EI Communication Error Info			
FI command	Reading the	e counts for the	communication er	rors recorded ir	the protocol.
	BR_CEI1		(Single	Read)	
Response Structure	The following table shows the general structure of the response to the F command "CEI1". A line of 5 columns is output.				
		Line 1	Column 1		Column 5
Value Range/Meaning of Columns	1 = Error o PC side	counter:	Contains the com that time registere		r occurred until
	2 = Error o SIO side	counter:	Contains the con until that time reg		
	3 = Error o internal tir		Contains the interthat time – which FI, if applicable.		
	4 = Error of number of repeat tele	dispatches of	Contains the rep that time.	eat telegrams c	occurred until



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 comm	and	XX_BR_CEI1
Line	Column	Answer
1	1	1
	2	0
	3	0
	4	1
	5	0

Commands for Executing WIN32 Applications: CPR

Designation	CPR Create PRoces	S			
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	n 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	d 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.				
	<u>Assumption:</u> The VBDEMO.EXE D:\Programs\Indramat\Mtgu	program is in the subdirectory i\bin.			
	XX_BW_(FI command D:\Progra	CPR1_ ms\Indramat\Mtgui\bin\VBDEMO.EXE_1_1			
FI command	Execute a WIN32 application	n, that is <u>NOT</u> 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.
	<u>Assumption:</u> The program "TASKMGR.EXE" is in the subdirectory C:\Winnt\System32.
	FI command XX_BW_CPR2_C:\Winnt\System32\Taskmgr.exe_1

Removing Function Interface Jobs: DFJ

Designation	DFJ	Delete F	unction-Interf	ace 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 a	all FI jobs fro	om the manage	ement structure of the	e function interface.	
	XX_BR_C)FJ1	(\$	Single Read)		
Response Structure	command	1 "DFJ1". T	he response		e response to the FI imum of n=19 lines lumns.	
		Line 1n:		Column 1	Column 2	
Value Range/Meaning	1 = De	eleted job ID	D [01	.20]		
of Columns	2 = FI	command				
Example DFJ1	Delete all	FI jobs.				
	 <u>Assumption:</u> 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 comma	and	XX_BR_DFJ	1		
	Line	Column	Answer			
	1	1	01			
		2	02_BR_NCA	1_"D:\Download.ini" /3		
FI command	Remove function ir		ed FI job fro	om the manageme	ent structure of the	
	XX_BR_	DFJ2_(1)	(Single	Read)		
	(1) = Job	ID	[0120]		
Response Structure	The following table shows the general structure of the response to the I command "DFJ2". The response consists of one line with 13 columns.					
		Line 1		Column 1	Column 2	
Value Range/Meaning	1= De	eleted job ID	5	[0120]		
of Columns	2 = FI	command		[string, in accorda		



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 comma	and	XX_BR_DFJ2_01
Line	Column	Answer
1	1	01_BR_NCA1_"D:\Download.ini" /3

Deleting of the FI Command Stack Administration: DFS

MPCX Device Group

Designation	DFS	Delete IF	Command Sta	ack
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	BW_DFS	61		(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		XX_BW_DFS1 Value to be written: Reference information string	
	Line	Column	Answer	
	1	1	(P_ACK)	
Command for Terminat	ing WI	N32 App	lications:	DPR

Designation	DPR	Delete PRoc	ess
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) = Logl name	InIf Login	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	
------------	-----------------------	--

Error in mathematical expression.

Check mathematical expression.

Correct NC program and re-transmit

Error Information: ERI

MPCX Device Group

Designation	ERI	ERror Inf	formation	
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.			
	BR_ERI1_(1)_(2)		(S	ingle Read)
	(1) = Erro	r class	2	= NACK error number, = FI error code = reserved = Windows NT error code]
	(2) = Erro	r number	[L	ONG]
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 12			Column 1
Meaning of the Column	1 = Error text			[language-dependent]
	2 = Addit	ional text		[language-dependent]
Example ERI	Read the error text inc 26.		including the	e additional error text with error number
	FI command		XX_BR_ERI1_1_26	
	Line	Column	Answer	

Far Configuration Parameter: FCP

MPCX Device Group

Designation	FCP	Far Device Configuration Parameter
Explanation	PC. DifferePC is in	mand "FCP" returns the list of the addressable devices on the ntiation is made between two cases (A and B): the PC network and tand-alone.
Case A PC is in the PC Network	read (see	arDevices defined in the network configuration data on the PC is 'FAR_DEV.INI" file). In addition the local devices that are not FarDevices are output.
Case B PC stand-alone	The list of l apply:	ocal devices is outputted if one or more of the following points

1

1

1

2

	- Thora is no natwork our	figuration data a	n the DC (acc	
	 There is no network configuration data on the PC (see "FAR_DEV.INI" file). 			
	 The PC has been disab the "PC Network Active configurator. 		-	
	configurator.			
FI command	Read out the addressable of XX_BR_FCP1{_(1))	devices on the P0 (Single Read)	С.	
	(1) = Device selection	[L= only local, F	= only FAR] ! (Optional !
	Read-out of the addressa devices from the stipulated		the PC; but	only applies to
	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 addressab devices of the stipulated de		PC; however,	only applies to
	XX_BR_FCP3_(1){_(2)}	(Single Read)		
	(1)= Device type	[MTC200-P-G2 SERCANS-A, 3 ISP200-R-G2, MTA200-P]	SERCANS-P, I	ISP200-P-G2,
	(2) = Device selection	[L= only local, F	= only FAR] ! (Optional !
Response Structure	The following table shown "FCP1", "FCP2" and "FCP2" configuration.	3". The number of	of lines depend	
Response Structure	"FCP1", "FCP2" and "FCP2 configuration. Result when network config	3". The number of guration data is a	of lines depend	ds on the actual
Response Structure	"FCP1", "FCP2" and "FCP3 configuration.	3". The number of	of lines depend	
Value Range/Meaning of the	"FCP1", "FCP2" and "FCP2 configuration. Result when network config	3". The number of guration data is a	of lines depend vailable:	ds on the actual
	"FCP1", "FCP2" and "FCP2 configuration. Result when network config Line 1n:	3". The number of guration data is a Column 1	of lines depend vailable: 	ds on the actual
Value Range/Meaning of the	"FCP1", "FCP2" and "FCP2" configuration. Result when network config Line 1n: 1 = FarDevice address	3". The number of guration data is a Column 1 [0015]	of lines depend vailable: characters] MTC200-R-G2 ERCANS-P, IS	Column 10 Column 10
Value Range/Meaning of the	"FCP1", "FCP2" and "FCP3 configuration. Result when network config Line 1n: 1 = FarDevice address 2 = Device name	3". The number of guration data is a Column 1 [0015] [max. 28 ASCII [MTC200-P-G2, SERCANS-A, S ISP200-R-G2, T	of lines depend vailable: characters] MTC200-R-G2 ERCANS-P, IS	Column 10 Column 10
Value Range/Meaning of the	"FCP1", "FCP2" and "FCP3 configuration. Result when network config Line 1n: 1 = FarDevice address 2 = Device name 3 = Device type	3". The number of guration data is a Column 1 [0015] [max. 28 ASCII [MTC200-P-G2, SERCANS-A, S ISP200-R-G2, T MTA200-P]	of lines depend vailable: characters] MTC200-R-G2 ERCANS-P, IS	Column 10 Column 10
Value Range/Meaning of the	 "FCP1", "FCP2" and "FCP2" configuration. Result when network config Line 1n: 1 = FarDevice address 2 = Device name 3 = Device type 4 = Local device address 	3". The number of guration data is a Column 1 [0015] [max. 28 ASCII [MTC200-P-G2, SERCANS-A, S ISP200-R-G2, T MTA200-P] [0015]	of lines depend vailable: characters] MTC200-R-G2 ERCANS-P, IS	Column 10 Column 10
Value Range/Meaning of the	 "FCP1", "FCP2" and "FCP2" configuration. Result when network configuration. Line 1n: FarDevice address Device name Device type 4 = Local device address F PC No. 	3". The number of guration data is a Column 1 [0015] [max. 28 ASCII [MTC200-P-G2, SERCANS-A, S ISP200-R-G2, T MTA200-P] [0015] [0015, XX]	of lines depend vailable: characters] MTC200-R-G2 ERCANS-P, IS	Column 10 Column 10
Value Range/Meaning of the	 "FCP1", "FCP2" and "FCP2" configuration. Result when network configuration. Line 1n: 1 = FarDevice address 2 = Device name 3 = Device type 4 = Local device address 5 = PC No. 6 = Local device 	3". The number of guration data is a Column 1 [0015] [max. 28 ASCII [MTC200-P-G2, SERCANS-A, S ISP200-R-G2, T MTA200-P] [0015] [0015, XX] [YES, NO,]	of lines depend vailable: characters] MTC200-R-G2 ERCANS-P, IS	Column 10 Column 10
Value Range/Meaning of the	 "FCP1", "FCP2" and "FCP2" configuration. Result when network config Line 1n: 1 = FarDevice address 2 = Device name 3 = Device type 4 = Local device address 5 = PC No. 6 = Local device 7 = Device status 8 =Assignment of a 	3". The number of guration data is a Column 1 [0015] [max. 28 ASCII [MTC200-P-G2, SERCANS-A, S ISP200-R-G2, T MTA200-P] [0015] [0015] [0015, XX] [YES, NO,] ON, OFF	of lines depend vailable: characters] MTC200-R-G2 ERCANS-P, IS RA200-P, TRA	Column 10 Column 10
Value Range/Meaning of the	 "FCP1", "FCP2" and "FCP2" configuration. Result when network config Line 1n: 1 = FarDevice address 2 = Device name 3 = Device type 4 = Local device address 5 = PC No. 6 = Local device 7 = Device status 8 =Assignment of a simulation pair. 	3". The number of guration data is a Column 1 [0015] [max. 28 ASCII [MTC200-P-G2, SERCANS-A, S ISP200-R-G2, T MTA200-P] [0015] [0015, XX] [YES, NO,] ON, OFF [0015, NO]	of lines depend vailable: characters] MTC200-R-G2 ERCANS-P, IS RA200-P, TRA	Column 10 Column 10

Explanation of Column 7In case A, the "Disable" entry from the "FAR_DEV.INI" file is evaluated.Device StatusThe following assignment applies:



	OFF	if "Disable =		ntry is missing
	OFF if	f the PC is o	disabled. FarDevice, Disable = YES	FarDevice, Disable = NO
	PC, Disat	ole = YES	OFF	OFF
	PC, Disat	ole = NO	OFF	ON
	Note:		s disabled then its corresp ble" status.	oonding devices are also in
Explanation of Column 10 Online?	via which possible o • YES	the device cases: = The ne = The ne = The ı	can be addressed. Differe twork connection to the PC twork connection is down	
	Note:	YES is al	ways output for B.	
Example FCP1 Case A	"FAR_DE Assumpti The follow Devic	EV.INI" and on: wing device ce address ce address	 configuration of all "IND_DEV.INI" files. types have been defined: 15: MTCNC 11: MTVNC 12: MTVNC 	devices defined in the
	FI comm	and	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

FI comm	and	XX_BR_FCP1
Line	Column	Answer
	8	15
	9	МТСХ
	10	YES
3	1	12
	2	Drill right
	3	MTVNC
	4	02
	5	03
	6	NO
	7	OFF
	8	NO
	9	МТСХ
	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 Read the network configuration of all devices defined in the Case B "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 comma	and	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



FI comma	and	XX_BR_FCP1
Line	Column	Answer
2	1	01
	2	Drill left
	3	MTVNC
	4	01
	5	XX
	6	YES
	7	ON
	8	NO
	9	МТСХ
	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.

Far Device Configuration Parameter: FDC

Designation	FDC Far Device Configuration
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)

·	command "FDC1".	5		
	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	Line 1:			
of Columns	1 = PC network exists?		[YES, NO]	
	2 = Name of the PC netw	vork	[max. 28 ASC	II characters]
	3 = Max. number of PCs	(Integer)		
	4 = Max. number of device	(Integer)		
	Line 2:			
	1 = PC No.	[0015, XX]		
	Line 3:			
	1 = Hostname/ Ethernet possibly expanded by na	(string)		
	Line 4:			
	1 = Computer name/ NE 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 Ma (Head PC), NO	
Example FDC1	Read the general data of	the PC network.		
Case A	A			

The following table shows the general structure of the response to the FI **Response Structure**

Case A 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 comm	and	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 comma	and	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	

Further Info Text: FIT

Designation	FIT	Further I	Info T ext	
Explanation	Returns the additional text of an FI error code or a NACK error number.			
FI command	Read additional (further) text.			
	BR_FIT1	l_(1)_(2)	(S	ingle Read)
	(1) = Err	or class		= NACK error number, = FI error code]
	(2) = Err	or number	[L0	ONG]
Response Structure	One line	with one co	lumn is outpu	tted for the additional text.
		Line		Column
Meaning of the Column	Additior	nal text		[language-dependent]
Example FIT	Read the	e additional g	general error	text with the number 26.
	FI com	nand	XX_BR_FIT1	_1_26
	Line	Column	Answer	
	1	1		matical expression. rogram and re-transmit
	Note:	there is obtain the	an additiona e additional e	It line contains an "X" in Column 5 when I text, otherwise simply "". You can rror text by calling up the "XX_BR_FIT1" and 2 nd partial result.



Far PC Configuration Parameters: FPC

Designation FPC Far PC Configuration Parameter The FI-Command "FPC" outputs the list of PCs that are defined in the Explanation network. Differentiation is made between two cases (A and B): PC is in the PC network and PC Stand-Alone. Case A The list of PCs defined in the network configuration files on the PC (see PC is in the PC Network "FAR_DEV.INI" file) is outputted. The data of the local PC is outputted if one or more of the following points Fall B **PC Stand-Alone** 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 1 = PC No. [00...15, XX] Value Range/Meaning of the Columns 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] [OFF = PC is disabled, ON = PC is enabled] 5 = Device status 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** This column indicates whether there is currently a connection to the PC **Online?** 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.

MPCX Device Group



Example FPC1 Case A

Assumption: Two PCs are defined:

• PC1 with the IP address: 192.4.4.91

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

PC2 with the name: st100103

FI comma	and	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.

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

Example FPC1 Case B

<u>Assumption:</u> No PCs are defined:

FI comma	and	XX_BR_FPC1
Line	Column	Answer
1	1	XX
	2	
	3	
	4	YES
	5	ON
	6	
	7	YES



Writing/Reading the General FI Data Buffer: GDB

Designation	GDB	Global D	ata Buffer		
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.				
		<u>Note!</u> The buf applicatio		695 are available for external	
FI command	Write data	a into an Fl	data buffer.		
	BW_GD	B1_(1)	(Single Write)	
	(1) = Buf	fer ID	[686-695]		
	Value to	be written			
	Data to b	e transport	ed (max. 100 byte)		
Response Structure	(P_ACK)	is returned	following successfu	Il transmission.	
		Lin	e 1	Column 1	
Value Range/Meaning of Columns	1 = 5	Successfull	y completed (P	_ACK)	
	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.				
Example GDB1					
Example GDB1		value are		er ID 686.	
Example GDB1	as a write	value are	written with the Buff	er ID 686.	
Example GDB1	as a write	and	written with the Buff	er ID 686.	
Example GDB1 Fl command	as a write FI comma Line 1	and Column	written with the Buff XX_BW_GDB1_686 Answer	er ID 686.	
	as a write FI comma Line 1	and Column 1 a from an F	Written with the Buff XX_BW_GDB1_686 Answer (P_ACK)	er ID 686.	
	as a write FI comm Line 1 Read data	and Column 1 a from an F 31_(1)	Written with the Buff XX_BW_GDB1_686 Answer (P_ACK) I data buffer.	er ID 686.	
	as a write FI comma Line 1 Read data BR_GDE (1) = Buf The cont	and Column 1 a from an F 31_(1) fer ID	XX_BW_GDB1_686 Answer (P_ACK) I data buffer. (Single Write [686-695] e addressed FI da	er ID 686.	
FI command	as a write FI comma Line 1 Read data BR_GDE (1) = Buf The cont	and Column 1 a from an F 31_(1) fer ID ents of the	Written with the Buff XX_BW_GDB1_686 Answer (P_ACK) I data buffer. (Single Write [686-695] e addressed FI data sion.	er ID 686.	
FI command	as a write FI comma Line 1 Read data BR_GDE (1) = Buf The cont successfu 1 = Cont	and Column 1 a from an F 31_(1) fer ID ents of the ul transmiss Lin ontents of t	written with the Buff XX_BW_GDB1_686 Answer (P_ACK) I data buffer. (Single Write [686-695] e addressed FI da sion. e 1 he [Data	er ID 686.	
FI command Response Structure Value Range/Meaning	as a write FI comma Line 1 Read data BR_GDE (1) = Buf The cont successfu 1 = Ca	and Column 1 a from an F 31_(1) fer ID rents of the ul transmiss Lin ontents of t ddressed F	written with the Buff XX_BW_GDB1_686 Answer (P_ACK) I data buffer. (Single Write [686-695] e addressed FI da sion. e 1 he [Data	er ID 686.	
FI command Response Structure Value Range/Meaning of Columns	as a write FI comma Line 1 Read data BR_GDE (1) = Buf The cont successfu 1 = Ca	and Column 1 a from an F 31_(1) fer ID cents of the ul transmiss Lin ontents of t ddressed F general FI	xX_BW_GDB1_686 Answer (P_ACK) I data buffer. [686-695] e addressed FI data sion. e 1 he [Data I data buffer FI data buffer	er ID 686.	
FI command Response Structure Value Range/Meaning of Columns	as a write FI comma Line 1 Read data BR_GDE (1) = Buf The cont successfu 1 = Co ac Read the	and Column 1 a from an F 31_(1) fer ID cents of the ul transmiss Lin ontents of t ddressed F general FI	<pre>written with the Buff XX_BW_GDB1_686 Answer (P_ACK) I data buffer. (Single Write [686-695] e addressed FI data sion. e 1 he [Data I data buffer FI data data buffer using th </pre>	er ID 686.	



Initialization of a V24 Communication Address: ICA

Designation	ICA	Initializ	ation C ommuni	ication Address
Explanation	By means of this command, a defined communication address (that has been created by the system configurator – CommAddr entry in the configuration file IND_DEV.INI) is initialized with new parameters.			
FI command	BW_ICA	A1_(1)_(2)		(Single Write)
		Addr entry	munication add	
	specifica	ation (Com	string according ImAddr entry in ND_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 mes	ssage (P_ACK)	(P_ACK)
Example ICA1	The define		nunication add	lress 1 is initialized with the following
	COM-PC	ORT:	1	
	BAUD R	ATE:	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)	
Information regarding	Functio	on Inte	rface Jobs	S: IFJ MPCX Device Group

Designation	IFJ Information a	bout Function-Interf	ace 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	6.			
	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 1n:	Column 1		Column 16		

Value Range/Meaning	1 =	Job ID	[0120]	
of Columns	2 =	FI command	[string, in accordance to chapter entitled "Elements of the FI Command"]	
	3 =	Job classification	[1 = NC download, 2 = compile NC program package]	
	4 =	Job status	[RUN, READY, ERROR]	
	5 =	Number of error lines in the e	error info buffer	
	6 =	Max. processing time [ms] ur	ntil TIMEOUT	
	7 =	Start time of the job	[hh:mm:ss:ms]	
	8 =	Processing time up to now in	ms	
	9 =	Function interface connection	n (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		
	Note:	The results of the columns started.	depend on the FI job that has been	

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 comma	and	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	



	FI comm	and	XX_E	BR_IFJ1		
	Line	Column	Ansv	/er		
		14				
		15	0			
		16				
FI command	Return ir	formation re	egardi	ng the selected	and active FI jo	b.
	XX_BR_		-	Single Read)		
	(1) = Job	ID	[C	120]		
	Note:				ure of the respor described above	nse is available in e.
Reading of the FI Com	mand S	Stack Ac	lmin	istration: I		CX Device Group
Designation	IFS	IF Comm	nand \$	Stack Info		
Explanation				and, the curre ent can be read		status of the FI
FI command	BR_IFS	1			(Single Writ	e)
Response Structure	The resp columns.		e "IFS	1"I command o	consists of n lir	nes, each with 4
		Line 1n		Column 1		Column 4
Value Range/Meaning	1 =	IF command	d stack	index	[140]	
of Columns		IF command			[max. 500 characters	
			applic	ponds to LOGIN ation issued by t		
	4 =	Access cour	nter rea	ading	[LONG val	ue]
Example IFS1				the IF commar ntly in the mana		ement, with 3 FI
	FI comm	and	XX_E	R_IFS1		
	Line	Column	Ansv	ver		
	1	1	1			
		2	00_B	R_ASM2		
		3	VBD	MO.EXE		
		4	5467			
	2	1	2			
		2	00_B	R_AMM2_0		
		3	IND4	00T.EXE		
		4	4562	34		
	3	1	3			
		2	02_B	R_ASM2		
	I	L		-		



	FI comm	r	XX_BR_I	-S1		
	Line	Column	Answer			
		3	VBDEMO	.EXE		
		4	534892			
Reading of PC Date an	d PC Ti	ime: LD	т			
-					MPCX	Device Group
Designation	LDT	PC Loca	al Date Tim	ie		
FI command				nd time are rea is supplied.	ad. At the same	e time, the local
	BR_LDT	1		(Single	Read)	
Response Structure				general struct olumn is outp		oonse to the FI
		Line 1		Column 1		Column 3
Value Range/Meaning of Columns	1 = Inf date	ormation	on [Day.m	nonth.year]		
	2 = Inf time	ormation	on [hour:r	ninute:second]	
	3 = LON	G value	[intern	al LONG codir	ng]	
Example LDT11	Read the	current dat	te and time	e of the PC.		
	FI comm	and	XX_BR_L	DT1		
	Line	Column	Answer			
	1	1	25.04.200	2		
		2	07:26:06			
		3	7192241			
Explanation	This FI command sets PC date and time.					
FI command	BW_LDT	1		(Single	Write)	
Value to be written	Date and written	d time inforr	mation to b	· · · · · ·	nth.year nute:second]	
	Note:				d to the "acVa ansfer" routine.	lue" parameter
Response Structure				general struct of 1 column is		oonse to the FI
		Line 1			Column 1	
Value Range/Meaning of Columns	1 = Statu	is message	9	(P_ACK)		
Example LDT1		C clock to t ue: 25.04.2		5.04.2002 07:: 33	31:33.	



FI comma	and	XX_BW_LDT1
Line	Column	Answer
1	1	(P_ACK)

Activated Language of the Rexroth Indramat GUI: LNG

MPCX Device Group

Designation	LNG Acti	vated LaNGuage		
Explanation	The country cod output.	e of the activated language for the Rexroth Indramat GUI is		
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 cod	e 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 I NG	Read the count	ry code of the activated language in the Revroth Indramat		

Example LNG

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

FI comma	and	XX_BR_LNG
Line	Column	Answer
1	1	SE

Read System Messages: MSG

Designation	MSG	MeSsaGe
Explanation	Reading o	of system messages
FI command	Message CC_MSC (1) = SYS	G_(1) (Cyclic Read) S-Message number
	Note:	Exists only as a cyclic command
Response Structure	The respo data.	onse of the FI command 'MSG' consists of the system message



Example MSG

(64 = MSG_SYSERRGEN)

FI comma	and	00_CC_MSG_64/3
Line	Column	Answer
1	1	00

The following system messages: Limitation

00_CC_MSG_64

SYS Message

SYS Message	SYS Message number
MSG_PCLUPDBEG	52
MSG_PARUPDBEG	24
MSG_FWAUPDBEG	82

These commands cannot be used with the following programs:

- Indramat OPC server
- Indramat DDE server •

NT Shutdown Functions: NST

Designation	NST NT-ShuT-Down					
Explanation	This allows the	ne NT ope	erating system to be shut down.			
FI command	Triggers NT-Shut-Down (WITHOUT shutdown boxes).BW_NST1(Single Write)					
Response Structure	As this conce	erns a cor	mmand there is no response data.			
Example NST1	Triggers NT	-Shut-Dov	wn (WITHOUT shutdown boxes).			
	FI command	ł	XX_BW_NST1			
FI command	Trigger NT-S	Shut-Dow	n (WITH shutdown boxes).			
	BW_NST2_	<u>(</u> 1)	(Single Write)			
	(1) = Time in second		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	k	XX_BW_NST2_30			

Formatting a Parameter Download Data File: PAF

Designation	PAF PArameter File 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. Rexroth Indramat'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.						
	BW_PAF	1_(1)_(2)			(Single Write)		
	(1) = Cor name un		imeter downloa	[input must be available	file]		
	(2) = Complete parameter download file name sorted				[output is generated.	file]	
Desman e Otausture	Note:	no write v paramete	value is passe r download for	d, this co mat.	be passed as a write ommand requires the	MTCNC	
Response Structure					ture of the response to the response to the section of the section		
		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 comma	and	XX_BW_PAF1 PARDATA.SO		/IP\PARDATA.DAT"_"C	:\TEMP\	
	Line	Column	Answer				
	1	1	(P_ACK)				



Generating Physical Directory Names: PHD

Designation	PHD	PH ysical	Directory			
Explanation	Generate	s physical o	directory names a	according to the BDI data written.		
	Note:	This is ba	ised on BDI philo	sophy.		
FI command	Generate	physical di	rectory names.			
	BR_PHD	01_(1)_(2)_	(3)_(4)_(5)_(6)	(Single Write)		
	(1) = Pro			[-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) = Dev	vice addres	S	[-1= DEVADDR_NEUTRAL otherwise the required device address]		
	(4) = Pro	cess ID		[-1= PROCESS_NEUTRAL otherwise the required process number]		
	(5) =Data	a type ID		[possible write values see BDI documentation (BDI_DEFINITIONS.H)]		
	(6) = Lar	nguage ID		[possible write values see BDI documentation (WINNT.H)]		
Response Structure	The following table shows the general s command "PHD1".			al structure of the response to the FI		
		Lin	e 1	Column 1		
Value Range/Meaning of Columns	1 = 1	Physical dir	ectory 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 comm	and	XX_BR_PHD1	1_011_0_0		
	Line	Column	Answer			
	1	1	D:\Programme\Ir	ndramat\Mtgui\Bin		



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

Designation	РОВ	POrt Byt	e Access						
FI command	This comr	mand is use	ed for writing	a PC p	ort address (b	yte acces	s).		
	BW_PO	31_(1)_(2)			(Single Rea	d)			
	(1) = req	uested PC	port addres	5	Declaration address	format:	0x	port	
	(2) = PC	port value	port value to be written			format:	0x	port	
Response Structure	The following table shows the general structure of the response to the command "POB1". A line of 1 column is output.					the FI			
		Line 1			Column 1				
Value Range/Meaning of Columns	1 = Status message (P_AC			(P_AC	K)				
Example POW1	Write the	value 0x00	00 into the I	PC port	address 0x31	C.			
	FI comma	and	00_BW_PC	W1_0x3	0x31C_0x0000				
	Line	Column	Answer						
	1	1	(P_ACK)						
FI command	This comr	mand is use	d for reading	g a PC	port address (byte acces	ss).		
	BR_POE	B1_(1)			(Single Read)			
	(1) = req	uested PC	port addres		Declaration address	format:	0x	port	
Response Structure			shows the g line of 1 cc		structure of th output.	e respons	se to	the FI	
		Line 1			Colu	mn 1			
Value Range/Meaning of Columns	1= PC pc	ort value rea	ad						
Example POB1	Read the	PC port ad	dress 0x310	C.					
	FI comma	and	00_BR_PO	B1_0x3 ⁻	1C				
	Line	Column	Answer						
	1	1	0x00						



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

Designation	POW	POrt Wo	rd Access					
FI command	This comr	nand is use	d for writing	aPC	oort address (w	vord acces	ss).	
	BW_PO	N1_(1)_(2)			(Single Rea	d)		
	(1) = req	uested PC	port addres	S	Declaration address	format:	0x	port
	(2) = PC	port value to be written			Declaration value	format:	0x	port
Response Structure	The following table shows the general structure of the response to the command "POW1". A line of 1 column is output.					the FI		
		Line 1		Column 1				
Value Range/Meaning of Columns	1 = Status	s message ((P_ACK)	(P_A0	CK)			
Example POW1	Write the	value 0x00	00 into the l	PC por	t address 0x3 ²	1C.		
	FI comma	and	00_BW_PC)W1_0x	31C_0x0000			
	Line	Column	Answer					
	1	1	(P_ACK)					
FI command	This comr	nand is use	d for readin	g a PC	port address (word acce	ess).	
	BR_POV	V1_(1)			(Single Read)		
		= requested PC port address Declaration format: 0x address						
	(1) = req	uested PC	port addres	S		format:	0x	port
Response Structure	The follow	ving table s		eneral	address structure of th		-	•
Response Structure	The follow	ving table s	hows the g	eneral	address structure of th	e respons	-	•
	The follow	ving table s I "POW1". <i>I</i>	hows the g A line of 1 c	eneral	address structure of th is output.	e respons	-	•
Value Range/Meaning	The follow command 	ving table s I "POW1". <i>A</i> Line 1 ort value rea	hows the g A line of 1 c	eneral olumn	address structure of th is output.	e respons	-	•
Value Range/Meaning of Columns	The follow command 	ving table s I "POW1". <i>A</i> Line 1 ort value rea PC port add	hows the g A line of 1 c ad	eneral olumn	address structure of th is output. Colu	e respons	-	•
Value Range/Meaning of Columns	The follow command 1= PC pc Read the	ving table s I "POW1". <i>A</i> Line 1 ort value rea PC port add	hows the g A line of 1 c ad dress 0x310	eneral olumn	address structure of th is output. Colu	e respons	-	•



Rexroth
Indramat

Ready Message for a WIN32 Application: RP

MPCX Device Group R

Designation	RPR	Ready PRoc	cess
Explanation			I, WIN32 applications logged in the FI can inform at they are ready for operation.
			N32 application was generated by means of the FI PR1_(1)_(2)_(3)".
FI command	Inform the operation.	initiating prog	gram that the WIN32 program invoked is ready for
	XX_BW_	RPR1	(Single Write)

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

Triggering an FI Device Polling Cycle: SDP

Designation	SDP	Start Device Polling				
Explanation	This FI command triggers an FI device polling cycle.					
FI command	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_SD			21		
	Line	Column	Answer			
	1	1	(P_ACK)			

Focusing Commands: SFW

Designation	SFW Set Focus to Window						
Explanation	The screen	i can be fo	cused wit	h these FI commands.			
FI command	Focus the screen on the DOS-BOF user interface. XX_BW_SFW1 (Single Write)						
Response Structure	As this cond	cerns a co	mmand th	ere is no response data.			
Example SFW1	Focus on the DOD-BOF user interface screen. <u>Assumption:</u> 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 commar	nd	XX_BW_S	SFW1			
FI command	Focus the the FI via L		arget wind	low) of a WIN32 application connected to			
	XX_BW_S	6FW2_(1)_	_(2)_(3)	(Single Write)			
	(1) = LogIr	nlf Login n	ames	This refers to the login name entered at the FI (LogInIf()) during the login procedure.			
	(2) = Min I			(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			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 cond	cerns a co	mmand th	ere is no response data.			
Example SFW2	name "VBI	DEMO.EX	E". The c sing takes	n that has logged on in the FI with the login current screen window (output window) is place automatically at the end of the WIN32			
	El comman	hd	XX BW 9				

FI command	XX_BW_SFW2_VBDEMO.EXE_1_1
------------	---------------------------



Issuing a SYS Message: SSM

Designation	SSM	SM Set Sys Message						
Explanation	This allo	ws SYS message	s to be	issued.				
	Note:	The SYS mes	sage h	andling of t	he FI MUST BE	known!		
FI command		ows SYS messag d from the WIN32 e.						
		litional informatior ted simultaneousl			ength of 200 ch	aracters can be		
	BW_S	SM1_(1)_(2)	, (Sing	le Write)				
		YS message r (ALWAYS an umber)	Value range: 24000 <u>Note:</u> The SYS message number is ALWAYS even, while the acknowledgement number associated with it is always an odd number!					
	(2) = in acknow in msee	but Input acknowledgement time – the WIN32 applications that want to receive the SYS				the SYS		
		o be written nce information						
Response Structure		owing table show nd "SSM1".	s the g	eneral strue	cture of the res	ponse to the FI		
		Line 1	Co	olumn 1		Column 8		
Value Range/Meaning of Columns	1 =	Status report		correctly a application [ERROR= acknowled	SYS message h cknowledged b Is] SYS message l Iged by a WIN3 pre-set time]	y the WIN32 has NOT been		
	2 =	Task name (LogInIf name)		[Task name that has triggered the SYS message]				
	3 =	SYS message number		[contains the issued SYS message number]				
	4 =	Acknowledgeme time	ent	[contains t time]	he pre-set ackr	nowledgement		
	5 =	Reference inforr	nation		where applicab information trar 2]			
	6 =	Length of addition	onal	[0 where N been trans	IO additional in ferred]	formation has		
	7 =	Where applicabl LOG channel of that has NOT acknowledged		completed number of	wledgements h in time or the L the WIN32 app acknowledged i	OG channel		
	8 =	Where applicabl task name that h			wledgements h in time or the t			

NOT acknowledged in has NOT acknowledged in time] 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		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.

BW_SSM2_(1) (Single Write)

(1) = SYS message number (ALWAYS an even number)	Value range: 24000 <u>Note:</u> 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] [Task name that has triggered the SYS message]		as been
	2 =	Task name (LogInIf name)				ered the SYS
	3 =	SYS message number	SYS message number] ement [0] formation [contains, where applicable, the additional information transferred write value]			
	4 =	Acknowledgemen time				
	5 =	Reference inform				
	6 =	Length of additior information				ormation has
	7 =	Where applicable LOG channel of th 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		XX_BW_SSM2_3302
Line	Column	Answer
1	1	READY
	2	VBDEMO.EXE
	3	3302
	4	0
	5	00
	6	2
	7	
	8	



7.2 FI Commands for the MTCX Device Group

Com. Description	
------------------	--

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	[0063]

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 6.1 "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 "Overview of the FI Commands", "Overview of the MTCX Device Group" contains a summary of the possible FI commands for the MTCX device group.



7.3 FI Commands for the MWCX Device Group

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	[0063]
NI - 4 -	Discourse state that the state is in a laboration of	

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 "Elements of the FI Command").

Active Acceleration Value: AAC

MWCX device group

- DesignationAACActive ACcelerationExplanationThe 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	[06]

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 2 = Acceleration value 3 = Unit	[ACC] [010 [%]	•	

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].



Active Angle Dimension (RAD/DEG): AAD

MWCX device group

Designation	AAD	AAD Active Angle Dimension			
Explanation	of the an functions	the active angle dimension of an NC process is read out. The arguments the angle functions SIN, COS, TAN and the results of the inverse inctions of the angle functions ASIN, ACOS, ATAN can be specified or lculated 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)	(Sin	ngle Read)	
	CC_AAD	_(1)	(Cyd	clic Read)	
	CB_AAD	_(1)	(Bre	(Break Cyclic Read)	
	(1) = NC	process nui	mber [0(6]	
Response Structure	The response to the FI command "AAD" consists of one line with one column for the unit [RAD/DEG].			I "AAD" consists of one line with one	
		Line 1		Column 1	
Example AAD	Reads the active angle dimension in NC process 0 of device address 00.				
	FI command 00_CR_A		00_CR_AAD_0	_AAD_0	
	Line	Column Answer			
	1	1	RAD		
Reference to Literature	See chap	ter entitled	"Literature" [2].		

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

Designation	AAS Actual Axis Speed			
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 p	rocess number	[06]	
	(2) = Axis	meaning	[011] (see Chapter "Data Tables")	
	(3) = Rec (opt.)	uired measurement system	[mm, inch]	

FI command	Output th		ot ovio	anaad	of the	alastad davi	ion related to the	
Fi command	physical a			speed (Selected devi	ice related to the	
	conversio	on of the	result ir	nto mm/m	nin or in	ch/min. If, how	ble to pre-select wever, a spindle is ystem serves no	
	CR_AAS	62_(1){_(2)}			(Single Read))	
	CC_AAS	CC_AAS2_(1){_(2)}				(Cyclic Read))	
	CB_AAS	62_(1){_(2	2)}			(Break Cyclic	: Read)	
	(1) = Phy	/sical axis	s numb	er		[132, accord the system pa	ding to settings of rameters]	
	(2) = Re (opt.)	equired r	neasur	ement sy	ystem [[mm, inch]		
Response Structure	The following table shows the general structure of the response to the command "AASx". One line is output with 4 columns for the ax designation, axis speed, unit and the axis speed limited to "indicate decimal places".				mns for the axis			
	Line	e 1	Colu	mn 1	Column	2 Column	n 3 Column 4	
Meaning of the Columns	1 = Axis	name		[accordi	ng to se	ettings of axis	parameters]	
	2 = Spee	ed		[accordi	ng to se	ettings of axis	parameters]	
	3 = Unit	3 = Unit [according to			-	settings of process parameters 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 []. Reads the current axis speed of the Z axis in NC process of device							
Example AAS1		column e curren	s is []					
Example AAS1	Reads th	column le curren)0.	t axis		f the Z			
Example AAS1	Reads th address 0	column le curren)0.	t axis	speed of CR_AAS1	f the Z			
Example AAS1	Reads th address 0	column le curren)0.	is is [] t axis : 00 _0	speed of CR_AAS1	f the Z _0_2 Answer			
Example AAS1	Reads th address (FI comma	column e curren)0. and	ns is [] t axis = 00_0 nn 1	speed of CR_AAS1 A	f the Z _0_2 Answer nn 2	axis in NC	process of device	
Example AAS1 Example AAS1	Reads th address 0 FI comma Line 1	column e curren 00. and Colur Z1 e curren	t axis : 00_0 nn 1	speed of CR_AAS1 A Colur -1.23 speed of	f the Z _0_2 mswer nn 2 345 f the Z	axis in NC Column 3 [mm/min]	process of device	
	Reads th address 0 FI comma Line 1 Reads th	column e curren 00. and Colur Z1 e curren 00. Outpu	t axis : 00_0 nn 1 t axis : t of valu	speed of CR_AAS1 A Colur -1.23 speed of	f the Z 0_2 nnswer nn 2 345 f the Z ch/min.	axis in NC Column 3 [mm/min] axis in NC	Column 4 -1.235	
	Reads th address 0 FI comma Line 1 Reads th address 0	column e curren 00. and Colur Z1 e curren 00. Outpu	t axis : 00_0 nn 1 t axis : t of valu	Speed of CR_AAS1 A Colur -1.23 Speed of ues in inc CR_AAS1	f the Z 0_2 nnswer nn 2 345 f the Z ch/min.	axis in NC Column 3 [mm/min] axis in NC	Column 4 -1.235	
	Reads th address 0 FI comma Line 1 Reads th address 0	column e curren 00. and Colur Z1 e curren 00. Outpu	t axis = 00_0 nn 1 t axis = t of valu	Speed of CR_AAS1 A Colur -1.23 Speed of ues in inc CR_AAS1	f the Z 0_2 nnswer nn 2 345 f the Z ch/min. 0_2_in nswer	axis in NC Column 3 [mm/min] axis in NC	Column 4 -1.235	
	Reads th address (FI comma Line 1 Reads th address (FI comma	column e curren 00. and Colur Z1 e curren 00. Outpu and	t axis = 00_0 nn 1 t axis = t of valu 00_0	Speed of CR_AAS1 A Colur -1.23 Speed of ues in inc CR_AAS1 A	f the Z _0_2 mswer mn 2 345 f the Z ch/min. _0_2_in mswer mn 2	axis in NC Column 3 [mm/min] axis in NC ch	process of device Column 4 -1.235 process of device	
	Reads th address 0 FI comma Line 1 Reads th address 0 FI comma Line 1	column e curren 00. and Colum e curren 00. Outpu and Colum and Colum z1	t axis = 00_0 nn 1 t axis = t axis = 00_0	Speed of CR_AAS1 A Colur -1.23 Speed of ues in inc CR_AAS1 A Colur -0.04	f the Z 0_2 Answer nn 2 345 f the Z ch/min. 0_2_in Answer nn 2 486	axis in NC Column 3 [mm/min] axis in NC ch Column 3 [inch/min]	process of device Column 4 -1.235 process of device Column 4	
Example AAS1	Reads th address 0 FI comma 1 Reads th address 0 FI comma Line 1 Reads the	column e curren 00. and Colum 21 e curren 00. Outpu and Colum 21 e current 00.	s is [] t axis = 00_0 nn 1 t axis = t of valu 00_0	Speed of CR_AAS1 A Colur -1.23 Speed of ues in inc CR_AAS1 A Colur -0.04	f the Z _0_2 mswer mn 2 345 f the Z ch/min. _0_2_in mswer mn 2 486 S (e.g.,	axis in NC Column 3 [mm/min] axis in NC ch Column 3 [inch/min]	process of device Column 4 -1.235 process of device Column 4 -0.049	
Example AAS1	Reads th address 0 FI comma 1 Reads th address 0 FI comma Line 1 Reads the address 0	column e curren 00. and Colum 21 e curren 00. Outpu and Colum 21 e current 00.	s is [] t axis = 00_0 nn 1 t axis = t of valu 00_0	Speed of CR_AAS1 A Colur -1.23 Speed of ues in inc CR_AAS1 A Colur -0.04 f spindle	f the Z _0_2 mswer mn 2 345 f the Z ch/min. _0_2_in mswer mn 2 486 S (e.g.,	axis in NC Column 3 [mm/min] axis in NC ch Column 3 [inch/min]	process of device Column 4 -1.235 process of device Column 4 -0.049	
Example AAS1	Reads th address 0 FI comma 1 Reads th address 0 FI comma Line 1 Reads the address 0	column e curren 00. and Colum 21 e curren 00. Outpu and Colum 21 e current 00.	s is [] t axis = 00_0 nn 1 t axis = t of valu 00_0 nn 1 speed o	Speed of CR_AAS1 A Colur -1.23 Speed of ues in inc CR_AAS1 A Colur -0.04 f spindle	f the Z _0_2 nn 2 345 f the Z ch/min. _0_2_in nn 2 486 S (e.g., 2_4 nswer	axis in NC Column 3 [mm/min] axis in NC ch Column 3 [inch/min]	process of device Column 4 -1.235 process of device Column 4 -0.049	
Example AAS1	Reads th address 0 FI comma Line 1 Reads th address 0 FI comma Address 0 FI comma	column e curren 00. and Colum 21 e curren 00. Outpu and Colum 21 e current s 0. and	s is [] t axis = 00_0 nn 1 t axis = t of valu 00_0 nn 1 speed o 00_0	Speed of CR_AAS1 A Colur -1.23 Speed of ues in inc CR_AAS1 A Colur -0.04 f spindle CR_AAS2 A	f the Z _0_2 mswer nn 2 345 f the Z ch/min. _0_2_in mswer nn 2 486 S (e.g., 2_4 mswer nn 2	axis in NC Column 3 [mm/min] axis in NC ch Column 3 [inch/min] physical axis r	process of device Column 4 -1.235 process of device Column 4 -0.049 number 4) of device	

MWCX device group

Active NC Block: ABI

				5 1			
Designation	ABI	Active NC-Block Information					
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 nui	mber	[06]			
	(2) = Num	nber of prev	vious NC blocks	[14] ! Optional !			
	(3) = Num	nber of follo	wing NC blocks	[14] ! Optional !			
Response Structure	NC recor	current N	C record is output. (1n = 9) in the res	are not specified then only the sponse depends on the number of sists of a column containing the			
Example ABI	Note: If there is no valid NC program in the device then the value of all columns is []. Reads the active NC record and the two previous and two following NC						
			ss 0 of device addre				
	FI comma	and	00_BR_ABI_0_2_2				
	Line	Column	Answer				
	1	1	N0000 .START				
	2	1	N0001 T13 BSR .M6	6			
	3	1	N0002 G90 G41 G5	4 G17 F2000.0 S3200.00 M003			
	4	1	N0003 G00 X 60.00	00 Y -30.0000			
	5	1	N0004 Z -6.0000				

Reference to Literature

See chapter entitled "Literature" [4].



Active Cutting Speed of the Reference Spindle: ACS

MWCX device group

Designation	ACS Active Cutting Speed						
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)		(Sin	gle Read	l)	
	CC_ACS_	(1)		(Cyc	clic Read)	
	CB_ACS_	(1)		(Bre	ak Cycli	c Read)	
	(1) = NC p	rocess nur	nber	[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.					or the S number	
		Line 1		Column 1			Column 3
Value Range/Meaning	1 = S number of reference spindle			pindle	S1, S2,	S3	
of Columns	2 = Cutting speed			[format according to settings of the parameters]			
	3 = Unit				-	ng to setting parameters	5
	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 cuttin	ng speed	in NC pro	cess 0 of	device addre	ess 00.
	FI comma	nd	00_CR_	ACS_0			
				Ansv	wer		

Reference to Literature	5

See chapter entitled "Literature" [5].

Line

1

Active D-Correction Number: ADN

MWCX device group

Column 3

[m/min]

Designation	ADN Active D-Correctio	n 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) CC_ADN1_(1) CB_ADN1_(1) (1) = NC process number	(Single Read) (Cyclic Read) (Break Cyclic Read) [06]					

Column 1

S1

Column 2

200



Response StructureOne 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 R

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].

Active Event Monitoring: AEM

Designation	AEM	Active Event Monitoring				
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				t monitoring of an NC process of the device group.		
	CR_AEM	_(1)		(Single Read)		
	CC_AEM	_(1)		(Cyclic Read)		
	CB_AEM	_(1)		(Break Cyclic Read)		
	(1) = NC	process nur	mber	[06]		
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			as follows:		
			of event mor	0		
Example AEM	Read the status of the event monitoring of NC process 0 of device address 00.					
	FI comma	and	00_CR_AE	И_0		
	Line	Column	Answer			
	1	1	EEV			
Reference to Literature	See chap	ter entitled	"Literature"	[7].		

Active Edge Number: AEN

MWCX device group

Designation	AEN	Active E	Active Edge Number					
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			dge number of an NC process of the selected device vice group.					
	CR_AEN	_(1)	(Single Read)					
	CC_AEN	_(1)	(Cyclic Read)					
	CB_AEN	_(1)	(Break Cyclic Read)					
	(1) = NC	process nu	umber [06]					
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 [19] that is assigned the address letter "E".							
Example AEN	Read the active edge number of NC process 0 of device address 00.							
	FI comma	and	00_CR_AEN_0					
	Line	Column	Answer					
	1	1	E					
			1					

Reference to Literature See chapter entitled "Literature" [8].

2

1

Active Feedrate Override: AFO

Designation	AFO	Active Feedrate Override						
Explanation	Override is	t value of the feedrate override of an NC process is outputted. interpreted in the NC, irrespective of the mode; it has an effect 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_(1) (Single Read)						
	CC_AFO1	_(1)	(Cyclic Read)					
	CB_AFO1	_(1)	(Break Cyclic Read)					
	(1) = NC p	rocess number	[06]					



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 2 = Current value of the feedrate of 3 = Unit 	verride	[OVR=0 [0255 [%]	Override] 5]

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].

Actual (Current) Feedrate: AFR

Designation	AFR Actual Feed Rate						
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		current value the MWCX			of an N	C process o	of the selected
	-	optional se	•			possible	to pre-select
	CR_AFR_	(1){_(2)}			(Single	e Read)	
	CC_AFR_	(1){_(2)}			(Cyclic Read)		
	CB_AFR_	(1){_(2)}			(Break Cyclic Read)		
	(1) = NC F	rocess numb	ber		[06]		
	(2) = Rec (opt.)	equired measurement system			[mm, ir	nch]	
Response Structure	command	•	ine with	three col	umns i	•	onse to the FI the identifier,
		Line 1		Colum	าท 1		Column 3
Value Range/Meaning	1 = Identifi	er	[F = fee	edrate]			
of Columns	2 = Value		[format	accordin	g to set	tings of the	parameters]
	3 = Unit						S



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" [!)].

Active G Functions: AGF

Designation	AGF	Active G	Function		
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 num		mber [06]		
	(2) = G code group		[121] ! Optional !		
	Note:	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 <u>not</u> 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 comma	Fl command 00_CR_AGF_0_17			
	Line	Column	Answer		
	1	1	G30		
Reference to Literature	See chapter entitled "Literature" [11].				



Active M Functions: AMF

MWCX device group

Designation	AMF	Active M	Function		
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)}	_(1){_(2)} (Single Read)		
	CC_AMF	_(1){_(2)}	(Cyclic Read)		
	CB_AMF	_(1){_(2)}	(Break Cyclic F	Read)	
	(1) = NC	process nu	nber [06]		
	(2) = M fu	Inction Gro	on Group [116] ! 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 <u>not</u> 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 comma	Fl command 00_CR_AMF_0_2			
	Line	Column	Answer		
	1	1	M005		

Reference to Literature See chapter entitled "Literature" [12].

Active Mechanism Messages: AMM

MWCX device group

Designation AMM Active Mechanism Messages 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:	device gi		e, any	valid device a	ices within this address can be MM1).	
Response Structure	command	following table shows the general structure of the response to the F mand "AMM1". The response consists of a maximum of n=512 line 6 devices x 32 mechanisms = 512), each with 12 columns.				n of n=512 lines	
	L	ine 1n	Colum	n 1		Column 12	
Value Range/Meaning	1 = I	Device address		[000	[0063]		
of Columns	2 = 1	Device nam	е	- [max.	32 ASCII char	acters]	
	3 = 1	Mechanism	number	[03	1]	-	
	4 = 1	Mechanism	name	[max.	28 ASCII char	acters]	
	5 =	Type of mes	ssage	[F = f	ault/error, D = o	diagnosis]	
	6 = 1	Message so	ource	[CNC	, PLC]		
	7 =	Type of mee	ssage (2)		Status, O = Ope External, I = Inte		
	8 = 1	Message nu	ımber	[06	[00		
	9 = 1	Message te	xt	[max. 54 ASCII characters]			
	10 = I	Reference t	ext	[x= exists, = does not exist]			
	i	2 bytes of a nformation for the mess	dditional sage number	is required to resolve the information "@" (see AMM5)			
	i	Filename fo nformation text	r additional for message	e.g. ir	HTML format		
Example AMM1	<u>Assumpti</u>	on:	chanism messa addresses and	-			
		0					
	 Device address 01 with 2 mechanisms 0 and 1, and Device address 03 with one mechanism 0. 						
	FI comm	and	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 u	until tool	-change comma	nd has ended.	
		H	l				

10

11 12 x 0

	FI comm					
	Line	Column	03_BR_AI Answer			
	2	1	01			
		2	Drill center	r		
		3	1	·		
		4	Station 2			
		5	F			
		6	CNC			
		7	0			
		8	1			
		9	No externa	al 24V supply.		
		10	x	,		
		11	0			
		12				
	3	1	03			
		2	Milling cen	iter		
		3	0			
		4	Camshaft 30.40.25.0S			
		5	D			
		6	CNC			
		7	S:			
		8	71			
		9	Circular in	terpolation		
		10	х			
		11	0			
		12				
FI command	Output tl device.	he currentl	y pending	mechanisn	n messages c	of the selected
	BR_AMM	M2	(Sin	gle Read)		
	BC_AMM	N 2	(Cyc	lic Read)		
	BB_AMM	/ 12	(Bre	ak Cyclic R	ead)	
Response Structure	command	d "AMM2". `	The respo	general stru nse consists	cture of the res of up to a ma	ponse to the FI ximum of n=31
	lines, eac					
	-	_ine 1n	C	olumn 1		Column 12
Value Range/Meaning	L	ine 1n				Column 12
Value Range/Meaning of Columns	L 1 = 1		ress	[006	3]	
	1 = 1 2 = 1	₋ine 1n Device addi	ress	[006	3] 6CII characters]	

- [F = fault/error, D = diagnosis] Type of message Message source
- [CNC, PLC] Type of message (2)
 - [S = Status, O = Operator, E = External, I = Internal]

5 =

6 =

7 =

8 =	Message number	[0600]
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	e.g. in HTML format

Example AMM2 Reads the current mechanism messages of device address 01. <u>Assumption:</u>

text

Device address 01 with 2 defined mechanisms 0 and 1.

FI comma	and	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	0
	8	1
	9	No external 24V supply.
	10	x
	11	0
	12	

Reference to Literature

See chapter entitled "Literature" [13].

FI command				ing message /CX device g			ns listed for the
	BR_AM			0	(Single Read)		
	BC_AMM3_(1)					(Cyclic Re	-
	BB_AM	• •					vclic Read)
		,	ist fø	or a max	of	10 [0_1_2	-
	mechan				01	10 [0_1_2	0.]
Response Structure	comman	lowing table shows the genera and "AMM3". The number of r of requested mechanism mea mns.			lines	(1 n=32) d	epends on the
		Line 1n		Column '	1		Column 12
Value Range/Meaning	1 =	Device addr	ess		[00	.63]	
of Columns	2 =	Device nam	е		-	32 ASCII cha	racters]
	3 =	Mechanism	num	ber	[03		
	4 =	Mechanism	name	Э	-	. 28 ASCII cha	iracters]
	5 =	Type of mes	ssage	•	-	fault/error, D =	-
	6 =	Message so	ource		[NC,	SPS]	
	7 =	Type of mes	ssage	e (2)	[S = Status, O = Operator, E = External, I = Internal]		
	8 =	Message nu	ımbeı	ſ	[0600]		
		Message te			[max. 54 ASCII characters]		
	10 =	Reference text			[x= exists,		
		·			=	does not exist]	
		 2 bytes of additional is required to re- information information "@" for the message number 					
		Filename fo			e.g.	in HTML forma	t
Reference to Literature	See cha	oter entitled	"Liter	ature" [13].			
Example AMM3	Reads th 01.	ie current m	iessa	ges of mech	anisn	ns 0 and 1 of	device address
	<u>Assumpt</u> Device a		vith 2	defined mecl	hanisı	ms 0 and 1.	
	FI comm	command 01_BR_AMM3_0		BR_AMM3_0_	1		
	Line	Column	Ansv	wer			
	1	1	01				
		2	Drill	center			
		3	0				
		4	Stati	on 1			
		5	D				
		6	CNC				
		7	S:				
		8	79				
		9		on waite until t	tool_ch	nange command	has ended
		10	Jain		001-01	ange command	

10

х



FI comm	and	01_BR_AMM3_0_1
Line	Column	Answer
	11	0
	12	
2	1	01
	2	Drill center
	3	1
	4	Station 2
	5	F
	6	CNC
	7	0
	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
Х	Device address [000.63]
Y	Mechanism number [031]

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 1n	Column 1		Column 12
Value Range/Meaning	1 =	Device address	[00]	63]	
of Columns	2 =	Device name	[ma	ix. 32 ASCII cha	racters]
	3 =	Mechanism num	ber [0	.31]	
	4 =	Mechanism nam	e [ma	ix. 28 ASCII cha	racters]
	5 =	Type of message	e [F =	fault/error, D =	diagnosis]
	6 =	Message source	[CN	IC, PLC]	
	7 =	Type of message	•	= Status, O = Op = External, I = In	
	8 =	Message numbe	r [0	600]	
	9 =	Message text	[ma	ix. 54 ASCII cha	iracters]
	10 =	Reference text	-	exists, = does not exist]	
	11 =	2 byte additional for the message		equired to resolv rmation "@" (se	



12 = Filename for additional e.g. in HTML format information for message text

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.

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		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	0				
	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				



	FI comm	and	01_BR_AMM4_01.	0_01.1_03.0		
	Line	Column	Answer			
		9	Circular interpolatio	n		
		10	x			
		11	0			
		12				
FI command			sm related output o the devices of the M			
	BR_AMN	15_(1)_(2)_((3) (Sing	gle Read)		
	1 = Mech	anism numl	ber [03	1]		
	(2) = Mes	sage numb	er [06	00]		
	(3) = 2 k number	(3) = 2 bytes of additional information for the message number				
	Note:		parameter of AMM ands AMM1 AMM		1 th partial result	
Response Structure	commane mechanis	d "ĂMM5". sms = 512)	hows the general The number of line depends on the in turn consists of	es n=512 lines (n= number of reques	16 devices x 32	
	L	ine 1n	Column 1		Column 10	
Value Range/Meaning	1 =	device addre	255	[0063]		
of Columns		Device name		[max. 32 ASCII ch	aractersl	
		Mechanism		[031]		
	4 =	Mechanism	name	[max. 28 ASCII ch	aracters]	
	5 =	Type of mes	sage	F = fault/error, D =	= diagnosis]	
	6 =	Message so	urce	[CNC, PLC]		
	7 =	Type of mes	sage (2)	[S = Status, O = C E = External, I = I		
	8 =	Message nu	mber	[0600]		
	9 =	Reference te	ext	[max. [max. 14 line] 78 characters/line]		
		Filename fo	r additional for reference text	e.g. in HTML forn	nat	



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

ure See chapter entitled "Literature" [13].

Active NC Memory Size: ANM

MWCX device group

234567

Designation	ANM	Active NC	M emory Si	ze			
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 C						Column 3
Value Range/Meaning	1 = NC Memory Size [string]						
of Columns	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	Line Column 1 C				C	Column 3

NC Memory Size

654321

1



Active Machine Parameter Index: API

Designation	API Active Machine-Parameter Index							
Explanation	devices device a	of all defined are output: the me of creation of the active						
FI command	BR_AP	I1 (Single Read)					
	BC_AP	II (Cyclic Read))					
	BB_API1 (Break Cyclic Read)							
	Note:	The "API1" FI comman group. Therefore, any v command line (see Exa	alid device add					
Response Structure	comma	owing table shows the gen nd "API1". The response co th 8 columns.						
		Line 1n:	Column 1		Column 8			
Value Range/Meaning	1 =	device address		[0063]				
of Columns	2 =	Index of active paramete	r record	[0]				
	3 =	Identification string of the record	e parameter	[max. 84 AS characters]	SCII			
	4 =	Name of parameter reco	rd	[max. 32 ASCII characters]				
	5 =	Size of parameter record	[byte]	[max. 8 AS0 characters]	CII			
	6 =	Date of creation or of the in the parameter record.	last change	[8 ASCII cha format: DD				
	7 =	Time of creation or of the l the parameter record.	ast change in	[8 ASCII cha format: HH				
	8 =	Additional information (e. defined processes).	[max. 24 ASCII characters]					
	Note:	In cases where there is the device or where th been changed, Colun Columns 2 to 8 the val	ne active mac nn 1 is giver	hine paramet	ter record has			
Example API1	Reads the information on the active machine parameter records of all defined devices. <u>Assumption:</u> The following device addresses of the MWCX device group have been defined:							
		address 00: MTC200-P, address 01: MTC200-P, a	nd					
		address 02: MTVNC.						
	20100							



	FI comma	and	01_BR_API1			
	Line	Column	Answer			
	1	1	00			
		2	0			
		3	00MSD 0209	-15 15625 28.0)1.9913:29:10N	/123456
		4	MSD 0209-1	5		
		5	15625			
		6	28.01.99			
		7	13:29:10			
		8	M123456			
	2	1	01			
		2	0			
		3	88PCI 12.45.	12.34 10584 1	1.11.9811:11:1	1M12
		4	PCI 12.45.12	2.34		
		5	10584			
		6	11.11.98			
		7	11:11:11			
		8	M12			
	3	1	02			
		2	0			
		3	11Lab 5 DRV	/ 24464 01.03.9	9914:25:10M13	3456
		4	Lab 5 DRV			
		5	24464			
		6	01.03.99			
		7	14:25:10			
		8	M13456			
Reference to Literature	See chap	ter entitled	"Literature" [14].		
FI command	BR_API2	(Siı	ngle Read)			
	BC_API2	(Cy	clic Read)			
	BB_API2	(Br	eak Cyclic R	ead)		
Response Structure				neral structure nsists of a line		
		Line 1		Column 1		Column 8
Value Range/Meaning	1 = [Device add	ress		[0063]	
of Columns			tive paramete	r record	[0]	
	3 = 1		n string of the		[max. 84 AS characters]	CII
	4 = 1	Name of pa	rameter reco	rd	[max. 32 AS characters]	CII

in the parameter record.

format: DD.MM.YY]

- 7 = Time of creation or of the last change in [8 ASCII characters in the parameter record. [8 ASCII characters]
- 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 a	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].

Active Note in NC Program (Note and NC Record Number): APM

Designation	APM Active Part-Program Message						
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 Rexroth Indramat 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	l)			
	(1) = NC proces	s number	[06]				
Response Structure	The following table shows the general structure of the response to the F 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].

Active Note in NC Program (only NC Record Number): APN

Designation	APN	Active P	art-Program I	Messag	le- 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 Rexroth Indramat 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 chapter entitled "Literature"					
FI command	CR_APN	_(1)		(Single	Read)	
	CC_APN	_(1)		(Cyclic	Read)	
	CB_APN	_(1)		(Break	Cyclic Read)	
	(1) = NC	process nu	mber	[06]		
Response Structure	One line with one column is output for the NC record number of the active note.					
			_			
		Lin	e 1		Column 1	
Value Range/Meaning of Columns	1 = NC re		e 1 per of the not	e	Column 1 [00009999]	
	1 = NC re Note:	ecord numb	per of the not	gram de		
	Note:	If the cur result of 0	per of the not rrent NC prog Column 1 is [i	gram de 0000].	[00009999]	
of Columns	Note:	If the curresult of 0	per of the not rrent NC prog Column 1 is [i	gram de 0000]. ne active	[00009999] bes not contain a note, then the	
of Columns	Note: Read the address 0	If the curresult of 0	per of the not rrent NC prog Column 1 is [i number of th	gram de 0000]. ne active	[00009999] bes not contain a note, then the	
of Columns	Note: Read the address 0	If the cur result of (NC record)0.	per of the not rrent NC prog Column 1 is [i number of th	gram de 0000]. ne active	[00009999] bes not contain a note, then the	



Actual (Current) Position Value of an Axis: APO

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.					
		m or inches. If, ho	s possible to pre-select conversion wever, a spindle is selected as an serves no purpose.			
	CR_APO1_(1)_(2)_	-	(Single Read)			
	CC_APO1_(1)_(2)_		(Cyclic Read)			
	CB_APO1_(1)_(2)_		(Break Cyclic Read)			
	(1) = NC process n		[06]			
	(2) = Axis meaning		[011] (see Chapter "Data Tables")			
	(3) = System of coc	ordinates	[1 = machine coordinates2 = program coordinates]			
	(4) = Required measurement system [mm, inch] (opt.)					
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	[132, according to settings of the system parameters]			
	(2) = System of coc	ordinates	[1 = machine coordinates2 = program coordinates]			
	(3) = Required mea (opt.)	asurement system	[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 Col	umn 2 Column 3 Column 4			
Value Range/Meaning	1 = Axis name	[according to	settings of axis parameters]			
of Columns	2 = Position		settings of process parameters]			
	3 = Unit		settings of process parameters:			
	4 = Position		2, but rounded up or down the parameter "indicated decimal			



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 comma	and 00	_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 comma	and	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].

Active NC Program Number: APP

Designation	APP Active Part-Progr		n 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) (1) = NC process number		(Break Cyclic Read)		
			[06]		
Response Structure	command			he response to the FI t for the NC memory	
		Line 1	Column 1	Column 2	
Value Range/Meaning	1 = NC me	emory	[A = memory A; B = memory B]		
of Columns	2 = NC pro	ogram number	[0199]		



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

Reference to Literature See chapter entitled "Literature" [17].

Axis Reference Flags: ARF

Designation	ARF	ARF Axis Reference Flags								
Explanation		The reference flags for a process are to be displayed. These flags exfor the interpolation axes $\{X, Y, Z, U, V, W, A, B, C\}$					s exist			
FI command	CR_ARF	_(1)			(Single	Read)			
	CC_ARF	_(1)			(Cyclic	Read)			
	CB_ARF	_(1)			(Cyclic	Break	()			
	(1) = NC	process	s numbe	r	[06]					
Response Structure		A line with 9 columns is output, each for the axis meaning: X, Y, Z, U, V, W, A, B, C axis.					, U, V,			
	An axis re	An axis reference flag can have the following three values:								
	0	Axis r	not in ref	erence						
	1	Axis i	n referei	nce						
		Axis r	not prese	ent						
Example ARF	Displays t	he axis	referenc	e flags f	or proce	ss 0				
	<u>Assumpti</u>	<u>on</u> :								
	• X, Y, Z	z axes a	re in refe	erence,						
	• U, V, V	V axes a	are not i	n referer	nce,					
	• A, B, C	Caxes a	re not pi	resent						
	FI comma	and	00_0	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			



Actual (Current) Rapid Override: APO

Designation	ARO	Actual R	apid O verride				
Explanation	group is ou are execut	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.					C process of the	
	CR_ARO1	l_(1)	(S	ingle	Read)		
	CC_ARO1	l_(1)	(C	yclic	Read)		
	CB_ARO1	l_(1)	(B	reak	Cyclic I	Read)	
	(1) = NC p	orocess nu	mber [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		Col	umn 1		Column 3
Value Range/Meaning	1 = Identif	ier			[R	OV= rapio	l override]
of Columns	2 = Curre	nt value of	the rapid over	ride	-	255]	
	3 = Unit				- [%	-	
	Note: The valid range of override weighting by the PLC program i between 0 and 255%. The NC limits the axis and/or processor speed to the maximum values set in the parameters if an overrid value is set that is too large.					and/or processor	
Example ARO1	Read the address 00		lue of the rapi	d ove	erride in	NC proc	ess 0 of device
	FI comma	nd	00_CR_AFO1_	0			
			Ai	nswer	•		
	Li	ne	Column 1		Colu	mn 2	Column 3
	1		ROV		10	00	[%]
Reference to Literature	See chapte	er entitled	"Literature" [18	8].			



Axis Reference Table: ART

MWCX device group

Designation	ART Axis Reference Table						
Explanation	be used to and with wi	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	parameter	s input, f			imited to or	ne process.	If the optional
	CR_ART1			(antio		(Single Read)	
	(1) = NC p	rocess n	umber	optio	iai)	[06]	
Response Structure	having the receives th X,Y,Z,U,V,V	Seven lines are output (1 optional line) each with 12 columns and each naving 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 [].					
	the first col	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	Colum	n 1	Column 2	2	Column 12
	With the fol	lowing m	neaning				
	Line 1, 2	7:	Axis re	eferen	ce table for	r process 0, 1	6
	Column 1,	212:	Axis n	umbe	r for axis m	eaning X, YS	S3
Example ART1	Reads the	complete	e axis re	feren	ce table for	device 00	
	Assumption						
	Processes and the axe		4 are p	resen	t		
	1 (X axis	s in proc	ess 0),				
	2 (Y axis	s in proc	ess 0 or	proc	ess 1),		
	,	s in proc	,				
		s in proc					
	FI comman	d	00_CR_	ART1			
	Line	Colur			Answer		Column 12
	Line 1				2 2		
	2				- *Y		4
	3						
	4		-				
	5	3	3				*S
	6		-				
	7		-				

Example ART1 Reads the axis reference table for process 1 of device 00:

	FI command (00_CR_4	ART1_1		
				Answer		
	Line	Colum	nn 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.					oonse to the FI
		Line 1		Column 1		
Value Range/Meaning of Columns	1 = Binary axis reference table data					
Example: ART	Read the	binary axis	referen	ce table data of	the device 00.	
	FI comma	and	00_BR_ART			
	Line	Column	Answe	r		

Read axis reference table data

1

1

Actual (Current) Spindle Data: ASD

Designation	ASD Actual Spindle Data			a				
Explanation	MWCX de	The current spindle data of an NC process of the selected de MWCX device group is read out. This command is a compilation of PSS, ASS, MSS, ASG ASG.						
FI command	Output th number.	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 numbe	er	[06]				
	(2) = Spir	ndle number		[13]				
Response Structure	The following table shows the general structure of the response to the Fl 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							
	current s speed, ar	pindle speed, nd the unit ac	prograr cording	nmed sp to settir	oindle speed, m	aximum spindle ess parameters,		
	current s speed, ar current sp level.	pindle speed, nd the unit ac	prograr cording maxim	nmed sp to settir	oindle speed, m	aximum spindle ess parameters,		
Value Range/Meaning of Columns	current sp speed, an current sp level. 1 = Axis f 2 = Curre 3=progra speed	bindle speed, ad the unit ac bindle override, _ine 1	prograr cording maxim Colu [S ed [ad [ad	nmed sp to settir um spinc umn 1 , S1, S2, ccording	bindle speed, m ngs of the proce lle override, and 	aximum spindle ess parameters, the current gear Column 9 s parameters] s parameters]		

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	
		Answ	/er
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].

Active Spindle for Process: ASF

Designation	ASF Active Spine	lle For 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 	C C			
	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	er [06]			
Response Structure	The response to the F column for the selected	-Command "ASF" consists of one line with one 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].

Actual (Current) Spindle Gear Level: ASG

Designation	ASG	Actual S	pindle G ear			
Explanation	The current spindle gear level of an NC process of the selected of from the MWCX device group is read out. The control signals of the selection are only evaluated by the CNC when one gear with at leas gear levels has been entered within the axis parameters.					
FI command	CR_ASG	_(1)_(2)	(5	(Single Read)		
	CC_ASG	_(1)_(2)	(0	Cyclic Read)		
	CB_ASG	_(1)_(2)	(E	Break Cyclic Read)		
	(1) = NC	process nu	imber [C	6]		
	(2) = Spir	ndle numbe	er [1	3]		
Response Structure		entifier and	for the curren	and consists of one line t spindle gear level o	f the selected NC	
	Line 1			Column 1	Column 2	
Value Range/Meaning	1 = Identifier			[g = gear]		
of Columns	2 = Current spindle gear level [13, -]					
	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	GG Read the current spindle gear level of device address 00.			el of the 1 st spindle in	NC process 0 of	
FI command			00_CR_ASG_	0_1		
	Line	Column	Answer			
	1	1	g			
		2	1			

Reference to Literature	See ch	apter entitled "Lite	rature" [20]			
				•		
Active System Error M	essay	185. AJIVI			N //\\//	CX device group
						CX device group
Designation	ASM	Active System	n Message	es		
Explanation	The active system error messages 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 the current system error messages pending for all active devices from the MWCX device group.					tive devices from
	BR_AS	M1	(Single R	lead)		
	BC_AS	M1	(Cyclic R	ead)		
	BB_AS	SM1	(Break C	yclic R	ead)	
	Note:	device group.	This mea	ns that		vices within this ice address can le ASM1).
Response Structure	The following table shows the general structure of the response to the F 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.			depends on the		
	address	s, device name, m	essage nu	mber, i	message statu	s, short text and or message.
	address	s, device name, m	essage nu	mber, erence t	message statu	s, short text and
Value Range/Meaning	address	s, device name, m on of whether ther	essage nu e is an refe	mber, erence t	message statu text for this erro	s, short text and or message.
Value Range/Meaning of Columns	address indicatio	s, device name, m on of whether ther Line 1n	essage nu e is an refe	mber, i erence f in 1 [00	message statu text for this erro	s, short text and or message. Column 8
	address indication 1 =	s, device name, m on of whether ther Line 1n Device address	essage nu e is an refe Colum	mber, i erence f in 1 [00	message statu text for this erro 63] . 32 ASCII cha	s, short text and or message. Column 8
	address indication 1 = 2 =	s, device name, m on of whether ther Line 1n Device address Device name	essage nu e is an refe Colum er	imber, i erence f in 1 [001 [max [01]	message statu text for this erro 63] . 32 ASCII cha	s, short text and or message. Column 8 racters]
	address indication 1 = 2 = 3 =	s, device name, m on of whether ther Line 1n Device address Device name Message numbe Type of message Message text	essage nu e is an refe Colum er	mber, i erence f in 1 [001 [max [01] [F = f [max	message statu text for this erro 63] . 32 ASCII cha 50] ault/error, D = . 54 ASCII cha	s, short text and or message. Column 8 racters] diagnosis] racters]
	address indication 1 = 2 = 3 = 4 = 5 = 6 =	s, device name, m on of whether ther Line 1n Device address Device name Message numbe Type of message Message text Reference text	essage nu e is an refe Colum er	mber, i erence i in 1 [001 [max [F = f [max [x= e»	message statu text for this erro 63] . 32 ASCII cha 50] ault/error, D = . 54 ASCII cha kists, = does n	s, short text and or message. Column 8 racters] diagnosis] racters] not exist]
	address indication 1 = 2 = 3 = 4 = 5 =	s, device name, m on of whether ther Line 1n Device address Device name Message numbe Type of message Message text	essage nu e is an refe Colum er e	mber, i erence i in 1 [001 [max [F = f [max [x= e» is rec	message statu text for this erro 63] . 32 ASCII cha 50] ault/error, D = . 54 ASCII cha	s, short text and or message. Column 8 racters] diagnosis] racters] tot exist] re the
	address indication 1 = 2 = 3 = 4 = 5 = 6 =	s, device name, m on of whether ther Line 1n Device address Device name Message numbe Type of message Message text Reference text 2 bytes of additio information	essage nu e is an refe Colum er e onal number litional	[001 [001] [max [01] [F = f [max [x= ex is rec inforr	message statu text for this erro 63] . 32 ASCII cha 50] ault/error, D = . 54 ASCII cha kists, = does n juired to resolv	s, short text and or message. Column 8 racters] diagnosis] racters] tot exist] e the e ASM5)
	address indication 1 = 2 = 3 = 4 = 5 = 6 = 7 = 8 = Reads	s, device name, m on of whether ther Line 1n Device address Device name Message numbe Type of message Message text Reference text 2 bytes of addition information for the message Filename for add information for m	essage nu e is an refe Colum er e onal number ditional nessage	(001 (001 (max (01 (F = f (max (x= e)) is rec inforr e.g. ii	message statu text for this error 63] . 32 ASCII cha 50] ault/error, D = . 54 ASCII cha kists, = does n juired to resolv nation "@" (sec n HTML format	s, short text and or message. Column 8 racters] diagnosis] racters] tot exist] re the e ASM5)
of Columns	address indication 1 = 2 = 3 = 4 = 5 = 6 = 7 = 8 = Reads MWCX <u>Assum</u>	s, device name, m on of whether ther Line 1n Device address Device name Message numbe Type of message Message text Reference text 2 bytes of addition for the message Filename for addo information for m text	essage nu e is an refe Colum r e onal number ditional nessage m error me	Imber, i Prence f In 1 [001 [max [01] [F = f [max [x= ex is rec inforr e.g. ii	message statu text for this error 63] . 32 ASCII cha 50] ault/error, D = . 54 ASCII cha kists, = does n juired to resolv nation "@" (sec n HTML format	s, short text and or message. Column 8 racters] diagnosis] racters] tot exist] re the e ASM5)
of Columns	address indication 1 = 2 = 3 = 4 = 5 = 6 = 7 = 8 = Reads MWCX <u>Assum</u> The foll	s, device name, m on of whether ther Line 1n Device address Device name Message numbe Type of message Message text Reference text 2 bytes of addition for the message Filename for addition for the message Filename for addition for the message the current system device group. ption:	essage nu e is an refe Colum r e onal number ditional nessage m error me	Imber, i Prence f In 1 [001 [max [01] [F = f [max [x= ex is rec inforr e.g. ii	message statu text for this error 63] . 32 ASCII cha 50] ault/error, D = . 54 ASCII cha kists, = does n juired to resolv nation "@" (sec n HTML format	s, short text and or message. Column 8 racters] diagnosis] racters] tot exist] re the e ASM5)
of Columns	address indication 1 = 2 = 3 = 4 = 5 = 6 = 7 = 8 = Reads MWCX <u>Assum</u> The foll • Dev	s, device name, m on of whether ther Line 1n Device address Device name Message number Type of message Message text Reference text 2 bytes of addition information for the message Filename for addo information for m text the current system device group.	essage nu e is an refe Colum er e onal number ditional nessage m error me es are defir	Imber, i Prence f In 1 [001 [max [01] [F = f [max [x= ex is rec inforr e.g. ii	message statu text for this error 63] . 32 ASCII cha 50] ault/error, D = . 54 ASCII cha kists, = does n juired to resolv nation "@" (sec n HTML format	s, short text and or message. Column 8 racters] diagnosis] racters] tot exist] re the e ASM5)

	FI comm	T	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	Х		
		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			ystem error message that is pending for the selected CX device group.		
	BR_ASM		(Single Read)		
	BC_ASM	12	(Cyclic Read)		
	BB_ASN	12	(Break Cyclic Read)		
Response Structure	command device ad	d "ÁSM2". Idress, devi	shows the general structure of the response to the FI The answer consists of a line of 8 columns for the ce name, message number, message status, short text ether there is an reference text for this error message.		
	L	ine 1n	Column 1 Column 8		
Value Range/Meaning	1 = 1	Device add	ress [0063]		
of Columns	2 =	Device nam			
	3 =	Message n			
	4 =	Type of me	ssage [F = fault/error, D = diagnosis]		
	5 =	Message te	[max. 54 ASCII characters]		
	6	Deference	toxt [v_ovietedeep not oviet]		

6 = Reference text [x= exists, -- = does not exist]



7 =	2 bytes of additional	is required to resolve the
	information	information "@" (see ASM5)
	for the message number	

8 = Filename for additional e.g. in HTML format information for message text

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	Answer
1	1	01
	2	Drill center
	3	71
	4	F
	5	PLC battery voltage too low.
	6	Х
	7	0
	8	

FI command Output the current system error messages 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 error message.

		Line 1n	Column 1		Column 8
Value Range/Meaning	1 =	Device address	[0063]		
of Columns	2 =	Device name	[max. 32 A	[max. 32 ASCII characters]	
	3 =	Message number	[0150]		
	4 =	Type of message	[F = fault/e	ror, D = d	iagnosis]
	5 =	Message text	[max. 54 A	SCII chara	acters]
	6 =	Reference text	[x= exists,	= does no	t exist]
	7 =	2 byte additional information for the message number	is required information		
	8 =	Filename for additional information for message text	e.g. in HTM	IL format	



Example ASM3 Reads the current system error messages of the selected MWCX devices. Assumption:

The following devices addresses are defined:

- Device address 01,
- Device address 07 and
- Device address 10.

FI comm	and	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	Х
	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 current system error messages 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 error message.

		Line 1n	Column 1		Column 8
Value Range/Meaning	1 =	Device address	[0063]		
of Columns	2 =	Device name	[max. 32 ASCII characters]		
	3 =	Message number	[0150]		
	4 =	Type of message	[F = fault/error,	D = diag	nosis]
	5 =	Message text	[max. 54 ASCII	characte	ers]
	6 =	Reference text	[x= exists, = de	oes not ex	kist]
	7 =	2 bytes of additional information	is required to re "@" (see ASM5		e information



for the message number

8 = Filename for additional e.g. in HTML format information for message text

Example ASM4 Reads the current system error messages of all defined devices of the MWCX device group.

<u>Assumption:</u> 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	Х
	7	0
	8	
2	1	10
	2	Milling center 2
	3	1
	4	D
	5	Error has been corrected.
	6	х
	7	0
	8	

FI command Output the reference text for the currently pending error message, related to the device and the message number.

	(1) =	•	[0150]	(Single Read) [0150] mation for the message			
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 1n		Column 1		Column 6		
Value Range/Meaning	1 =	Device address	[00.	63]			
of Columns	2 =	Device name	[ma	[max. 32 ASCII characters			
	3 = Message number		[0	[0150]			
	4 =	Type of message	[F =	[F = fault/error, D = diagno			
	5 =	Reference text	-	[max. [max. 14 lines with a m 78 characters/line]			
	6 =	Filename for additi	ional e.g.	in HTML forma	ıt		



information for reference text

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].

Actual (Current) NC Sequence Number: ASN

MWCX device group

Designation	ASN	Actual Sequence Number						
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 Re	ead)				
	CC_ASN	_(1)	(Cyclic Re	ead)				
	CB_ASN	_(1)	(Break Cy	clic Read)				
	(1) = NC p	process nun	nber [06]					
Response Structure	The response to the "ASN" FI command consists of one line with one column for the active NC sequence number [N0000N9999].							
			•					
			ne 1	Column 1				
	Note:	Lir	ie 1 id NC program exis					
Example ASN	Note:	Lir If no val value [N0	id NC program exis 0000].	Column 1				
Example ASN	Note:	Lir If no val value [N0 active NC s	id NC program exis 0000].	Column 1 ts then Column 1 receives the				
Example ASN	Note:	Lir If no val value [N0 active NC s	id NC program exis 0000]. equence number of N0	Column 1 ts then Column 1 receives the				
Example ASN	Note: Read the FI comma	Lir If no val value [NC active NC s	ie 1 id NC program exis 0000]. equence number of NC 00_CR_ASN_0	Column 1 ts then Column 1 receives the				

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



Actual (Current) Spindle Override: ASO

MWCX device group

Designation	ASO A	ctual S pindle O v	erride				
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		urrent value of t e group related to				ed device of the bindle number.	
	CR_ASO1_(1)_(2)	(Sir	ngle Read)		
	CC_ASO1_(1		, (Cy	clic Read)		
	CB_ASO1_(1			eak Cyclic			
	(1) = NC proc		[0	.6]			
	(2) = Spindle		- [1	-			
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 [%].						
	L	ine 1	Col	umn 1		Column 3	
Value Range/Meaning	1 = Identifier			[S= Spindle]			
of Columns	2 = Current v	value of the override with unit [0255]					
	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 cur device addres		e overrid	le of Spind	dle 1 in I	NC process 0 of	
	FI command	00_CR_ASO	1_0_1				
			Answ	/er			
	Line	Columr	1	Colun	nn 2	Column 3	
	1	S:		60)	[%]	

Reference to Literature

See chapter entitled "Literature" [21].



Actual (Current) Spindle Speed: ASS

Designation	ASS Actual Spindle Speed							
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) (S	ingle Read)					
	CC_ASS_(1)_(2) (C	yclic Read)					
	CB_ASS_(1)_(2) (В	reak Cyclic	Read)				
	(1) = NC process	number [0	6]					
	(2) = Spindle nur	nber [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.							
	Lii	ne 1	Column 1		Column 3			
Value Range/Meaning	1 = Axis name	[S, S1, S	1, S2, S3]					
of Columns	2 = Spindle spee	-	cording to settings of axis parameters]					
			nin; according to parameter setting]					
	Note: If the spindle number is not defined in the selected process, then the result in Column 1 is [], the result Column 2 is [0.0] and that in Column 3 is [1/min].							
Example ASS	Read the current address 00.	axis speed of the	e 1 st spindle ir	NC proces	s 0 of device			
	FI command	00_CR_ASS_0_1						
		Α	nswer					
	Line	Column 1	Colum	n 2 (Column 3			
	1	S1	4000.	0	1/min			
Reference to Literature	See chapter entitled "Literature" [22].							



Active Tool Number: ATN

MWCX device group

Designation	ATN	Active Too	ol- N umber			
Explanation			per of an NC p is read out.	rocess of the select	ted device from the	
FI command	CR_ATN_	_(1)	(S	(Single Read)		
	CC_ATN_	_(1)	(C	yclic Read)		
	CB_ATN_	_(1)	(E	reak Cyclic Read)		
	(1) = NC p	process nur	nber [0	6]		
Response Structure				mmand consists of umber of the active		
		Line 1		Column 1	Column 2	
Value Range/Meaning of Columns	1 = Ident 2 = Num	ifier ber of activ	e tool	[T = tool] [1999999	99]	
	Note:			e selected NC proce nd Column 2 the val		
Example ATN	Read the	number of	the active tool	in NC process 0 of (device address 00.	
	FI comma	and	00_CR_ATN_0			
	Line	Column	Answer			
	1	1	Т			
		2	4			
Reference to Literature	See chap	ter entitled	"Literature" [23].		

Reading Actual (Current) Tool Place Information: ATP

MWCX device group

Designation ATP Actual Tool Place Information

Explanation Information regarding the tool place and the current edge of the preselected 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 ar e			n of the tool magazine and of the active tool			
	CR_ATP1_(1)			(Single Read)			
	CC_ATP	1_(1)		(Cyclic Read))		
	CB_ATP	• •		(Break_Cycli	c Read)		
	(1) = NC	process nu	mber [[06]			
Response Structure				vith 4 column		esponse to the for the returned	
		Line 1		Column 1		Column 4	
Value Range/Meaning of the Columns	2 = Ac	ctual positio	f magazine n of magazin				
			ge number	[19]		t [4 000]	
		ool place /pe + place	number)	[Mx= ma Sx = spi Gx = gri		t [x=1999] [x=14] [x=14]]	
	Note:		e refer to t			sition of the tool the magazine	
Example ATP1				magazine p n NC process		and tool-place 00.	
	FI comm	and	00_CR_ATP	1_0			
	Line	Column	Answer				
	1	1	3				
		2	3				
		3	1				
		4	S1				
FI command	Edge and	l place infor	mation for the	e active tool.			
	CR_ATP2	•		(Single Read	I)		
	CC_ATP2	2_(1)		(Cyclic Read)		
	CB_ATP2	2_(1)		(Break_Cycli	ic Read)		
	(1) = NC	process nur	mber	[06]			
Response Structure						esponse to the for the returned	
		Line 1		Column	1	Column 2	
Value Range/Meaning of the Columns	Line 1 1 = Active tool edge number 2 = Tool place (type + place number)			[19] [Mx= magaz	ine/turret	[x=1999]	

Example ATP2	P2 Reads the edge and tool place information of the active tool from process 0 of device 00.						
	FI comm	FI command 00_CR_ATP2_0					
	Line	Column	Answer				
	1	1	1				
		2	S1				
FI command	Output of	the positio	n informatior	of the tool magazine			
	CR_ATP	•		(Single Read)			
	CC_ATP	• •		(Cyclic Read)			
	CB_ATP	P3_(1)		(Break_Cyclic Read	i)		
	(1) = NC	process nu	umber	[06]			
Response Structure	The following table shows the general structure of the response to th "ATP3" FI command. One line with 2 columns is output for the returne values.						
		Line 1		Column 1	Column 2		
Value Range/Meaning of the	1 = Com	mand posit	ion of magaz	zine [1999]			
Columns	2 = Actu	al position o	of magazine	[1999]			
	Note: Details of the current command and actual position of the tool magazine refer to the reference point of the magazine controller.						
Example ATP3		e command) of device (position of the tool	magazine from NC		
	FI comm	and	00_CR_ATP	93_0			
	Line	Column	Answer				
	1	1	3				
		2	3				
Reference to Literature	See chap	ter entitled	"Literature" [8].			
Access to Actual (Curr	ent) To	ol Data	Record:	ATR			
,	,				MWCX device group		
Designation	ATR	Actual T	ooldata R eco	ord			
Explanation		a complete rocessing to		record and/or cutter	data record of the		
FI command	Read bas	sic data rec	ord or cutter	data record of the cu	rrent tool.		
	CR_ATR	R_(1)_(2)		(Single Read)			
	CC_ATR	R_(1)_(2)		(Cyclic Read)			
	CB_ATR	R_(1)_(2)		(Break Cyclic Read)			
	(1) = NC	process nu		[06]			
	(0) D-4	ha waaawa					

[0 = base tool data, 1...9 = cutter data]



Response Structure	"CR_ATR	" FI comma	e shows the general structure of the response to the nand. One line is output with 28 (basic data) or 40 (cutter ne returned values.			
	Line	e1 (Column 1			Column 28/40
Value Range/Meaning of the Columns		-	base tool da tool cutter da		[max. 28 data el (see basic value [max. 40 data el (see value range	e range data) lements]
Example TDR1	Read the processin		IC process 0 of	the tool currently		
	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	М			
		13				
		14	М			
		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].

Accepting the Data Record for the Current Tool: ATU

1

MWCX device group

Designation	ATU Actual Tooldata Update						
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]		
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 comma	and	00_CR_AT	J_0			
	Line	Column	Answer				

(P_ACK)

1

Active Zero-Offset Bank: AZB

Designation	AZB Active Zero-Offset 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_AZB	1_(1)	(5	Single Read)			
	CC_AZB	1_(1)	(0	Cyclic Read)			
	CB_AZB	1_(1)	(E	Break Cyclic Read)			
	(1) = NC	process nu	mber [0	6]			
Response Structure		or the iden		ommand consists o set) and the number			
		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 comma	and	00_CR_AZB1	_0			
	Line	Column	Answer				
	1	1	0				
		2	2				
Reference to Literature	See chap	ter entitled	"Literature" [24	4].			

MWCX device group

NC Cycle Download: CCA

Designation	CCA	NC-Cycle Access			
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 [0120] (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		

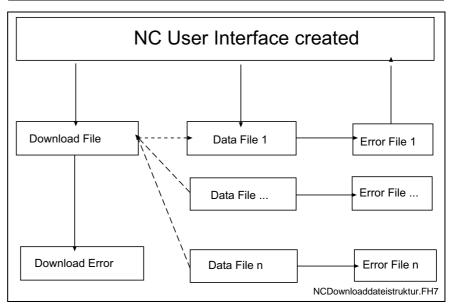


Fig. 7-1: File structure of the download file

0

1

3

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	3 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:

	or the NC cycle programs.			
	ZZZZZZ XX YYY	Proces	pe (CYC-PRG) s number m number of the cycle programs (1 – 255)	
		ision car	be freely selected. ".dat" has been used in the	
	Examples:			
	CYC-PRG-00- CYC-PRG-01-		Cycle program for process 0 program 86 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:\Progi	ram 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: Ca	re must l	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]

Version	
Process	[06]
No	[0255]
Name	max. 32 characters
Size	
Time	max. 8 characters
Date	max. 8 characters
ShortID	max. 8 characters
i,	
(optional)	
	Process No Name Size Time Date ShortID

Information on date and time is given in the format Date : dd.mm.yy Time: hh:mm:ss

Status flag	Description	
С	Compiled	
E	Error	
The marked section is then printed out.	Not compiled	
No details	No compiler call	

Fig. 7-2: 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

NC Cycle Upload: CCA

MWCX device group Designation CCA NC-Cycle Access NC cycles are uploaded via all active processes. During upload, a basic Explanation 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 00 BR CCA1 "D:\Program Files\Indramat\Mtgui\ **FI** command Temp\upload.ini"/3 Column Line Answer 1 1 01 2 1 00_BR_CCA1_"D:\Program Files\Indramat\Mtgui\ Temp\upload.ini" /3



3

1

0

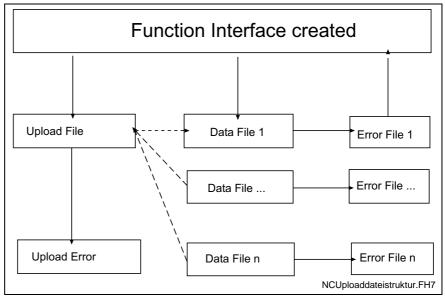


Fig. 7-3: File structure of the upload file

Structure of Upload File The

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	3 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 =1PackageName =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)
хх	Process number
ууу	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	[06]
Program number:	No	[0255]
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,	I	
	(optional)	

Information on date and time is given in the format Date : dd.mm.yy Time: hh:mm:ss





Status flag	Description
С	Compiled
E	Error
The marked section is then printed out.	Not compiled
No details	No compiler call

Fig. 7-4: 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

Position Set point of an Axis: CPO

Designation	СРО	Command POsition			
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		e command value of the select the code of the axis meaning.	ted axis of the device specified,		
	of the res		possible to pre-select conversion ever, a spindle is selected as an erves no purpose.		
	CR_CPC	D1_(1)_(2)_(3){_(4)}	(Single Read)		
	CC_CPC	D1_(1)_(2)_(3){_(4)}	(Cyclic Read)		
	CB_CPC	D1_(1)_(2)_(3){_(4)}	(Break Cyclic Read)		
	(1) = NC	process number	[06]		
	(2) = Axi	s meaning	[011] (see Chapter "Data Tables")		
	(3) = System of coordinates [1 = machine coordinates 2 = program coordinates]		-		
	(4) = Re (opt.)	equired measurement system	[mm, inch]		
FI command		e command position of an axis cal axis a	of the device specified, related to		
	of the res		possible to pre-select conversion ever, a spindle is selected as an erves no purpose.		
	CR_CPC	D2_(1)_(2){_(3)}	(Single Read)		
	CC_CPC	D2_(1)_(2){_(3)}	(Cyclic Read)		



	CB_CPC	CB_CPO2_(1)_(2){_(3)}		(Break Cyclic R	Read)	
	(1) = Physical axis number			[132, accordin the system para		
	(2) = Sys	tem of coo	rdinat	es	[1 = machine co 2 = program co	
	(3) = Rec (opt.)	quired meas	surem	nent system	[mm, inch]	
Response Structure	commanc axis desi	The following table shows the general str commands "CPO1" and "CPO2". One line axis designation, position, unit and the decimal places".		ne is output with 4	columns for the	
	Li	ne 1	Co	lumn 1 Col	umn 2 Column	3 Column 4
Value Range/Meaning	1 = Axis	name	[2	according to se	ettings of axis para	ameters
of Columns	2 = Posit	ion	-	-	ettings of process	-
	3 = Unit		- [2	-	e settings of proc	
	4 = Posit	ion	a		out rounded up or e parameter "indic	
	Note:			axis is not de	fined in the selections is [].	ted NC process
Example CPO1					s in machine co	ordinates in NC
Example CPO1		of device a	addre	ss 00.		ordinates in NC
Example CPO1	process 0	of device a	addre		1	ordinates in NC
Example CPO1	process 0	of device a	addre: 00_C	ss 00. CR_CPO1_0_2_	1	ordinates in NC
Example CPO1	process 0 Fl comma	of device a	addre: 00_C	ss 00. CR_CPO1_0_2_ Answe	1	
Example CPO1	FI comma Line 1 Read the	of device a and Column Z1 current po	addres 00_C 1	ss 00. CR_CPO1_0_2 Answe Column 2 -5.98975 n of the Z axi	1 Column 3	Column 4 -5.990 ordinates in NC
	FI comma Line 1 Read the	of device a and Column Z1 current po of device a	addres 00_C 1 ositior addres	ss 00. CR_CPO1_0_2 Answe Column 2 -5.98975 n of the Z axi	1 Column 3 [mm] s in machine coult is to be output	Column 4 -5.990 ordinates in NC
	FI comma Line 1 Read the process 0	of device a and Column Z1 current po of device a and	1 osition addres 00_C	ss 00. CR_CPO1_0_2 Answe Column 2 -5.98975 a of the Z axiss 00. The res CR_CPO1_0_2 Answe	1 Column 3 [mm] s in machine coult is to be output 1_inch	Column 4 -5.990 ordinates in NC in inches.
	FI comma Line 1 Read the process 0 FI comma Line	of device a and Column Z1 current po of device a and Column	1 osition addres 00_C	ss 00. CPO1_0_2 Answe Column 2 -5.98975 a of the Z axiss 00. The rest CR_CPO1_0_2 Answe Column 2	1 Column 3 [mm] s in machine coult ult is to be output 1_inch Column 3	Column 4 -5.990 ordinates in NC in inches. Column 4
	FI comma Line 1 Read the process 0	of device a and Column Z1 current po of device a and	1 osition addres 00_C	ss 00. CR_CPO1_0_2 Answe Column 2 -5.98975 a of the Z axiss 00. The res CR_CPO1_0_2 Answe	1 Column 3 [mm] s in machine coult is to be output 1_inch	Column 4 -5.990 ordinates in NC in inches.
	process 0 FI comma Line 1 Read the process 0 FI comma Line 1 Read the process 0 FI comma Line 1 Read the process 0	Column Z1 current po of device a and Column Z1 Current po	addres 00_C 1 osition addres 00_C 1 1 osition e add	ss 00. CPO1_0_2 Answe Column 2 -5.98975 a of the Z axiss 00. The ress CPO1_0_2 Answe Column 2 -0.23582 a of the Z axis	1 Column 3 [mm] s in machine coult ult is to be output 1_inch Column 3	Column 4 -5.990 ordinates in NC in inches. Column 4 -0.236 ordinates in NC
Example CPO1	process 0 FI comma Line 1 Read the process 0 FI comma Line 1 Read the process 0 FI comma Line 1 Read the process 0	Column Z1 current po of device a and Column Z1 current po 0 of device coordinates	addres 00_C 1 osition addres 00_C 1 00_C 1 osition addres 00_C	ss 00. CPO1_0_2 Answe Column 2 -5.98975 a of the Z axiss 00. The ress CPO1_0_2 Answe Column 2 -0.23582 a of the Z axis	1 Column 3 [mm] s in machine coult is to be output 1_inch r Column 3 [inch] s in machine coult	Column 4 -5.990 ordinates in NC in inches. Column 4 -0.236 ordinates in NC
Example CPO1	process 0 FI comma Line 1 Read the process 0 FI comma 1 Read the process 0 FI comma 1 Read the process 0 FI comma 1	Column Z1 current po of device a and Column Z1 current po 0 of device coordinates and	addres 00_C 1 osition addres 00_C 1 00_C 1 osition addres 00_C	ss 00. CPO1_0_2 Answe Column 2 -5.98975 a of the Z axiss 00. The ress COLUMN 2 -0.23582 a of the Z axist Column 2 -0.23582 a of the Z axist Column 2 Column 3 Column 2 Column 2 Column 3 Column 2 Column 3 Column 4 Column 5 Column 5 Column 6 Column 6 Column 7 Column 7 Column 6 Column 7 Column 7 Column 6 Column 7 Column 7 Column 7 Column 7 Column 7 Column 7 Column 8 Column 8 Column 8 Column 9 Column 9 Column 9 Column 1 Column 1	1 Column 3 [mm] s in machine coult is to be output 1_inch r Column 3 [inch] s in machine coult . physical axis r	Column 4 -5.990 ordinates in NC in inches. Column 4 -0.236 ordinates in NC number = 3) in
Example CPO1	process 0 FI comma Line 1 Read the process 0 FI comma Line 1 Read the process 0	Column Z1 current po of device a and Column Z1 current po 0 of device coordinates	addres 00_C 1 osition addres 00_C 1 00_C 1 osition addres 00_C	ss 00. CPO1_0_2 Answe Column 2 -5.98975 a of the Z axiss 00. The ress CR_CPO1_0_2 Answe Column 2 -0.23582 a of the Z axidress 00 (e.g	1 Column 3 [mm] s in machine coult is to be output 1_inch r Column 3 [inch] s in machine coult . physical axis r	Column 4 -5.990 ordinates in NC in inches. Column 4 -0.236 ordinates in NC

Reference to Literature See chapter entitled "Literature" [25].



Trigger Control Reset: CRT

Designation	CRT C	ontrol-Reset			
Explanation		eset allows the sele If there is no syste			
		Carrying out a device. During initializat interrupted (inhere	on, commur	•	
FI command	CW_CRT	(Sing	le Write)		
Value to be written	Trigger reset	0			
		e value to be writte the "DataTransfer" r		the "acVa	ue" parameter
Response Structure	has been su information in the routine "R	lue of the "DataTrar uccessfully complet the form of a gene eadGroupItem" (ref r Result Line").	ed. In the e ral error result	event of an	n error, more e requested by
Example CRT	Triggers a co	ntrol reset on the se	ected device.		
	FI command	00_CW	_CRT		
	Value to be wi	itten 0			
Reference to Literature	See chapter e	entitled "Literature" [2	26].		
Device Axis Configurat	tion Paran	neter: DAC			
				MWC	X device group
Designation	DAC De	vice A xis C onfigurat	on 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 cu	rrent parameters of	all configured	device axes	
	BR_DAC1	(S	ingle Read)		
Response Structure	"DAC1" FI co	table shows the g mmand. The numbe he number of config	r of answer lin	es [132 pe	er NC process]
	1	.ine 1n:	Column 1	1	Column 11



	Note:			chine parameter record in the device are not applicable.
Value Range/Meaning	1 =	Physical axi	s number	[132]
of Columns	2 =	NC process		[06]
	3 =	Assigned pr	ocesses	[06,]
	4 =	Type of axis	i	[see Chapter 6.2 "Data Tables"]
	5 =	APR numbe	er	[15]
	6 =	APR axis nu	ımber	[18]
	7 =	Main axis m	eaning	[see the chapter entitled "Data Tables"]
	8 =	Secondary a	axis meaning	[see the chapter entitled "Data Tables"]
	9 =	Main axis na	ame	[Xi, Yi, Zi, Ui, Vi, Wi, Ai, Bi ,Ci, Si,] (i=[], [13])
	10 =	Secondary a	axis name	[Xi, Yi, Zi, Ui, Vi, Wi, Ai, Bi ,Ci, Si,] (i=[], [13])
	11 =	Assigned ax	is number	[132,]
Reference to Literature	See cha	pter entitled	"Literature" [27]	l.
Example DAC1			arameters of al record of device	I configured device axes of the active address 00.
	<u>Assump</u> The follo		levice axes hav	e 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	х	
		10		
		10		
	2	1	2	
	2	2	0	
		3		
		4	81	
		5	1	
		6	2	



	FI comm	and	00_BR_DAC1	
	Line	Column	Answer	
		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		
			<u> </u>	
FI command				e selected device axis type.
	BR_DAC	• •	(Single Read	
	(1) = axis	з туре	$1^{1} = 0$ niy digita	al axes, 2 = only analog axes]
Response Structure	"DAC2" F	I command	d. The number	neral structure of the response to th of answer lines [132] depends on th Each line consists of 11 columns.
		Line 1	.n	Column 1 Column 11
	Note:			chine parameter record in the devic] are not applicable.
Value Range/Meaning	1 =	Axis numbe	r	[132]
of Columns	2 =	NC process	number	[06]
	3 =	Assigned pr	ocesses	[06,]
	4 =	Type of axis	3	[see chapter entitled "Data Tables", Axis Types]
	5 =	APR numbe	er	[15]
	6 =	APR axis nu	umber	[18]
	7 =	Main axis m	eaning	[see chapter entitled "Data Tables" Axis Meanings]
	8 =	Secondary a	axis meaning	[see chapter entitled "Data Tables" Axis Meanings]
	9 =	Main axis na	ame	[Xi, Yi, Zi, Ui, Vi, Wi, Ai, Bi ,Ci, Si,] (i=[], [13])
	10 =	Secondary a	axis name	[Xi, Yi, Zi, Ui, Vi, Wi, Ai, Bi ,Ci, Si,] (i=[], [13])

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 comma	and	00_BR_DAC2
Line	Column	Answer
1	1	1
	2	0
	3	
	4	81
	5	1
	6	1
	7	0
	8	
	9	Х
	10	
	11	

NC D-Correction Download: DCA

	MWCX device group							
Designation	DCA	NC- D-C 0	prrection Access					
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 p	bath 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 [0120] (see Chapter "FI Commands for the MPCX Device Group", IFJ).							
		I command	d ance with Chapter "Elements of the FI Command"]					
		FI Job Error Chapter "Err						
Example DCA1	00_BW_[DCA1_"D:\F	Program Files\Indramat\Mtgui\Temp\download.ini"/3					
	FI comma	and	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 I	be taken in the u	se of upper and lower case letters.				
	This is cu	rocess, then		cessing, i.e., if an error is detected <i>ror</i> key is written with "YES" within				
	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 Package Package Package	size: " Packa time: " Packa date: " Packa	ageName" ageSize" max. 8	characters left-justified max. 8 characters max. 8 characters				
	Total:			max. 84 characters				
	Informatic Date : Time:	n on date an dd.mm hh:mm	••	a the format				
	Example:							
	Package	lame = Size = Time =	1					
		DCorrection	_ A] a process numb	per [06]				
			rection_A]) is an s not regarded as	optional entry, i.e., if a section for s an error.				
	values ar	e the write va		correction numbers [199] and the ctions (L1, L2, L3, R, unit optional).				
	[DCorrect 001=L1 1	ion_0] .0 L2 2.0 L3 3	3.0 R 4.0					
	099=L1 1).0 L2 20.0 L	3 30.0 R 40.0					
	[DCorrect 001=L1 1 	ion_1] .0 L2 2.0 L3 3	3.0 R 4.0 mm					
	050=L1 1).0 L2 20.0 L	3 30.0 R 40.0 mi	m				
	[DCorrect 001=L1 1 	ion_6] .0 L2 2.0 L3 3	3.0 R 4.0					
	099=L1 1).0 L2 20.0 L	3 30.0 R 40.0					

NC D-Correction Upload: DCA

MWCX device group

Designation	DCA NC-D-Correction Access					
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 f	ile 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 [0120] (see Chapter "FI Commands for the MPCX Device Group", IFJ). 					
	-	2 = FI comr g, in accord	nand dance with Chapter "Elements of the FI Command"]			
		3 = FI Job E Chapter "Eı	Error Code rror Codes")			
Example DCA1	00_BR_DCA1_"D:\Program Files\Indramat\Mtgui\Temp\Upload.ini"/3					
	FI comm	and	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	•	ad file is str [Common]	uctured in the Windows – "Ini" format structure.			
			is stored in the COMMON section.			
		whether o nly set in th	r not an error has occurred during uploading. This ne event of an error.			
	[Commor UploadEr		; error			
	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 identifications is described in the following: Package number"PackageNo" max. 2 characters Package name "PackageName" max. 32 characters					



Total:			ma	ax. 84 chara	acters		
Information Date : Time:		m.yy	time	is given	in	the	forma
Example:							
[DCorrection	Package	e_Info]					
PackageNo =		1					_
PackageNan		D 1234				CC	orrection
PackageSize PackageTim		1234	1				
PackageDate		24.12.00					

[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

Reading D-Correction Data: DCD

Designation	DCD	D-Correction Data					
Explanation	The values of a D-correction register of the selected NC process are read out.						
	that effects	e D-corrections are additive to the tool geometry data of the register at effects the tool management, i.e. they are additive to the existing cometry registers L1, L2, L3 and R.					
	processes. L2, L3 and	e 99 D-correction numbers available for each of the seven NC es. Each D-correction number therefore contains the registers L1, and R. Value assignment of the D-correction register is via the Indramat GUI or via the function interface.					
FI command	Reading of a D-correction register value of an NC process of the selected device.						
		<u>(1)_(2)_(3){_(4)}</u> <u>(1)_(2)_(3){_(4)}</u>	(Single Read) (Cyclic Read)				



		4 (4) (2) (2)(<i>(</i> 4))		(P	rook Cuolio F			
		1_(1)_(2)_(process nur			•	(Break Cyclic Read) [06]			
	• /	prrection nu			-	[00]			
	()			on re					
	(3) = Number of the D-correction (4) = Required measurements				• • •				
	(4) = Required measurement system				n (opt.) [m	(opt.) [mm, inch]			
		Using the optional fourth parameter it is pos of the result into mm or inches.					t conversion		
Response Structure	The response consists of on (length correction L1 to L3 a requested D-correction regis settings of the process parame			and ra ster, a	adius correc	tion R), the v	value of the		
	Line				Column 1	Column 2	Column 3		
Value Range/Meaning	1 = Identifier			[] 1	L2, L3, R]				
of Columns		alue of D-cc	rrection	-	-	dina to settina	of the		
				[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-corregister is not assigned a value then the value 0 is our response, formatted according to the settings in the parameters.								
Example DCD1						device addre correction R).	ss 00 in NC		
	FI comma		00_CR_I		•	,			
					swer				
	Line	Colum	n 1	Column 2		Colu	mn 3		
	1	R			0.0860	[m	ım]		
Reference to Literature	See chap	ter entitled	"Literatur	e" [6].					
e Configuration F	aramet	ter: DCF	2						
						MWCX of	device group		
Designation	DCP	Device Co	onfiguratio	on P ar	ameter				
Explanation	machine p The confi the device	parameter i guration pa	record as arameters evice typ	well a s of th	as in the "INE ne device inc	e entered in D_DEV.INI" file clude the dev mber, mecha	e are output. ice address,		
FI command	-	-	tion paraı		s of all define	ed devices.			
	BR_DCP	1		(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).

Device



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 1n: Column 1 Column 7

Note:	If no active machine parameter record exists in the device,
	then the columns [17] for the respective device are not applicable.

Value	Range/Meaning	
	of Columns	

1 =	device address	[0063]
2 =	Device name	[max. 32 ASCII characters]
3 =	Device Type	[MTC200-P-G2, MTC200-R-G2, MTVNC]
4 =	Mechanism number	[031]
5 =	Mechanism name	[max. 28 ASCII characters]
6 =	Process type	[1= internal, 2 = external process]
7 =	Process type	[1 = NC Process, 2 = PLC Process]

Read the device configuration parameters of all defined devices. Example DCP1

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 comma	and	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_DCF	22		(Single Rea	d)			
Response Structure			ows the genera ponse consists			e to the "DCP2"		
		Line 1		Column 1		Column 7		
	Note:	Note: If no active machine parameter record exists in the device, then the columns [17] for the respective device are not applicable.						
Value Range/Meaning	1 =	device addr	ess	[0063]				
of Columns	2 =	Device nam	е	[max. 32 AS	CII charac	ters]		
	3 =	Device Type	e	[MTC200-P- MTVNC, M				
	4 =	Mechanism	number	[031]				
	5 =	Mechanism	name	[max. 28 ASCII characters]				
	6 =	Process type		[1= internal, 2 = external process]				
	7 =	Process typ	e	[1 = NC Process, 2 = PLC Process]				
Example DCP2	address <u>Assumpt</u>	01). <u>ion:</u>		ameter of the	e selected	device (device		
			been defined					
			00 (MTC200-R)					
			01 (MTC200-P) 02 (MTC200-P)					
	FI comm Line	Column	01_BR_DCP2					
	1	1	Answer 01					
		2	0					
		3	MTC200-P					
		4	0					
	5 Milling machine 01							
		6	1					
		7	1					
		<u> </u>	<u> </u>	•				

Reference to Literature

See chapter entitled "Literature" [28].

C Correction Register: DCR

MWCX device group

Designation DCR D-Correction Record

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 exisitng 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 Rexroth Indramat GUI or via the function interface.

FI command	Readin device	ng of a D-correction	record	of an NC	process of	the selected		
	CR_DO	CR1_(1)_(2){_(3)}		(Single	Read)			
	CC_DC	CR1_(1)_(2){_(3)}		(Cyclic	(Cyclic Read)			
	CB_DC	CR1_(1)_(2){_(3)}		(Break ((Break Cyclic Read)			
	(1) = N	C process number		[06]	[06]			
	(2) = D	-correction number		[199]	[199]			
	(3) = R (opt.)	equired measurement	systen	n [mm, inc	[mm, inch]			
		the optional third para esult into mm or inche		it is possible	to pre-sele	ct conversion		
	Note:	If the value of a s the command "CR				be read then		
Response Structure	identifie the rec	esponse consists of for er (length correction L quested D-correction re s of the process param	l to L3 egister	and radius c	orrection R)), the value of		
		Line 10.4		Column 1		Column 3		
Value Range/Meaning of Columns	1 = 2 =	Identifier Value of D-correction	- [Form					
	3 =	Unit	[mm,	settings of process parameters] [mm, inch; according to parameter settings]				
	Note:	If the requested register is not ass response, formatt parameters.	signed	a value then	the value () is output as		
Example DCR1	Read t	he value of all D-corre	ection	registers at c	levice addre	ess 00 of NC		

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

without optional Parameters

Write all D-correction register values of an NC process of the selected device.

CW_DCR_(1)_(2)

(1) = NC process number

(Single Write) [0...6]



	(2) = D-correction num		ber	[199]		
Value to be written	D correc	tion register	[L1 <val opt. uni</val 	ue> L2 <value> L3<value> R<value>] t]</value></value></value>		
	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 conv unavoidable		rom mm \rightarrow inch, <u>rounding errors are</u> ision is lost!		
		formatting s	should be s param	are separated by a space, whereby the e carried out according to the settings of neters. (see example DCR1: write D		
Response Structure	One line is output with a column for acknowledgement of whether or not the FI command has been executed successfully.					
	(P_ACK)) = Positive AC	CKnowled	dge Value has been written		
Example DCR Write D-Correction Register	D-correct	ion number 1		at device address 00 of NC process 0 of ollowing 5 values:		
		e L1: 1.2586				
		e L2: 3.5892				
		e L3: 0.0000 ai	na			
	 Value R: 0.0860 Unit of the values: mm (optional) 					
	5. Office	n the values. I				
	Note:		he "acVa	ritten are passed in the "Data Transfer" alue" parameter and must be separated space " ".		

FI command		00_CW_DCR_0_1 Value to write: L1 1.2586 L2 3.5892 L3 0 R 0.086 mm
Line	Column	Answer
1	1	(P_ACK)

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

Setting the Communication Timeout Time DCTMWCX 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_DCT (1) = requ	-1_(1) uested timeout time in m	(Single Write) s			
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 = 5	Status mess	sage (P_ACk	(P_ACK)	
Example DCT1	For the device 00, the timeout time is set 1500 ms.				
	FI comma	and	00_BW_DCT	1_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_DC	Г2		(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 de	evice 00, th	e timeout time	e is reset to the default value.	
	FI comma	and	00_BW_DCT	2	
	Line	Column	Answer		
	1	1	(P_ACK)		
Long Identification of I	NC/PLC	Data R	ecords: [DIS	
				MWCX device group	
Designation	DIS	Data Ide	ntification S tri	ng	
Explanation	Reads the long identification (directory entries) of NC/PLC data records.				

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:
	o i

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

	Line 1 C		Column 1		Column 5
Value Range/Meaning of Columns		Number in NC parameter Name of the NC parameter		[0199] [max. 32 AS characters]	CII



- 3 = Length of the NC parameter record [byte]
- 4 = Date of creation/last change to NC [DD.MM.YY] parameter record 5 = Time of creation/last change to NC [HH:MM:SS]
- 5 = Time of creation/last change to NC [HH:MM:SS] parameter record
- **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 comma	and	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	[0199]		
	2 =	Name of the PLC program	[max. 8 ASCII characters]		
	3 =	Length of the PLC progra	[byte]		
	4 =	Date of creation/last change to PLC program		[DD.MM.YY]	
	5 =	Time of creation/last cha PLC program	[HH:MM:SS]		
	6 =	Date of creation/last char program	nge to PLC	[DD.MM.YY	YY]
	Note:	If there is no valid PLC program in to columns contain [].		ne selected de	evice then all



Rexroth Indramat

	Assumption: There is a valid PLC program in the selected device.					
	FI comm		00_BR_DIS2			
	Line	Column	Answer	<u> </u>		
	1	1	01			
		2	KEY1			
		3	20018			
		4	10.05.99			
		5	12:42:00			
		6	10.05.1999			
Reference to Literature	See chap	ter entitled	"Literature" [3	30].		
FI command	Output the directory entries of the valid NC package of the select memory.			selected NC		
	BR_DIS	3_(1)	(Single Read)			
	BC_DIS	3_(1)	(Cyclic R	ead)		
	BB_DIS	3_(1)	(Break Cyclic Read)			
	(1) = NC	1) = NC memory [1 = NC memory A; 2 = NC memory B]			B]	
Response Structure	e The following table shows the general structure of the response to the FI command. The response consists of a line with five columns.				e to the "DIS3"	
	Line 1			Column 1		Column 5
Value Range/Meaning	1 = Number in NC package directory		[0199]			
of Columns	2 = Name of the NC package			e	[max. 32 ASC characters]	
	3 = Length of the NC package			[byte]		
	4 = Date of creation/last change to NC [DD.MM.YY] package					
	5 = Time of creation/last change to NC [HH:MM:SS] package					
	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 a address 00. <u>Assumption:</u> 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			

Example DIS2 Read the directory entries of the PLC program at address 00. Assumption:

Reference to Literature See chapter entitled "Literature" [31].



FI command	Output the directory entries of the valid tool list of the selected N process.					selected NC
	BR_DIS4	↓_(1)	(Single Rea	ld)		
	BC_DIS4	L_(1)	(Cyclic Rea	d)		
	BR_DIS4	Ⅰ_(1)	(Break Cyc	lic Read)		
	(1) = NC	process nu	mber [()6]		
Response Structure				eral structure of s of a line with		e to the "DIS4"
		Line 1		Column 1		Column 5
Value Range/Meaning	1 =	Number in th	ne tool list inde	x	[0199]	
of Columns	2 =	Name of the	e tool list		[max. 32 AS characters]	CII
	3 =	Length of th	e tool list		[byte]	
		Date of creatool list	ation/last char	nge to the	[DD.MM.YY]
		Time of created tool list	ation/last cha	nge to the	[HH:MM:SS	5]
	Note:		s no valid tool contain [] .	list in the sel	ected NC pro	ocess then all
Example DIS4	Read the directory entries of the tool list of NC process 0 at device address 00. <u>Assumption:</u> There is a valid tool list in NC process 0 of the selected device.					
	There is	a valid tool	· ·		elected devic	e.
	FI comm	and	00_BR_DIS4		elected devic	e.
	FI comm Line	and Column	00_BR_DIS4 Answer		elected devic	:e.
	FI comm	nand Column 1	00_BR_DIS4 Answer 01		elected devic	:e.
	FI comm Line	and Column	00_BR_DIS4 Answer 01 KEY1		elected devic	:e.
	FI comm Line	Column 1 2	00_BR_DIS4 Answer 01		elected devic	:e.
	FI comm Line	Column 1 2 3	00_BR_DIS4 Answer 01 KEY1 2048		elected devic	:e.
Reference to Literature	FI comm Line 1	Column 1 2 3 4 5	00_BR_DIS4 Answer 01 KEY1 2048 17.09.99	_0	elected devic	:e.
Reference to Literature FI command	FI comm Line 1	Column 1 2 3 4 5 ter entitled he directory	00_BR_DIS4 Answer 01 KEY1 2048 17.09.99 10:45:08 "Literature" [3	_0		
	FI comm Line 1 See chap Output t	And Column 1 2 3 4 5 5 0ter entitled he directory device.	00_BR_DIS4 Answer 01 KEY1 2048 17.09.99 10:45:08 "Literature" [3	_ 0 32]. the valid ma		
	FI comm Line 1 See chap Output t selected	And Column 1 2 3 4 5 5 5 5 5	00_BR_DIS4 Answer 01 KEY1 2048 17.09.99 10:45:08 "Literature" [3 y entries of	_ 0 32]. the valid ma		
	FI comm Line 1 See chap Output t selected BR_DIS	A Column 1 2 3 4 5 bter entitled he directory device. 5 5	00_BR_DIS4 Answer 01 KEY1 2048 17.09.99 10:45:08 "Literature" [3 y entries of (Single Real	0 32]. the valid main id) d)		
	FI comm Line 1 See chap Output t selected BR_DIS BB_DIS The follow	A column Column 1 2 3 4 5 5 5 5 5 wing table sh	00_BR_DIS4 Answer 01 KEY1 2048 17.09.99 10:45:08 "Literature" [3 y entries of (Single Rea (Cyclic Rea (Break Cycle)	0 32]. the valid main id) d)	chine data r	ecord in the
FI command	FI comm Line 1 See chap Output t selected BR_DIS BB_DIS The follow	A column Column 1 2 3 4 5 5 5 5 5 wing table sh	00_BR_DIS4 Answer 01 KEY1 2048 17.09.99 10:45:08 "Literature" [3 y entries of (Single Rea (Cyclic Rea (Break Cycle)	0 32]. the valid main id) d) lic Read) eral structure of	chine data r	ecord in the
FI command	FI comm Line 1 See chap Output t selected BR_DIS BB_DIS The follow FI comma	A Column 1 2 3 4 5 5 5 5 5 5 5 5 5 5 5 5 5	00_BR_DIS4 Answer 01 KEY1 2048 17.09.99 10:45:08 "Literature" [3 y entries of (Single Rea (Cyclic Rea (Break Cycle)	_0 32]. the valid man id) d) lic Read) vral structure of s of a line with Column 1	chine data r	e to the "DIS5"
FI command	FI comm Line 1 See chap Output t selected BR_DIS BB_DIS The follow FI comma	A Column	00_BR_DIS4 Answer 01 KEY1 2048 17.09.99 10:45:08 "Literature" [3 y entries of (Single Rea (Cyclic Rea (Break Cycl nows the gene ponse consist	_0 32]. the valid man id) d) lic Read) vral structure of s of a line with Column 1	f the response five columns.	e to the "DIS5"



4 =	Date	of	creation/last	change	to	the	[DD.MM.YY]
	data r	eco	ord				

5 = Time of creation/last change to the [HH:MM:SS] data record

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 comm	and	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) = NC memory	[1 = NC memory A, 2 = NC memory B]
(2) = NC process number	[06]
(3) = NC program number	[199]

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	[0199]		
	2 =	Number of the NC progra	am	[0199]	
	3 =	Name of the NC program	1	[max. 32 ASCII characters]	
	4 =	Length of the NC program	[byte]		
	5 =	Date of creation/last char program	nge to NC	[DD.MM.YY]]
	6 =	Time of creation/last cha program	nge to NC	[HH:MM:SS]
	Note:	If there is no valid No then all columns conta		the selected	NC process



Example DIS6

e DIS6 Read the directory entries of the third NC program (NC package number 2, NC memory A, NC process 0) at device address 00. <u>Assumption:</u>

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].

Delete NC Program: DPN

Designation	DPN	Delete P	rogram NC			
Explanation	An NC pro	An NC program located in an NC package directory is deleted.				
FI command	BW_DPN	BW_DPN_(1)_(2)_(3)_(4) (Single Write)				
	(1) = NC	package di	rectory number	[199]		
	(2) = NC	process nu	mber	[0 6]		
	(3) = NC	program nu	umber	[199]		
	(4) = with	check / wit	thout 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)					
		()		NC program not delated		
			_CREATED)	NC program not deleted.		
	Without of	check (0)				
		CT_OK) = n CT ion_ O	K	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 chap	ter entitled	"Literature" [31].			



Delete NC Program Package: DPP

MWCX device group

Designation	DPP	DPP Delete Program Package					
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 p	orogram pa	ackage [199]				
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) =Program package has been deleted.BOF_FunCTion_OK						
Example DPP	The NC program package numbered 1 in the NC package directory is to be deleted.						
	FI comma	and	00_BW_DPN_1_2_3_0				
	Line	Column	Answer				
	1	1	(BOF_FCT_OK)				

Reference to Literature See chapter entitled "Literature" [17].

Reading the Device Status Information: DSI

Designation	DSI	Device S	tatus Information		
Explanation		vs the most i information is		atus information	to be read out. The
	Type of	information	Status	Statement	
	System e	error informa	tion		
	Informati	on on mecha	anism error		
	Machine	key informat	ion	valid	Yes/No
	Machine	key informat	ion		
	Machine status information				
	Sercans information				
	Paramet	er download		running	Yes/No
	PLC dov	nload		running	Yes/No
	Firmware	e download		running	Yes/No
	Offline/C	nline informa	ation		
	Device s	imulation		switched on	Yes/No
	Device status information ON, OFI				
FI command	Read out	device statu	us information for	ALL defined dev	ices.
	BR_DSI1		(Single Read)		
	BC_DSI1		(Cyclic Read)		
	BB_DSI1		(Break Cyclic Re	ead)	



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 show "DSI1" FI command.	s the general s	structure of the i	response to the
		<u> </u>		

		Line 1n	Colum	n 1		Column 13
Value Range/Meaning	1 =	device address		[0063]		
of Columns	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 inform	Machine key information [4 byte in HEX coding]			
	5 =	ls machine key info valid?	ormation	[0 = not valid, 1=valid]		
	6 =	Machine status info	ormation	[4 byte	in HEX coding]	
	7 =	Sercans information		[4 byte in HEX coding]		
	8 =	ls parameter down active?	load		rameter downloa rameter downloa	-
	9 =	Is PLC download a	active?		C download not C download runi	
	10 =	ls firmware downlo active?	ad	[0 = PLC download not running] 1 = PLC download running]		
	11 =	Offline/Online infor	mation	-	vice connection i	
	12 = Device sim switched o		1	[0 = NO Simulation mode 1 = simulation mode]		ode
	13 =	Current device st information	atus		eviceStatus=OF eviceStatus=ON	

 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 comm	and	01_BR_DSI1
Line	Column	Answer
1	1	01
	2	0
	3	0
	4	0000000
	5	0
	6	0000000
	7	0000000
	8	0
	9	0
	10	0
	11	1
	12	0
	13	1
2	1	03
	2	1
	3	0
	4	0000000
	5	0
	6	0000000
	7	0000000
	8	0
	9	0
	10	0
	11	1
	12	0
	13	1

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 1n	Column	1		Column 13
Value Range/Meaning	1 =	device address		[0063]		
of Columns	2 =	System error information		[0 = there is no system error 1 = there is a system error]		
	3 =	Information on me	chanism		ere is no mechar ere is a mechani	
	4 =	Machine key inform	nation	[4 byte	in HEX coding]	
	5 =	Is machine key info valid?	ormation	[0 = no	ot 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]

Example DSI2 Read the current device status information for the selected device.

FI comma	and	01_BR_DSI2
Line	Column	Answer
1	1	00
	2	0
	3	0
	4	0000000
	5	0
	6	0000000
	7	0000000
	8	0
	9	0
	10	0
	11	1
	12	0
	13	1

Tool Management Configuration Data: DTC

Designation	DTC Device Tool Management Configuration				
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	5		N], [PROGR	4M]]	

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	
BC	DTC2	

(Single Read) (Cyclic Read)

Response Structure

One line with 48 columns is output for the returned values.

		Line 1	Column 1		Column 48
eaning of the	1 =	Tool Management	[YES,	NO]	
Columns	2 =	Setup list	[[STA	TION], [PROC	GRAM]]
	3 =	Max. number of tool edge	s [19]		
	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	[[TUR	N./MILL.], [GF	RINDING]]
	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]	

Value Range/Meaning of the Columns

Note:	If there is no tool manage	ment (Column 1: NO), then all pa
48 =	Cutter user date 10	[Cutter user date,]
47 =	Cutter user date 10	[YES, NO]
46 =	Cutter user date 9	[Cutter user date,]
45 =	Cutter user date 9	[YES, NO]
44 =	Cutter user date 8	[Cutter user date,]
43 =	Cutter user date 8	[YES, NO]
42 =	Cutter user date 7	[Cutter user date,]
41 =	Cutter user date 7	[YES, NO]
40 =	Cutter user date 6	[Cutter user date,]
39 =	Cutter user date 6	[YES, NO]
		[Cutter user date,]
37 = 38 =	Cutter user date 5	[YES, NO]
30 = 37 =	Cutter user date 5	[Cutter user date,]
35 = 36 =	Cutter user date 4	
34 = 35 =	Cutter user date 3	[YES, NO]
33 = 34 =	Cutter user date 3	[Cutter user date,]
32 = 33 =	Cutter user date 3	[Cutter user date,] [YES, NO]
31 = 32 =	Cutter user date 2	
30 = 31 =	Cutter user date 2	[YES, NO]
29 = 30 =	Cutter user date 1	[Cutter user date,]
20 = 29 =	Cutter user date 1	[YES, NO]
27 = 28 =	Tool user date 9	[Tool user date,]
20 – 27 –	Tool user date 9	[YES, NO]
20 = 26 =	Tool user date 8	[Tool user date,]
24 – 25 =	Tool user date 8	[YES, NO]
23 – 24 =	Tool user date 7	[Tool user date,]
23 =	Tool user date 7	[YES, NO]
22 =	Tool user date 6	[Tool user date,]
20 =	Tool user date 6	[YES, NO]
20 =	Tool user date 5	[Tool user date,]
19 =	Tool user date 5	[YES, NO]
18 =	Tool user date 4	[Tool user date,]
17 =	Tool user date 4	[YES, NO]
16 =	Tool user date 3	[Tool user date,]
16 =	Tool user date 3	Tool user date1

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



FI comma	and	00_BR_DTC2
Line	Column	Answer
	40	
	41	NO
	42	
	43	NO
	44	
	45	NO
	46	
	47	NO
	48	

Reference to Literature See chapter entitled "Literature" [8].

Distance to Go of Axis Movement: DTG

Designation	DTG	Distance To Go				
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		e distance to go of the selec he code of the axis meaning.	ted axis of the device specified,			
	of the resu		possible to pre-select conversion ever, a spindle is selected as an serves no purpose.			
	CR_DTG	1_(1)_(2)_(3){_(4)}	(Single Read)			
	CC_DTG	1_(1)_(2)_(3){_(4)}	(Cyclic Read)			
	CB_DTG	1_(1)_(2)_(3){_(4)}	(Break Cyclic Read)			
	(1) = NC p	process number	[06]			
	(2) = Axis	meaning	[011; 20]; (see chapter "Data Tables")			
	(3) = Syst	em of coordinates	[1 = machine coordinates2 = program coordinates]			
	(4) = Req (opt.)	uired measurement system	[mm, inch]			
FI command		e distance to go of the move cified related to the physical a	ment of the selected axis of the xis number.			
	Using the optional third parameter it is possible to pre-select conversio of the result into mm or inches. If, however, a spindle is selected as a axis, indicating a measurement system serves no purpose.					
	CR_DTG	2_(1)_(2){_(3)}	(Single Read)			
	CC_DTG	2_(1)_(2){_(3)}	(Cyclic Read)			
	CB_DTG	2_(1)_(2){_(3)}	(Break Cyclic Read)			
	(1) = Phys	sical axis number	[132, according to settings of the system parameters]			
	(2) = Syst	em of coordinates	[1 = machine coordinates			



	(3) = Required measurement system [mm, inch] (opt.)					
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	1 = Axis	name	[according	g to settings of	axis param	eters]
of Columns	2 = Dista	nce to go		ing to settings of process parameters]		
	3 = Unit		[mm, inch]		
	4 = Dista	nce to go		in 2, but round to the parame		
	Note:	If the specifie NC process th				the selected
Example DTG1		distance to g es in NC proces				in machine
	FI comma	and 00_	CR_DTG1_	0_2_1		
			Ar	iswer		
	Line 1	Column 1	Colum		mn 3	Column 4
	1	Z1	-1.234	15 [m	m]	-1.235
Example DTG1		distance to g es in NC proces				
	FI comma	and 00_	CR_DTG1_	0_2_1_inch		
			Ar	iswer		
	Line 1	Column 1	Colum	n 2 Colu	mn 3	Column 4
	1	Z1	-0.048	36 [in	ch]	-0.049
Example DTG2		distance to go 3) in machine o				
	FI comma	and 00_	CR_DTG2_	_3_1		
			Ar	iswer		
	Line 1	Column 1	Colum			Column 4
	1	Z1	-1.234	15 [m	m]	-1.235
Reference to Literature	See chap	ter entitled "Lite	rature" [16].		



Device Type and Accompanying Components: DTY

MWCX device group

Designation	DTY	Device	ТҮре			
Explanation	The device type and the accompanying components of the selected device address are output.					he selected
FI command	BR_DT	Y1	(Single Rea	d)		
Response Structure	The following table shows the ge "DTY1" FI command. A line with type, as well as the names of the f the second device component.			three columns	is output fo	r the device
		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		nt type1 INE	DEV.INI entry	: Componer	nttype1=
	3 = Component type 2			IND_DEV.INI entry: Componenttype2=		
Example DTY1	Output the device type and the accompanying components of device address 00.					s of device
	FI command 00_BR_DTY1					
	Answer					
	L	ine	Column 1	Column	2 C	olumn 3
		1	MTC200-P-G2	2 MTS-P		MTC-P

Diagnosis Window Data: DWD

Designation	DWD	Diagnosis Wind	ow 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 o	liagnostic messa	ges.		
	BR_DWD	1_(1){_(2)}	(Single Read)		
	BC_DWD	1_(1){_(2)}	(Cyclic Read)		
	(1) = Type window	of diagnosis	 [1 = NC error, 2 = sequence errors, 3 = general errors, 4 = messages, 10 = start requirements, 11 = warnings, 12 = setup diagnosis] 		
	(2) = Mod	ule number	[199] ! only for window type 1 -4 !		
	Output first	diagnostic mess	ages.		
	BR_DWD	2_(1){_(2)}	(Single Read)		
	BC_DWD	2_(1){_(2)}	(Cyclic Read)		
	(1) = Typ window	be of diagnosis	 [1 = NC error, 2 = sequence errors, 3 = general errors, 4 = messages, 10 = start requirements, 		



11 = warnings, 12 = setup diagnosis]

	(2) = M	lodule number [1	99] !	only for v	vindow type 1	-4 !
Response Structure	FI comr pending	owing table shows the g nands. The number of . Different columns are re no messages, the nu	lines o valid	depends o according	n the numbe to the type (r of messages
		Line 1n	Co	lumn 1		Column 12
Meaning of the Columns	1 =	Message text		[ASCII c	haracters]	
	2 =	Time stamp day		[mm.dd.]	уууу]	
	3 =	Time stamp hour		[hh:mm:	ss]	
	4 =	Reference text availab	le	[YES, N	D]	
	5 =	Type of diagnosis		-	√i, 2 = SFC, C-NC, 4 = MT	A-NC]
	6 =	Message number		[ASCII c	haracters]	
	7 =	Message ID			haracters] D, decimal) (l	ProVi)
	8 =	Mechanism number		[031] (N	/TC-NC) [0]	(MTA-NC)
	9 =	2 byte additional inform	nation	[ASCII c	haracters] (N	ITC NC)
	10 =	Message group		[19999] (MTA-NC)	
	11 =	SFC entity name		[ASCII c	haracters]	
	12 =	NC note		[ASCII c	haracters] (N	ITC NC)
	13 =	Analysis of criteria ava	ilable	[YES, N	O] (ProVi, SF	C)
	14 =	Message HTML file		[ASCII c NC)	haracters] (P	roVi, MTC-

Example DWD1 All diagnostic messages from Module 3 in Control unit 0.

There are two m	essages	s 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

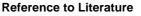
See chapter entitled "Literature" [13].



MWCX device group

Existing errors: EDE

Designation	EDE	DE Existing Diagnosis Error					
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 EDE		(Cyclic Read)				
Response Structure	ructure The following table shows the general structure of the "EDI 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.						
i i command	BR_EDE2_(1) (Single Read)						
	BC_EDE		(Cyclic Read)				
	$(1) = Module number \qquad [199]$						
Response Structure	ucture The following table shows the general structure of the "EDE2 command.						
	Line 1			Column 1			
Meaning of the Columns	1 = Error exists [YES, NO]				10]		
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].						



See chapter entitled "Literature" [13].



Existing Diagnosis Window: EDW

Designation	EDW Existing Diagnosis Window					
Explanation	Which types of diagnosis window exist is queried.					
FI command	Output al	types of di	agnosis w	indow.		
	BR_EDV		-	(Single Read)		
Response Structure	command	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 0n		Column 1	Column 2	
Meaning of the Columns	1 = Type	of diagnos	is window	3 = general errors 10 = start requirer	, 4 = messages,	
	2 = Modu	ule number		[ASCII characters] 0 = Diagnosis winc belong to any mod		
Example EDW1	All types	of diagnosis	s window i	n control unit 0.		
	There are	e three diag	nosis winc	lows.		
	FI comm		00_BR_E	DW1		
	Line	Column	Answer			
	1	1	10 0			
	2	1	1			
	2	2	3			
	3	1	2			
		2	3			
FI command	BR_EDV	V2_(1)	-	indow for one module (Single Read) [199]	3.	
Response Structure	 (1) = Module number [199] The following table shows the general structure of the "EDW2" F command. The number of lines depends on the number of types o window existing. 					
		Line 0n		Column 1	Column 2	
Meaning of the Columns	window	of diagnos		1 = NC error, 2 = sequence errors, 3 = general errors, 4 = messages]		
	0 =			[ASCII characters] 0 = Diagnosis window type does not belong to any module		

	FI command		00_BR_EDW2_3		
	Line	Column Answer			
	1	1	1		
		2	3		
	2	1	2		
		2	3		
FI command	Query a s	pecific type	of diagnosis wir	ndow.	
	BR_EDV	V3_(1){_(2)	} (Single R	lead)	
	(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[199] ! only for window type 1 -4 !				
Response Structure	The follo	-	shows the ge	neral structure of the "EDW3" FI	
		Line '	1	Column 1	
Meaning of the Columns	1 = Type	of diagnos	is window exists	[YES, NO]	
Example EDW3	Query whether or not a NC error window exists in module 3, control unit 0.				
	FI comma	and	00_BR_EDW3_1	_3	
	Line	Column	Answer		
	1	1	YES		

Example EDW2 All types of diagnosis window in Module 3, Control unit 0.

There are two diagnosis windows.

Existing NC Diagnoses: END

Designation	END Existing NC Diagnosis					
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 type	s are available in a module.				
	BR_END1_(1)	(Single Read)				
	(1) = Module number	[199]				
Response Structure	The following table shows the command.	e general structure of the "END1" FI				
	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 comma	and	00_BR_END1_2		
	Line	Column	Answer		
	1	1	NO		
		2	YES		
FI command	Query a s	pecific NC	diagnostic type.		
	BR_END	02_(1)_(2)	(Single Read)	
	(1) = Me:	ssage type	[^	1 = error, 2 = messages]	
	(2) = Mo	dule numbe	er [1	199]	
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]			YES, NO]	
Example END2	Are there any messages in module 4 in control unit 0?				
	FI comma	and	00_BR_END2_2	_4	
	Line Column Answer				

YES

1

1

Example END1 Query the NC diagnostic types in Module 2 on Control unit 0.

Existing PLC Diagnoses: EPD

Designation	EPD Existing PLC Diagnosis					
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	requiremer	nt exists	[YES, NO]		
	2 = Warr	ning exists		[YES, NO]		
	3 = Setu	p diagnosis	exists	[YES, NO]		
Example EPD1	Query PL	C diagnosti	c types in control	l unit 0.		
	FI comma	and	00_BR_EPD1			
	Line	Column	Answer			
	1	1	YES			
		2	NO			
		3	YES			
FI command	Query wh	ich PLC dia	agnostic types are	e available in a module.		
	BR_EPD2_(1)			Single Read)		
	$(1) = Module number \qquad [199]$					



Response Structure	The following table shows the general structure of the "EPD2" F command.				
	Line 1			Column 1-3	
Meaning of the Columns	1 = Mess	ages exist		[YES, NO]	
	2 = Error	s exist		[YES, NO]	
	3 = Step	chains exis	st	[YES, NO]	
Example EPD2	Query the	PLC diagn	ostic types in Mc	dule 2 on Control unit 0.	
	FI comma	and	00_BR_EPD2_2		
	Line	Column	Answer		
	1	1	NO		
		2	YES		
		3	YES		
FI command	Query a s	pecific PLC	diagnostic type.		
	BR_EPD	3_(1){_(2)}		•	
	(1) = Mes	ssage type		= messages, 3 = SFC, gs, 11 = start requirements, liagnosis]	
	(2) = Moo	dule numbe	er [199] ! 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 = Diagi	nosis type e	exists [YES	S, NO]	
Example EPD3	Are there	any messa	ges in module 4	in control unit 0?	
	FI comma	and	00_BR_EPD3_2_	4	
	Line	Column	Answer		
	1	1	YES		

End Point of an Axis Movement: EPO

MWCX device group

- Designation EPO End POsition
- **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)} CC_EPO1_(1)_(2)_(3){_(4)} CB_EPO1_(1)_(2)_(3){_(4)} (1) = NC process number (2) = Axis meaning (3) = System of coordinates (4) = Required measurement system (opt.)					(Cyc) (Brea [06] [01 (see [1 = r 2 = p	Jle Read) lic Read) ak Cyclic Re] 1; 20]; chapter "Dat nachine coo orogram coo inch]	a Tables") rdinates
FI command	Output the	(opt.) Output the end point of the selected axis to the physical axis number.				of the	device spec	ified, related
	Using the of the res	optional t	hird para n or inch	es. I	f, howe	ever, a		ect conversion selected as an
	CR_EPC	2_(1)_(2){	_(3)}	(S	ingle R	ead)		
	CC_EPC	92_(1)_(2){	_(3)}	(C	yclic R	ead)		
	CB_EPC	2_(1)_(2){	_(3)}	(В	reak C	yclic I	Read)	
	(1) = Phy	vsical axis r	number			-	2, according stem param	to settings of eters]
	(2) = Sys	tem of coo	rdinates				nachine coor rogram coor	
	(3) = Re (opt.)	quired me	asureme	nt s	ystem	[mm,	inch]	
Response Structure	The following table shows the general structure of the response to the commands "EPO1" and "EPO2". One line is output with 4 columns for axis designation, end point of the movement, unit and the position line to "indicated decimal places".					columns for the		
	Lir	ne 1	Column	n 1	Colur	nn 2	Column 3	Column 4
Value Range/Meaning	1 – Δxis	name		ordi	na to se	ottinas	of axis nara	metersl
of Columns		Axis name [according to setting end point [according to setting			-		-	
	3 = Unit				0			
	4 = end p	point	acc		ng to the	, but rounded up or down he parameter "indicated decimal		
	Note:	If the spe then the						ed NC process
Example EPO1		end point o cess 0 of d				ne Z a	ixis in machi	ne coordinates
	FI comma	and	00_CR_	EPO	1_0_2_1			
			·		Answer			
	Line	Column	1	Colu	mn 2	С	olumn 3	Column 4
	1	Z1		-1.2	345		[mm]	-1.235
Example EPO1							ixis in machi are displayed	ne coordinates d in inches:



FI comma	and 0	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 comma	and 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].

Existing ProVi Types: EPT

Designation	EPT Existing ProVi Types				
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 type	S.		
	BR_EPT	1		(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 0n		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 t	ypes in cor	ntrol unit 0	. There are three dia	gnosis windows:
	FI comma	and	00_BR_E	PT1	
	Line	Column	Answer		
	1	1	20		
		2	0		
	2	1	11		
		2	3		
	3	1	12		
		1	3		



Error Status: EST

MWCX device group

Designation	EST	Error ST	ate		
Explanation	Queries the error state of a variable.				
FI command	Query the	<u>frozen</u> erro	or status of a vari	able.	
	BR_EST	1!(1)!(2)	(Single	Read)	
	BC_EST	1!(1)!(2)	(Cyclic	Read)	
	(1) = Erro	or ID	[ASCII	characters] (DWORD, decimal)	
	(2) = Var	iable name	[ASCII	characters]	
	Note:	The sepa	rator "!" is used i	n this command.	
Deenenee Cómieture	The following table shows the general structure of the "EXD1" FI command.				
Response Structure		-	shows the ge	neral structure of the "EXD1" FI	
Response Structure		-		neral structure of the "EXD1" FI	
Meaning of the Columns		Line			
	command 1 = Error Read the	Line	1 'inPcl variable "IE		
Meaning of the Columns	command 1 = Error Read the at device Assumption	Line state value of W address 00 on:	1 'inPcl variable "IE).	Column 1	
Meaning of the Columns	command 1 = Error Read the at device <u>Assumption</u> The WinF	Line state value of W address 00 on: Pcl variable	1 "inPcl variable "IE). "IB_EXT24" is d	Column 1 8_EXT24" in WinPcl program "Prog",	
Meaning of the Columns	command 1 = Error Read the at device <u>Assumption</u> The WinF "Prog".	Line state value of W address 00 on: Pcl variable	1 "inPcl variable "IE). "IB_EXT24" is d	Column 1 B_EXT24" in WinPcl program "Prog", eclared as BOOL in WinPcl program	

Execution Display: EXD

Designation	EXD	EXecution Display					
Explanation	Information for displaying the execution of a movement is output.						
FI command	Query the	Query the execution of a step or of an action.					
	BR_EXD	1!(1)!(2)!(3)	(Single Read)				
	BC_EXD	1!(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 follow command.		eneral structure of the "EXD1" FI				
		Line 1	Column 1				



1 = Exec	ution	[1 – ca execut	an be executed, 0 – cannot be ted]
		of the step "oper	n" for the chain "clamp" in control unit
FI comma	and	00_BR_EXD1!St	ation03A.Clamp!Open!1
Line	Column	Answer	
1	1	1	
Query wh enabled.	ether the o	condition analysi	s (control image) of a step chain is
BR_EXD	2!(1)	(*	Single Read)
(1) = SF0	C entity nan	ne [/	ASCII characters]
Note:	The sepa	rator "!" is used i	n this command.
	•	shows the ge	eneral structure of the "EXD2" FI
	Line	1	Column 1
1 = Enab	led		[1 - enabled, 0 – not enabled]
Query wh enabled.	nether the	condition analys	sis of the "clamp" chain has been
FI comma	and	00_BR_EXD2!St	ation03A.Clamp
Line	Column	Answer	
1	1	1	
	Query the 0 for all m FI comma Line 1 Query wh enabled. BR_EXD (1) = SFO Note: The follo command 1 = Enab Query wh enabled. FI comma Line	0 for all modes. FI command Line Column 1 1 Query whether the orenabled. BR_EXD2!(1) (1) = SFC entity name Note: The sepa The following table command. Line 1 = Enabled Query whether the enabled. FI command Line Column	execution of the step "oper 0 for all modes. FI command 00_BR_EXD1!St Line Column Answer 1 1 1 Query whether the condition analysi enabled. BR_EXD2!(1) ((1) = SFC entity name) Note: The separator "!" is used if The following table shows the ge command. Image: Column analysi enabled. Image: I

Global Process Parameter Configuration: GPC

MWCX device group

Designation GPC Global Process Configuration

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	"GPC1" maximur	FI commar	nd. The responses (n= max. nu	nse consists	of betwe	esponse to the een one and a ocesses [06] =
		Line 1		Column 1		Column 12
	Note:		s no active ma columns [10.1			d in the device
Value Range/Meaning	1 = 1	NC process	number		[06]	
of Columns	2 =	Name of the	NC process		[max. charae	20 ASCII cters]
	3 = 1	Basic coordii	nate system		[mm,	inch]
		Programmeo decimal poin	l number of pos t	itions after	[4, 5]	
	5 = 1	Displayed po	sitions after the	decimal point	[04]	
	6 = 1	Max. zero-po	oint-data bank n	umber	[09]	
	7 =	D corrections	3		[YES,	NO]
	8 =	Home positio	on required		[YES,	NO]
	9 =	Reference required			[YES,	NO]
	10 = 0	Cartesian-polar coordinate transformation			[YES,	NO]
		Manual axis jogging causes reset			[YES,	-
	12 =	Tool storage	axis repositioni	ng	[YES,	NO]
	Assumpt The follo Sled 1 (N Turret 1	<u>ion:</u> wing three N IC process I (NC process	e machine para NC processes a number 0), s number 1) an s number 3).	are defined:		iduless oo.
	FI comm	and	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]			
	L	1	1			



	FI comm	and	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		
Deference to Literature	See ober	tor optitlad	"Litoroturo" [24]		
Reference to Literature	See chap		"Literature" [34].		
FI command			ation of the global process pecord of the selected device re		
	BR_GPC	2_(1)	(Single Read)		
	(1) = NC	process nui	mber [06]		
Response Structure			nows the general structure of the sponse consists of a line with 1:		
		Line 1	Column 1		Column 12
	Note:	the selec	no active machine paramete cted NC process is not de re not applicable.		
Value Range/Meaning	1 = N	NC process	number	[06]	
of Columns			NC process	[max. 20 /	
	3 = E	Basic coordi	nate system	[mm, incl	-
	4 = F		d number of positions after	[4, 5]	
			altions after the sheater at a stat	10 41	

5 = Displayed positions after the decimal point [0...4]



6 =	Max. zero-point-data bank number	[09]
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 comma	and	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].

Global Process Parameters: GPP

Designation	GPP	Global Process Parameter
Explanation	the select includes point for	al process parameters of the active machine parameter record of cted device from the MWCX device group is read out. This the programmable and actually displayed digits after the decimal the displacement, the name of the CC process, the basic the 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		f the configuration of the global process parameters of all defined esses of the active machine parameter record. P1 (Single Read)





Response Structure	"GPC1" F	I command (n= max. n	shows the ger d. The respons umber of defin	e consists o	f one up to a	a maximum of
		Line 1		Column 1		Column 6
	Note:		s no active ma columns [16]			in the device
Value Range/Meaning	1 = 1	NC process	number		[06]	
of Columns			NC process			20 ASCII
	3 = E	Basic coordi	nate system		[mm, inch]	
	4 = F		d number of po			
	5 = D		ositions after	the decimal	[04]	
	•		oint-data bank n	umber	[09]	
Example GPP1	machine p <u>Assumpti</u> The follov Sled 1 (N Turret 1 (oarameter re on: wing three N C process NC process	ess parameters ecord of device NC processes a number 0), s number 1) an s number 3).	address 00. are defined:	NC process	es of the active
	FI comm	and	00_BR_GPP1			
	Line	Column	Answer			
	1	1	0			
		2	Sled 1			
		3	[mm]			
		4	4			
		5	3			
		6	0			
	2	1	1 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			ation of the glo ecord of the sele			of the active NC process.
	BR_GP	P2_(1)	(S	Single Read)		
	(1) = NC	process nu	umber [0	6]		
Response Structure			ows the genera ponse consists			
		Line 1		Column 1		Column 6
					•	
	Note:	the selec		ess is not d		n the device or the columns
Value Range/Meaning	1 =	NC process	number		[06]	
of Columns	2 =	Name of the NC process			[max. 20 A characters]	
	3 =	Basic coordii	nate system		[mm, inch]	
		Programmed number of positions after decimal point			[4, 5]	
		Displayed po point	sitions after the	decimal	[04]	
	6 = 1	Max. zero-po	oint-data bank n	umber	[09]	
Example GPP2		the global process parameter in NC process 0 of the active parameter record of device address 00.				
			VC processes a number 0),	are defined:		
		· ·	s number 1) an	d		
			s number 3).	•		
	FI comm Line	Column	00_BR_GPP2_ Answer	_0		
	1	1	0			
		2	Sled 1			
		3	[mm]			
		4	4			
		5	3			
		6	0			
Reference to Literature	See chap	oter entitled	"Literature" [35	j].		

Insert NC Program Package: IPP

Designation	IPP Insert NC-Program	Package
Explanation	Inserts an NC program package	e into the NC package directory.
FI command	BW_IPP_(1){_(2)}	(Single Write)
	(1) = Number in NC package directory	[199]
	(2) = Is the NC package	[0 = without check (preset);

	dir	ectory entr	y empty?	1 = with check] ! Optional !
	Note:	number o	of the NC packa	age already exists at the selected age directory, an error is signaled if s been selected.
Value to be written	Name o	f the NC pa	ackage	[max. 32 ASCII characters]
	Note:			s passed to the "acValue" parameter DataTransfer" routine.
Response Structure			column is outpu g of the elements	t to acknowledge the FI command is as follows:
	(P_ACK	() = P ositive	e ACKnowledge	NC package has been entered.
Example IPP		NC progra IC package		the designation "KEY1" into number
	<u>Assumpti</u>	on:		
	It is to be is emp		hether the select	ted entry in the NC package directory
	FI comma	and	00_BA_PPP_1 Value to be writ	ten: KEY1
	Line	Column	Answer	
	1	1	(P_ACK)	

Module Assignment of a Process: MAP

Designation	MAP Modul Assign of I	Process
Explanation	"Moduldef.ini" file. This	ular process is assigned is read out from the file is located in the directory //TGUI\CustomData\Resource" and le configurations.
	The process data is located in	three sections:
	[DeviceAddrX\ModulY	\Process]
	whereby "X" stands for the c module numbers.	evice addressed and "Y" for the configured
FI command		ch the process belongs. Information is read ation of the MWCX device group.
	BR_MAP1_(1)	(Single Read)
	BC_MAP1_(1)	(Cyclic Read)
	BB_MAP1_(1)	(Break Cyclic Read)
	1 = Mechanism number	[031]
Response Structure		e general structure of the response to the with one column is output for the module ined.
	Line 1	Column 1
Value Range of the Column	1 = Module number	[099]

Example MAP1	Road the	module n	umber to which NC	process number 4 is assigned	
		module con		process number 4 is assigned	
	<u>Assumpti</u> The mode		NC process 4 is ass	signed has module number 5.	
	FI comm	and	00_BR_MAP1_4		
	Line	Column	Answer		
	1	1	5		
Reference to Literature	See chap	ter entitled	"Literature" [36].		
Read Reference Name	of a PL	.C Varia	ble: MAR		
				MWCX device group	
Designation	MAR	Map Abs	solute PCL- R eference	e	
PLC Explanation	The abso	lute referer	nce name of a symbo	lic PLC variable is read out.	
FI command	Read the	absolute re	ference name of a PL	.C variable.	
	BR_MAR	_(1)	(Single R	ead)	
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.				
		Lin	ne 1	Column 1	
Meaning of the Column	1 = Ident	ifier of the l	PLC variable		
	Read the absolute reference name of the PLC variable with the identifier "abref" at device address 00.				
PLC – Example MAR				e PLC variable with the identifier	
PLC – Example MAR	"abref" at <u>Assumpti</u>	device add <u>on:</u>	lress 00.	e PLC variable with the identifier	
PLC – Example MAR	"abref" at <u>Assumpti</u>	device add <u>on:</u> variable wit	lress 00.		
PLC – Example MAR	"abref" at <u>Assumpti</u> The PLC	device add <u>on:</u> variable wit	lress 00. th the identifier "abret		
PLC – Example MAR	"abref" at <u>Assumpti</u> The PLC FI comm	device add <u>on:</u> variable wit and	Iress 00. th the identifier "abret 00_BR_MAR_abref		
PLC – Example MAR WinPcl-Explanation	"abref" at <u>Assumpti</u> The PLC FI comm Line 1 The abso	device add on: variable wit and Column 1	Iress 00. th the identifier "abref 00_BR_MAR_abref Answer %M100.0 nce name of a sym		
	"abref" at <u>Assumpti</u> The PLC FI comm Line 1 The abso program	device add on: variable wit and Column 1 blute refere entity is rea	Iress 00. th the identifier "abref 00_BR_MAR_abref Answer %M100.0 nce name of a sym	" is of the type "INTEGER".	
WinPcl-Explanation	"abref" at <u>Assumpti</u> The PLC FI comm Line 1 The abso program	device add on: variable wit and Column 1 blute refere entity is rea absolute re	Iress 00. th the identifier "abref 00_BR_MAR_abref Answer %M100.0 nce name of a sym d out.	f" is of the type "INTEGER".	
WinPcl-Explanation	"abref" at <u>Assumpti</u> The PLC FI comm Line 1 The abso program Read the BR_MAN	device add on: variable wit and Column 1 blute refere entity is rea absolute re	Iress 00. th the identifier "abref 00_BR_MAR_abref Answer %M100.0 nce name of a sym d out. ference name of a W	f" is of the type "INTEGER".	
WinPcl-Explanation	"abref" at <u>Assumpti</u> The PLC FI comm Line 1 The abso program Read the BR_MAH (1) = Ide Read the	device add on: variable wit and Column 1 blute refere entity is rea absolute re R1_(1) entifier of th e absolute	Iress 00. th the identifier "abref 00_BR_MAR_abref Answer %M100.0 nce name of a sym d out. ference name of a W (Single R e PLC variable	f" is of the type "INTEGER".	
WinPcI-Explanation Fl command	"abref" at <u>Assumpti</u> The PLC FI comm Line 1 The absor- program Read the BR_MAH (1) = Ide Read the identifier <u>Assumpti</u> The Win	device add on: variable wit and Column 1 Dute refere entity is rea absolute re R1_(1) entifier of th e absolute "Prog.abref on: Pcl variab	Iress 00. th the identifier "abref 00_BR_MAR_abref Answer %M100.0 nce name of a sym d out. ference name of a W (Single R e PLC variable reference name of " at device address 0	" is of the type "INTEGER".	
WinPcI-Explanation Fl command	"abref" at <u>Assumpti</u> The PLC FI comm Line 1 The absor- program Read the BR_MAH (1) = Ide Read the identifier <u>Assumpti</u> The Win	device add on: variable wit and Column 1 olute refere entity is rea absolute re R1_(1) entifier of th e absolute "Prog.abref on: Pcl variab R" and is pr	Iress 00. th the identifier "abref 00_BR_MAR_abref Answer %M100.0 nce name of a sym d out. ference name of a W (Single R e PLC variable reference name of " at device address 0 le with the identifie	f" is of the type "INTEGER".	
WinPcI-Explanation Fl command	"abref" at <u>Assumpti</u> The PLC FI comm Line 1 The absor- program Read the BR_MAR (1) = Ide Read the identifier <u>Assumpti</u> The Win "INTEGE	device add on: variable wit and Column 1 olute refere entity is rea absolute re R1_(1) entifier of th e absolute "Prog.abref on: Pcl variab R" and is pr	Iress 00. th the identifier "abref 00_BR_MAR_abref Answer %M100.0 nce name of a sym d out. ference name of a W (Single R e PLC variable reference name of " at device address 0 le with the identifier resent in WinPcl prog	f" is of the type "INTEGER".	
WinPcI-Explanation Fl command	"abref" at <u>Assumpti</u> The PLC FI comm Line 1 The absor- program Read the BR_MAH (1) = Ide Read the identifier <u>Assumpti</u> The Win "INTEGE FI comm	device add on: variable wit and Column 1 blute refere entity is rea absolute re R1_(1) entifier of th e absolute "Prog.abref on: Pcl variab R" and is pr and	Iress 00. th the identifier "abref 00_BR_MAR_abref Answer %M100.0 nce name of a sym d out. ference name of a W (Single R e PLC variable reference name of " at device address 0 le with the identifier resent in WinPcl prog 00_BR_MAR1_:Prog	f" is of the type "INTEGER".	

Reference to Literature See chapter entitled "Literature" [30].

Device Data of the Module Configuration: MCD

MWCX device group

Designation	MCD	Modul Confi	guration	: D evice Infor	mation	
Explanation	"Moduldef.ir "[LW]:\Prog following st	ni" file ramme\Indra tandard inst	wh mat\MT allation.	ich is GUI∖CustomD	stored ata\Resourc data is ir	the sections
FI command	Read out o device grou		within t	he module c	onfiguration	of the MWCX
	BR_MCD1		ingle Re	ead)		
	BC_MCD1	(C	yclic Re	ead)		
	BB_MCD1	(В	reak Cy	clic Read)		
	ſ	MWCX devic	e group		ny valid devi	ces within the ce address can 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.					
	Li	ne 1	С	olumn 1		Column 4
Value Range of the Columns	1 = Device	address			[063]	
	2 = PLC-FE	3 name for th	ne setup	diagnostics	[max. character	9 ASCII s]
	3 = PLC-FE	3 name for th	ne warni	ng messages	[max. character	9 ASCII s]
	4 = PLC-FE	3 name for th	ne start i	requirements	[max. character	9 ASCII s]
Example MCD1	Read all device data of the module configuration.					
	Assumption: The following devices have been configured in the MWCX device group:					
	Device a	ddress 01 (N	/ITC200	-P)		
	Device a	ddress 03 (N	/ITC200	-P)		
	FI comman	d 03_E	BR_MCD	1		
	Answer			-		
	Line	Col	umn 1	Answer Column 2	Column 3	Column 4
	Line 1	Col	umn 1 01		Column 3 PVWarn_1	
		Col		Column 2		PVStart_1

Reference to Literature

See chapter entitled "Literature" [36].



Module Data of the Module Configuration: MCM

MWCX device group

Designation	МСМ	Modul Con	figuration: M odul Infor	mation	
Explanation	All module data for module configuration is read out from the "Moduldef.ini" file which is stored in the "[LW]:\Programme\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			from the module co device group.	nfiguration wi	th respect to a
	BR_MCM1		Single Read)		
	BC_MCM1	()	Cyclic Read)		
	BB_MCM1	(1	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				Column 4
	1 = Module number [099]				
Value Range of the Columns	1 = Module	e number		[099]	
Value Range of the Columns	1 = Module 2 = Module			[099] [max. character:	28 ASCII s]
Value Range of the Columns	2 = Module	e name	general module errors	[max. character	s] 9 ASCII
Value Range of the Columns	2 = Module 3 = PLC-F	e name B name for	general module errors module messages	[max. character [max.	s] 9 ASCII s] 9 ASCII
Value Range of the Columns Example MCM1	2 = Module 3 = PLC-F 4 = PLC-F Read the m <u>Assumption</u> The followin	e name B name for B name for nodule data <u>n:</u> ng modules	-	[max. character: [max. character: [max. character:	s] 9 ASCII s] 9 ASCII s]
	2 = Module 3 = PLC-F 4 = PLC-F Read the m <u>Assumption</u> The followin • Module	e name B name for B name for nodule data <u>n:</u> ng modules number 5	module messages of device 03 from the i	[max. character: [max. character: [max. character:	s] 9 ASCII s] 9 ASCII s]
	2 = Module 3 = PLC-F 4 = PLC-F Read the m <u>Assumption</u> The followin • Module	e name B name for B name for nodule data ng modules number 5 number 7	module messages of device 03 from the i have been defined:	[max. character: [max. character: [max. character:	s] 9 ASCII s] 9 ASCII s]
	2 = Module 3 = PLC-F 4 = PLC-F Read the m <u>Assumption</u> The followin • Module	e name B name for B name for nodule data ng modules number 5 number 7	module messages of device 03 from the have been defined: _ BR_MCM1	[max. character: [max. character: [max. character:	s] 9 ASCII s] 9 ASCII s]
	2 = Module 3 = PLC-F 4 = PLC-F Read the m <u>Assumption</u> The followin • Module • Module	e name B name for B name for nodule data ng modules number 5 number 7 nd 03_	module messages of device 03 from the have been defined: _BR_MCM1 	[max. character: [max. character: [max. character: module config	s] 9 ASCII 9 ASCII s] uration:
	2 = Module 3 = PLC-F 4 = PLC-F Read the m <u>Assumption</u> The followin Module Module Fl comman <u>Line</u>	e name B name for B name for nodule data nodule data number 5 number 7 nd 03_ Column 1	module messages of device 03 from the have been defined: BR_MCM1 Answer Column 2	[max. character: [max. character: [max. character: module config	s] 9 ASCII 9 ASCII s] uration: Column 4
	2 = Module 3 = PLC-F 4 = PLC-F Read the m <u>Assumption</u> The followin Module Module	e name B name for B name for nodule data ng modules number 5 number 7 nd 03_	module messages of device 03 from the have been defined: _BR_MCM1 	[max. character: [max. character: [max. character: module config	s] 9 ASCII 9 ASCII s] uration:

Reference to Literature

See chapter entitled "Literature" [36].



Process Data of the Module Configuration: MCP

MWCX device group

Designation	МСР	Modul C	onfiguration: P roc	ess Information
Explanation	All process data of a certain module which is si "[LW]:\Programme\Indramat\MTGUIV following standard installation. This data. The process data [DeviceAddrX\ModulY\Process], who addressed and "Y" for the selected mod			red in the customData\Resource" directory le contains all module configuration is located in sections eby "X" stands for the device
FI command	BR_MCP	1_(1)	(Sing	le Read)
Ti command	BC_MCP	1_(1)	(Cycl	ic Read)
	BB_MCP	1_(1)	(Brea	k Cyclic Read)
	(1) = Mod	lule numbe	r [099	9]
Response Structure	The response to the FI comman maximum number of n=32 lines wit process or of the external mechanis			1 column for the number of the NC
	p100000 0			5.
		Line 1.		S. Column 1
Value Range of the Column			32	Column 1
Value Range of the Column Example MCP1	1 = Mech	Line 1. nanism num NC proces	32 nber [031	Column 1
-	1 = Mech Read the configurat Assumptio	Line 1. nanism num NC procestion. on:	32 nber [031	Column 1
-	1 = Mech Read the configurat <u>Assumption</u> The follow	Line 1. nanism num NC procestion. on:	32 hber [031 ss number of mo ocesses are define	Column 1
-	1 = Mech Read the configurat <u>Assumption</u> The follow NC proce	Line 1. nanism num NC proces tion. <u>on:</u> ving NC pro	32 hber [031 ss number of mo ocesses are define 1	Column 1
-	1 = Mech Read the configurat <u>Assumption</u> The follow NC proce	Line 1. nanism num NC procestion. on: ving NC pro ss number ss number	32 hber [031 ss number of mo ocesses are define 1	Column 1
-	1 = Mech Read the configurat <u>Assumptio</u> The follow NC proce	Line 1. nanism num NC procestion. on: ving NC pro ss number ss number	32 hber [031 ss number of mo ocesses are define 1 4	Column 1
-	1 = Mech Read the configurat <u>Assumption</u> The follow NC proce NC proce FI comma	Line 1. nanism num NC procestion. <u>on:</u> ving NC pro ss number ss number	32 hber [031 ss number of mo ocesses are define 1 4 00_BR_MCP1_5	Column 1
-	1 = Mech Read the configurat <u>Assumptio</u> The follow NC proce NC proce FI comma Line	Line 1. nanism num NC procestion. on: ving NC pro ss number ss number and Column	32 hber [031 ss number of mo bcesses are define 1 4 00_BR_MCP1_5 Answer	Column 1

Reference to Literature See chapter entitled "Literature" [36].

SFC Data of the Module Configuration: MCS

MWCX device group

MCS Modul Configuration: SFC Information Designation All SFC data of a certain module is read out from the "Moduldef.ini" file Explanation which is stored in the "[LW]:\Programme\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_MCS	1_(1)		(Cyclic Read)	
	BB_MCS			(Break Cyclic Read)	
	(1) = Mod	ule number	r	[099]	
Response Structure	chains fo		. Each	s on the number of configured Indrastep step line contains a column for the name of the	
Value Range of the Column	1 = Name	e of the Indr	astep st	tep chain [format W.X.Y.Z]	
	Fo	rmat W.X.Y.	Z	Value Range	
		W		Max. 9 ASCII characters	
		Х		Max. 9 ASCII characters ! OPTIONAL !	
		Y		Max. 9 ASCII characters ! OPTIONAL !	
		Z		Max. 9 ASCII characters ! OPTIONAL !	
	 Read the name of the Indrastep step chain of module 5 from device 03 of the module configuration. <u>Assumption:</u> 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 comm	and	03_BR	_MCS1_5	
	Line	Column	Answe		
	1	1	ISFB_1		
	2	1		.ISFB_3	
	3	1		.ISFB_3.SW1	
	4	1	FB_US	.ISFB_3.SW1.ABBA	
Reference to Literature	See chap	ter entitled	"Literatu	ıre" [36].	
Machine Data Downloa	d: MDA	4			
				MWCX device group	
Designation	MDA	Machine	Data A	ccess	
Explanation	Complete	e machine d	lata reco	ords are downloaded by means of a download	

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:

file.

Long MachineDataDownloadBegin(Long IProjectNumber,

Long IDeviceNumber,

Long IIndexNumber,

Char* pcMDLFileName,

Long IMaxLengthFileNameBuffer,

Char* pcErrorText,

Long IMaxLengthErrorTextBuffer)



Pass parameters: IProjectNumber: Currently selected project number IDeviceNumber: Currently selected device address IIndexNumber: Currently selected machine data record directory number [1..99] Contains the complete file names for the pcMDLFileName: created machine data record download file IMaxLengthFileNameBuffer: Max. length of the buffer for the name the machine data of record download file. pcErrorText: Text of user error, if applicable IMaxLengthErrorTextBuffer: 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 IResult) Pass parameters: Contains the complete file names for the pcMDLFileName: created machine data record download file. **IResult:** 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) = (1) = Machine data record directory number; the two functions to be implemented are located in INDIF410.DLL. (2) = Complete DLL name, if required, in (2) = which the two functions to be implemented are located. The response to the "MDA1" FI command consists of three lines, each **Response Structure** 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_011_1		D:\UserDir\USER.DLL"			
	FI comma	and	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			download command whereby the machine data record tly indicated.			
	BW_MD	A2_(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 [0120] (see Chapter "FI Commands for the MPCX Device Group", IFJ). 					
	[String		lance with Chapter "Elements of the FI Command"]			
		3 = FI Job E Chapter "Ei	rror Code rror Codes")			
	Note:	File and p	bath details must be enclosed in inverted commas.			
Example MDA2			DOWNLOAD.DAT"			
	FI comma		00_BW_MDA2_"D:\DOWNLOAD.DAT"			
	Line	Column	Answer			
		4				
	1	1				
	2	1	00_BW_MDA2_"D:\DOWNLOAD.DAT"			
		-				
Structure of Download File	2 3 The struc file. V21_Mad detailed a Summary Section [1 ture of the Rexroth at_Downlo ccount of the D_MACHI	00_BW_MDA2_"D:\DOWNLOAD.DAT" 0 download file corresponds to that of a Windows Ini Indramat's own description in ad_Upload_01.doc, is recommended for a more he structure of the download file.			
Structure of Download File	2 3 The struc file. V21_Mad detailed a <u>Summary</u> Section [Informatic Section [1 1 ture of the Rexroth at_Downloa ccount of the D_MACHI D_ Concerni ID_TYPE_	00_BW_MDA2_"D:\DOWNLOAD.DAT" 0 download file corresponds to that of a Windows Ini Indramat's own description in ad_Upload_01.doc, is recommended for a more he structure of the download file. NE_DATA] ng the identification of the machine data record. DEFINITION]			
Structure of Download File	2 3 The struc file. V21_Mad detailed a <u>Summary</u> Section [Informatic Section [1 ture of the Rexroth at_Downloa ccount of the ID_MACHI on concerni ID_TYPE_ on concerni TYPE_DEF	00_BW_MDA2_"D:\DOWNLOAD.DAT" 0 download file corresponds to that of a Windows Ini Indramat's own description in ad_Upload_01.doc, is recommended for a more he structure of the download file. NE_DATA] ng the identification of the machine data record.			
Structure of Download File	2 3 The struc file. V21_Mad detailed a Summary Section [Informatic Section [Max. data Section [1 1 ture of the Rexroth at_Downloa ccount of the D_MACHI on concerni ID_TYPE_DEI type identi TYPE_DEI	00_BW_MDA2_"D:\DOWNLOAD.DAT" 0 download file corresponds to that of a Windows Ini Indramat's own description in ad_Upload_01.doc, is recommended for a more he structure of the download file. NE_DATA] ng the identification of the machine data record. DEFINITION] ng the identification of the type definition. FINITION_INFO]			
Structure of Download File	2 3 The struc file. V21_Mad detailed a <u>Summary</u> Section [Informatic Section [Max. data Section [Data for th Section [1 1 ture of the Rexroth at_Downloa ccount of the D_MACHI on concerni ID_TYPE_DEI type identi TYPE_DEI	00_BW_MDA2_"D:\DOWNLOAD.DAT" 0 download file corresponds to that of a Windows Ini Indramat's own description in ad_Upload_01.doc, is recommended for a more he structure of the download file. NE_DATA] ng the identification of the machine data record. DEFINITION] ng the identification of the type definition. FINITION_INFO] ification number FINITION_XXX] type definitions. fo]			
Structure of Download File	2 3 The struc file. V21_Mad detailed a <u>Summary</u> Section [Informatic Section [Max. data Section [Data for th Section [Max. defin Section [1 ture of the Rexroth at_Downloa ccount of the ID_MACHI on concerni ID_TYPE_ on concerni TYPE_DEI type identi TYPE_DEI he various PAGE_INF ned page n D_PAGE_I	00_BW_MDA2_"D:\DOWNLOAD.DAT" 0 download file corresponds to that of a Windows Ini Indramat's own description in ad_Upload_01.doc, is recommended for a more he structure of the download file. NE_DATA] ng the identification of the machine data record. DEFINITION] ng the identification of the type definition. FINITION_INFO] ification number FINITION_XXX] type definitions. fo]			
Structure of Download File	2 3 The struc file. V21_Mad detailed a <u>Summary</u> Section [Informatic Section [Max. data Section [Data for th Section [Max. defin Section [Section [Max. defin Section [Section [Max. defin]	1 1 ture of the Rexroth at_Downloa ccount of th D_MACHI on concerni ID_TYPE_DEI on concerni TYPE_DEI type identi TYPE_DEI he various PAGE_INF ned page n D_PAGE_I on concerni	00_BW_MDA2_"D:\DOWNLOAD.DAT" 0 download file corresponds to that of a Windows Ini Indramat's own description in ad_Upload_01.doc, is recommended for a more he structure of the download file. NE_DATA] ng the identification of the machine data record. DEFINITION] ng the identification of the type definition. FINITION_INFO] fication number FINITION_XXX] type definitions. CO] umber. DEFINITION_XXX] ng the identification of the page definition. FINITION_XXX]			

00_BW_MDA1_3_"D:\UserDir\USER.DLL"



	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.				
	With this command, ALL machine data page definitions in the selected device are deleted.				
	BW_MD	A4		(Siı	ngle Write)
	The respo column.	onse to the	"MDA4" FI co	mmand consi	sts of one line with one
		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				
	The following machine data page definitions have been deleted for the device 00.				
	FI comma	and	00_BW_MDA	4	
	Line	Column	Answer		
	1	1	1,2,10,11,12,2 3,104	21,30,40,50,60,	61,62,90,91,92,101,102,10

Machine Data Upload: MDA

Designation	MDA	Machine Data Access				
Explanation	Uploads complete machine data records from a selected device. The d read is written into an upload file with an identical structure to that o download file.					
FI command	are to be	programmed by the user.	nand whereby two predefined functions These two functions concern: plete name of the upload file:			
	Long IVIa	achineDataUploadBegin(
			Long IDeviceNumber,			
			Char* pcUploadFileName,			
			Long IMaxLengthFileNameBuffer,			
			Char* pcErrorText,			
			Long IMaxLengthErrorTextBuffer)			
	Pass par	ameters:				
	 IProj 	ectNumber:	Currently selected project number			
	 IDev 	iceNumber:	Currently selected device address			
		bloadFileName: nachine	Contains the complete file name for data record upload file to be created.			

	• IMax	LengthFileN	ameBuffer:	max. length of the buffer for the name of the machine data record upload file.
	• pcEr	rorText:		If necessary, user error text
	 IMax 	LengthError	TextBuffer:	Max. length of the buffer for the user error text.
	2. Func	tion called u	p at the end	of the machine data record upload:
	Long Ma	achineDataU	ploadEnd(Char* pcUploadFileName,
				Long IResult)
	<u>Pass par</u>	ameters:		
	pcUpload	dFileName:		ontains the complete file names for the nachine data record upload file.
	IResult:			ontains the status message of the
			uploading Here,	process of the machine data record.
			,	neter upload procedure O.K.
				or has occurred
	The two exported		ust be progr	ammed in a DLL by the user and also
	BR_MD	A1_(1)_{(2)}	}	(Single Read)
	 (1) = Machine data record directory number; the two functions to be implemented are located in 			
	i	number; the	two function d are located	s to be
	(2) =	number; the implemented INDIF410.DI Complete DI in which the	two function d are located	s to be in equired, s to be
Response Structure	(2) = The resp	number; the implemented INDIF410.DI Complete DI in which the implemented ponse to the	two function d are located LL. LL name, if ru two functions d are located "MDA1" FI	s to be in equired, s to be
Response Structure	(2) = The resp with one • Line	number; the implemented INDIF410.DI Complete DI in which the implemented oonse to the column. The 1 = Job ID 20] (see Cha	two function d are located LL. LL name, if re two functions d are located "MDA1" FI meaning of	s to be in equired, s to be command consists of three lines, each
Response Structure	(2) = The resp with one • Line [01 IFJ"). • Line	number; the implemented INDIF410.DI Complete DI in which the implemented bonse to the column. The 1 = Job ID 20] (see Cha. $2 = FI comm$	two function d are located LL. LL name, if re- two functions d are located "MDA1" FI meaning of apter "FI Co	s to be in equired, s to be command consists of three lines, each the elements is as follows:
Response Structure	 (2) = The resp with one Line [01 IFJ"). Line [Strin Line 	number; the implemented INDIF410.DI Complete DI in which the implemented bonse to the column. The 1 = Job ID 20] (see Cha. $2 = FI comm$	two function d are located LL. LL name, if re- two functions d are located "MDA1" FI meaning of apter "FI Co nand ance with Ch rror Code	s to be in equired, s to be command consists of three lines, each the elements is as follows: mmands for the MPCX Device Group:
Response Structure	 (2) = The resp with one Line [01 IFJ"). Line [Strin Line 	number; the implemented INDIF410.DI Complete DI in which the implemented oonse to the column. The 1 = Job ID 20] (see Cha 2 = FI comm g, in accord 3 = FI Job E Chapter "En	two function d are located LL. LL name, if re- two functions d are located "MDA1" FI e meaning of apter "FI Co nand ance with Ch rror Code ror Codes")	s to be in equired, s to be command consists of three lines, each the elements is as follows: mmands for the MPCX Device Group:
Response Structure	(2) = The resp with one • Line [01 IFJ"). • Line [Strin • Line (see Note:	number; the implemented INDIF410.DI Complete DI in which the implemented oonse to the column. The 1 = Job ID 20] (see Cha 2 = FI comm g, in accord 3 = FI Job E Chapter "Em File and p	two function d are located LL. LL name, if re- two functions d are located "MDA1" FI e meaning of apter "FI Co nand ance with Ch rror Code ror Codes")	s to be in equired, s to be command consists of three lines, each the elements is as follows: mmands for the MPCX Device Group: mapter "Elements of the FI Command"]
	(2) = The resp with one • Line [01 IFJ"). • Line [Strin • Line (see Note:	number; the implemented INDIF410.DI Complete DI in which the implemented oonse to the column. The 1 = Job ID 20] (see Cha 2 = FI comm g, in accord 3 = FI Job E Chapter "En File and p	two function d are located LL. LL name, if re- two functions d are located "MDA1" FI e meaning of apter "FI Co nand ance with Ch rror Code ror Codes") ath details m	s to be in equired, s to be command consists of three lines, each the elements is as follows: mmands for the MPCX Device Group: mapter "Elements of the FI Command"]

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	lumn 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. Rexroth Indramat'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.

Inputting an NC Record: MDI

Designation	MDI	Manual Data Input	
FI command	Input an N	C record for direct execu	ution in manual mode.
	CW_MDI_((1)	(Single Write)
	(1) = NC pr	rocess number	[06]
Value to be written	NC record		(see DOK-MTC200-NC**PRO*V)
	Note:		is passed to the "acValue" parameter e "DataTransfer" routine.
Response Structure		output with a column f mand has been execute	or acknowledgement of whether or not ed successfully.
	(P_ACK) =	= Positive ACKnowledge	e Data element has been set
Example MDI	Write an N	C record for direct exect	ution in NC process 0.
Conditions	The contro exist.	ol unit must be in "Setu	p" ("Manual") mode. Axes X1 and Y1
Value to be written	G01 X1 50	0.45 Y1 35.456 F 1000	

FI comma	and	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	0x0000000 texts)	(=additional information for some		
	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	0xFFFFFFC texts)	(=additional information for some
	4	Unrecognized e	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	18	(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	18	(For process information see documentation BR_AMM3)		
	9	Ready to start for processing of MDI record		

Possible error codes: Assumption:

Example 3

It is not possible to process the NC record because of an erroneous expression.

FI command		00_BR_AMM3_0
Line	Column	Answer
1	18	(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	18	(For process information see documentation BR_AMM3)		
	9	Failure of external 24 Volt supply		

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

Reading and Writing Machine Data: MDS

Designation	MDS	IDS Machine Data Single				
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		nachine data. IDS1_(1)_{(2)(10)} Info string for the	(Single		nfo strings:	
		machine data date to be written	 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: 129) Page number: Value according to page definition Data element: Value according to page definition Travel variable1: Value according to page definition Travel variable2: 			data value\
			Value according to page definition Data value: Value to be written Data unit: String of units		'n	
Response Structure		lowing table shows " FI command. Ther				
	Line 1n		Colun	nn 1		Column 5
Value Range/Meaning of Columns		Status message with regard to write		Vrite pro successfu	cedure has beer Illy.	n executed
		procedure		Write proexecuted.	cedure could NC	DT be
	2 = In m to		See syr	ntax of the	e MDS comman	d
	3 =	Error class	0 = 1 = Con 2 = Fl e 3 = Erro	rror	r tion error (NACK	()

Example MDS1

4 = Short message If the write procedure has been executed successfully, then --, otherwise the short error text text is given. 5 = Reference If the write procedure has been executed information successfully, then --, otherwise reference information is given. 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: POS (ID number: 14) Data type: Page number: 103

Data element:3Travel variable1:1Travel variable2:2Data value:100.00

mm

Data unit:

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	BR_M (1) =	hachine data. DS1_(1)_{(2)(10)} Info string for the machine data date to be read	 (Single 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: 129) 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 type n n
Response Structure		lowing table show " FI command. The				
		Line 1n	Colum	n 1		Column 7
Value Range/Meaning of Columns	1 =	Data value read as	s a string	lf an e given.	rror occurs during	g reading – is
	2 =	Data unit read as a	a string	lf an e given.	rror occurs during	g reading – is
	3 =	Number of places decimal point	after the	If an error occurs during reading – is given.		
	4 =	Info string for the r data date to be rea		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 tex	ĸt	If the read procedure has been executed successfully, then, otherwise the short error text is given.		
	7 =	Reference informa	ation	execut	ead procedure h ted successfully, <i>r</i> ise reference inf	then,
Example MDS1	Three r 1. Valu	nachine data value e:	es are reac	1:		
	Data ty	pe: DREAL (I	D number:	: 13)		
	Page n					
	Data el	ement: 1 variable1: 0				
		variable2: 0				
	2. Valu					
	Data ty	pe: POS (ID r	number: 14	4)		
	Page n	umber: 122				
		ement: 3				
		variable1: 1				
	Iravel	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].

Downloading Message Texts: MFD

Designation	MFD Message Files Download
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





	BW_MFD1	(Single Write)			
Response Structure	The response to the "MFD1" FI command consists of three lines, each w one column. The meaning of the elements is as follows:				
	Line 1 = Job ID	[0120] (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 comma	and	00_BW_MFD1	
Line	Column	Answer	
1	1	01	
2	1	00_BW_MFD1	
3	1	0	

Maximum Feedrate Override: MFO

Designation	MFO Ma	aximal	Feedrate Over	ride		
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 Rea	ad)	
	CB_MFO1_(1))		(Break Cyc	lic Read)
	(1) = NC proce	ess nur	mber	[06]		
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 [%].					the identifier, the
	Line 1			Column 1		Column 3
Value Range/Meaning	1 = Identifier			[MAX]		
of Columns	2 = Value of maximum feedrate override [0100]]	
	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				ımn 2	Column 3
	1		MAX	1	00	[%]
Reference to Literature	See chapter er	ntitled '	"Literature" [9].			



Maximum Feedrate: MFR

MWCX device group

Designation	MFR Maxima	l Feed Rate			
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.				to pre-select
	CR_MFR_(1){_(2)}		(Single	e Read)	
	CC_MFR_(1){_(2)}		(Cyclie	c Read)	
	CB_MFR_(1){_(2)}		(Break	Cyclic Rea	ad)
	(1) = NC process n	umber	[06]		
	(2) = Required m (opt.)	easurement sys	stem [mm, i	nch]	
Response Structure	e The following table shows the general structure of the response to command "MFR". One line with three columns is output for the id 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 address 00.	the maximum	feedrate in	NC process	s 0 of device
	FI command	00_CR_MFR_0)		
		Ar	nswer		
	Line	Column 1	Colu	mn 2	Column 3
	1	F	1200	0.00	[mm/min]
Example MFR	Read the value of address 00. The dis				
	FI command 00_CR_MFR_0_inch				
		Ar	nswer	r	
	Line	Column 1	Colu	mn 2	Column 3
	1	F	472	24.4	[inch/min]
Poforonco to Litoraturo	See chanter entitled	"Literature" [18	1		

Reference to Literature

See chapter entitled "Literature" [18].



Reading Machine Key Information : MKS

MWCX device group

				101	Werk device group	
Designation	MKS	Machine	Key Status			
Explanation	Current r	nachine key	ed device.			
FI command	Read ma	achine key ir	nformation for s	selected device.		
	BR_MKS	S	(Single Read	I)		
	BC_MKS	S	(Cyclic Read)		
	BB_MKS	S	(Break Cyclic	c Read)		
Response Structure		owing table s nd "MKS".	shows the gene	eral structure of the	response to the FI	
		Line	1	Column 1	Column 2	
Value Range/Meaning	1 =	Machine ke	y information	[4 byte in I	HEX coding]	
of Columns	2 =	Information	-		llid, 1=valid]	
Example MKS	Read the current machine key information			mation for device 0.		
	FI command 00_BR_M		00_BR_MKS			
	Line	Column	Answer			
	1 1 0000000		0000000			
	2 0		0			
Writing the GUI-SK Blo	ock: Mł	κт				
				N/I)	WCX device group	
					Work device group	
Designation	МКТ	Machine	Key Table			
Explanation	Writes the GUI-SK16 block in the PLC.					
FI command	Write Gl	JI-SK16 bloc	ck.			
	BW_MKT1_(1)			(Single Write)		
	 List of the 48 PLC variables for writing the GUI-SK16 block. 			A distinction is made between the		
				ollowing cases: 1. Clear GUI-SK1	16 block.	
	_				SK16 block with riables, filling	
Response Structure	(P ACK) is returned	following succ	essful transmission.		
· · · · · · · · · · · · · · · · · · ·	()	,				

Value Range/Meaning of the

Line 1 Column 1 1 = Successfully completed (P_ACK)

Columns

1. Example MKT1

1.Clear GUI-SK16 block:

FI comm	and	00_BW_MKT1 Value to write: \$EMPTY
Line	Column	Answer
1	1	(P_ACK)



2. Exa

mple MKT1	Write GUI-SK16 block:
-----------	-----------------------

FI comm	and	00_BW_MKT1 Value to write: SPSVAR1,SPSVAR2,\$SPACE,
Line	Column	Answer
1	1	(P_ACK)

Write the GUI-SK16 block, writing only those PLC variables which are **FI** command 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,....

	Line 1		Column 1	Column 2
Value Range/Meaning of Columns	1 =	Status report [0 = at least 1 PLC varia current PLC program is 1 = ALL PLC variables o written]		gram is <u>NOT</u> defined
	2 =	List of the NON-defin PLC variables in current PLC program	the written, or else li that could not be	st of the PLC variables written.] LC variables are

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 comma	and	00_BW_MKT2 Value to be written: SPSVAR1,SPSVAR2,SPSVAR48
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.

Maxim Rapid Override: MRO

MWCX device group

Designation	MRO	Maximal	Rapid Overric	le			
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 Rea	d)		
	CC_MRO1	_(1)		(Cyclic Rea	d)		
	CB_MRO1	_(1)		(Break Cycl	ic Read)		
	(1) = NC pr	ocess nui	mber	[06]			
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 [%].					t for the identifier,	
		Line 1		Column 1		Column 3	
Value Range/Meaning	1 = Identif	ier			[RN	[RMAX]	
of Columns	2 = Value	e of maximum rapid override		ide	de [0100]		
	3 = Unit			[%]			
Example MRO1	Read the n address 00		value of the ra	apid override i	in NC pro	cess 0 of device	
	FI command 00_CR_MRO1_0						
	Answer						
	Line Column 1 Column 2 Column 3						
	1		RMAX	10	00	[%]	
Reference to Literature	See chapter entitled "Literature" [18].						
System Message	s: MSG				MW	CX device group	
Designation							
	MSG	MeSsaG	ie				
Explanation	MSG Reading of						
Explanation Fl command							

Read S

Designation	MSG	MeSsaGe			
Explanation	Reading of system messages				
FI command	Message CC_MSG (1) = SYS	G_(1) (Cyclic Read) G-Message number			
	Note:	Exists only as a cyclic command			
Response Structure	The respo data.	onse of the FI command 'MSG' consists of the system message			

(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:

Indramat OPC-Server

00_CC_MSG_64

Indramat DDE-Server

Maximum Spindle Override: MSP

Example MSG

MWCX device group

MSO Maximal Spindle Override						
The value of the maximum spindle override of the selected device of the MWCX device group is read.						
CR_MSO1_(1)_(2)			(Single Read)			
CC_MSO1_(1)_(2)			(Cyclic Read)			
CB_MSO1_(1)_(2)			(Break Cyclic Read)			
(1) = NC process number(2) = Number of spindle			[06]			
			[13]			
The following table shows the general structure of the response "MSO1" FI command. One line with three columns is output for the ide the value of the maximum spindle override and the unit [%].						
Line 1			Column 1		Column 3	
1 = Identifier [SMAX]						
2 = Value of maximum rapid overr			ride [0100]			
3 = Unit			[%]			
Read the maximum value of the spindle override in NC process 0 of device address 00.						
FI command	command 00_CR_MSO1_0_1					
Answer						
Line		Column 1	Colu	mn 2	Column 3	
1		SMAX	1(00	[%]	
	The value of MWCX device CR_MSO1_(CC_MSO1_(CB_MSO1_((1) = NC prod (2) = Number The following "MSO1" FI cont the value of the value of the 2 = Value of 3 = Unit Read the modevice addres FI command	The value of the ma MWCX device group $CR_MSO1_(1)_(2)$ $CC_MSO1_(1)_(2)$ $CB_MSO1_(1)_(2)$ (1) = NC process nu (2) = Number of spin The following table "MSO1" FI command the value of the maxim I = Identifier 2 = Value of maxim 3 = Unit Read the maximum device address 00. FI command Line	The value of the maximum spindle MWCX device group is read. CR_MSO1_(1)_(2) CC_MSO1_(1)_(2) CB_MSO1_(1)_(2) (1) = NC process number (2) = Number of spindle The following table shows the ger "MSO1" FI command. One line with the value of the maximum spindle over Line 1 1 = Identifier 2 = Value of maximum rapid overrial 3 = Unit Read the maximum value of the device address 00. FI command 00_CR_MSO1_ Ar Line Column 1	The value of the maximum spindle override of the MWCX device group is read. CR_MSO1_(1)_(2) (Single Read Cyclic Read CB_MSO1_(1)_(2) CB_MSO1_(1)_(2) (Break Cyclic Read Cyclic Re	The value of the maximum spindle override of the selecter MWCX device group is read. CR_MSO1_(1)_(2) (Single Read) CC_MSO1_(1)_(2) (Cyclic Read) CB_MSO1_(1)_(2) (Break Cyclic Read) (1) = NC process number [06] (2) = Number of spindle [13] The following table shows the general structure of the ref "MSO1" FI command. One line with three columns is output to the value of the maximum spindle override and the unit [%]. Line 1 Column 1 1 = Identifier [SMAX] 2 = Value of maximum rapid override [0100] 3 = Unit [%] Read the maximum value of the spindle override in NC device address 00. Image: Column 1 FI command 00_CR_MSO1_0_1 Line Column 1 Column 2	

Reference to Literature

See chapter entitled "Literature" [21].



Maximum Spindle Speed: MSS

MWCX device group

Designation	MSS	Maximal Spindle Speed					
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			[06]			
	(2) = Numb	per of spin	dle	[13]			
Response Structure	The following table shows the general structure of the response to the "MS FI command. One line with three columns is output for the identifier, the spe and the unit [1/min].						
	Line 1		Column 1		Column 3		
Value Range/Meaning	1 = Identifi	ier	[S = spindle]				
of Columns	2 = Speed		[format according to settings of the parameters]			ne parameters]	
	3 = Unit		1/min				
Example MSS	Read the maximum value of the speed of the 1 st spindle in NC process 0 of device address 00.						
	FI command 00_CR_MSS_0_1						
	Answer						
	Lin	ie	Column 1	Colu	mn 2	Column 3	
	1		S:	750	0.0	1/min	

Reading the Firmware Identification: MTC

Designation	MTC MT-CNC 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)		
	(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	Example MTC Read the firmware identification of slot number 1 (CPU) of device 00				



	FI command		DO_BR_MTC_1		
	Line	Column	Answer		
	1	1	CPU01/0004-20V00		
FI command	This command is used to read the firmware identification from the various control components (slot numbers).				
	CR_MTC	C_(1)	(Sing	gle 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 comma	and	00_CR_MTC_1		
	Line	Column	Answer		

CPU01/0004-20V00

User Machine Data: MTD

MWCX device group

Designation	MTD	Machine Table Data	
FI command	Output of u	user machine data.	
	CR_MTD1	_(1)_(2)_(3)_(4)_(5)	(Single Read)
	CC_MTD1	_(1)_(2)_(3)_(4)_(5)	(Cyclic Read)
	(1) = Page	e number	[1299]
	(2) = Run v	variable 1	[-1000 +1000]
	(3) = Run v	variable 2	[-1000 +1000]
	(4) = Elem	ent number	[11000]
	(5) = Name	e	[113]

1

1

Answer

Data element

Data element		
10110100		
	LV1: 0 and LV2: 1 the 13 th element 50	of the type UDINT
FI command	Write machine table data.	
	CW_MTD1_(1)_(2)_(3)_(4)_(5)	(Single Write)
	(1) = Page number	[1299]

(i) = i ago namboi	[1200]
(2) = Run variable 1	[-1000 +1000]
(3) = Run variable 2	[-1000 +1000]
(4) = Element number	[11000]
(5) = identifier code	[113]

			_		
	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. 22	20 bytes	I
	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	e value	[ad	cc. to the display fo	ormat of the BOF]
	Note:	The value to in the "Data"			e "acValue" paramete
•	One line with one column is output to acknowledge the FI command issued. The meaning of the elements is as follows: (P_ACK) = Positive ACKnowledge Value has been successfully transmitted				
NC Program Compile a	(P_ACI	() = Positive A	CK nowledg	ents is as follows: ge Value has bee	-
NC Program Compile a	(P_ACI	() = Positive A	CK nowledg	ents is as follows: ge Value has bee	en successfully
NC Program Compile a	(P_ACI	() = Positive A	CK nowledg	ents is as follows: ge Value has bee	-
NC Program Compile a	(P_ACI	() = Positive A	CKnowledg	ents is as follows: ge Value has bee	en successfully
	(P_ACH and Do NCA NC prog	(x) = Positive A (c)	CKnowledg ICA m Access nloaded via	ents is as follows: ge Value has bee transmitted	en successfully MWCX device grou
Designation	(P_ACH and Do NCA NC prog and via a NC prog BW_NC	(x) = Positive A (x)	CKnowledg ICA m Access nloaded via sses.	ents is as follows: ge Value has bee transmitted a a download file (Single	en successfully MWCX device grou
Explanation	(P_ACH and Do NCA NC prog and via a NC prog BW_NC	(x) = Positive A (x) = Positive A (x) Program (x) P	CKnowledg ICA m Access nloaded via sses. h path deta	ents is as follows: ge Value has bee transmitted a a download file (Single	en successfully MWCX device group and NC program files



Example NCA1 00_BW_NCA1_"D:\Program Files\Indramat\Mtgui\Temp\download.ini"/3

FI comma	and	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	

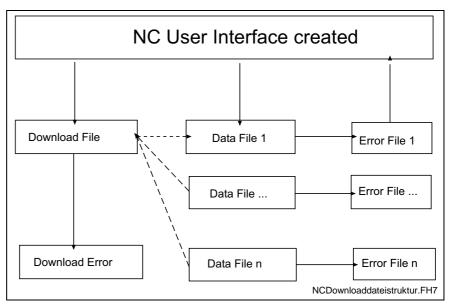


Fig. 7-5: File structure of the download file

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" Pac Package name " Pack	-	max. 2 characters max. 32 characters
5	-	3 characters left-justified
Package time: "Pack	-	max. 8 characters
Package date: "Pack	ageDate"	max. 8 characters
Package default:"Pack	kageDefault"	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 [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)
ХХ	Process number
ууу	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-00-86	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	[06]
Program number:	Νο	[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, (opt	ional)
Information on data an	d time ie	aivon in the format

Information on date and time is given in the format Date: dd.mm.yy Time: hh:mm:ss

Status flag	Description
С	Compiled
E	Error
The marked section is then printed out.	Not compiled
No details	No compiler call

Fig. 7-6: 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

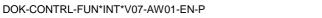
Five-digit number starting with 1.

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 progra	am compile.	
	BW_NC	A3_(1)	(Single Write)
	(1) = Administration file with path details.		
	Note:	Enclose f	ile 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 [0120] (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		
	00_BW_NCA3_"D:\Program Files\Indramat\Mtgui\ FI command Temp\compile.ini"/3		
	Line	Column	Answer
	1	1	01

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

NC Program Upload: NCA

Designation	NCA	NC-Program Access	
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.(Single Read)BR_NCA1_(1)_(2)(I = NC memory A, 2 = NC memory B](2) = Upload file with path details		
	Note:	In this command, the	letails in inverted commas. progress information is implemented in ted via the command IFJ of the MPCX
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 [0120] (see Chapter "FI Commands for the MPCX Device Group", IFJ). Line 2 = FI command [String, in accordance with Chapter "Elements of the FI Command"] 		



Example NCA

 Line 3 = FI Job Error Code (see Chapter "Error Codes")

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	

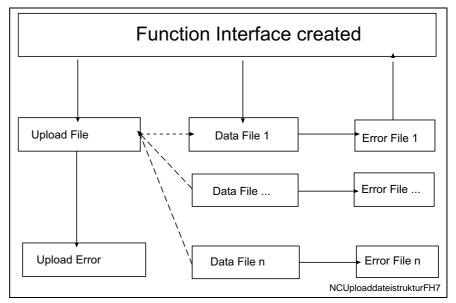


Fig. 7-7: File structure of the upload file

Structure of Upload File The upload file is str

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-justifiedPackage 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)
хх	Process number
ууу	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	[06]
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, (alwa	ays 'N')

Information on date and time is given in the format Date : dd.mm.yy Time: hh:mm:ss

Status flag	Description
С	Compiled
E	Error
The marked section is then printed out.	Not compiled
No details	No compiler call

Fig. 7-8: 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.

Example:

[Data] Max_Index_Data=25 00001=N0000 G0 X0 Y0 Z0 ...

00025=N0024 .Start



MWCX device group

NC Messages: NCM

Designation	NCM NC Messages	
Explanation	Indramat NC messages are output. These messages are assigned to a specific module and message type.	
FI command	Output all NC messages.	
	BR_NCM1_(1)_(2)	(Single Read)
	BC_NCM1_(1)_(2)	(Cyclic Read)
	(1) = Message type	[1 = error, 2 = messages]
	(2) = Module number	[199]
	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	[199]
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	e number of lines is 0.
	If there are no messages, th	e number of lines is 0. Column 1 Column 8
Meaning of the Columns	Line 1n	Column 1 Column 8
Meaning of the Columns		Column 4 Column 9
Meaning of the Columns	Line 1n 1 = Message text	Column 1 Column 8 [ASCII characters]
Meaning of the Columns	Line 1n 1 = Message text 2 = Message number	Column 1 Column 8 [ASCII characters] [ASCII characters]
Meaning of the Columns	Line 1n 1 = Message text 2 = Message number 3 = Time stamp day	Column 1Column 8[ASCII characters][ASCII characters][mm.dd.yyyy]
Meaning of the Columns	Line 1n 1 = Message text 2 = Message number 3 = Time stamp day 4 = Time stamp time	Column 1Column 8[ASCII characters][ASCII characters][mm.dd.yyyy][hh:mm:ss][031]
Meaning of the Columns	Line 1n 1 = Message text 2 = Message number 3 = Time stamp day 4 = Time stamp time 5 = Mechanism number	Column 1Column 8[ASCII characters][ASCII characters][mm.dd.yyyy][hh:mm:ss][031]
Meaning of the Columns	Line 1n 1 = Message text 2 = Message number 3 = Time stamp day 4 = Time stamp time 5 = Mechanism number 6 = 2 byte additional information	Column 1Column 8[ASCII characters][ASCII characters][mm.dd.yyyy][hh:mm:ss][031]ation[ASCII characters]
Meaning of the Columns	Line 1n 1 = Message text 2 = Message number 3 = Time stamp day 4 = Time stamp time 5 = Mechanism number 6 = 2 byte additional informa 7 = NC note	Column 1 Column 8 [ASCII characters] [ASCII characters] [ASCII characters] [mm.dd.yyyy] [hh:mm:ss] [031] ation [ASCII characters] [ASCII characters] [ASCII characters] [YES, NO] [e.g.HTML format]
Meaning of the Columns	Line 1n 1 = Message text 2 = Message number 3 = Time stamp day 4 = Time stamp time 5 = Mechanism number 6 = 2 byte additional informa 7 = NC note 8 = Reference text exists 9 = Filename for additional	Column 1Column 8[ASCII characters][ASCII characters][Mm.dd.yyyy][hh:mm:ss][031][031]ation[ASCII characters][ASCII characters][YES, NO][e.g.HTML format]t



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	



NC Events Download: NEA

MWCX device group Designation NEA NC-Event Access NC events are downloaded by means of the download file via all processes. Explanation Download NC events. **FI command** 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 00 BW NEA1 "D:\Program Files\Indramat\Mtgui\ **FI** command 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.

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.

Care must be taken in the use of upper and lower case letters.

Example:

Note:

[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	er" PackageNo "
Package name	"PackageName"
Package size:	"PackageSize"
Package time:	"PackageTime"

max. 2 characters max. 32 characters max. 8 characters left-justified 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
```



NC Events Upload: NEA

Designation	NEA	NC-Ever	nt Access			
Explanation	NC event	s are uploa	ded through all processes and external events.			
FI command	Upload NC events. BR_NEA1_(1) (Single Read) (1) = Upload file with path details					
	Note:	Enclose f	ile and path details in inverted commas.			
			ommand, the progress information is implemented in be interrogated via the command IFJ of the MPCX oup.			
Response Structure			NEA1 FI command consists of three lines, each with aning of the elements is as follows:			
		I = Job ID Chapter "F	[0120] FI Commands for the MPCX Device Group", IFJ).			
		2 = FI comr g, in accord	nand dance with Chapter "Elements of the FI Command"]			
	 Line 3 = FI Job Error Code (see Chapter "Error Codes") 					
Example NVA	00_BR_N	EA1_"D:\P	rogram Files\Indramat\Mtgui\Temp\Upload.ini"/3			
	FI comma	and	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.					
		Common] nformation	is stored in the COMMON section.			
		whether or	not an error has occurred during uploading. This ne event of an error.			
	<u>Example:</u> [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]

corresponds to a process number [0..6] A:

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



Status of NC Events: NEV

Designation	NEV	NC-EVe	nt		
FI command	Read the status of an NC event of the selected device device group.			of the selected device of the MWCX	
	CR_NEV	_(1){_(2)}		(Single Read)	
	(1) = NC	process nu	mber	[06]	
	(2) = Nun	nber of the	NC event	[031] ! Optional !	
	Note:		ional paramete s is output.	er is not specified then the status of all	
Response Structure	One line is output, whereby the number of columns depends on the number of event statuses requested. When the optional parameter has <u>not</u> 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 00.	status of t	he 17"' NC eve	ent in NC process 0 of device address	
	FI comma	and	00_CR_NEV_0)_17	
	Line	Column	Answer		
	1	1	0		
	L				
FI command	Write the device gro		an NC event	of the selected device of the MWCX	
FI command		oup.	an NC event	of the selected device of the MWCX (Single Write)	
FI command	device gro CW_NEV	oup.			
FI command	device gro CW_NEV (1) = NC	oup. /_ (1)_(2)		(Single Write)	
FI command Value to be written	device gro CW_NEV (1) = NC (2) = Even	oup. /_ (1)_(2) process nu	mber	(Single Write) [06]	
	device gro CW_NEV (1) = NC (2) = Even Status of One line	oup. /_(1)_(2) process nu nt number NC Event is output w	mber 0 = delet	(Single Write) [06] [031] e NC event; 1 = set NC event or acknowledgement of whether or not	
Value to be written	device gro CW_NEV (1) = NC (2) = Even Status of One line the FI con	oup. 7_(1)_(2) process nu nt number NC Event is output w nmand has	mber 0 = delet ith a column fo	(Single Write) [06] [031] e NC event; 1 = set NC event or acknowledgement of whether or not d successfully.	
Value to be written	device gro CW_NEV (1) = NC (2) = Even Status of One line the FI con (P_ACK)	oup. 7_(1)_(2) process nu nt number NC Event is output w nmand has) = P ositive	mber 0 = delet ith a column fo been execute ACK nowledge	(Single Write) [06] [031] e NC event; 1 = set NC event or acknowledgement of whether or not d successfully. NC event has been deleted or	
Value to be written Response Structure	device gro CW_NEV (1) = NC (2) = Even Status of One line is the FI cor (P_ACK) Set the 17 FI comma	oup. /_(1)_(2) process nu nt number NC Event is output w nmand has) = P ositive 7 th NC even	mber 0 = delet ith a column fo been execute ACK nowledge it in NC proces 00_CW_NEV_ Value to write	(Single Write) [06] [031] e NC event; 1 = set NC event or acknowledgement of whether or not d successfully. NC event has been deleted or set. s 0 at device address 00. 0_17	
Value to be written Response Structure	device gro CW_NEV (1) = NC (2) = Even Status of One line in the FI corr (P_ACK) Set the 17 FI comma Line	oup. 7_(1)_(2) process nu nt number NC Event is output w nmand has 0 = Positive 7 th NC even and Column	mber 0 = delet ith a column for been executer ACKnowledge it in NC proces 00_CW_NEV_ Value to write Answer	(Single Write) [06] [031] e NC event; 1 = set NC event or acknowledgement of whether or not d successfully. NC event has been deleted or set. s 0 at device address 00. 0_17	
Value to be written Response Structure	device gro CW_NEV (1) = NC (2) = Even Status of One line is the FI cor (P_ACK) Set the 17 FI comma	oup. /_(1)_(2) process nu nt number NC Event is output w nmand has) = P ositive 7 th NC even	mber 0 = delet ith a column fo been execute ACK nowledge it in NC proces 00_CW_NEV_ Value to write	(Single Write) [06] [031] e NC event; 1 = set NC event or acknowledgement of whether or not d successfully. NC event has been deleted or set. s 0 at device address 00. 0_17	



Selection of NC Memory: NMM

MWCX device group

MWCX device group

Designation	NMM	NC-MeM	lory				
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	Λ	(Single Write)				
Value to be written	NC memo	ory	[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) = Positive ACKnowledge The selected NC memory has been selected.						
Example NMM	Select NC memory B at device 00 for processing the NC program.						
	00_CW_NMM FI command Value to write: 2						
	Line	Column	Answer				
	1	1	(P_ACK)				

Reference to Literature

See chapter entitled "Literature" [37].

Reading NC Parameters: NPA

Designation	NPA	NC-PArameter						
FI command	Read a par	Read a parameter line.						
	BR_NPA1	_(1)_(2)	(Single	Read)				
	(1) = Parai	meter record num	ber [199]					
	(2) = Parai	meter number	[A00.00	0Cxx.120]				
Response Structure	The following table shows the general structure of the response command "NPA1". One line is output with 3 columns for the ide value and the name respectively.							
		Column 2	Column 3					
Value Range/Meaning of Columns	1 = Identif 2 = Value 3 = Name	[AS	ameter ID [max. CII text]	. 32 ASCII char value or empty]	-			



Rexroth Indramat

Example NPA1	Return the parameter line from parameter record 10 with parameter number B00.007.					ith parameter
	In this pla	r record 10 ce, the follo) has been cre owing informat on 75 mm/sec	ion is located		been defined.
	FI comma	and	00_BR_NPA1	_10_B00.007		
	Line	Column	Answer			
	1	1	B00.007			
		2	75			
		3	mm/sec^2			
FI command	Read out	several pa	rameter lines f	rom a parame	eter record.	
		(1)(2)_(3		(Single Rea		
		meter reco	-	[199]	,	
	(2) = Para	meter numl	ber [from]	[A00.000C	xx.120]	
	(3) = Para	meter numl	ber [to]	[A00.000C	xx.120]	
Response Structure	The following table shows the general structure of the response to th command "NPA2". As many lines as are requested are output, each three columns for the identifier, the value and the name respectively.				put, each with	
		Line 1	n:	Column 1		Column 3
Value Range/Meaning	1 = Ident	ifier	[max. 32]	ASCII charact	ers]	
of Columns	2 = Value		- [ASCII te>		-	
	3 = Nam	е	[unit, relat	ted to the valu	e or empty]	
Example NPA2			ter lines from parameter num			of parameter
	Assumption: Parameter record 10 has been created and contains the following information in this location:					the following
	FI comma	and	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 pa	articular ele	ement of a para	ameter line.		
	Note:	Comman	d NPA3 is not	supported in	FI Version 0	6!

BR_NPA3_(1)_(2)_(3)	(Single Read)
(1) = Parameter record number	[1.99]
(2) = Parameter number	[A00.000Cxx.120]
(3) = Element number	[11000]



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		Colur	nn 1
Value Range/Meaning of Columns	1 = Nam	e/value/des	signated name		[ASCII-Text]	
Example NPA3	Return element 1 of the parameter line from parameter record 10 with parameter number C01.079. <u>Assumption:</u> The parameter record has been created and contains the following					
	FI comma	on in this loo	00_BR_NPA3_	10 001 070	10	
	Line	Column	Answer	_10_001.079_	_19	
	1	1	Required value table of axis 1.	(here 19) fro	m existing com	pensation
FI command	Read all e	elements fro	om a paramete	r line (such a	as "NPA1").	
	Note:	Comman	d "NPA4" is no	t supported i	n FI Version ()6!
	. ,		ord number	(Single Rea [199] [A00.000C		
Response Structure	command	I "NPA4". C	shows the gene One line is outp respectively.			
		Line 1		Column 1	Column 2	Column 3
Value Range/Meaning	1 = Ident	ifier	[max. 32	ASCII chara	cters]	
of Columns	2 = Value	e	[ASCII te	ext]	-	
	3 = Name		[unit, related to the value]			
Example NPA4	Return th number A		ter line from	parameter r	ecord 10 wit	h parameter
	<u>Assumption:</u> The parameter record has been created and contains the following information in this location: Master.					
	FI comma	and	00_BR_NPA4_	10_A00.000		
	Line	Column	Answer			
	1	1	A00.000			
		2	Master			
		3				
	Note:		mands support d "00_NPA1_?		ersion are list	ed using the
Explanation	•		a list with a maxi process parame			he same type



FI command	Read NC	parameters	s for a seled	cted device.			
	BR_NPA	5_(1)_(2)_	{(3)(12)}	(Single R	ead)		
	 (1) = Parameter type (2) = Process number or axis number 			1 = System parameter 2 = Process parameter 3 = Axis parameter			
				If "system parameter" has been selected as the type of parameter, then this parameter is NOT evaluated – set to 0.			
	(3)(12 L paramete	ist of reque	ested	A maximum of 10 parameters of th same type may be listed here. Plea take the parameter number from th general description of parameters f the control unit.			
Response Structure		The following table shows the g command "NPA5".			cture of the res	ponse to the FI	
	L	ine 1n	C	olumn 1		Column 3	
Value Range/Meaning of Columns	1 = F	arameter n	lumber	Parameter requested.	number that ha	as been	
	2 = F	Parameter value		Data setup – see general description of parameters.			
	3 = F	Parameter u	ınit	Data setup parameters	– see general s.	description of	
Example NPA5	NC paran	neter reque	st for syste	m paramete	ers 0,52,53.		
	FI comma	and	00_BR_NP	A5_1_0_0_5	52_53		
	Line	Column	Answer				
	1	1	0				

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].



MWCX device group

Activate NC Compiler: NPC

Designation	NPC NC-Pac	kage C ompiling	I		
FI command	Compiles the select	ed NC package			
	BR_NPC1_(1)		(Single Read)		
	(1) = Number in NC	package direc	tory [1.	99]	
Response Structure	The following table shows the general construction of the answer of the F 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		(refer to chap PCX Device G	ter entitled "FI Group", IFJ).	Commands
	2 = FI command [string, in accordance to chapter entitled "Elements of the FI Command"]				ntitled
	3 = FI job error code (see chapter ent			Error Codes"	')
Example NPC	Compile the 2 nd NC	package.			
	FI command	00_BR_NPC1_	_2		
	Answer				

Column 1

01

Line

1

Activate NC Download: NPD

MWCX device group

Column 3

0

Column 2

00_BR_NPC1_2

Designation	NPD	NC-Package Down	load			
FI command	Downloads the selected NC package into the identified device without the setup lists.					
	BW_NPD	1_(1)_(2)		(Single Write)		
	(1) = NC I	memory		[1 = NC memory A, 2 = NC memory B]		
	(2) = Num	ber in NC package d	irectory	[199]		
Value to be written	Initializatio	on 1 = Trigger	NC downloa	ad		
	Note:	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 = J	ob ID		efer to chapter entitled "FI s for the MPCX Device Group",		
	Line 2 = F	l command		accordance to chapter entitled of the FI Command"]		
	Line 3 = F	I job error code	(see chapt	er entitled "Error Codes")		



Load the 2nd NC package (**without setup lists**) into the NC memory A of the device with device address 00 1

		e with devic	Т		
	FI comma	and	00_BW_NF	PD1_1_2 e written: 1	
	Line	Column	Answer		
	1	1	02		
	2	1	00_BW_N	PD1_1_2	
	3	1	0		
	Note:	which is terminate	already in t	le to transfer once again an NC package the device, the "DataTransfer" routine will ess with error code 1030 (see chapter s").	
FI command	Download setup list		cted NC pa	ckage into the identified device with the	
	BW_NPI	D2_(1)_(2)		(Single Write)	
	(1) = NC	memory		[1= NC memory A, 2= NC memory B]	
	(2) = Nur	mber in NC	package d	irectory [199]	
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				NPD2 consists of three lines, each with one ents is as follows:	
	Line 1 =	Job ID		[0120] (refer to chapter entitled "FI Commands for the MPCX Device Group", IFJ).	
	Line 2 =	FI commar	nd	[string, in accordance to chapter entitled "Elements of the FI Command"]	
	Line 3 =	FI job error	code	(see chapter entitled "Error Codes")	
Example NPD2		3 rd NC pacl th device ad		setup lists) into the NC memory B of the	
	FI comma	and	00_BW_N Value to b	PD2_2_3 e written: 1	
	Line	Column	Answer		
	1	1	03		
	2	1	00_BW_NF	PD2_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

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:	period (re		utine remains for all the transmission - transmission period). This is only valid
FI command	Downloads the selected NC package into the identified device without the setup lists.			ge into the identified device without the
	BW_NPI	D3_(1)_(2)		(Single Write)
	(1) = NC	memory		[1 = NC memory A, 2 = NC memory B]
	(2) = Nur	nber in NC	package dire	ctory [199]
Value to be written	Initializat	ion	1 = Trigger NC	C download
	Note:			n is passed to the "acValue" parameter e "DataTransfer" routine.
Response Structure				D3 consists of three lines, each with one ts is as follows:
				0120] (refer to chapter entitled "FI commands for the MPCX Device Group", FJ).
	Line 2 =	FI commar	-	string, in accordance to chapter entitled Elements of the FI Command"]
	Line 3 =	FI job error	code (s	see chapter entitled "Error Codes")
Example NPA3			kage (withou t ce address 00.	t setup lists) into the NC memory A of
	FI comm	and	00_BW_NPD3 Value to be w	-
	Line	Column	Answer	
	1	1	03	
	2	1	00_BW_NPD3	3_1_2
	3	1	0	
	Note:	which is terminate	already in the	to transfer once again an NC package device, the "DataTransfer" routine will with error code 1030 (see chapter
FI command	Download setup list		cted NC pack	age into the identified device with the
	BW_NPI	D4_(1)_(2)		(Single Write)
	(1) = NC	memory		[1 = NC memory A, 2 = NC memory B]
	(2) = Nur	mber in NC	package dire	ctory [199]
Value to be written	Initializat	ion	1 = Trigger NC	C download



	Note:	ote: The value to be written is passed to the "acValue" parame 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			[0120] (refer to chapter entitled "FI Commands for the MPCX Device Group", IFJ).	
	Line 2 =	FI commar	nd	[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 3 rd NC package (with setup lists) into the NC memory device with device address 00.			setup lists) into the NC memory B of the	
	00_BW_NPD4_2 FI command Value to be writted				
	Line				

i i command		
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").

Read NC Package Directory: NPI

Designation	NPI	NC-Package Directory			
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 Fl command NPI. The response consists of up to a maximum of n=99 lines, each with 5 columns.				
		Line 1n:	Column 1		Column 5
Value Range/Meaning	1 = Number in NC package directory			[0199]	
of Columns	2 =	Name of the NC package	[max. 32 AS characters]	CII	
	3 =	Length of the NC packag	[byte]		
4 = Date of creat package		Date of creation/last char package	nge to NC	[DD.MM.YY]]
	5 =	Time of creation/last change to NC [HH:MM:SS] package]



Example NPI

Read the entries in the	NC package directory	v at device address ()()
	The package an obter	

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].

Selection of the NC Program in the Active NC Memory: NPS

Designation	NPS NC-Program Selection			
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 p	process number	[06]	
Value to be written	Number ir	NC package directory	[199]	
	Note:	valid NC program pack Otherwise, the request is a	ct an NC program when there is a age in the active NC memory. acknowledged by an error message. passed to the "acValue" parameter 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)	= Positive ACKnowledge	The selected NC program has been selected.	
Example NPS	NC memo Assumptic	iry.	essing NC program 01 in the active in the active NC memory.	



FI com	mand	00_CW_NPS_0 Value to write: 1
Line	Column	Answer
1	1	(P_ACK)

Reference to Literature See chapter entitled "Literature" [37].

Next Tool Number: NTN

MWCX device group

Designation	NTN	Next Too	ol- N umber	
Explanation	Returns the next pre-selected tool number of the selected device of the MWCX device group.			
FI command	CC_NTN_ CB_NTN_	CR_NTN_(1) CC_NTN_(1) CB_NTN_(1) (1) = NC process number		(Single Read) (Cyclic Read) (Break Cyclic Read) [06]
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.			C process 0 of device address 00.
	FI comma	and	00_CR_NTN_0	
	Line	Column	Answer	
	1	1	Т	
		2	1	

Reference to Literature See chapter entitled "Literature" [43].

NC Zero Point Download: NUA

Designation	NUA	NC-Offset Data Access				
Explanation	Zero points processes.	s are downloaded by means of the download file via all active				
FI command	BW_NUA	NC zero points. 1_(1) (Single Write) hload file with path details.				
	Note:	Enclose file and path details in inverted commas.				
Response Structure		nse to the "NUA1" FI command consists of three lines, each lumn. The meaning of the elements is as follows:				
	 Line 1 = Job ID [0120] (see Chapter "FI Commands for the MPCX Device Group: IFJ"). 					
		= FI command in accordance with Chapter "Elements of the FI Command"]				
		= FI Job Error Code hapter "Error Codes")				



Example NUA1

JA1 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" Package size: "PackageSize" max. 8	max. 32 characters
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 =1PackageName =Offset DataPackageSize =1234PackageTime =13:10:10PackageDate =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]

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	Meanin	ng Explanation
	3	Genera	al acts additive to all offset types
	4	G54	adjustable zero offset
	9	G59	adjustable zero offset
	Note:	The axis Tables".	meanings are contained in chapter 6.2, "Data
	[OffsetDat 03=1.0000	-	;Process 0, zero-point data bank 0, axis X
	03=1.0000		;Gen. offset ;G54
	05=3.0000		;655
	06=4.0000		;G56
	07=5.0000)	;G57
	08=6.0000		;G58
	09=7.0000)	;G59
	[OffsetDat 03=1.0000	-	;Process 0, zero-point data bank 3, axis X ;Gen. offset
	04=2.0000		;654
	05=3.0000		;G55
	06=4.0000		;G56
	07=5.0000)	;G57
	08=6.0000)	;G58

;G59

09=7.0000



NC Zero Point Upload: NUA

NUA NC-Offset Data Access			
Zero-points are uploaded via all active processes.			
BR_NUA (1) = me	\1_(1)_(2) mory	(Single Read) [1 = memory A; 2 = memory B] n 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.			
 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 [0120] (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") 			
00_BR_NUA1_1_"D:\Program Files\Indramat\Mtgui\Temp\Upload.ini"/3			
00_BR_NUA1_"D:\Program Files\Indramat\Mtgui\ FI command Temp\upload.ini"/3			
Line	Column	Answer	
1	1	01	
2	1	00_BR_NUA1_"D:\Program Files\Indramat\Mtgui\ Temp\upload.ini"/3	
3	1	0	
The uploa	ad file is stru	uctured in the Windows – "Ini" format structure.	
General in Key Uplo Indicates value is o <u>Example:</u> [Commor UploadEr Section N Key Mem Identifies Memory=	nformation adError whether or nly set in th a] ror = YES NC variable ory the memor 1 ;Mer	not an error has occurred during uploading. This ie event of an error. ; error es information [OffsetDataPackage_Info] y from which the NC package is loaded. nory	А
	Zero-poin Zero-poin BR_NUA (1) = met (2) = Upl Note: The response one colum • Line 2 (see • Line 2 (see • Line 2 (see • Commer • Line 3 (see • Commer • Line 3 (see • Commer • Line 3 (see • Commer • Com	Zero-points are uploa Zero-point upload. BR_NUA1_(1)_(2) (1) = memory (2) = Upload file with Note: Enclose f In this co %. It can device grow The response to the one column. The me In this co %. It can device grow The response to the one column. The me I Line 1 = Job ID (see Chapter "F Line 2 = FI comm [String, in accord Line 3 = FI Job E (see Chapter "Er 00_BR_NUA1_1_"Di FI command Line Column 1 1 2 1 3 1 The upload file is strue Section [Common] General information Key UploadError Indicates whether or value is only set in the Example: [Common] UploadError = YES Section NC variable Key Memory Identifies the memor Memory=1 ;Mer	Zero-points are uploaded via all active processes. 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 %. It can be interrogated via the command IFJ of the MPC device group. The response to the NUA1 Fl command consists of three lines, each wi one column. The meaning of the elements is as follows: • Line 1 = Job ID [0120] (see Chapter "FI Commands for the MPCX Device Group", IFJ). • Line 2 = Fl command [String, in accordance with Chapter "Elements of the FI Command"] • Line 3 = FI Job Error Code (see Chapter "Error Codes") 00_BR_NUA1_1_"D:\Program Files\Indramat\Mtgui\Temp\Upload.ini"/3 Fl command 00_BR_NUA1_"D:\Program Files\Indramat\Mtgui\ Templupload.ini"/3 Line Column Answer 1 1 01 2 1 00_BR_NUA1_"D:\Program Files\Indramat\Mtgui\ Templupload.ini"/3 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



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:

	characters.	The length	of the individua	l identifications is described below:
	Package nu	mber" Pack a	ageNo"	max. 2 characters
	Package na			max. 32 characters
	Package siz			max. 8 characters left-justified
	Package tim			max. 8 characters
	Package da			max. 8 characters
	-		ageDefault"	
	Total:			max. 84 characters
	Information	on date and	l time is given i	n the format
	Date :	dd.mm.	-	in the format
	Time:	hh:mm:		
	Example:			
	OffsetData	Package In	fol	
	Memory=	uonago_m	.0]	
	PackageNo	=	1	
	PackageNa			
	PackageSiz		1234	
	PackageTim PackageDat		13:10:10 24.12.00	
	i uchugobu	.0 –	21112.00	
			ta download	
		-		on and is structured as follows:
		_A\Number		bank\code of axis meaning]
	A: Number of z	ero-point b		ss number [06]
	Code of axis			
		,		gle of rotation "PHI"
	Key values o	correspond	to the types of	offset [39] and values are the read
	values of the	e types of of	ffset in the base	e unit.
Offset Type	Code	Meaning	Explanatio	n
	3	General		e to all offset types
	4	G54	adjustable	zero offset
	 9	G59	adjustable	zero offset
			eanings are c	ontained in chapter entitled "Data
	I	ables".		
	[OffsetData_	_0\0\0]	;Process 0, ze	ro-point data bank 0, axis X
	03=1.0000		;Gen. offset	
	04=2.0000		;G54	
			;G55	
	05=3.0000			
	06=4.0000		;G56	
	06=4.0000 07=5.0000		;G56 ;G57	
	06=4.0000		;G56	



[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

NC Variables Download: NVA

Designation	NVA	NC-Varia	able Access		
Explanation	NC variables are downloaded by means of the download file via all processes.				
FI command	Download NC variables.				
	BW_NV	A1_(1)	(Single Write)		
	(1) = Dov	wnload file	with path details.		
	Note:	Enclose f	ile and path details in inverted commas.		
Response Structure			e "NVA1" FI command consists of three lines, each e meaning of the elements is as follows:		
		1 = Job ID Chapter "F	[0120] FI Commands for the MPCX Device Group: IFJ").		
		2 = FI comr g. in accord	nand lance with Chapter "Elements of the FI Command"]		
	-	3 = FI Job E			
	(see	Chapter "Ei	rror Codes")		
Example NVA1	00_BW_N	NVA1_"D:\F	Program Files\Indramat\Mtgui\Temp\download.ini"/3		
	00_BW_NVA1_"D:\Program Files\Indramat\Mtgui\ FI command Temp\download.ini"/3				
	FI comma	and	Temp\download.ini"/3		
	FI comma Line	and Column	Temp\download.ini"/3 Answer		
	Line	Column	Answer		
	Line	Column 1	Answer 01 00_BW_NVA1_"D:\Program Files\Indramat\Mtgui\		
Structure of Download File	Line 1 2 3 The struc	Column 1 1 1 1	Answer 01 00_BW_NVA1_"D:\Program Files\Indramat\Mtgui\ Temp\download.ini"/3 0 "download.ini" file used in this example corresponds		
Structure of Download File	Line 1 2 3 The struc	Column 1 1 1 1 ture of the ile in Windo	Answer 01 00_BW_NVA1_"D:\Program Files\Indramat\Mtgui\ Temp\download.ini"/3 0 "download.ini" file used in this example corresponds		



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 =1PackageName =NC variablesPackageSize =1234PackageTime =13:10:10PackageDate =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



NC Variables Download: NVA

MWCX device group

Designation	NVA	NC-Varia	able Access			
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:		•	in inverted commas.		
			be interrogated vi	ess information is implemented in a the command IFJ of the MPCX		
Response Structure			NVA1 FI comman eaning of the eleme	d consists of three lines, each with nts is as follows:		
		1 = Job ID Chapter "F	[0120] El Commands for th	ne MPCX Device Group", IFJ).		
	• Line 2	2 = FI comr	mand	"Elements of the FI Command"]		
		3 = FI Job E Chapter "Ei	Error Code rror Codes")			
Example NVA	00_BR_N	IVA1_"D:\P	rogram Files\Indra	mat\Mtgui\Temp\Upload.ini"/3		
	00_BR_NVA1_"D:\Program Files\Indramat\Mtgui\ FI command Temp\upload.ini"/3					
	Line	Column	Answer			
	1	1	01			
	2	1	00_BR_NVA1_"D:\I Temp\upload.ini"/3	Program Files\Indramat\Mtgui\		
	3	1	0			
Structure of Upload File	The uploa	ad file is str	uctured in the Wind	lows – "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 varia	bles inforn	nation section [NC	VariablesPackage_Info]		
	The pack of all pack	age identifi kage identif	fications must not e	rom several keys. The total length xceed a maximum of 84 identifications is described below:		
	-		nckageNo"	max. 2 characters		

Package name "PackageName"

max. 2 characters max. 32 characters



Package size: "Packa Package time: "Packa Package date: "Packa Package default:"Packa	ageTime" ageDate"	max. 8 characters left-justified max. 8 characters max. 8 characters max. 26 characters (optional)
Total:		max. 84 characters
Information on date and Date : dd.mm Time: hh:mm	.уу	n the format
Example: [NCVariablesPackage_ PackageNo = PackageName = PackageSize = PackageTime =	_Info] 1 NC variables 1234 13:10:10	

Section NC variables download [NCVariables_A]

24.12.00

A: corresponds to a process number [0..6]

Key values correspond to the variable numbers $\left[0..255\right]$ 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

PackageDate =





Reading and Writing NC Variables: NVS

				MV	VCX device group
Designation	NVS	NC-Varia	able S ingle		
Explanation	Reads the NC variables of the selected device of the MWCX device group.				
FI command	CR_NVS_(1)_(2){_(3)} CC_NVS_(1)_(2){_(3)} CB_NVS_(1)_(2){_(3)} (1) = NC process number (2) = NC variable number {from} (3) = NC variable number {to}			(Single Read) (Cyclic Read) (Break Cyclic Rea [06] [0255] [0255] !Optional	
	Note:		otional paramet are output.	ter is specified the	en up to 20 NC
Response Structure			ximum of 20 co ed NC variable is	olumns containing t s output.	he corresponding
	Note:		uested NC varia	able does not exist t ımn.	hen [] is entered
Example NVS without optional Parameter	Read the NC proce		ne NC variable	numbered 1 at dev	ice address 00 in
	FI comm	and	00_CR_NVS_0	_1	
	Line	Column	Answer		
	1	1	1.111000		
Example NVS with optional Parameter	Read the value of the 1 st NC variable to the 3 rd NC variable at device address 00 in NC process 0.				
	Assumpti The 2 nd N	<u>on:</u> IC variable i	is not defined.		
	FI comm		00_CR_NVS_0	_1_3	
			An	swer	
	L	.ine	Column 1	Column 2	Column 3
		1	1.111000		23.100000
Explanation	Writes ar	NC variabl	e of the selecte	d device of the MW	CX device group.
FI command	CW_NVS	6_(1)_(2)	(Sin	gle Write)	
	. ,	process nu	-		
	(2) = NC	variable nu	mber [0	255]	
Value to be written	NC varial	ble	[For	mat, long, or double	real]
	Note:	accordan	ce with the e	es is set to long ntered format. With used by all means.	

	Note:	written is	ned NC variables can be written. The value to be passed to the "acValue" parameter as an ASCII he "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 1 st NC variable in NC process 0 at device address 00.			
	FI command		00_CW_NVS_0_1 Value to be written: 1.111000	
	Line	Column	Answer	
	1	1	(P_ACK)	

Reference to Literature See chapter entitled "Literature" [39].

Optimum Position Distance from Axes: POD

Designation	OPD	Optimal Position Distance			
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	L_(1)_(2){_(3)}	(Break Cyclic Read)		
	(1) = NC p	process number	[06]		
	(2) = Axis	meaning	[011; 20] (see Chapter 6.2, "Data Tables")		
	(3) = Requ (opt.)	uired measurement system	[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-sel conversion of the result into mm or inches. If, however, a spindle selected as an axis, indicating a measurement system serves purpose.				
	CR_OPD2_(1){_(2)}		(Single Read)		
	CC_OPD2_(1){_(2)}		(Cyclic Read)		
	CB_OPD2	2_(1){_(2)}	(Break Cyclic Read)		
	(1) = Phys	ical axis number	[132]		
	(2) = Rec (opt.)	quired measurement system	[mm, inch]		

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".						
	Lin	e 1	Colu	mn 1	Column	2 Column	3 Column 4
Value Range/Meaning of the Columns	1 = Axis	name				Zi, Ui, Vi, Wi, <i>I</i> [,1,2,3]	Ai, Bi, Ci, Si]
	2 = Optir	num positio	on dis	tance	[acc. to parame	settings in the ters]	e process
	3 = Unit [mm, inch]						
	4 = Optimum position distance [as Column 2, but rounded up or down according to the parameter "Indicated decimal places"]						e parameter
	Note: If the specified axis is not defined in the selected NC process then the response in all columns is [].						cted NC process
Example OPD1	Read the optimum position distance of the Z axis in NC process 0 of device address 00.					NC process 0 of	
	FI comma	and	00_0	R_OPD	1_0_2		
					Answer		-
	Line	Columr	า 1	Colu	mn 2	Column 3	Column 4
	1	Z		-5.9	897	[mm]	-5.990
Example OPD1		optimum dress 00. V					NC process 0 of
	FI comma	and	00_0	R_OPD	1_0_2		
					Answer		
	Line	Columr	า 1	Colu	mn 2	Column 3	Column 4
	1	Z		-0.2	358	[inch]	-0.236
Example OPD2		optimum p ce address		n distan	ce of the	Z axis (physic	al axis number =
	FI comma	and	00_0	R_OPD	2_3		
					Answer		
	Line	Columr	า 1	Colu		Column 3	Column 4
	1	Z		-5.9	897	[mm]	-5.990

Parameter Download: PAA

Designation	ΡΑΑ	PArameter Acces	S					
Explanation	Complete parameter records are downloaded by means of a download file.							
FI command	Parameter download command whereby two predefined functions are to programmed by the user. These two functions concern:							
	1. Fur	nction for creating the c	downlo	ad file itself:				
	LONG	ParameterDownloadB	egin(Long IProjectNumber,				
				Long IDeviceNumber,				
			Long IIndexNumber,					
				Char* pcPDLFileName,				
				Long IMaxLengthFileNameBuffer,				
				Char* pcErrorText,				
				Long IMaxLengthErrorTextBuffer)				
	Pass pa	arameters:						
	• IPro	ojectNumber:	Cur	rently selected project number				
	• IDe	viceNumber:	Cur	rently selected device address				
	• Ilno	lexNumber:	Currently selected parameter directory number [199]					
	• pcF	PDLFileName:	Contains the complete file name for the created parameter download file. Buffer: max. length of the buffer for the name of the parameter download file.					
	• IMa	axLengthFileNameBuff						
	• pcE	ErrorText:	If necessary, user error text					
	• IMa	axLengthErrorTextBuffe	•					
	2. Fur	nction called up at the e						
	Long F	ParameterDownloadEn						
			L	ong IResult)				
	Pass pa	arameters:						
		PDLFileName:						
	Co file	•	name	es for the created parameter download				
			essage of the parameter download procedure eter download procedure O.K. has occurred					
		o functions must be p ed from it.	rogran	nmed in a DLL by the user and also				
	BW_P	AA1_(1)_{(2)}		(Single Write)				
	 Parameter directory number; the two functions to be implemented are located in INDIF410.DLL. 							
		Complete DLL name, i which the two functions implemented are locate	s to be					



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:								
		I = Job ID	[0120]						
		2 = FI comr	FI Commands for the MPCX Device Group: IFJ").						
			dance with Chapter "Elements of the FI Command"]						
		 Line 3 = FI Job Error Code (see Chapter "Error Codes") 							
	Note:	File and p	bath details must be enclosed in inverted commas.						
Example PAA1	00_BW_F	PAA1_2_"D	2:\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	Paramete directly inc		d command whereby the parameter download file is						
	BW_PAA2_(1) (Single Write)								
	_	·(·)	(Single Write)						
			ameter download file						
Response Structure	(1) = Cor name The resp	mplete para							
Response Structure	(1) = CornameThe responsible of the responsible of the responsible of the response of the res	nplete para onse to the column. Th I = Job ID	ameter download file e "PAA2" FI command consists of three lines, each						
Response Structure	 (1) = Corname The response Line 2 Line 2 	nplete para onse to the column. Th I = Job ID Chapter "F 2 = FI comr	ameter download file e "PAA2" FI command consists of three lines, each e meaning of the elements is as follows: [0120] FI Commands for the MPCX Device Group", IFJ). mand						
Response Structure	 (1) = Corname The response Line 2 Line 2 [String 	nplete para onse to the column. Th I = Job ID Chapter "I 2 = FI comr g, in accord	ameter download file e "PAA2" FI command consists of three lines, each e meaning of the elements is as follows: [0120] FI Commands for the MPCX Device Group", IFJ). mand dance with Chapter "Elements of the FI Command"]						
Response Structure	 (1) = Corname The response Line 2 (see Line 2 [String Line 3 	mplete para onse to the column. Th I = Job ID Chapter "I 2 = FI comr g, in accord 3 = FI Job E	ameter download file e "PAA2" FI command consists of three lines, each e meaning of the elements is as follows: [0120] FI Commands for the MPCX Device Group", IFJ). mand dance with Chapter "Elements of the FI Command"]						
Response Structure	 (1) = Corname The response Line 2 (see Line 2 [String Line 3 	mplete para onse to the column. Th I = Job ID Chapter "I 2 = FI comr g, in accord 3 = FI Job E Chapter "El	ameter download file e "PAA2" FI command consists of three lines, each e meaning of the elements is as follows: [0120] FI Commands for the MPCX Device Group", IFJ). nand dance with Chapter "Elements of the FI Command"] Error Code						
Response Structure	 (1) = Corname The response of the response o	mplete para onse to the column. Th 1 = Job ID Chapter "F 2 = FI comr g, in accord 3 = FI Job E Chapter "E File and p PAA2_"D:\[ameter download file e "PAA2" FI command consists of three lines, each e meaning of the elements is as follows: [0120] FI Commands for the MPCX Device Group", IFJ). mand dance with Chapter "Elements of the FI Command"] Error Code rror Codes") bath details must be enclosed in inverted commas.						
	 (1) = Corname The response Line 2 (see Line 2 (String) Line 3 (see Note: 00_BW_F FI command 	mplete para onse to the column. Th I = Job ID Chapter "H 2 = FI comr g, in accord 3 = FI Job E Chapter "El File and p PAA2_"D:\[E and	ameter download file e "PAA2" FI command consists of three lines, each e meaning of the elements is as follows: [0120] FI Commands for the MPCX Device Group", IFJ). mand dance with Chapter "Elements of the FI Command"] Error Code rror Codes")						
	 (1) = Corname The response of the response o	mplete para onse to the column. Th I = Job ID Chapter "F 2 = FI comr g, in accord 3 = FI Job F Chapter "E File and p PAA2_"D:\E and Column	ameter download file e "PAA2" FI command consists of three lines, each e meaning of the elements is as follows: [0120] FI Commands for the MPCX Device Group", IFJ). mand dance with Chapter "Elements of the FI Command"] Error Code rror Codes") path details must be enclosed in inverted commas. OWNLOAD.DAT" 00_BW_PAA2_"D:\DOWNLOAD.DAT" Answer						
	 (1) = Corname The response Line 2 (see Line 3 (see 0) Note: 00_BW_F FI commandation 1 	mplete para onse to the column. Th I = Job ID Chapter "H 2 = FI comr g, in accord 3 = FI Job E Chapter "En File and p PAA2_"D:\E and Column	ameter download file e "PAA2" FI command consists of three lines, each e meaning of the elements is as follows: [0120] FI Commands for the MPCX Device Group", IFJ). mand dance with Chapter "Elements of the FI Command"] Error Code rror Codes") Dath details must be enclosed in inverted commas. DOWNLOAD.DAT" 00_BW_PAA2_"D:\DOWNLOAD.DAT" Answer 01						
	 (1) = Corname The response of the response o	mplete para onse to the column. Th I = Job ID Chapter "F 2 = FI comr g, in accord 3 = FI Job F Chapter "E File and p PAA2_"D:\E and Column	ameter download file e "PAA2" FI command consists of three lines, each e meaning of the elements is as follows: [0120] FI Commands for the MPCX Device Group", IFJ). mand dance with Chapter "Elements of the FI Command"] Error Code rror Codes") path details must be enclosed in inverted commas. OWNLOAD.DAT" 00_BW_PAA2_"D:\DOWNLOAD.DAT" Answer						

Structure of Download FileThe structure of the download file corresponds to that of a Windows Ini
file. Indramat'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.

Parameter Upload: PAA

Designation	PAA	PArameter Access				
Explanation	Uploads complete parameter records from a selected device. read is written into an upload file with an identical structure to download file.					
Fl command	Programi 1. The fu LONG F Pass pa IProj IDev Curr PcU Max PcEr IMax max 2. Funct	ter upload command whereby two predefine amed by the user. These two functions concern unction supplies the complete name of the u ParameterUploadBegin(Long IProjectNum Long IDeviceNum Char* pcUploadF Long IMaxLength Char* pcErrorTex Long IMaxLength Char* pcErrorTex Long IMaxLength Char* pcErrorTex Long IMaxLength arameters: ojectNumber: Currently selected project number viceNumber: rently selected device address UploadFileName: Contains the complete file name for the para upload file to be created. xLengthFileNameBuffer: k. length of the buffer for the name of the para frorText: If necessary, user error text xLengthErrorTextBuffer: k. length of the buffer for the user error text tion called up at the end of the parameter up ParameterUploadEnd(Char* pcUploadFile Long IResult)	i: ipload file: nber, nber, iileName, FileNameBuffer, ct, ErrorTextBuffer) ameter rameter upload file.			



Pass parameters:

- pcUploadFileName: Contains the complete file names for the created parameter upload file.
- IResult: 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.

	exponed	nom it.						
	BR_PAA	1_(1)_{(2)}	}	(Single Read)				
	fur		ectory number; the two e implemented are located DLL.					
	wł	 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 [0120] (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	B = FI Job E	Error Code (see Chapter "E	rror Codes")				
	Note:	File and p	path details must be enclos	ed in inverted commas.				
Example PAA1	00_BR_PAA1_2_"D:\UserDir\USER.DLL"							
	FI comma	and	00_BR_PAA1_2_D:\UserD	ir\USER.DLL				
	Line	Column	Answer					
	1	1	01					
	2	1	00_BR_PAA1_2_D:\UserDi	USER.DLL				

FI command Parameter upload command whereby the parameter upload file is directly indicated.

BR_PAA2_(1)

3

(Single Read)

(1) = complete name of the parameter upload file

0

1

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.





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. Rexroth Indramat'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.						
Process Axis Configur	ation D	ata: PA	C				
			MWCX device group				
Designation	PAC Process Axis Configuration 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	(1) = NC	process nu	ımber [06]				
of Columns	2 = Phys	ical axis nu	mber [132]				
	3 = Main	axis mean					
	4 = Main	axis name	[Xi, Yi, Zi, Ui, Vi, Wi, Ai, Bi ,Ci, Si,] (i=[], [13])				
	5 = Axis	type	[see Chapter 6.2 "Data Tables"]				
Example PAC1	Read all p	process axis	s configuration data of device address 00.				
	FI comma	and	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				

Example PAA2 00_BR_PAA2_"D:\UPLOAD.DAT"

FI command



	FI command		00_BR_PAC1				
	Line	Column	Answer				
		5	0x82				
	3	1	2				
		2	3				
		3	5				
		4					
		5					
FI command	Output the	Output the axis configuration data of an NC process.					
	BR_PAC	BR_PAC2_(1) (Single Read)					
	(1) = NC process number [06]						
Response Structure	The following table shows the general structure of the response to the F 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	1 Column 1 Column 5				
Value Range/Meaning of Columns	. ,	process nu ical axis nu					
	3 = Main	axis meani	ning [see the chapter entitled "Data Tables"]				
	4 = Main	axis name	[Xi, Yi, Zi, Ui, Vi, Wi, Ai, Bi ,Ci, Si,] (i=[], [13])				
	5 = Axis	type	[see Chapter 6.2 "Data Tables"]				
Example PAC2	Read the a	axis configu	uration data of process 0 at device address 00.				
	FI comma	and	00_BR_PAC2_0				
	Line	Column	Answer				
	1	1	0				
		2	1				
		3	0				

Deactivate Parameters for an Offline Device PAD

MWCX device group

Designation	PAD	PArameter Deactivate)		
Explanation	If a device is in offline mode (DeviceStatus=OFF), this FI comm deactivates the parameter record in the offline device; then, NO v parameter record is present.				
FI command	BW_PAD	1	(Single Write)		
Response Structure	The response to the "PAD1" FI command consists of one line with o column.				
		Line 1	Column 1		

X1

0x81

4 5



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

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)	

Setting Parameters Active for an Offline Device: PAS

MWCX device group

Designation	PAS	PA ramet	er S et Active			
-	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					
	 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 [0120] (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:		es must be enclosed in inverted commas. This I is an FI job command.			
	For the offline device 00, the parameter data of the parameter download file D:\DOWNLOAD.DAT are set active.					
	FI comma	Ind	00_BW_PAS1_"D:\DOWNLOAD.DAT"			
	Line	Column	Answer			
	1	1	01			
	2	1	00_BW_PAS1_"D:\DOWNLOAD.DAT"			
	3	1	0			

ProVi Diagnosis Data: PDD

Designation	PDD	Provi Diagnosis Data					
Explanation	Data for F	Data for ProVi criteria analysis is output.					
FI command	BR_PDD (1) = Me	files to indicate the deta 01_(1)_(2){_(3)} ssage ID ssage type	ail in the editor. (Single Read) [ASCII characters] [1 = errors, 2 = messages, 10 = warnings, 11 = start requirements, 12=setup diagnosis]				



	(3) = Mo	dule numbe	er	[[199] ! only for message type 1 -2!				
Response Structure	The follo	-	e shows t	he g	eneral	structure	of the	PDD1	FI
	Lin	e 1	Column	1			C	olumn 5	
Meaning of the Columns	3 = Error 4 = POU	il morphem	[ASCII characters] (DWORD, decimal) [ASCII characters]						
Example PDD1	Indication of data of a ProVi error with ID 43923028 from module 3 in control unit 0.								
	FI comm	and	00_BR_PD	D1_4	3923028	3_1_1			
	Line	Column	Answer						
	1	1	STATION_	1_2					
		2	98243823						
		3	34985304						
		4	Station2.Mo	odule3	3				
		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[199] ! only for message type 1 -2								
Response Structure	command	J.	e shows ti			structure			FI
	L	.ine 1-n		Colu	ımn 1		Colu	nn 2	
Meaning of the Columns	1 = Varia 2 = I/O a	able morphe ddress	eme	-	CII char CII char	acters] (DV acters]	VORD,	decima	I)
Example PDD2	module 3	in control u				rror with I	D 4392	23028 fi	rom
	r		e an I/O add						
	FI comma	and Column	00_BR_PD	DD2_4	3923028	5_1_1			
	1	1	98243823						
		2	% 3.2.0						
	2	1	40923423						
		2	%Q23.21.7	,					
	3	1	34985304						
	_	2	% 100.3.5						
		_							

FI command

Determine the multilingual comments for displaying a detail.



	(1) = Me	03_(1)_(2){ ssage ID ssage type		(Single Read) [ASCII characters] [1 = error, 2 = messages, 10 = warnings, 11 = start requirements, 12 = setup diagnosis]		
	(3) = Mo	dule numbe	er	[199] ! only	for message type 1 -2!	
Response Structure	The following table shows the g command.		general struc	ture of the PDD3 FI		
	L	ine 1-n	Co	olumn 1	Column 2	
Meaning of the Columns		ment morpl comment	-	SCII characters	s] (DWORD, decimal) s]	
Example PDD3	from mod	lule 3 in cor			error with ID 43923028	
	FI comm	and	00_BR_PDD3	_43923028_1_1		
	Line	Column	Answer			
	1	1	98243823			
		2	Clamp open			
	2	1	40923423			
		Z	Clamp closed			
	(1) = Me (2) = Me	04_(1)_(2){ ssage num ssage type dule numbe	ber	(Single Read [ASCII charac [1 = error, 2 = 10 = warning 11 = start rec 12 = setup di [199] ! only	rters] messages, s, quirements,	
Response Structure	The follo		e shows the	general struc	ture of the PDD4 FI	
	Line	1-n	Column 1		Column 2	
Meaning of the Columns		sage is pre ria analysis		[YES, NO] [YES, NO]		
Example PDD4	Query of control 0.	the status	s of a ProVi e	error, number 1	1001 from module 3 in	
	This mes analysis.	sage is n	ot present at	the moment,	and there is a criteria	
	FI comm		00_BR_PDD4	_1001_1_1		
	Line 1	Column 1	Answer NO			
		2	YES			
FI command			MessageID of	a certain mess (Single Read	•	



 (1) = POU entity name (2) = Nw ID (3) = Message number 	[ASCII characters] [ASCII characters] [ASCII characters]
(4) = Message type	 [1 = error, 2 = messages, 10 = warnings, 11 = start requirements, 12 = setup diagnosis]
(5) = Module number	[199] ! only for message type 1 -2!

Note: The separator "!" is used in this command.

The following table shows the general structure of the PDD5 FI **Response Structure** command. Line 1-n Column 1 Column 3 ... [ASCII Meaning of the Columns 1 = Message ID 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 comma	and	00_BR_PDD5!Station2.Modul3!43493454!1001!1!1
Line	Column	Answer
1	1	240872342
	2	NO
	3	YES

Reading the Parameter Definition Table: PDT

Designation	PDT Parameter Definition Table				
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 p BR_P	Darameter definition table for the DT (Single Read)	ne selected device.		
Response Structure		llowing table shows the gener and "PDT".	al structure of the response to the FI		
		Line 1	Column 1		
Value Range/Meaning of Columns	1 =	1 = Parameter definition table in binary form Binary encoding of the parameter definition table in accordance with conventional control			
Example PDT	Read t	he parameter definition table f	or device 0.		



FI command		00_BR_PDT1			
Line	Column	Answer			
1	1	Binary data for the parameter definition table			

Programmed Feed Velocity: PFR

	-		M	NCX device group	
Designation	PFR Program	nmed Feed Rate			
Explanation	The value of the p MWCX device grou		te of the selec	ted device of the	
FI command	Output the current v Using the optiona conversion of the re	I second paramet	er it is possi	•	
FI command	CR_PFR_(1){_(2)} CC_PFR_(1){_(2)} CB_PFR_(1){_(2)} (1) = NC process n (2) = Required m (opt.)		(Single Read) (Cyclic Read) (Break Cyclic [06] [mm, inch]		
Response Structure	command "PFR". Or	The following table shows the general structure of the response to the F command "PFR". One line is output with three columns for the identifier, th current value of the programmed feedrate and the unit.			
	Line	1 Co	olumn 1	Column 3	
Value Range/Meaning of Columns	1 = Identifier 2 = Feedrate 3 = Unit	[form para	eedrate] at according to s meters] ording to settings	-	
	0 - 0111		meters]		
Example PFR	Read the programm	ned feedrate in NC p	rocess 0 of devi	ce address 00.	
	FI command	00_CR_PFR_0			
		Answe	r	-	
	Line	Column 1	Column 2	Column 3	
	1	F	30000.0	[mm/min]	
Example PFR	Read the programmed the displayed value			evice address 00.	
	FI command 00_CR_PFR_0				
		Answe	er	_	
	Line	Column 1	Column 2	Column 3	
	1	F	1181.1	[inch/min]	



Designation	PHD	PH ysical	Directory			
Explanation Fl command	Generates physical directory names according to the BDI data written. <u>Note:</u> This is based on BDI philosophy.					
Fi command	Generate physical directory names. BR_PHD1_(1)_(2)_(3)_(4)_(5)_(6)			(Cin		
	(1) = Pro		(3)_(4)_(3)_(0)	-	gle Write) PROJECT_NEUTRAL	
		,		-	PROJECT_DEFAULT]	
	(2) = Sec	ection ID			SECT_NEUTRAL SECT_BIN SECT_BASIC_DATA SECT_OEM_DATA SECT_CUSTOM_DATA SECT_PROG_DATA]	
	(3) = Dev	vice addres	S	othe	DEVADDR_NEUTRAL erwise the required ice address]	
	(4) = Pro	= Process ID			PROCESS_NEUTRAL prwise the required cess number]	
	(5) = Dat	(5) = Data type ID			sible write values see documentation I_DEFINITIONS.H)]	
	(6) = Lan	= Language ID			sible write values see documentation (WINNT.H)]	
Response Structure		The following table shows the general structure of the response to th command "PHD1".				
		Lir	ne 1		Column 1	
Value Range/Meaning of Columns	1 = F	 Physical directory name [complete physical directory name in accordance with the BDI data written] 			cordance with the BDI data	
Example PHD1	Requestir	ng the phys	ical directory nam	ne for		
·	PROJECT_NEUTRAL SECT_BIN DEVADDR_NEUTRAL PROCESS_NEUTRAL DATATYPE_NEUTRAL LANG_NEUTRAL					
	FI command XX_BR_PHD11_011_0_0				1_0_0	
	Line	Column	Answer			
	1	1	1 D:\Programme\Indramat\Mtgui\Bin			



Active NC Program Information: PPA

MWCX device group

Designation	PPA	Part Program Active					
Explanation		e active NC am number		n information a	bout the NC	memory and	
FI command	BR_PPA	_(1)	(Singl	e Read)			
	BC_PPA	_(1) (Cy		c Read)			
	(1) = Proc	ess numbe	er [06]				
Response Structure	command	I "PPA". Or		neral structure out with 3 colu ram name.			
		Line 1		Column 1	Column 2	Column 3	
Value Range/Meaning	1 = NC m	nemory		[A = memory A, B = memory B]			
of Columns	2 = NC p	rogram nur	nber	ber [0199]			
	3 = NC p	rogram name		[max. 32 ASCII characters]			
Example PPA	Read in N	IC process	0 at device a	ddress 00.			
	<u>Assumption:</u> The NC program numbered 01 and the with the name "Block4" is located in NC memory A; the memory is currently active.				4" is located		
	FI comma	mand 00_BR_PPA_0					
	Line	Column	n Answer				
	1	1	А				
		2	01				
		3	Block4				

Reference to Literature

See chapter entitled "Literature" [37].

Read NC Program Directory: PPD

Designation	PPD	Part-Program Director	У			
FI command	Reads the	e entries of the NC progr	am direct	ory.		
	BR_PPD	0_(1)_(2)		(Single Read)		
	(1) = Nur	mber in NC package dire	ctory	[1.	99]	
	(2) = NC	process number		[0.	6]	
Response Structure	command	ving table shows the ger I "PPD". The response co 5 columns.				
		Line 1n:	Colum	n 1		Column 5
Value Range/Meaning	1 = NC p	orogram number	[0	099]	
of Columns	2 = Prog	ram designation	[n	nax. 3	2 ASCII char	racters]
	3 = Prog	ram length	[b	yte]		
	4 = Date	of creation/last change of	of [C	D.MN	1.YY]	



program

5 = Time of creation/last change of [HH:MM:SS] program

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 comma	and	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

Expert or Import NC Program: PPN

Designation	PPN	Part-Pro	-Program NC			
FI command	Transfers an NC program from the NC program directory into an ASCII file (export).			ogram directory into an ASCII		
	BR_PPN	l_(1)_(2)_(3	3)_(4)	(Single Read)		
	(1) = Nur	mber in NC	package directory	[199]		
	(2) = NC	process nu	umber	[06]		
	(3) = Nur	mber of the	NC program	[199]		
	(4) = NC	block num	bering	[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	program	Without any NC block numbering, import the NC program with the NC program number 1 of the 2 nd NC package of the NC process 0 at the device address 00 into a file.				
	FI comma	and	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":

	Line 1	Column	Answer (P_ACK)			
	FI comma		00_BW_PPN_2_0	0_1_0_1_C:\Data\T1010001.TMP		
Example PPN	From the program device ad	number 1 o	ata\T1010001.TM of the 2 nd NC pa	P", export the NC program in NC ckage of the NC process 0 at the		
	issued. The meaning of the elements is as follows: (P_ACK) = P ositive ACK nowledge NC programs was exported.					
Response Structure				to acknowledge the FI command		
	Note:	Note: This FI command does not have any "Value to be written".				
		ectory	formation on the [DRIVE:\\X.Y]			
	en	try empty?	kage directory	[0 = without check (preset); 1 = with check] ! Optional !		
	(4) = NC	C block num	nbering	[0 = without number; 1 = with numbers] !		
	(3) = Nu	umber of the	e NC program	[199]		
	(2) = NC	C process n	umber	[06]		
		Imber in NC ectory	, pacкage	[199]		
			3)_(4)_(5)_(6)	(Single Write)		
FI command	Transfers an NC program from an ASCII file into the NC prog directory (import).					
	PROGR	AM END				
	BST .ST	ART				
	T0 BSR	M6				
	M05 [spi	ndle stop]				
	Z100					
	X5 Y0					
		Y0 F2000				
	G00 X60 Z-6 [infe	Y-30 ed motion]				
	G90 G96	G54 S1 200	00 F5000 M03			
	T1 BSR					
	SPF 1 [s	elect referen	ce spindle]			
	START					



MWCX device group

Renaming of an NC Part Program: PPN

Reference to Literature See chapter entitled "Literature" [37].

Renaming of an NC Program Package: PPP

Designation	PPP	Part Program Package			
Explanation		The name of an NC program package of the selected device of the MWCX device group is changed.			
FI command	BA_PPP_((1) = NC pr	1) rogram package	(Single Alternate) [199]		
Value to be written	Name of th	e NC program package	[max. 32 ASCII characters]		
	Note:	The value to be written is passe as an ASCII value in the "DataTi	•		
Response Structure	One line with one column is output to acknowledge the FI comma 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

PP The name of the NC program package numbered 1 in the NC package directory is to be renamed "FORM1".

FI comma	and	00_BA_PPP_1 Value to be written: FORM1
Line	Column	Answer
1	1	(BOF_FCT_OK)

Reference to Literature See chapter entitled "Literature" [37].

Reading an NC Record: PPS

MWCX device group

Designation	PPS	S Part Program Sequence			
Explanation	An NC record of an NC program from the selected device of the MWCX device group is read out.				
FI command	CR_PPS	CR_PPS_(1)_(2)_(3)_(4) (Single Read)			
	(1) = NC	memory	[1=memory A, 2=memory B]		
	(2) = NC	process nu	mber [06]		
	(3) = NC	program nu	umber [099]		
	(4) = NC	record number [09999]			
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].

Issuing SYS Messages Specific to the PCL: PSM

MWCX device group

Designation PSM PCL Sys Message

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_P	°SM1_(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	e to be written: Dev	vice addres	S		
Response Structure		lowing table show and "PSM1".	s the gene	eral structure	of the respo	onse to the FI
		Line 1		Column 1		Column 8
Value Range/Meaning 1 = of Columns		Status report		correctly ac WIN32 app [ERROR=3 been ackno	BYS message cknowledged blications] BYS messag owledged by within the pr	l by the e has NOT a WIN32
	2 =	Task name (LogInIf name)		[Task name that has triggered the SYS message]		
	3 =	SYS message n	umber	[contains th SYS mess	ne issued age number]
	4 =	Acknowledgeme	nt time	[contains th acknowled	ne pre-set gement time]
	5 =	Reference inform	ation		where applic nformation tr /alue]	
	6 =	Length of referer information	nce	[0 where NO reference information has been transferred]		
	7 =	Where applicable channel of the Fl NOT acknowledg	I that has	completed channel nu application	wledgement in time or the imber of the that has NC ged in time]	WIN32
	8 =	Where applicable name that has N acknowledged in	ОТ	completed		s have been e task name edged in

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 – value to be written: 00
Line	Column	Answer
1	1	READY
	2	WINPCL.EXE
	3	14
	4	30000
	5	00
	6	2
	7	
	8	



Programmed Spindle Speed: PSS

MWCX device group

Designation	PSS Programmed Spindle Speed						
Explanation	The value of the programmed spindle speed of the selected device of the MWCX device group is read out.			device of the			
FI command	CR_PSS_(1)_(2)	(S	(Single Read)			
	CC_PSS_(CC_PSS_(1)_(2)		yclic Read)			
	CB_PSS_(1)_(2)	(E	Break Cyclic	Read)		
	(1) = NC pr	ocess nur	mber [0	6]			
	(2) = Numb	er of spin	dle [1	3]			
Response Structure	The following table shows the general structure of the respons command "PSS". One line with three columns is output for the axi speed and the unit [1/min].			kis name, the			
		Line 1					Column 3
Value Range/Meaning	1 = Identifi	er	[S = spindle	e]			
of Columns	2 = Speed [format according to settings of the parameters]				arameters]		
	3 = Unit		1/min				
Example PSS	Read the speed of the 1 st spindle in NC process 0 of device address 00.			ess 00.			
	FI comman	d	00_CR_PSS_0)_1			
			Ar	nswer			
	Lin	е	Column 1	Colu	mn 2	С	olumn 3
	1		S:	750	0.0		1/min

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

Process Tool Management Configuration: PTC

Designation	PTC Process Tool Management Configuration					
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			Read) Read)		
	BC_PTC1	(Cycl	(Cyclic Read)			
Response Structure	ure The following table shows the general structure of the recommand "PTC1". The number of lines depends on the r CN processes. Each line consists of 9 columns for the return		on the nui	mber of defined		
	Line 1n:	Colum	n 1		Column 9	
Value Range/Meaning of the Columns	1 = NC process number 2 = Process name					
	3 = Tool management		[YES, I	NO]		
	4 = Tool memory		[[MAG/	AZINE], [1	[URRET]]	
	5 = Endlessly turning tool memory [YES, NO]					



6 = Number of tool memory locations	[0999]
7 = Number of tool spindles	[04]
8 = Number of tool grabbers	[04]
9 = Axis number of tool axis	[020]

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	

FI command Read tool management data of an NC process.

BR_PTC2_(1)	(Single Read)
BC_PTC2_(1)	(Cyclic Read)
(1) = NC process number	[06]
 The following table chows the	non-oral atmostering of the s

Response Structure

e 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 2 = Process name	[0	6]	
	3 = Tool management	[YE	S, NO]	
	4 = Tool memory	[[M/	AGAZINE], [T	URRET]]
	5 = Endlessly turning tool memory [YES, NO]			
	6 = Number of tool memory loca	ations [0	999]	



7 = Number of tool spindles	[04]
8 = Number of tool grabbers	[04]
9 = Axis number of tool axis	[020]

Note: 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	

Edit PROVI Message Files: PVA

Designation	PVA	PROVI-Messages Access			
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	1 (Single Write)			
	Note:	This command is an FI job command.			
Value to be written	No write value exists Write value exists		PROVI message files according to the current PLC project.		
			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 [0120] (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					
Example PVA1	 No write value is passed, i.e. the PROVI message files are generated according to the current PLC project.					
	FI comm	nand	00_BW_PVA	1		
	Line	Column	Answer			
	1	1	01			
	2	1	00_BW_PVA	1		
	3	1	0			
Explanation		d command PROVI mess		e most s	significant information on the	
FI command	BR_PVA	.1	(Sing	gle Read)	
Response Structure	comman		. For each av		truction of the answer of the FI ROVI message file, 1 line with	
		Line 1n	Colu	ımn 1	Column 10	
Value Range/Meaning	1 =	PROVI diag	nosis type	[120]		
of Columns	2 =	PROVI diag			owing designations can be	
		designation		returned StartCor ning, Se	ndition, Error, Message, War-	
	3 =	Module num	nber	[199]		
	4 =	PROVI diag and module			diagnosis type: module , see write value for A2]	
	5 =	Complete na PROVI mes	ame of the sage text file	[max. 20	00 ASCII characters]	
	6 =		uired for the sages in the	[figure ir	ASCII format]	
		Complete na PROVI inde		[max. 20	00 ASCII characters]	
		Memory required for the PROVI index data in the control				
		Total memo index) requi control	ry (for text + red in the	[figure ir	n ASCII format]	
		Total memo PROVI data index) requi control	(text +	[figure ir	n ASCII format]	



Example PVA1

The most significant information of 2 available PROVI message files are returned.

comm	and	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	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_PVA	2	(Single Write)
	Note:	This command is a	n 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:		ritten is passed to the "acValue" parameter 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	

Formatted Input / Output of PLC Variables: PVF

MWCX device group

Designation	PVF PLC Variable Formatted			
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) = Ident	ifier 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 1n: 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	read. The	following table indicates acted when reading out	depends on the data type of the variable the range in which the results string is 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	[-128127]	char
INT	[-3276832767]	short
DINT	[21474836482147483647]	long
USINT	[0255]	unsigned char
UINT	[065535]	unsigned short
UDINT	[04294967295]	unsigned long
BYTE	[0x000xFF]	unsigned char
WORD	[0x00000xFFFF]	unsigned short
DWORD;	[0x00000000xFFFFFFF]	unsigned long
TIME	[04294967295]	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</string></string>	Char[xx+1]] +1 i.e. room for the zero byte
REAL	[-3.402823567E+383.402823567E+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. –128127	1
INT	[8000 н (-32768)7FFF н (32767)]	2
DINT	[80000000 _н (-2147483648) 7FFFFFF _н (2147483647)]	4
USINT	[00 _н (0)FF _н (255)]	1
UINT	[00 _н (0)FFFF _н (65535)]	2
UDINT	[04294967295]	4
BYTE	[0x000xFF]	1
WORD	[0x00000xFFFF]	2
DWORD;	[0x00000000xFFFFFFF]	4
TIME	[04294967295]	4
CHAR	[\$00\$20,!~,\$7F\$FF]	1



	Data Typ	e Value	Range	Length (bytes)
	STRING	with a	g> bby <string> string is a character string maximum of as many characters as eclared for the string in the PLC</string>	XX+1
	REAL	[-3.40	2823567E+383.402823567E+38]	4
	Note:	Note: Binary array and structure elements will without space between (1 Byte Alignment).		
PLC - Example 1 PVF	address 00.			format from device
	<u>Assumpti</u> The "STK		able is declared as STRING in the F	PLC program.
	FI comma	and	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 program "Prog" at device address 00. <u>Assumption:</u> The WinPcl variable "STK_TXT" is declared in WinPcl program "I STRING.			
	FI comm	and	00_CR_PVF_:Prog.STK_TXT/1	
	Line	Answer		
	1	1	Repeat counter	
PLC - Example 2 PVF	address (<u>Assumpti</u>)0. <u>on:</u>	e PLC array "BEG_END" in ANSI fo iable is declared as BYTE with 2 ele	
	FI comma	and	00_CR_PVF_BEG_END/3	
	Line	Column	Answer	
	1	1	0x00	
	2	1	0x1F	
WinPcl - Example 2 PVF	program '	'Prog" at de	VinPcI array "BEG_END" in ANSI evice address 00.	format in WinPc
			"BEG_END" is declared in WinPcl pents.	program "Prog" as
	FI comma	and	00_CR_PVF_:Prog.BEG_END/3	
	Line	Column	Answer	
	1	1	0x00	
	2	1	0x1F	
PLC - Example 3 PVF		dress 00.	he PLC structure "MSTRCT" in A	SCII format from

Assumption:

The "MSTRCT" variable is declared as a structure in the PLC program as follows:



TYP STRUCT

T1	BOOL
T2	CHAR
Т3	STRING[16]
T4	TIME

END

FI comma	and	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

Write PLC variable.

END

FI comma	and	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

CW_PVF_(1) (Single Write) (1) = Identifier of the PLC variable [acc. to declaration part of the PLC1 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". One line is output with a column for acknowledgement of whether or not **Response Structure** the FI command has been executed successfully. Data element has been set (P_ACK) = Positive ACKnowledge Value Range of the The value ranges agree for the most part with the ANSI / ASCII resultvalue to be written in value ranges during read access. ANSI umlauts are thereby converted **ANSI / ASCII Format** 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 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 PLC - Example 4 PVF The value ranges agree with the binary result-value range during read access. For deviations to this, please refer to the following note:

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 comma	and	00_CR_PVF_BEG_END/3		
	Line	Column	Answer		
	1	1	(P_ACK)		
	Value to b	e written:			
	Value of d	lata elemer	nt 0x20 0x3f		
	Data code)	/3		
WinPcl - Example 5 PVF	PVF Write into the WinPcl byte array "BEG_END" in WinPcl progradevice address 00. The value is passed in ANSI format. Assumption:				
		cl variable n two eleme	"BEG_END" is declared in WinPcl program "Prog" as ents.		
	FI comma	and	00_CW_PVF_:Prog.BEG_END/3		
	Line	Column	Answer		
	1	1	(P_ACK)		
	Value to b				
		lata elemer			
	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.				
		<u>Assumption:</u> The "MSTRCT" variable is declared as a structure in the PLC program as follows:			
	TYP STR	UCT			
	T T				
	T:		ING[16]		
	T				
	END				
	FI comma		00_CW_PVF_MSTRCT.T3/2		
	Line	Column	Answer		
	1	1	(P_ACK)		
	Value to b	e written:			
	Value of	data eleme	ent Binary sequence: 43 4F 55 4E 54 45 52 00		
	Data cod	е	/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 "Prog" as follows:				
	TYP STR	-			
	T				
	T: T:		ING[16]		
	T				
	END				



	FI command		00_CW_PVF_:Prog.MSTRCT.T3/2			
	Line	Column	Answer			
	1	1	(P_ACK)			
	Value to b	be written:				
	Value of o	data elemer	nt Binary sequence: 43 4F 55 4E 54 45 52 00			
	Data code	Data code /2				
PLC - Example 7 PVF			the PLC structure "MSTRCT" from the structure ored in the C program at device address 00.			
	Assumption The "MST follows:		able is declared as a structure in the PLC program as			
	TYP STR					
	T	1 BOO 2 CHA				
			ING[16]			
	T END	4 TIME				
	To excha be used:	nge binary	data in a C program, the following "C" data type can			
	#pragma	pack(1)	//Write all elements			
			//without spaces next to each other.			
	typeder	struct				
	{	nsigned (char Tl:			
		har	T2;			
	C	har	T3[17]; //Space for zero byte			
	u	nsigned]	long T4;			
	} Tymst		// Declare structure			
	Tymstrc	t mstrct	; // Apply structure			
	FI comma	and	00_CW_PVF_MSTRCT/2			
	Line	Column	Answer			
	1	1	(P_ACK)			
	Value to b	be written: a	ddress of the C structure.			
	Value of	data eleme	nt &mstrct			
	Data code /2					
WinPcl - Example 7 PVF	/F Write the value of the WinPcl structure "MSTRCT" from the str "mstrct" previously stored in the C program at device address 00.					
	Assumption: The WinPcl variable "MSTRCT" is declared as a structure in Wi programm "Prog" as follows: TYP STRUCT T1 BOOL T2 CHAR T3 STRING[16]					
	T 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;
                  T3[17]; //Space for zero byte
      char
      unsigned long T4;
                  // Declare structure
} Tymstrct;
Tymstrct mstrct; // Apply structure
```

F	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 /2

Data code

ProVi Messages: PVM

Designation	PVM	ProVi Message	S			
Explanation		sages are output. ype or module.	These	messages a	re assigned t	o a particular
FI command	Output all I	ProVi messages.				
	BR_PVM	1_(1){_(2)}	(Singl	e Read)		
	BC_PVM'	1_(1){_(2)}	(Cycli	c Read)		
	(1) = Mess	sage type	10 = v 11 = s	ror, 2 = mess varnings, start requirem setup diagnos	ients,	
	(2) = Mod	ule number	[199]] ! only for me	essage type 1	-2!
	Output first	t ProVi messages	S.			
	BR_PVM2	2_(1){_(2)}	(Singl	e Read)		
	BC_PVM2	2_(1){_(2)}	(Cycli	c Read)		
	(1) = Mess	sage type	10 = v 11 = s	ror, 2 = mess varnings, start requirem setup diagnos	ients,	
	(2) = Mod	ule number	[199]] ! only for me	essage type 1	-2!
Response Structure		ring table shows nd "PVM2". The pending.				
	If there are	e no messages, th	ne numb	per of lines is	0.	
		Line 1n		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 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.



Response Structure	(1) = N (2) = N (3) = N	VM3_(1)_(2){_(3)} Aessage ID Aessage type Aodule number Ilowing table shows	(Single Read) [ASCII characters] [1 = error, 2 = messages, 10 = warnings, 11 = start requirements, 12 = setup diagnosis] [199] ! only for message type 1 -2! s the general structure of the "PVM3" FI				
	comma	Ind.		Colun	n 1		Column 16
		Line i		Colui			Column 10
Meaning of the Columns	1 =	Message text			[ASC	CII characte	ers]
	2 =	Message number			[ASC	CII characte	ers]
	3 =	Error category				CII characte pty no cate	
	4 =	Time stamp day			[mm	.dd.yyyy]	
	5 =	Time stamp hour			[hh:r	nm:ss]	
	6 =	Reference text avail	lable		[YES	S, NO]	
	7 =	Reference text			[ASC	CII characte	ers]
	8 =	Message ID			[AS0 deci		ers] (DWORD,
	9 =	Diagnosis source			[ASC CNC	CII characte	ers] (PLC,
	10 =	POE name			[ASC	CII characte	ers]
	11 =	Detail name				CII characte Implemen	
	12 =	Detail type			3 =	Action bloc Transition, Implement	
	13 =	Network number			[ASC	CII characte	ers]
	14 =	Variable name			[ASC	CII characte	ers]
	15 =	POU entity name			[ASC	CII characte	ers]
	16 =	POU type				program, function blo	ock]
	17 =	Analysis of criteria a	availabl	е	[YES	5, NO]	
	18 =	File name for addition information for mess		ext	[e.g.	HTML form	nat]
	19 =	File name for addition information for refer		ext	[e.g.	HTML form	nat]



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:\Programme\Indramat\MtGui\Project_000\ ProgramData\HMTL\DE\Error34.html		

Example PVM3	Reference text of a ProVi error with ID 43923028 from module 3 in control
	unit 0.

One after the other, all active ProVi messages are output. In the result, **FI command** 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.

	5	0	
	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	[199] ! only for messag	e type 1 -2!
	(3) = Clock frequency	[ASCII characters] Time	in ms
Response Structure	The following table shows the command.	general structure of the	e "PVM4" FI
	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 2^{nd} 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	

Download of PLC Retain Variables: PVR

Designation	PVR	PLC Variable Retain Backup		
Explanation	Download of PLC retain variables.			
FI command	dBW_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 [0120] (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 mus	st be taken in the use of upper and lower case letters.	
Upload of PLC Retain	Variable		MWCX device group	
Designation	PVR	PLC Varia	able R etain Backup	
Explanation	PLC retain variables are uploaded via all active processes.			
FI command	BR_PVR	R1!(1)	(Single Read)	
	(1) = Upl	oad file with	n 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:			
		1 = Job ID	[0120]	
	(see Chapter "FI Commands for the MPCX Device Group", IFJ).			
	 Line 2 = FI command [String, in accordance with Chapter "Elements of the FI Command"] 			
		-	Error Code (see Chapter "Error Codes")	
Example PVR	00_BR_P	vR1!"D:\Pr	ogram Files\Indramat\Mtgui\Temp\Upload.ini"/3	
	FI comm	and	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.			



Reading the PLC Variable Declaration: PVT

Designation	PVT PLC Variable Type			
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) (1) = Identifier of the PLC variable [acc. to declaration part of the PLC variable]			•
Response Structure	One line with 2 colur	nns is output fo	r each eleme	ent of the variables.
	Line 1	.n:	Colum	n 1 Column 2
	n = number of eleme	ents.		
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	e is declared as	WORD in t	he PLC program.
	FI command	00_BR_PVT_1	TEST	
		Α	nswer	
	Line	Column 1		Name
	1	TES	T	WORD
WinPcI: Reading a Variable	le <u>Assumption:</u> The WinPcl variable "TEST" is declared as WORD in WinPcl pro "Prog".			WORD in WinPcl program
	FI command	00_BR_PVT_:	Prog.TEST	
			nswer	
	Line	Column 1		Name
	1	TES	Τ	WORD
PLC: Reading a Structure	Assumption: The "TEST1" variable is declared as STRUCT in the PLC program.			in the PLC program.
	STRUCT E1 BOOL E2 INT E3 SINT END FI command 00_BR_PVT_TEST1			
		Α	nswer	
	Line	Colum		Column 2
	1	TEST1	I.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

BOOL
INT
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

<u>Assumption:</u> The "TEST2" variable is declared as ARRAY in the PLC program.

0..3

BOOL] OF

FI command	00_BR_PVT_TEST2	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 [01] 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 "Prog". ARRAY ["TEST3" is declared as a	ARRAY in WinPcl program	

RRAY	r

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	

<u>Assumption:</u> The data types are output according to IEC1131.

See also command PVF.



Repositioning Data: REP

MWCX device group

Designation	REP	REPositioning 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 1 st 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 2 nd parameter.			
FI command	CR_REP1	_(1)_(2)	(Single Read)	
	(1) = NC p	rocess number	[06]	
	(2) = Refer	ence system	[12] 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	• Line 1 contains the repos status (5 bit values).		
	positio	• 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	



	FI command		00_CR_REP1_0_1			
	Line	Column	Answer			
	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		
	314	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 rep	ositioning data.			
	CR_REP	2_(1)_(2)	(Cyclic Rea	d)		
	CC_REP	2_(1)_(2)	(Cyclic Rea	d)		
		2_(1)_(2) process nu	(Cyclic Brea umber [06]	ak)		
	(2) = Ref	erence sys	tem [12] Machir	e/ 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.



FI command		00_CR_REP2_0_1				
Line	Column	Answer	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			
314	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			

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.

Reference to Literature

See chapter entitled "Literature" [11].



SFC Diagnosis Data: SDD

Designation	SDD	SFC Dia	agnosis D ata			
Explanation	this data	can conce		steps, a		n the FI command ons or a definite ID
FI command	BR_SDE (1) = Mo			(Sin) [19	gle Read)	
	Note:	The sepa	arator "!" is u	sed in th	is command.	
Response Structure	The follo "SDD1".	wing table	e shows the	genera	l structure of	the FI command
	Lin	e 1	Column 1		•••	Column 7
Meaning of the Columns	1 = Step 2 = Deta			[1 = act]		action network,
	3 = Deta	il nomo			nsition] characters]	
	3 = Dela 4 = POU			-	characters]	
		il morphen	ne	[ASCII characters] (DWORD, decimal)		
	6 = Error	ID				WORD, decimal)
	7 = POU	entity nam	ne	[ASCII	characters]	
Example SDD1	Query dis	rupted ste	p of the "clan	np" chair	n in module 3 ir	n control unit 0.
	FI comma	and	00_BR_SDI	D1!3!Stat	tion03A.Clamp	
	Line	Column	Answer			
	1	1	Open			
		2	1			
		3	Aopen			
		4	SFC_1_2			
		5	98243823			
		6	34985304			
		7	Station2.Mo	dule3		
FI command	•					disrupted step.
)2!(1)!(2)!(dule numb	-	(Sing [19	gle Read)	
	. ,	C entity na		-	CII characters]	
	(2) = SF(3) (3) = Ste	•		-	CII characters]	
	(0) = 010			_,.oc		
	Note:	The sepa	arator "!" is u	sed in th	iis command.	



Response Structure	The following table shows the general structure of the FI command "SDD2".			e FI command		
	Lin	e 1	Column	1		Column 6
Meaning of the Columns	1 = Detail type			[1 = actions) 3 = trans	n block, 2 = act sition]	ion network,
	2 = Deta	il name		[ASCII cł	naracters]	
	3 = POU	ID		[ASCII ch	naracters]	
	4 = Deta	il morphem	е	[ASCII ch	naracters] (DWC	ORD, decimal)
	5 = Error	ID		[ASCII ch	naracters] (DWC	ORD, decimal)
	6 = POU	entity name	е	[ASCII cł	naracters]	
Example SDD2		ulty action of in control u		upted step	"open" of the	"clamp" chain in
	FI comma	and	00_BR_SE	D1!3!Stati	on03A.Clamp_O	pen
	Line	Column	Answer			
	1	1	1			
		2	AOpen			
		3	SFC_1_2			
		4	98243823			
		5	34985304			
		6	Station2.M	odule3		
FI command	Output the	definite ID t	o display the	e action, mo	onitor error or tran	sition.
		93!(1)!(2)!(3		Single Re	ead)	
	. ,	dule numbe		199]		
	. ,	C entity nan		ASCII cha	-	
	(3) = Det	ail type	[1 = action block, 2 = action network, 3 = transition]			n network,
	(4) = Det	ail name	I	ASCII cha	racters]	
	Note:	The sepa	rator "!" is	used in this	s command.	
Response Structure	The follo "SDD3".	wing table	shows the	e general	structure of th	ne FI command
		Line 1	Co	lumn 1		Column 4
Meaning of the Columns	1 = POU	ID		[ASCII cł	naracters]	
-		il morphem	е	-	naracters] (DWC	ORD, decimal)
	3 = Error				naracters] (DWC	
	4 = POU	entity name	е	[ASCII cł	naracters]	



Example SDD3	control unit 0.				
	FI comm	and	00_BF	R_SDD3!3!Station03A.C	lamp!1!aOpen
	Line	Column	Answ	er	
	1	1	SFC_	1_2	
		2	98243	823	
		3	34985	304	
		4	Station	n2.Module3	
FI command	Output th	e I/O addre	sses to	display a detail.	
	•	04!(1)!(2)!(3		(Single Read)	
		dule numbe		[199]	
	. ,	C entity nan		[ASCII characters]	
	(3) = Det	•		[1 = action block, 2] 3 = transition]	
	(4) = Det	ail name		[ASCII characters]	
	Note:	The sepa	rator "!	" is used in this comm	and.
Response Structure	The following table shows the general structure of the FI command "SDD4".			re of the FI command	
	Line 1-n Column 1			Column 1	Column 2
Meaning of the Columns	1 = Varia	ble morphe	eme	[ASCII character	s] (DWORD, decimal)
5	2 = I/O a			[ASCII character	- , , ,
Example SDD4		addresses		play the action "aOpen	" of the "clamp" chain in
	Three var	iables have	e an I/C	address.	
	FI comm	and	00_BF	R_SDD4!3!Station03A.C	lamp!1!aOpen
	Line	Column	Answ	er	
	1	1	98243	823	
		2	%I3.2.	0	
	2	2 1		423	
		2	%Q23	.21.7	
	3	1	34985	304	
		2	%I100	.3.5	
FI command	Determin	e the multili	ngual c	comments for displayin	g a detail.
	BR_SDD	05!(1)!(2)!(3	8)!(4)	(Single Read)	
	(1) = Module number [199]				
	(2) = SFC entity name			[ASCII characters]	
	(3) = Det	ail type		[1 = action block, 2] 3 = transition]	2 = action network,
	(4) = Det	= Detail name		[ASCII characters]	
	Note: The separator "!" is used in this command.				and.

Response Structure	The follo "SDD5".	wing table	shows the	general structu	ire of the FI command
	L	.ine 1-n		Column 1	Column 2
Meaning of the Columns		ment morp comment	heme	[ASCII character [ASCII character	rs] (DWORD, decimal) rs]
Example SDD5	module 3	in control u	init 0.		of the "clamp" chain in
	Two com	ments are r	eplaced by	another text.	
	FI comm	1		D5!3!Station03A.0	Clamp!1!aOpen
	Line	Column	Answer		
	1	1	98243823		
		2	Clamp oper	1	
	2	1	40923423		
		2	Clamp close	ed	
FI command			at has not b the online s		or the transition of a step
	BR_SDE	06!(1)!(2)!(3	5)	(Single Read	d)
	()	dule numbe		[199]	
	. ,	C entity nar	ne	[ASCII chara	-
	(3) = Ste	p name		[ASCII chara	cters]
	Note:	The sepa	rator "!" is u	sed in this comm	and.
Response Structure	The follo "SDD6".	wing table	shows the	general structu	ire of the FI command
	Lin	e 1	Column 1		Column 6
Meaning of the Columns	1 = Deta	il type		[1 = action block	ck, 3 = transition]
meaning of the oolumns	2 = Deta			[ASCII charact	-
	3 = POU			[ASCII charact	
	4 = Deta	il morphem	е	-	ers] (DWORD, decimal)
	5 = Error	· ID		[ASCII charact	ers] (DWORD, decimal)
	6 = POE	entity nam	е	[ASCII charact	ers]
Example SDD6			it has not b lule 3 in con		or the step "open" of the
	FI comm	and	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.Mc	odule3	



Set the Device Status Information: SDS

Designation	SDS	Set Devi	ce 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:	message	s are generated:	s transmitted, the following system
FI command	With this o be set.	command, th	ne device status i	nformation of ALL defined devices can
	BW_SD	S1_(1)	(Singl	e Write)
	Device s be set	tatus inform		evice status information OFF evice status information ON
Response Structure		wing table : I command		eral structure of the response to the
		Line	1	Column 1
Value Range/Meaning of Columns	1 = 5	Status repoi	rt	[(P_ACK)]
Example SDS1	Set devic	e status info	ormation to OFF	for ALL defined devices.
	FI comma	and	00_BW_SDS1_0)
	Line	Column	Answer	
	1	1	(P_ACK)	
FI command	With this can be se		the device stat	tus information for a selected device
	BW_SD	S2_(1)	(Sing	Jle Write)
		vice status mation to b		evice status information OFF vevice 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)]			[(P_ACK)]
Example: SDS2	Set devic	e status info	ormation to OFF	for the selected device 00.
	FI command 00_BW_SDS2_0			
	FI comm	and	00_BW_SDS2_0)
	FI comma Line	and Column	00_BW_SDS2_0 Answer)



Sequencer Data: SFD

				Ν	IWCX device group
Designation	SFD	SFC Da	ita		
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	BR_SFD (1) = Mod		er	(Single Read) [199] [ASCII characters]	
	Note:	The sepa	arator "!" is used	in this command.	
Response Structure	The follo	-	e shows the g	eneral structure	of the "SFD1" FI
		Line 1	Colu	umn 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 comma	and	00_BR_SFD1!3	Station03A.Clamp	
	Line	Column	Answer		
	1	1	Clamping device)	
		2	CLAMP		
FI command	Query the	data of a	step.		
	BR_SFD	2!(1)!(2)!(3)	(Single Read)	
	(1) = Moo	dule numb	er	[199]	
	(2) = SF0	C entity na	me	ASCII characters]	
	(3) = Ste	p name	I	ASCII characters]	
	Note:	The sep	arator "!" is used	in this command.	
Response Structure	command transitions	I. The nur S.	nber of lines dep	pends on the num	of the "SFD2" FI ber of actions and
	If there ar	e no detai	s the line numbe	r is 1.	
	Lin		Column 1		Column 3
	Line	2n:	Column 1		Column 6
Meaning of the Columns	Line 1				
	1 = Step	comment	[AS	CII characters]	
	2 = Maxi	mum time	[AS	CII characters]	
	3 = Minin	num time	[AS	CII 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	[199]
(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

Sequencer Messages: SFE

Designation	SFE	SFC Error				
Explanation	The step	chain messages of a r	nodule are	outpu	t.	
FI command	Output all	SFC messages.				
	BR_SFE	1_(1)	(Singl	e Rea	d)	
	BC_SFE	1_(1)	(Cycli	c Rea	d)	
	(1) = Mo	dule number	[199]		
	Output first SFC messages.					
	BR_SFE	2_(1)	(Singl	e Rea	d)	
	BC_SFE	2_(1)	(Cycli	c Rea	d)	
	(1) = Mo	dule number	[199]		
Response Structure	"SFE1" a	wing table shows the nd "SFE2". The num s pending.				
	If there ar	e no messages, the n	umber of lir	nes is	0.	
		Line 1n:	Colur	nn 1		Column 7
Meaning of the Columns	1 = Mess	sage text	[ASCI	l chara	acters]	
	2 = SFC	entity name	[ASCI	l chara	acters]	
	3 = Step	name	[ASCI	l chara	acters]	
	4 = Time	stamp day	[mm.d	d.yyyy	/]	
	5 = Time	stamp time	[hh:mr	n:ss]		
	6 = Туре	of error	-		or, 2 = monite event]	or error,
	7 = Is the	ere condition analysis?	[YES,	NO]		



Example SFD1

All SFC messages from module 2 in control unit 0. There are two messages.

FI comma	and	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 comm	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			



Sequencer Mode: SFM

MWCX device group

Designation	SFM	SFC Mod	de			
Explanation	Queries step chain mode.					
FI command	Query the mode of a step chain.					
	BR_SFM	1!(1)!(2)	(\$	Single Read)		
	BC_SFM	1!(1)!(2)	(0	Cyclic Read)		
	(1) = Moo	dule numbe	e r [1	199]		
	(2) = SF0	C entity nam	ne [/	ASCII characters]		
	Note:	The sepa	rator "!" is used i	n this command.		
-	The follo command	-	shows the ge	neral structure of the "SFM1" FI		
	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			Station03A.Clamp		
	Line	Column	Answer			
	1	1	1			

Software Installation Data: SID

Designation	SID Software Installation Data					
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)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	3 = Data c 4 = GBO v 5 = IF-DLI 6 = IF vers	allation directory lirectory version mode sion e pack info	[FI dii [in ac [from [from [from [from	files of the B rectory] cordance with INDRAMAT. INDRAMAT. INDRAMAT. INDRAMAT.	n BOF] ini] ini] ini from DLL	mode 420]



Line	• •	
-	Column	Answer
1	1	
	2	D:\Programme\Indramat\MTGUI\Bin
	3	
	4	005-22Vxx
	5	07.20
	6	07V00
	7	
	8	

Example SID1 Return information on the current installation.

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

Servo Lag of an Axis: SLA

Designation	SLA	Servo LAg				
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		e servo lag of the selected axis of the selected axis of the selected axis meaning.	ne device specified, related to			
	of the res	optional third parameter it is poss ult into mm or inches. If, however ating a measurement system serve	, a spindle is selected as an			
	CR_SLA	1_(1)_(2){_(3)}	(Single Read)			
	CC_SLA	1_(1)_(2){_(3)}	(Cyclic Read)			
	CB_SLA	1_(1)_(2){_(3)}	(Break Cyclic Read)			
	(1) = NC	process number	[06]			
	(2) = Axis	meaning	[011; 20]; (see Chapter 6.2 "Data Tables")			
	(3) = Req	uired measurement system (opt.)	[mm, inch]			
FI command		e servo lag of the selected axis of the alaxis number.	ne device specified, related to			
	conversior	e optional second parameter in n of the result into mm or inche as an axis, indicating a measu	es. If, however, a spindle is			
	CR_SLA	2_(1){_(2)}	(Single Read)			
	CC_SLA	2_(1){_(2)}	(Cyclic Read)			
	CB_SLA	2_(1){_(2)}	(Break Cyclic Read)			
	(1) = Phys	sical axis number	[132]			
	(2) = Req	uired 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	Colum	าท 1	Column	2	Column 3	Column 4
Value Range/Meaning of Columns	1 = Axis 2 = Serve			[according to settings of axis parameters] [according to settings of process parameters]				
	3 = Unit	5		- [acco	•	•	•	s parameters:
	4 = Servo	o lag			ding to the		unded up o ameter "indi	r down icated decimal
	Note: If the specified axis is not defined in the selected NC proces then the response in all columns is [].					ted NC process		
Example SLA1	Read the	servo lag	of the	Z axis	in NC pro	cess	0 of device	address 00.
	FI comma	and	00_0	CR_SL	A1_0_2			
					Answer			
	Line	Colum	n 1	Co	lumn 2	Co	olumn 3	Column 4
	1	Z1		2	.9124		[mm]	2.912
Example SLA1	Read the Values ar				is in NC p	roces	ss 0 of dev	ice address 00.
	FI comma	and	00_0	CR_SL	A1_0_2_in	ch		
					Answer			
	Line	Colum	n 1	Co	lumn 2	Co	olumn 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.				er = 3) at device			
	FI comma	and	00_0	CR_SL	A2_3			
				•	Answer			
	Line	Colum	n 1	Co	lumn 2	Co	olumn 3	Column 4
	1	Z1		2	.9124		[mm]	2.912

See chapter entitled "Literature" [40]. **Reference to Literature**



PLC Long Identification: SLI

MWCX device group

Designation	SLI SPS Long Identification				
Explanation	Returns the unit data from the PLC long identification.				
FI command		LC long identification.	(Single P	ad)	
	BR_SL	.1	(Single Re	ead)	
Response Structure	One lin	e with 15 columns is outpu	t for the retur	ned values.	
		Line 1	Column 1	Column	Column 15
Value Range/Meaning of the	1 =	Device address	[00.	63]	
Columns	2 =	Program number	[01.	99]	
	3 =	Project name	[ma:	x. 8 ASCII ch	aracters]
	4 =	Program name	[ma:	x. 8 ASCII ch	aracters]
	5 =	User name	[acc	. to passwore	d entry]
	6 =	Program length	[byte	es]	
	7 =	Compilation time	[LOI	NG] (coded ii	n long value)
	8 =	Compilation date	[8 A	SCII characte	ers]
	9 =	Compilation time	[8 A	SCII charact	ers]
	10 =	Download time	[LOI	NG] (coded ii	n long value)
	11 =	Download date	[8 A	SCII charact	ers]1
	12 =	Download time	[8 A	SCII charact	ers]
	13 =	Version of PLC long identification	[LOI	NG]	
	14 =	RUN flags	[HE]	X value]	
	15 =	Compiler info	[LOI	NG]	
Essential OLI	Deed th	a unit data from the DLC I	on a idontifico	tion	

Example SLI Read the unit data from the PLC long identification.

FI comma	and	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 comma	and	00_BR_SLI
Line	Column	Answer
	15	13

Reference to Literature see chapter entitled "Literature" [30].

Υ

Ζ

SERCOS Parameters: SPA

MWCX device group

Designation	SPA	SERCOS PArameter			
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	[17]		
	(2) = Parar	neter No.	in the format: X-Y-ZZZZ		
	(3) = Elem	ent coding	[standard or advanced format]		
Parameter No.	Format X-	Y-ZZZZ	Value Range		
	Х		S = standard data P = product data		

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) \rightarrow 48.

[0..00.15] = parameter record

[0...4095] = data block no.

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	н	20H	Decimal word	20000
Operating date	D	40H	(see Displaying the O	perating 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 ad allowed.	dditional zeros or plus and minus signs are not		
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	[00.7] = parameter record		
	Z	[04095] = data block no.		
	(see example SPA1/w	rite).		
Lists of Variable Length	 Lists always begin with two decimal numbers for the actual length an 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 b calculated using the attribute. The list elements are displayed according t the attribute. All parts of the list are separated from each other by a lin feed ("\n"). <u>Example:</u> 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 sh command "SPA1". Lin	ows the general structure of the response to the FI le 1 is output both when reading and when writing. Inly output when reading depending on the element		
	then the fir Line 1 is a displays th	ent coding has been requested in standard format st line is not applicable. status line that either contains the Sercos error or e 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=""></sercos>	<drive no.<br="">SERCOS error></drive>	0x0000	0x0000
2	Read: 1. Element corresponding to the element coding.			
n	Reading: (n-1). Element corresponding to the element coding.			

Example SPA1/ Read parameter S-0-0003 of the 3rd drive (element coding 0x48) read

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 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	00_BW_SPA1_3_S-0-0305_40 Value to be written: : P-0-0037				
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].

Active SERCOS Phase Switch-Over: SPH

Designation	SPH SERCOS PHase			
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	CR_SPH_(1)	(Single Read)		
	CW_SPH_(1)	(Single Write)		
	(1) = Physical axis number	[132]		
Value to be written	Phase	[2, 4]		
Response Structure	One line with one column is output for the returned value.			
	Line 1	Column 1		



MWCX device group

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

	Example SPH	
Read	SERCOS Phase	

Read the active phase of the first axis at device address 00.

FI command		00_CR_SPH_1
Line	Column	Answer
1	1	2

Example SPH Write SERCOS Phase Switch-over the first axis (write) after phase 4; phase 2 is active.

00_CW_SPH_1 Fl command Value to write: 4					
Answer					
Line	Column 1	Column 2			
1	52	1			

Note: Switching over from phase 2 to phase 4 returns the value [52] as the result in column 1. On switching over from phase 4 to phase 2, column 1 contains the value [50]. The result of column 2 is the physical axis number in both cases.

Reference to Literature See chapter entitled "Literature" [42].

Selected NC Program: SPP

Designation	SPP Selected	Part-Program	Number		
Explanation	The selected NC pro	gram is read.			
	CR_SPP_(1) CC_SPP_(1) CB_SPP_(1) (1) = NC process number		(Single Read) (Cyclic Read) (Break Cyclic Read) [06]		
Response Structure	The response to the FI command "SPP" cor columns for the identifier of the NC memor selected NC program.				
	Line 1		Colum	in 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 parameter		ngs of process
Example SPP	Read the selected NC program in NC process 0 of device address 00.			ce address 00.	
	FI command 00_CR_SPP_0		_0		
		swer			
	Line	Colum	n 1		Column 2
	1	В			55

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	record is a	ead the complete tool data record. For this FI command, the tool data accord is addressed via the NC process number, the tool memory and the acction number.			
	BR_TDA1	L_(1)_(2)_(3)	(Single Read)		
	(1) = NC process		[06]		
	(2) = Tool	memory	[M = magazine/turret, S = spindle, G = grabber]		
	(3) = Loca	ation number	[1999]		
sponse Structure		llowing table shows the general structure of the response to the FI and "BR_TDA1". The number of lines depends on the number of			

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 comma	and	03_BR_TDA1_0_M_21
Line	Column	Answer
1	01	10156
	02	Cutter head D80
	03	Μ
	04	21
	05	1
	06	1
	07	2
	08	1
	09	-p
	10	0
	11	M 21
	12	Μ
	13	
	14	Μ
	15	
	16	[cycl]



FI comma	and	03_BR_TDA1_0_M_21
Line	Column	Answer
-	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



FI comm	and	03_BR_TDA1_0_M_21
Line	Column	Answer
	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	[06]
(2) = Tool memory	[M = magazine/turret, S = spindle, G = grabber]
(3) = Location number	[1999]

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 19 = cutter data record			two- data eleme	0
	Note: The character " " (= 0x7D) is use number of the data element and individual lines of the table are also number n> < > <value n=""> < > <ele m> < ></ele </value>		t and the value t re also separated	to be written. The by a " ". <element< th=""></element<>	
Example TDA1 Write Tool Data Record		following data elemen ent number 002:		e tool data record: (ID) "drill Z72"	:
		ent number 008: ent number 107:		er of tool edges "1 L1 "100"	" and



	Tool magazine:	0 2	M = magazine and
	FI command		03_BW_TDA1_0_M_2
	Values to be written 002 Drill Z72 008 1 107 100	0	
FI command			record. For this FI command, the tool data C process number, the tool number and the
	BR_TDA2_(1)_(2)_(3)		(Single Read)
	(1) = NC process number	-	[06]
	(2) = Tool number		[19999999]
	(3) = Index number		[19999]

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

FI comma	and	03_BR_TDA2_0_1_1
Line	Column	Answer
1	01	10156
	02	Cutter head D80
	03	Μ
	04	21
	05	1
	06	1
	07	2
	08	1
	09	-р
	10	0
	11	M 21
	12	Μ
	13	
	14	Μ
	15	
	16	[cycl]
	17	[mm]

Example TDA2 Read the complete tool data record



FI comma	and	03_BR_TDA2_0_1_1
Line	Column	Answer
	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



FI comma	and	03_BR_TDA2_0_1_1
Line	Column	Answer
	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].

Loading Tool Data into the Control Unit: TDD

MWCX device group

Designation	TDD	Tool Data Download		
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 b	oasic data or cutter data	of a tool data record.	
	CW_TDD	_(1)_(2)_(3)_(4)	(Single Read)	
	(1) = NC p	process number	[06]	
	(2) = Tool memory [M = magazine/turret, S =spindl G = grabber, P = change positi			
	(3) = Tool	Fool memory locationIn the magazine/turret:[1999]In the spindle:[14]In the gripper:[14]In the change position:[14]		
	(4) = Cutte	er number	[0 = basic data, 19 = cutter data]	
Value to be written	Tool data	record	[basic and cutter data]	
	Note:		n is passed to the "acValue" parameter e "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. $2 = 1^{st}$ 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) = Positive ACKnowledge 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=""></data>			
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 3^{rd} cutter of the tool in the magazine at location number 2.

FI command		00_CW_TDD_0_M_2_3 <data record=""></data>
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.

Access to Tool Data Record: TDR

Designation	TDR	Tool Data Record			
Explanation	Returns a co	omplete basic data re	ecord and/or cutter data record of a tool.		
FI command	Read the t memory.	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	CC_TDR1_(1)_(2)_(3)_(4) (Cyclic Read)			
	CB_TDR1	_(1)_(2)_(3)_(4)	(Break Cyclic Read)		
	(1) = NC p	rocess number	[06]		
	(2) = Tool	memory	[M = magazine/turret, S = spindle, G = grabber, P = change position, X = index address]		



	(3) = Tool location(4) = Data record			In the s In the s In the As an [0 = to	magazine/turret: spindle: gripper: change position: index address: ol basic data, = cutter data]	[14] [14]
	Note: The index address reason, during the fir tool memories M, S, addressed via the rec			st acce G and	ess, access can P. Thereafter, the	only be made via
Response Structure	The following table shows the general structure of the response to th "CR_TDR1" FI command. One line is output with 28 (basic data) or 4 (cutter data) columns for the returned values.					
	L	ine 1	Colum	n 1		Column 28/40
Value Range/Meaning of the Columns	C	requested I lata		(see	. 28 data elemen basic value rang	e data)
		requested	tool cutter		. 40 data elemen value range of cu	
	Data elements 20 to 28 of the basic data and data elements 31 to 4 the cutter data are only available as options (depending on the sys parameters).			ng on the system		
Example TDR1	Read the process 0		data record	d of the	e 2 ^m tool in the	magazine in NC
	FI comma	and	00_CR_TDF	R1_0_M	_2_0	
	Line	Column	Answer			
	1	1	928 Miller D20			
		3	M			
		4	2			
		5	1234567			
		6				
		0	1234			
		7	1234 2			
		-				
		7	2	p		
		7 8	2	p		
		7 8 9	2 1 +	p		
		7 8 9 10 11 12	2 1 + 0 M1 M	p		
		7 8 9 10 11 12 13	2 1 + 0 M1 M 	p		
		7 8 9 10 11 12 13 14	2 1 + 0 M1 M M	p		
		7 8 9 10 11 12 13 14 15	2 1 + 0 M1 M M 	p	····	
		7 8 9 10 11 12 13 14 15 16	2 1 + 0 M1 M M [cycl]	p		
		7 8 9 10 11 12 13 14 15	2 1 + 0 M1 M M 	p		

FI comma	and	00_CR_TDR1_0_M_2_0
Line	Column	Answer
	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 Tool Data Record

FI command Read basic data record or cutter data record of a tool. Addressing is by means of the tool number and index number.

<u>Attention:</u> 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	[06]
(2) = Tool number	[19999999]
(3) = Index number	[19999]
(4) = Data record	[0 = tool basic data, 19 = 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	Colum	nn 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)		
	10.40 = requested tool data			40 data elementa alue range of cut	

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).



FI comma	and	00_CR_TDR2_0_2_1_0
Line	Column	Answer
1	1	928
	2	Miller D20
	3	Μ
	4	2
	5	2
	6	1
	7	2
	8	1
	9	-p
	10	0
	11	МО
	12	Μ
	13	0
	14	Μ
	15	0
	16	θ [cycl]
	17	θ [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

Example TDR2 Read the basic tool-data record of tool 2 / duplo number 1 in NC process 0.

Reference to Literature Se

See chapter entitled "Literature" [43].



Tool Insert Finish: TIF

Designation	TIF	Tool Inse	ert F inish	
Explanation	Complete		ion of a	tool. The reservation of the tool memory
Refer also to:	CR_TII a	nd CW_TLI	D1	
FI command	Complete insertion.(Single Read)(1) = NC process number[06](2) = Tool memory[M = magazine/turret, S =spindle, G = grabber, P = change position](3) = Location number in the tool storagein the magazine/turret: [1999] in the spindle: [14] in the change position: [14]			
Response Structure	One line is output with a column for acknowledgement of whether or not the FI command has been executed successfully.			
		= P ositive		•
Example TIF	Finish the insertion of a tool at location 5 in magazine in NC process 0 of device 00.			
	FI comma	and Column	00_CR_I Answer	ΊF_0_M_5
	Line 1		(P_ACK)	
Reference to Literature	See chap	ter entitled	. ,	
Tool Insert Initiate: TII				
				MWCX device group
Designation	тіі	Tool Inse	ert Initiate	

Designation	TII Tool Insert Initiate				
Explanation	Initiate the insertion of an individual tool. Reserves a location in the to memory.				
	inputting of "CW_TLD1". After	, the basic data and the cutter data are to be entered by repeated of "CW_TLD1". After the tool has actually been inserted in the ory, the procedure is completed by "CR_TIF".			
FI command	Initiate insertion.				
	CR_TII_(1)_(2)_(3)	(Single Read)			
	(1) = NC process number	[06]			
	(2) = Tool memory	[M = magazine/turret, S =spindle, G = grabber, P = change position]			
	(3) = Location number in the tool storage	in the magazine/turret: [1999] in the spindle: [14] in the gripper: [14] in the change position: [14]			

Response Structure One line is output with a column for acknowledgement of whether or not the FI command has been executed successfully.

(P_ACK) = Positive ACKnowledge 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	I (= 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].

Tool Basic Data List: TLB

Designation	TLB TooL Basic 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		[06]		
	(2) = Tool memory		[M = magazine/turret, S = spindle, G = grabber]		
	(3) = Location from		[1999]		
	(4) = Location to		[1999]		
	(5) = Data element		[128]		
	If more than one element is required as the 5 th entry parameter then th 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 1n:	Column 1	Column 2		Column 29
Value Range/Meaning of the Columns			[xxx = magazin/turret, SPx = spindle, GRx = gripper]		
	229 = Requested base tool data		[max. 28 data elements] (see value range)		



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. 7-287 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 r	number	[06]			
	(2) = Data element	t	[128]			
	If more than one element is required as the 2 nd entry parameter these are attached to the command with "_" and corresponding numbers					
Response Structure	The following table command "BR_TL tools. One line with more than one da increases accordin	.B2". The nun 2 columns is ta element is	nber of lines output per to	depends on to of the retu	the number of Irned values. If	
	Line 1n:	Column 1	Column 2		Column 29	
Value Range/Meaning of the Columns	1 = Tool memory		[xxx = magazine/turret, SPx = spindle, GRx = gripper]			
	229 = Requested base tool data		[max. 28 data elements] (refer to basic data value range, p. 7-287)			



Example TLB2Read 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



FI command		00_BR_TLB2_0_2_5_6_7
Line	Column	Answer
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].

Tool Data Record Elements: TLD

MWCX device group

Designation	TLD	TooL Data							
Explanation	memory. Cone telegra	Returns elements of the basic data or cutter data of a tool in the tool memory. Only basic data or data from <u>one</u> 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 com "CR_TDR"	nplete data record of basic data or cutting data, please refer to 							
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 Rea	d)					
	(1) = NC p	process number	[06]						
	(2) = Tool	memory	[M = magazine/turret, S G = gripper, P = chan X = index address]	•					
	(3) = Tool	location	in the magazine/turret: in the spindle: in the gripper: in the change position: as Index address:	[14] [14] [14]					
	(4) = Data	record	[0 = tool basic data, 19 = cutter data]						
	(5) = Data	element	The basic data: [128] of the tool edge 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). 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 5^{th} 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 structu command "CR_TLD1". One line with one returned value. If more than one data elen number of columns increases correspondingly			e column is ement is req	output for the		
	L	ine 1	Colum	n 1	. Up to	column 28/40	
Value Range/Meaning of the Columns	• •			-	nax. 28 data elements] ee basic value range data)		
	10.40 = requested tool cutter data[max. 40 data elements] (see value range of cutter data					-	
Example TLD1	Read the process 0	the name (basic data 2) of the 4 th tool in the magazine in Ne s 0.			magazine in NC		
	FI comma	and	00_CR_TLD	1_0_M_4_0	_2		
	Line	Column	Answer				
	1	1	MILLER D24	ŀ			
FI command	Write sin memory.	gle elemer	nt of basic c	lata or cutt	er data of a	tool in the tool	
	CW_TL	01_(1)_(2)_	(3)_(4)_(5)	(Single	Write)		
	(1) = NC	process nu	umber	[06]			
	(2) = Too	ol memory		[M = magazine/turret, S = spindle, G = gripper, P = change position, X = index address]			
	(3) = Too	bl memory l	ocation	in the spir in the grip	per: nge position:	[14] [14]	
	(4) = Dat	a record		[0 = tool b 19 = cu			
	(5) = Dat	a element		The basic data: [128] of the tool edge data: [140]			
	Data elements 20 to 28 of the basic data and data elements 31 to 40 the cutter data are only available as options (depending on the syste parameters). The response to access to data elements that are n available is N_ACK (Negative Acknowledge).					g on the system	
	Note:	reason, c tool mem	luring the fire	st access, a G and P. Th	access can one reafter, the	device. For this only be made via tool can also be	

Value to be written				see value ranges for basic and cutter data			
	Note:			is passed to the "acValue" parameter e "DataTransfer" routine.			
Response Structure			ith a column for been execute	or acknowledgement of whether or not d successfully.			
	(P_ACK)	= P ositive	ACK nowledge	Data element has been set.			
Example TLD1	Write data element 4 (warning limit magazine position in cutter 1.			it) in NC process 0 for the tool at the 3 rd			
	00_CW_TLD1_ Fl command Value to be wr						
	Line Column Answer						
	1 1 (P_ACK)						
Designation	TLD2	TooL Da	ita				
Explanation	means of	the tool nu	mber and inde	element(s) of a tool. Addressing is by x number. s executed, a tool identification run is			
FI command		2 (1) (2)	(3)_(4)_(5)	(Single Read)			
			(3)_(4)_(5)	(Cyclic Read)			
		(1)_(2)_((Break Cyclic Read)			
		-Process n		[06]			
	. ,	l number		[19999999]			
	(3) = Inde	ex number		- [19999]			
	(4) = Dat	a record		[0 = tool basic data, 19 = cutter data]			
	(5) = Dat	a element		of the base data: [128] of the tool edge data: [140]			
	Data elements 20 to 28 of the basic data and data elements 31 to 40 o the cutter data are only available as options (depending on the system parameters). The response to access to data elements that are no available is N_ACK (Negative Acknowledge).						
	If more th are attach	an one elei ned to the c	ment is require command with	ed as the 5 th entry parameter then these "_" and corresponding numbers.			
Response Structure	command returned	i "ČR_TLE value. If m	02". One line	eral structure of the response to the FI with one column is output for the data element is requested then the espondingly.			
	L	ine 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)			
	10.40 =	requested	tool cutter dat	a [max. 40 data elements] (see value range of cutter data)			

	FI comma	and	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 tool number + index number.					
	CW_TL	D2_(1)_(2)_	_(3)_(4)_(5)	(Single Write)		
	(1) = NC	-Process n	umber	[06]		
	(2) = Too	ol number		[19999999]		
	(3) = Ind	ex number		[19999]		
	(4) = Dat	ta record		[0 = tool basic data, 19 = cutter data]		
	(5) = Dat	ta element		of the base data: [128] of the tool edge data: [140]		
	the cutter paramete	r data are ers). The re	only available	sic data and data elements 31 to 40 of as options (depending on the system ccess to data elements that are not nowledge).		
Value to be written	Value of data element			see value ranges for basic and cutter data		
	Note:	as an AS	CII value in the	e "DataTransfer" routine.		
Response Structure	One line	as an AS is output w	CII value in the	e "DataTransfer" routine.		
Response Structure	One line the FI cor	as an AS is output w mmand has	CII value in the	e "DataTransfer" routine. for acknowledgement of whether or not ed successfully.		
Response Structure Example TLD2	One line the FI cor (P_ACK) Write da	as an AS is output w mmand has) = P ositive	CII value in the ith a column f been execute ACKnowledge 4 (warning lim	e "DataTransfer" routine. for acknowledgement of whether or not ed successfully.		
	One line the FI cor (P_ACK) Write da	as an AS is output w mmand has) = P ositive ta element number 1 in	CII value in the ith a column f been execute ACKnowledge 4 (warning lim	e "DataTransfer" routine. for acknowledgement of whether or not ed successfully. e Data element has been set it) in NC process 0 for tool number 2_0_3_1_1_4		
	One line the FI cor (P_ACK) Write da 3/index r	as an AS is output w mmand has) = P ositive ta element number 1 in	CII value in the rith a column for been execute ACKnowledge 4 (warning lim cutter 1.	for acknowledgement of whether or not ed successfully. e Data element has been set it) in NC process 0 for tool number 2_0_3_1_1_4		
	One line the FI cor (P_ACK) Write da 3/index r	as an AS is output w mmand has) = P ositive ta element number 1 in and	CII value in the ith a column for been execute ACKnowledge 4 (warning lim ocutter 1. 00_CW_TLD2 Value to be w	e "DataTransfer" routine. for acknowledgement of whether or not ed successfully. e Data element has been set it) in NC process 0 for tool number 2_0_3_1_1_4		
	One line the FI cor (P_ACK) Write da 3/index r FI comma Line	as an AS is output w mmand has) = P ositive ta element humber 1 in and Column	CII value in the ith a column for been execute ACKnowledge 4 (warning lime cutter 1. 00_CW_TLD2 Value to be we Answer (P_ACK)	e "DataTransfer" routine. for acknowledgement of whether or not ed successfully. e Data element has been set it) in NC process 0 for tool number 2_0_3_1_1_4		
Example TLD2	One line the FI cor (P_ACK) Write da 3/index r FI comma Line 1 TLD3	as an AS is output w mmand has) = Positive ta element number 1 in and Column 1 TooL Da	CII value in the rith a column for been execute ACKnowledge 4 (warning lime cutter 1. 00_CW_TLD2 Value to be we Answer (P_ACK)	e "DataTransfer" routine. for acknowledgement of whether or not ed successfully. e Data element has been set it) in NC process 0 for tool number 2_0_3_1_1_4		
Example TLD2 Designation	One line the FI cor (P_ACK) Write da 3/index r FI comma Line 1 TLD3 Returns a In contras element i data reco	as an AS is output w mmand has) = P ositive ta element number 1 in and Column 1 TooL Da any element st with the c is extended ord (0 = bas	CII value in the rith a column for been execute ACKnowledge 4 (warning lime cutter 1. 00_CW_TLD2 Value to be we Answer (P_ACK) ata t of the basic of command "TLI t to three pos	e "DataTransfer" routine. for acknowledgement of whether or not ed successfully. P Data element has been set it) in NC process 0 for tool number 2_0_3_1_1_4 written: 6.5 data or cutter data of a tool in any order. D1", for this FI command addressing an itions. The first position addresses the I, 1-9= cutter data) and the second and		
Example TLD2 Designation Explanation	One line the FI cor (P_ACK) Write da 3/index r FI comma Line 1 TLD3 Returns a In contras element i data reco third positi	as an AS is output w mmand has) = P ositive ta element number 1 in and Column 1 TooL Da any element st with the c is extended ord (0 = bas	CII value in the rith a column for been execute ACKnowledge 4 (warning lime cutter 1. 00_CW_TLD2 Value to be we Answer (P_ACK) ata t of the basic of command "TLI t to three possic data record ss the actual of	e "DataTransfer" routine. for acknowledgement of whether or not ed successfully. P Data element has been set it) in NC process 0 for tool number 2_0_3_1_1_4 written: 6.5 data or cutter data of a tool in any order. D1", for this FI command addressing an itions. The first position addresses the I, 1-9= cutter data) and the second and		
Example TLD2 Designation	One line the FI cor (P_ACK) Write da 3/index r FI comma Line 1 TLD3 Returns a In contras element i data reco third posit	as an AS is output w mmand has) = P ositive ta element humber 1 in and Column 1 TooL Da any element st with the c is extended ord (0 = bas tions addre	CII value in the rith a column for been execute ACKnowledge 4 (warning lime cutter 1. 00_CW_TLD2 Value to be we Answer (P_ACK) ata t of the basic of command "TLI t to three possic data record ss the actual of	e "DataTransfer" routine. for acknowledgement of whether or not ed successfully. Data element has been set it) in NC process 0 for tool number 2_0_3_1_1_4 written: 6.5 data or cutter data of a tool in any order. D1", for this FI command addressing an itions. The first position addresses the I, 1-9= cutter data) and the second and lata element.		

	Note: You should always make sure when requesting tool date the maximum net data length of 240 bytes is not exceed more than 240 bytes are requested then the contract returns the error message (NACK) /FI (1014).							
FI command	-	of basic dat 3_(1)_(2)_		ter data of a tool in the tool memory. (Single Read)				
		3_(1)_(2)_		(Cyclic Read)				
		3_(1)_(2)_		(Break Cyclic Read)				
	(1) = NC	-Process n	umber	[0 6]				
	. ,	l memory		[06] [M = magazine/turret, S = spindle, G = gripper, P = change position, X = index address]				
	(3) = Too	ol memory l	ocation	In the sp In the gr In the ch		[14] [14]		
	(4) = Dat	a element		[00194	40]			
	Data elements 020 to 028 of the basic data and data elements x31 to x of the cutter data are only available as options (depending on the syster parameters). The response to access to data elements that are r available is "N_ACK" (Negative Acknowledge).							
	If more than one element is required as the 4 th 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							
		addresse	d via the re	eceived ind	dex address.			
Response Structure	command returned	I "ČR_TLE	03". One nore than	line with one data	one column i element is re	esponse to the FI is output for the quested then the		
	Lin	e 1n:	Colu	mn 1		Column xxx		
Value Range/Meaning of the Columns		requested cutter data	basic tool o	data and	see value ra and cutter d	anges for basic ata		
Example TLD3		name of th				the magazine and		
	FI comma	and	00_CR_TL	_D3_0_M_4	4_002_103			
	Line	Column	Answer					
	1	1	MILLER D	24				
		2	100.00					
Designation	TLD4	TooL Da	ata					
Explanation	Returns a	ny element	t of the bas	ic data or	cutter data of a	tool in any order.		
-	In contras	st with the c	command "	TLD2", foi	r this FI comma	and 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 B	asic data –	tool name				
	103 C	utter 1 – re	maining tool li	l life			
	203 C	utter 2 – re	maining tool li	fe			
	Note: You should always make sure when requesting tool data that the maximum net data length of 240 bytes is not exceeded. more than 240 bytes are requested then the control un returns the error message (NACK) /FI (1014).						
FI command	Read the basic and cutter data of a tool according to the tool number index number.					tool number and	
	CR_TLD4_(1)_(2)_(3)_(4)			(Single	e Read)		
	CC_TLD	4_(1)_(2)_((3)_(4)	(Cyclic	Read)		
	CB_TLD	4_(1)_(2)_((3)_(4)	(Break	Cyclic Read)		
	(1) = NC	-Process nu	umber	[06]			
	(2) = Too	ol number		[19999999]			
	(3) = Inde	ex number		[19999]			
	(4) = Dat	a element		[0019	[001940]		
	of the cut paramete	ter data are rs). The re	e only availabl	pasic data and data elements x31 to x40 le as options (depending on the system access to data elements that are not nowledge).			
					e 4 th entry para corresponding	meter then these numbers.	
Response Structure	command returned	d "ČR_TLD value. If m	94". One line	general structure of the response to the F ine with one column is output for the one data element is requested then the orrespondingly.			
	Line	ə 1n:	Column 1			Column xxx	
Value Range/Meaning of the Columns		requested cutter data	basic tool data	ta and see value ranges for basic and cutter data			
Example TLD4			ool number 3/ process 0 of c			ne remaining tool	
	FI comma	and	00_CR_TLD4	_0_3_1_	002_403		
	Line	Column	Answer				
	1	1	TAPPER M5				
		2	100.00				
Deference to Literature	See abor	tor optitical	"Litoroturo" [4	01			

Reference to Literature

See chapter entitled "Literature" [43].



Tool Edge Data List: TLE

MWCX device group

Designation	TLE TooL Edge Data List						
Explanation	Returns th	ne cutter da	ata of the tool I	list.			
FI command	Read sele	ected cutter	data of the to	ol list.			
	BR_TLE1	_(1)_(2)_(3	8)_(4)_(5)_(6)	(Single	e Read)		
	(1) = NC p	process nur	nber	[06]	[06]		
	(2) = Tool	edge		[19]			
	(3) = Tool	memory			[M = magazine/turret, S = spindle, G = gripper]		
	(4) = Loca	tion from		[099	9]		
	(5) = Loca	ition to		[099	[0999]		
	(6) = Data			[140]			
	If more th are attach	an one eler led to the c	ment is require ommand with	ed as the 6 th ("_" and corre	entry parame esponding nu	ter then these mbers.	
Response Structure	The following table shows the general structure of the response to the F command "BR_TLE1". The number of lines depends on the number o tools. One line with 2 columns is output per tool for the returned values. I more than one data element is requested then the number of columns increases accordingly.					he number of med values. If	
	Line	e 1n:	Column 1	Column 2		Column 41	
Value Range/Meaning of the Columns	1 =	Tool merr	iory	SPx = sp	xxx = magazine/turret, SPx = spindle, GRx = gripper]		
	241 =	Requeste data	d tool cutter	-	max. 40 data elements] see value range Tool edge data)		
Example TLE1	Element r	number 002	2: Tool edge	status is req	uested.		
	Assumption	on:					
	NC pr	ocess num	ber: 0				
	• Tool e	edge:	1				
	• Tool r	nagazine:	M = n	nagazine and	1		
	 location 	on number	from: 1				
	 location 	on number	to: 3				
	Read data	a elements	2 and 3.				
	FI comma	-	00_BR_TLE1	_0_1_M_1_3_	2_3		
	Line	Column	Answer				
	1	1	001				
		2	d (tool worn o	ut)			
	2	1	002				
		2	_ (tool ok)				
	3	1	003				
		2	w (fallen below	v warning limit)		

FI command Read all cutter data of the tool list.



Response Structure	these are attached to the comma The following table shows the ge command "BR_TLE2". The num cutters. One line with 2 column values. If more than one data el			(Single Read) [06] [08] [140] equired as the 3 rd entry parameter then and with "_" and corresponding numbers. eneral structure of the response to the FI nber of lines depends on the number of ns is output per cutter for the returned		
	values. If columns	more than increases a	one data eleccordingly.	ement is reque		the number of
	Line	1n:	Column 1	Column 2		Column 41
Value Range/Meaning of the Columns	1 = Tool memory		[00 = magazine/turret, SP = spindle, GR = gripper]			
		Requested data	base tool	[max. 40 data (see value rar		ge data)
Example TLE2	• Elem	ent number	· 003: Residu	al tool life [0.00	00100.00	00]
	• Elem	ent number	004: Warnir	ng limit [0.110	0.00]	-
	• Elem	ent number	· 005: Maxim	um period of us	e [099999	999]
	• Elem	ent number	009: Length	L3 [-9999.9999	99999.999	99]
	Read in I position 1		0 the data	elements 3, 4,	5, 9 for all	tools at cutter
	FI comm	and	00_BR_TLE	2_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	5 1	106.8500 002			
	3	5 1 2	106.8500 002 100.0000			
	3	5 1 2 3	106.8500 002 100.0000 5.0000			
	3	5 1 2 3 4	106.8500 002 100.0000 5.0000 0.0000			
		5 1 2 3 4 5	106.8500 002 100.0000 5.0000 0.0000 132.9600			
	3	5 1 2 3 4 5 1	106.8500 002 100.0000 5.0000 0.0000 132.9600 003			
		5 1 2 3 4 5 1 2	106.8500 002 100.0000 5.0000 0.0000 132.9600 003 48.0000			
		5 1 2 3 4 5 1	106.8500 002 100.0000 5.0000 0.0000 132.9600 003			

FI comma	and	00_BR_TLE2_0_1_3_4_5_9
Line	Column	Answer
	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].

Tool Move : TMV

MWCX device group

Designation	тмv	Tool MoVe				
Explanation	data is mo	complete tool data record consisting of basic data and defined cutter ata is moved. This corresponds to the Rexroth Indramat BOF function fool Move".				
FI command	Move the s	elected tool data record.				
	CR_TMV_	<u>(1)_(2)_(3)_(4)_(5)</u>	(Single Read)			
	(1) = NC p	rocess number	[06]			
	(2) = Curre	ent tool memory	[M = magazine/turret, S = spindle, G = grabber]			
	(3) = Curre	ent location number	[1999]			
	(4) = Targe	et tool memory	[M = magazine/turret, S = spindle, G = grabber]			
	(5) = Targe	et location number	[1999]			
Response Structure		with one column is outpute meaning of the elements	ut to acknowledge the FI command is as follows:			
	(P_ACK) =	Positive ACKnowledge	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].

Information on Grippers/Spindles/Tool Magazine Locations: TPI

MWCX device group

Designation	TPI Tool Position Information					
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.</bis></von>					
	BR_T	⁻ PI1_(1)_{(2)_(3)}		(Single Read)		
	(1) =	Process number		[06]		
	(2) =	<von location=""> - Start location index in the tool storage location administration (optional para- meter) [11007] 14 = Gripper 14 58 = Spindle 14 91007 = Magazin 1999</von>		4		
	(3) =	administration (optional para- 58 = Spindle		14 = Gripper 1 58 = Spindle 1 91007 = Magaz	4	
Response Structure	The following table shows the general structure of the response to the command "TPI1". N lines, each with 3 columns, are output. Each corresponds to one occupied tool storage location.					
	Line 1n Column 1		I Column 2	Column 3		
Value Range/Meaning of Columns	1 =	Location name	-	[G1G4 = Gripper 14 S1S2 = Spindle 14 M = Tool storage location]		
	2 =	Location name	[1999]		
	3 =	Index address of the t LONG value	tool as a [LONG value]		



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	Μ
	2	1
	3	1456
4	1	Μ
	2	3
	3	3456
5	1	М
	2	9
	3	1678
	·	· · · · · · · · · · · · · · · · · · ·

Example TPI1	Reads the index addresses of ALL occupied tool storage locations of the
	process 0 of device 00.

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> ('fom 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)}	(S	ingle Read)	
	(1) = Process number	[0.	6]	
	(2) = <von location=""> - Star index in the tool storag administration (option meter)</von>	ge location 1. al para- 5. 9.	1007] 4 = Gripper 14 8 = Spindle 14 1007 = Magazi 999	4
	(3) = <von location=""> - End index in the tool storage administration (option meter)</von>	ge location 1. al para- 5. 9.	1007] 4 = Gripper 14 8 = Spindle 14 1007 = Magazi .999	4
Response Structure	The following table shows the general structure of the response to the command "TPI2". A line of n columns is output. Here, the column inc corresponds to the location index.			
	Line 1	Column 1		Column n
Value Range/Meaning of Columns	1 = Location status byte for I 2 = Location status byte for I		[0x00-0xF 1 [0x00-0xF	-

.... [0x00-0xFF] n = Location status byte for location index+n [0x00-0xFF]

3 = Location status byte for location index+2



[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)

Torque: TQE

MWCX device group

Designation	TQE	T or Q u E				
Explanation	command axis mean	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 grourelated to the code of the axis meaning.					device group,
	CR_TQE1	_(1)_(2)	(S	ingle Read)		
	CC_TQE1	_(1)_(2)	(0	yclic Read)		
	CB_TQE1	_(1)_(2)	(E	Break Cyclic R	ead)	
	(1) = NC pr	ocess numbe	r [0	6]		
	(2) = Axis r	neaning	[0	11; 20];		
Response Structure	The following table shows the general structure of the response command "TQE1". One line with three columns is output for the axis, the torgue and the unit [%].					
		Line 1		Column 1		Column 3
Value Range/Meaning	1 = Axis n	ame	e [according to settings of axis paramet			neters]
of Columns	2 = Torqu	2 = Torque		[format acc. to settings of the process parameter]		
	3 = Unit		[%]			
	Note:			ot defined in th columns is [].		d NC process
Example TQE1	Read the to					



	FI command	00_CR_TQE1_	_0_2			
	Line	Column 1	Column	2	Column 3	
	1	Z	-25.6		[%]	
FI command	Output the torque the physical axis n		xis of the dev	ice spe	cified, related to	
	CR_TQE2_(1)	(Single Read)				
	CC_TQE2_(1)		(Cyclic Rea	ad)		
	CB_TQE2_(1)		(Break Cyc	lic Rea	d)	
	(1) = Physical axis	number	[132]			
Response Structure	The following table shows the general structure of the response to the F command "TQE2". One line with three columns is output for the name of the axis, the torque and the unit [%].					
	Line	91	Column 1		Column 3	
Value Range/Meaning	1 = Axis name	to settings of	to settings of axis parameters]			
of Columns	•		t acc. to settings of the process eter]			
	3 = Unit	[%]				
	Note: If the specified axis is not defined in the selected NC process then the response in all columns is [].					
	Read the torque at the Z axis (physical axis number = 3) at device address 00 .					
Example TQE2	•	at the Z axis (p	hysical axis i	number	= 3) at device	
Example TQE2	•	at the Z axis (p	•	number	= 3) at device	
Example TQE2	address 00.	00_CR_TQE2_	•	number	= 3) at device	

Removing Tool Data Record: TRM

1

MWCX device group

[%]

-25.6

Designation	TRM	Tool ReMove			
Explanation	A complete tool data record consisting of basic data and defined cutter data is removed from the device. This corresponds to the Rexroth Indramat BOF function "Remove Tool from the Magazine List".				
FI command	Remove th	e selected tool data	record.		
	CR_TRM_	<u>(1)_(2)_(3)</u>	(Single Read)		
	(1) = NC p	rocess number	[06]		
	(2) = Tool	memory	[M = magazine/turret, S = spindle, G = gripper]		
	(3) = Locat	ion number	[1999]		
Response Structure			output to acknowledge the FI command ements is as follows:		
	(P_ACK) =	Positive ACKnowled	edge Data record has been removed		

Ζ



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].

Resetting Remaining Tool Life of a Tool: TRS

MWCX device group

Designation	TRS	Tool Res	Set			
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 ren	naining tool	l life of a tool:			
	CR_TRS_	_(1)_(2)_(3)) (Single Read)			
	(1) = NC p	process nur	mber [06]			
	(2) = Tool	(2) = Tool memory [M = magazine/turret, S = spindle, G = gripper, P = change position, X = index address]				
	(3) = Tool locationin the magazine/turret:[1999]in the spindle:[14]in the gripper:[14]in the change position:[14]as Index address:[09999999]					
	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			ith a column for acknowledgement of whether or not been executed successfully.			
	(P_ACK) :	= P ositive /	ACKnowledge 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 comma	and	00_CR_TRS_0_P_1			
	Line	Column	Answer			
	1	1	(P_ACK)			
Reference to Literature	See chapter entitled "Literature" [43].					



Requesting Watch List Allocations: WLA

MWCX device group

Designation	WLA	Watch Lis	st A llocati	on		
Explanation	Requests free watch list allocations. A maximum of ten free watch list allocations can be requested with one FI command.					free watch list
	BR_WL	A1_(1)	(Single Read)			
	(1) = Number of the requested free watch list numbers				d number of fre s identified her : 110	
Response Structure		wing table sl d "WLA1".	hows the	general strue	cture of the res	ponse to the FI
		Line 1	(Column 1	•••	Column n
Value Range/Meaning	1 = 1	I. free watch	n list alloc	ation	Value range:	116
of Columns	2 = 2	2 = 2. free watch list allocation			Value range:	116
	3 = 3	3 = 3. free watch list allocation			Value range:	116
	n =r	nth free watch list allocation			Value range:	116
Example WLA1	Request f	our free wat	tch list alle	ocations.		
	Assumption: Watch list allocations 3 and 5 are already assigned!					
	FI comm	and	00_BR_W	/LA1_4		
	Line	Column	Answer			
	1	1	1			
		2	2			
		3	4			
	1					

4

6

Freeing Watch List Allocations: WFL

MWCX device group

Designation	WLF Watch List Free							
Explanation	Previously requested watch list allocations are freed again.							
FI command	Free ALL a	assigned watch li	st allocations for th	e selected devic	Э.			
	BR_WLF1	(Sin	gle Read)					
	Note:	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	1 = 1. freed watch list allocation Value range: 116							
of Columns	2 = 2.	. freed watch lis	t allocation	Value ra	inge: 116			
	3 = 3.	. freed watch lis	t allocation	Value ra	inge: 116			



Value renge: 1 16

	n = r	nth freed watch list allocation Value range: 116						
Example WLF1	Free ALL	Free ALL assigned watch list allocations.						
	<u>Assumpti</u>	Assumption: The following watch list numbers have been allocated: 1, 2, 3, 4.						
	FI comma	FI command 00_BR_WLF1						
	Line	Column Answer						
	1	1	1 1					
		2	2 2					
		3	3					
		4	4					
FI command	Free the r	equired wat	ch list allocations for a se	elected device.				
	BR_WLF	⁻ 2_(1)_{(2)	(10)} (Single Rea	d)				
		= List of wa ns to be rele		of 10 watch lis ferred here to	t allocations be freed again.			
Response Structure	The follow		shows the general struc	cture of the res	ponse to the FI			
		Line 1	Column 1	•••	Column n			
Value Range/Meaning	1 = 1	l. freed wat	ch list allocation	Value ra	ange: 116			
of Columns	2 = 2	2. freed wat	ch list allocation	Value ra	ange: 116			
	3 = 3	 freed wat 	ch list allocation	Value ra	ange: 116			
	n = r	nth freed wat	tch list allocation	Value ra	ange: 116			
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 comma	and	00_BR_WLF2_1_3_4_8	}				
	Line	Column	Answer					
	1	1	1					
		2	3					
	1	-	1					

nth froad watch list allocation

Data of the Zero Offset Table: ZOD

MWCX device group

Designation ZOD Zero Offset Data

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:

3

4

- 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			
	49	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 ze	ero offset.				
	CW_ZOI	D_(1)_(2)_(3)_(4	l)_(5)	(Single Write)		
	(1) = NC			[1 = memory A; 2 = memory B]		
	(2) = NC	process number	r	[06]		
	(3) = Off:	set table number	r	[09]		
	(4) = Offs	set type		[offset type code]		
	(5) = Coo	de of the axis me	eanings	[08] 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 in the "DataTr		s passed to the "acValue" parameter tine.		
Offset Type	Code	Maguina	Evelopetic			
Oliset Type	Code 3	Meaning General offset	Explanatio			
	3	General onset	· · · · · · · · · · · · · · · · · · ·			
	4	CE 4	Adjustable	zara affaat		
	4	G54	Adjustable	zero offset		
				zero offset		
		 G59	 Adjustable			
Response Structure	9 Note: One line issued. T	 G59 The axis mea Tables". with one colur he meaning of th	Adjustable anings are nn is outpu	zero offset contained in chapter entitled "Data ut to acknowledge the FI command s is as follows:		
Response Structure Example ZOD	9 Note: One line issued. T (P_ACK) Write into	 G59 The axis mea Tables". with one colur he meaning of the P ositive ACK r	Adjustable Adjustable anings are nn is outp he element nowledge le O2 the v	zero offset contained in chapter entitled "Data ut to acknowledge the FI command s is as follows: Value has been written alue of the general offset of axis X in		
	9 Note: One line issued. T (P_ACK) Write into NC memo	 G59 The axis mea Tables". with one colurn he meaning of the meaning of the Positive ACK r o zero offset table ory A of NC proc	Adjustable Adjustable anings are nn is outp he element nowledge le O2 the v	zero offset contained in chapter entitled "Data ut to acknowledge the FI command s is as follows: Value has been written		
	9 Note: One line issued. T (P_ACK) Write into NC memory Assumption	 G59 The axis mea Tables". with one colur he meaning of the meaning of the P ositive ACK r to zero offset tab ory A of NC proc	Adjustable Adjustable anings are nn is outpo he element nowledge le O2 the v cess numbe	zero offset contained in chapter entitled "Data ut to acknowledge the FI command s is as follows: Value has been written alue of the general offset of axis X in er 0 at device address 00.		
	9 Note: One line issued. T (P_ACK) Write into NC memory Assumpti • There	 G59 The axis mea Tables". with one colur he meaning of the meaning of the P ositive ACK r to zero offset tab ory A of NC proc	Adjustable Adjustable anings are nn is outputhe element nowledge le O2 the v cess number	zero offset contained in chapter entitled "Data ut to acknowledge the FI command s is as follows: Value has been written alue of the general offset of axis X in		



FI comm	and	00_CW_ZOD_1_0_2_3_0 Value to be written: 0.111
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.

(Single Read)
(Cyclic Read)
(Break Cyclic Read)
[1 = memory A; 2 = memory B]
[06]
[09]
[09 Code of offset type]
[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	Code Meaning Explanation						
	0	Total		Sum of all active offset values				
	1	G50/G51		Programmable absolute / incremental zero offset				
	-							
	2	G52	Pro	Programmable work piece zero point				
	3	General offse	et Acts	t Acts additive to all offset types				
	4	G54	Adju	Adjustable zero offset				
	9	G59	Adju	istable zero offset				
Response Structure	The following table shows the general structure of the response to the F command "ZOD1". The answer consists of one to a maximum of n=1 lines (1 per axis), each with three columns for the name of the axis, valu of zero offset and the unit.							
		Line 1n:		Column 1 Column 3				
Value Range/Meaning of Columns	1 = Axis	designation		o settings of the a Zi, Ui, Vi, Wi, Ai,				
	2 = Valu	e		acc. to paramete angle PHI always		.xxxx]		
	3 = Unit	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.							
	<u>Assumpti</u>	<u>on</u> :						
	There	is a valid para	ameter	record in the devi	ce and			
	 the axes X_Y_Z (assigned at certain times) are defined 							

• the axes X, Y, Z (assigned at certain times) are defined.



FI command	00_CR_ZOD1_1_0_2_3						
Answer							
Line	Line Column 1 Column 2 Column 3						
1	Х	0.111	[mm]				
2	Y	0.000	[mm]				
3	*Z	0.000	[mm]				
4	PHI	0.0000	[deg]				

FI command

nd Output all zero offset values for the axes selected in a list.

CR_ZOD2_(1)_(2)_(3)_(4)_(5){_(6)} CC_ZOD2_(1)_(2)_(3)_(4)_(5){_(6)} CB_ZOD2_(1)_(2)_(3)_(4)_(5){_(6)}	(Single Read) (Cyclic Read) (Break Cyclic Read)
(1) = NC memory	[1 = memory A; 2 = memory B]
(2) = NC process number	[06]
(3) = Offset table number	[09]
(4) = Offset type	[offset type code]
(5) = Selection list for a max. of 10 elements	[08] 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 offse	et types			
	4	G54	Adjustable zero offset				
	9	G59	Adjustable zero offset				
Response Structure	Note: The axis meanings are contained in chapter 6.2, "Data Tables". 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.						
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=[13])						
	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].



Value Ranges

Basic Data

MWCX device group

Element No.	Name of the File Element		Writable?
1	Index address	09999999	No
2	Name (ID)	Max. 28 ASCII characters	Yes
3	Memory	M = magazine/turret, S = spindle, G = grabber	No
4	Location	0999	No
5	Tool number	19999999	Yes
6	Duplo number	19999	Yes
7	Correction type	15	Yes
8	Number of tool edges	19	Yes
9	Tool status	32 status bits with 0/1 (see following table)	Yes
10	Unassigned half-location	04	Yes
11	Former tool location	Memory [M/S/G] location [0999]	No
12	Memory of the next replacement tool	M = magazine/turret, S = spindle, G = grabber	No
13	Location of the next replacement tool	0999	No
14	Memory of the previous replacement tool	M = magazine/turret, S = spindle, G = grabber	No
15	Location of the previous replacement tool	0999	No
16	Time unit	0 = min, 1 = cycle	Yes
17	Unit of length	0 = mm, 1 = inch	Yes
18	Tool code	09	Yes
19	Display type	065535	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

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".



Bit	Symbol	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	е	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	В	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	р	1 0	Alternate tool identification	Processing tool No processing tool	No	There is a processing tool for every sister tool group

Tool status bits



Bit	Symbol	Value	Group name	Group information	Change able	Comment
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	С	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
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	08	Yes
2	Tool edge status	16 status bits with 0/1 (see following table)	Yes
3	Remaining tool life	-99.99100.00	Yes
4	Warning limit	0.1100.00	Yes
5	Max. life time	09999999	Yes
6	Time used	09999.999	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

Element Number	Name of the Data Element	Value Range	Writable?
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
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	Wrongcutting 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



Bit	Symbol	Value	Group name	Group information	Change able	Comment
12	*				No	Reserved
13	A	1 0	ANW 1	User cutter status bit 1	Yes	Any meaning
14	В	1 0	ANW 2	User cutter status bit 2	Yes	Any meaning
15	С	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

Flow Diagram for Command Groups

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.

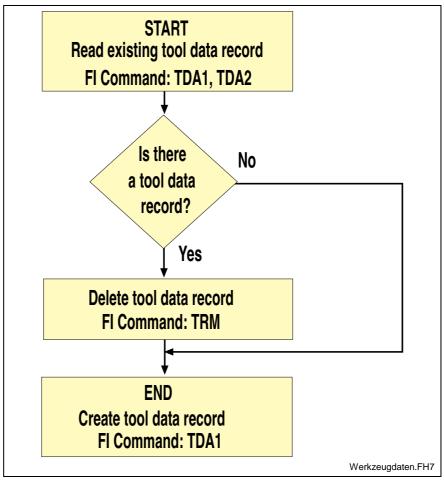


Fig. 7-9: Structure for handling tool data records



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. Returns one or more elements of the tool cutter data of several tools from **BR_TLE** the tool memory. No additional command required. Note: Returns a complete basic data record or cutter data record of a tool in the CR_TDR 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). No additional command required. Note:

Release Tool Location

 CR_TIF
 The indicated tool location is released after a temporary block.

 Note:
 No additional command required.



Remove Tool

Note:	Removing	а	tool	means	deleting	the	tool	from	the	tool
	database. the user.	The	e actu	al tool it	self must	be re	emove	ed prev	vious	ly by

- **CR_TRM** The tool data at this tool location is deleted from the database.
 - 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.

<u>Note:</u> 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.
- <u>Note:</u> 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.

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.

<u>Note:</u> 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. <u>Note:</u> No additional command required.



7.4 FI Commands for the MSCX Device Group

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:

Grou	р	Device Type	Address
MSC	Х	SERCANS-A, SERCANS-P	[00]
Note	re (r	lease note that the device address must be se espective FI command, e.g., 00_BR_SPA1_3_S efer also here to Chapter 6.1 "Elements ommand").	S-0-0003_48

Determining the Actual (Current) System Error: ASE

MSCX Device Group

Designation	ASE Actual System Error						
Explanation		The current system error is read out, whereby the response 0x0000 indicates that the SERCANS card is functioning correctly.					
FI command	CR_ASE	CR_ASE (Single Read)					
	CC_ASE		(Cyclic Rea	d)			
	CB_ASE		(Break Cycl	lic 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 t	he current	system error re	turns LWL rir	ng interrupted.		
	FI comma	and	00_CR_ASE				
	Line	Column	Answer				
	1	1	0x0000				
		2 0x0000					
		3	0x8009				
		4	0x0000				
Reference to Literature	See chapter entitled "Literature" [42].						

Deleting the Actual (Current) System Error: CSE

MSCX Device Group

Designation	CSE Clear System Error						
Explanation	An error reported by the SERCANS card is deleted.						
FI command	CW_CSE		(Single V	Vrite)			
	Value to b	e written:	The content of the co		alue paramete	er is not	
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 t	he actual (o	current) system	error:			
	FI comma	and	00_CW_CSE				
	Line	Column	Answer				
	1	1 0x0000					
	2 0x0000						
	3 0x0000						
	4 0x0000						
Defenses (s. 1.) (sectors	and charter entitled "Literature" [45]						

Reference to Literature see chapter entitled "Literature" [45].

Setting the Communication Timeout Time DCT

MSCX 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) = 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 c	device 00, the timeout time	e 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_DC	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 comma	and	00_BW_DCT	2			
	Line	Column	Answer				
	1 1 (P_ACK)						

Reading the Device Status Information: DSI

MSCX Device Group

Designation	DSI	Device Status Information					
Explanation	This allows the most important device status information to be read out. T following information is returned:						
	Type of i	information	Status	Statement			
	System e	error information					
	Mechanis	sm error information					
	Machine	key information	valid	Yes/No			
	Machine	key information					
	Machine	status information					
	Sercans i	information					
	Paramete	er download	running	Yes/No			
	PLC dow	nload	running	Yes/No			
	Firmware	e download	running	Yes/No			
	Offline/O	nline information	Communication?	Yes/No			
	Device si	mulation	switched on	Yes/No			
	Device st	atus information		ON/ OFF			
FI command	Read out	device status information for	ALL defined devices	S.			
	BR_DSI1	(Single Read)					
	BC_DSI1	(Cyclic Read)					
	BB_DSI1	(Break Cyclic Re	ead)				
	Note:	The "DSI1" FI command re group. Therefore, any valid the command line (see exa mechanism MUST be switc	device address car ample DSI1). The F	be indicated in I device polling			

Response Structure	FI command.						
		Line 1n	Colum	า 1	•••	Column 11	
Value Range/Meaning	1 =	device address		[0063]			
of Columns	2 =	System error infor	[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 infor	mation	[4 byte in HEX coding]			
	5 =	Is machine key inf valid?	[0 = not valid, 1=valid]				
	6 =	Machine status inf	[4 byte in HEX coding]				
	7 =	Sercans information	[4 byte	in HEX coding]	I		
	8 =	Is parameter down active?	[0 = parameter download not running 1 = parameter download running]				
	9 =	Is PLC download	active?		LC download no LC download ru		
	10 =	Is firmware downle active?	bad	-	LC download no LC download ru	•	
	11 =	Offline/Online info	[0 = device connection interrupted 1 = device connection O.K.]				
	12 =	Device simulation on?	<pre>d [0 = NO Simulation mode 1 = simulation mode]</pre>				
	13 =	Current device status information		[0 = Device status=OFF 1 = Device status=ON]			

Response Structure

The following table shows the general structure of the response to the "DSI1"

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	0000000
	6	0000000
	5	0
	7	0000000
	8	0
	9	0
	10	0
	11	1
	12	0
	13	1

	FI comm	I command 01_BR_DSI1					
	Line	Column	Answer				
	2	1	03				
		2	1				
		3	0				
		4	0000000				
		5	0				
		6	0000000				
		7	0000000				
		8	0				
		9	0				
		10	0				
		11	1				
		12	0				
		13	1				
FI command	Read out	t device stat	us information fo	or a sel	ected device.		
	BR_DSI2	2	(Single Read)				
	BC_DSI2	2	(Cyclic Read)				
	BB_DSI2	2	(Break Cyclic	Read)			
Response Structure		following table shows the general structure of the response to 2" FI command.					
	L	.ine 1n	Colum	n 1		Column 11	
Value Range/Meaning	1 =	device addre	SS	[006	63]		
of Columns	2 =	System error	r information	[0 = there is no system error 1 = there is a system error]			
		Mechanism e information	error	[0 = there is no mechanism error 1 = there is a mechanism error]			
	4 =	Machine key	information		e in HEX coding	1	
	5 =	Machine key valid?			ot valid, 1=valid]	-	
	6 =	Machine stat	us information	[4 byte in HEX coding]			
	7 =	Sercans info	rmation	[4 byte	[4 byte in HEX coding]		
		ls parameter active?	download	[0 = parameter download not running 1 = parameter download running]			
	9 =	Is PLC down	load active?	[0 = PLC download not running 1 = PLC download running]			
		ls firmware d active?	firmware download		[0 = PLC download not running 1 = PLC download running]		
	11 =	Offline/Online information [0 = device connect 1 = device connect					
						1 O.N.J	
		Device simu on?	llation switched	[0 = N	IO Simulation n imulation mode	node	



FI command		00_BR_DSI2
Line	Column	Answer
1	1	00
	2	0
	3	0
	4	0000000
	5	0
	6	0000000
	7	0000000
	8	0
	9	0
	10	0
	11	1
	12	0
	13	1

Example DSI2 Read the current device status information for the selected device.

Device Type and Accompanying Components: DTY

MSCX Device Group

Designation	DTY	Device	TY pe						
Explanation	The device type and the accompanying components of the selected device address are output.								
FI command	BR_D	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.							vice type is	
		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_DE	V.INI entry: Co	mpone	ent type	e1=	
	3 =	Component	type 2	IND_DE	V.INI entry: Co	mpone	ent type	e2=	
Example DTY1	Output the device type and the accompanying components of device address 00.						s of device		
	FI command 00_BR_DTY1								
	Answer								
		Line	Col	umn 1	Column	2	Co	olumn 3	
		1	SERC	CANS-A	NONE		١	NONE	

Read System Messages: MSG

MSCX Device Group

Designation	MSG	MeSsaG	Ge	
Explanation	Reading of system messages			
FI command	Message CC_MSC	3_(1)	(Cyclic Read)	
	(1) = SYS	S-Message	enumber	
	Note:	Exists on	nly as a cyclic command	
Response Structure	The respo data.	onse of the	e FI command 'MSG' consists of the system message	
Example MSG	00_CC_N	1SG_64	(64 = MSG_SYSERRGEN)	
	FI comma	and	00_CC_MSG_64/3	
	Line	Column	Answer	
	1	1	00	
Restriction	The follov	ving system	n messages:	
	SYS Mes	sage	SYS message numbers	
	MSG_PC	LUPDBEG	52	
	MSG_PA	RUPDBEG	G 24	
	MSG_FWAUPDBEG 82			
	These co	mmands ca	annot be used with the following programs:	
		OPC serve		
	Indramat	DDE serve	er	

Generating Physical Directory Names: PHD

Designation	PHD I	PHysical Directory			
Explanation	Generates physical directory names according to the BDI data written. <u>Note:</u> This is based on BDI philosophy.				
FI command	Generate ph	ysical directory names.			
	BR_PHD1_	(1)_(2)_(3)_(4)_(5)_(6)	(Single Write)		
	(1) = Projec	t ID	[-1= PROJECT_NEUTRAL -2= PROJECT_DEFAULT]		
	(2) = Sectio	n 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]		

	ot			[-1= PROCESS_NEUTRAL, otherwise the required process number]
	(5) =Data	a type ID		[possible write values see BDI documentation (BDI_DEFINITIONS.H)]
	(6) = Lan	iguage ID		[possible write values see BDI documentation (WINNT.H)]
Response Structure		The following table shows the general structure of the response to the F command "PHD1".		
		Lin	ne 1	Column 1
Value Range/Meaning	1 = F	Physical dire	ectory name	[complete physical
of Columns				directory name in accordance with the BDI data written]
	PROJECT SECT_BII DEVADDI PROCES	T_NEUTRAN N R_NEUTRA S_NEUTRA PE_NEUTRAN	AL AL	the BDI data written]
of Columns	PROJECT SECT_BII DEVADDI PROCES DATATYF	T_NEUTRAN N R_NEUTRAS_NEUTRA S_NEUTRAPE_NEUTR EUTRAL		the BDI data written] ne for:
of Columns	PROJECT SECT_BII DEVADDI PROCESS DATATYF LANG_NE	T_NEUTRAN N R_NEUTRAS_NEUTRA S_NEUTRAPE_NEUTR EUTRAL	AL AL AL AL	the BDI data written] ne for:

Set the Device Status Information: SDS

Designation	SDS	Set Device Statu	S		
Explanation	•	By this command, the device status information can be set; here, the configuration file IND_DEV.INI is adjusted as well.			
	Note:	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_SDS	S1_(1)	(Singl	le Write)	
	. ,	evice status0 = Device status information OFFformation to be set1 = 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 = 5	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_SD	S2_(1)	(Sing	le Write)	
	Device status information to $0 = Device$ status information OFF be set $1 = Device$ status information ON				
Response Structure		wing table	-	ral structure of the response to the	
		Line 1		Column 1	
Value Range/Meaning of Columns	1 = Status report [(P_ACK)]			[(P_ACK)]	
Example: SDS2	Set devic	Set device status information to OFF for the selected device 00.			
	FI comm	and	00_BW_SDS2_0)	
	Line Column Answer				

(P_ACK)

1

1

Software Installation Data: SID

Designation	SID	Software Installation	Data		
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	e installation data.			
	BR_SID1		(Single Rea	ad)	
	BC_SID1		(Cyclic Rea	ad)	
Response Structure	One line w	ith 8 columns is outpu	t for the returne	ed values.	
		Line 1	Column 1		Column 8
Meaning of the Columns	1 = Basic	directory	[EXE files of	the DOS-BOI	F]
	2 = FI inst	allation directory	[FI directory]		
	3 = Data d	directory	[in accordan	ce with DOS-I	BOF]
	4 = GBO v	version	[from INDRA	MAT.ini]	
	5 = IF-DLI	_ mode	[from INDRA	MAT.ini]	
	6 = IF ver	sion	[from INDRAMAT.ini from DLL mode 400]		
	7 = Servic	ice pack info [from INDRAMAT.ini from DLL mode 420]			DLL mode
	8 = Relea	se info	[from INDRAI 420]	MAT.ini from E	OLL mode





Return information on the current installation.

FI command		00_BR_SID1
Line	Column	Answer
1	1	
	2	D:\Programme\Indramat\MTGUI\Bin
	3	
	4	005-22Vxx
	5	07.00
	6	07V00
	7	
	8	

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

SERCOS Parameters: SPA

Designation	SPA	SERCOS PArameter				
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	[0254]			
	(2) = Parar	meter No.	in the format: X-Y-ZZZZ			
	(3) = Elem	ent coding	[standard or advanced format]			
Parameter No.	Form	nat X-Y-ZZZZ	Value Range			
		Х	S = standard data P = product data Y = SERCANS parameter			
		Y	[000.15] = parameter record			
		Z	[04095] = data block no.			
Element Coding	operating d request, th operating d	Z $[04095] = data block no.$ 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) \rightarrow 48The 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	Н	20H	Decimal word	20000
Operating date	D	40H	(see Displaying the Operating Date	
Operating date, when no list		80H		

Displaying the Operating DateThe display of the operating date depends on the parameter number
requested.DecimalDecimal 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
Х	S = standard data P = product data
Y	[00.7] = parameter record
Z	[04095] = 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 dateStandard 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.

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=""></sercos>	<drive no.<br="">SERCOS error></drive>	<global sercans<br="">error></global>	<drive no.<br="">Global SERCANS error></drive>
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 comman	nd	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.



DOK-CONTRL-FUN*INT*V07-AW01-EN-P

00_BW_SPA1_3_S-0-0305_40FI commandValue to be written: : P-0-0037							
	Answer						
Line	Column 1	Column 2	Column 3	Column 4			
1	0x0000	0x0003	0x0000	0x0000			

Reference to Literature Se

See chapter entitled "Literature" [41].

See chapter entitled "Literature" [46].

Active SERCOS Phase Switch-Over: SPH

MSCX Device Group

Designation	SPH	SERCOS PHase		
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	CR_SPH	(Single Read)		
	CC_SPH	(Cyclic Read)		
	CB_SPH	(Break Cyclic Read)		
	CW_SPH	(Single Write)		
Value to be written/ Result	The phase	e conditions allowed are shown by the numbers [04].		
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.			

Line	Column ²	Column 1 Column 2 Colum		Column 3	Column 4
1	<serco< td=""><td>S error></td><td><drive no.<br="">SERCOS error></drive></td><td><global sercans<br="">error></global></td><td><pre><drive caused="" error="" global="" has="" no.="" sercans="" that="" the=""></drive></pre></td></serco<>	S error>	<drive no.<br="">SERCOS error></drive>	<global sercans<br="">error></global>	<pre><drive caused="" error="" global="" has="" no.="" sercans="" that="" the=""></drive></pre>
2	Read: Write:	current phase previously phase			

Example SPH Switch-over (write) of the SERCANS control after phase 4; phase 2 is active.

In this case, the single result "Drive No." is set to 0x0000.

FI commar	nd	00_CW_SPH Value to be written: 4		
		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].





7.5 FI Commands for the MWMX and MWSX Device Group

The FI Commands described in this chapter are valid for the MWMX and MWSX device group. 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	[0063]
MWSX	ISP200-P-G2, ISP200-R-G2	[0063]

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 entitled "Elements of the FI Command").

Active System Error Messages: ASM

						e ao neo gio apo
Designation	ASM Active System Messages					
Explanation	The active system error messages 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 reference text are all output.					
FI command		Dutput the current system error messages pending of all active devices rom the MWSX device group.				
	BR_AS	M1	(Single	Read)		
	BC_AS	M1	(Cyclic I	Read)		
	BB_AS	M1	(Break C	Cyclic Re	ead)	
Response Structure	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"). The following table shows the general structure of the response to the F					ddress can be SM1").
	comma defined device i	nd "ASM1". The nu devices. Each line name, message nu r there is an referen	mber of lir e consists mber, me	nes (1 r s of 7 cc ssage sta	=15) depends o lumns for the o atus, short text a	n the number of device address,
		Line 1n	Colur	nn 1		Column 7
Value Range/Meaning	1 =	device address		[0015]	
of Columns	2 =	Device name		- [max. 3	2 ASCII charac	ters]
	3 =	Message numbe	r	[0150]	
	4 =	Type of message	9	[F = fau	ılt/error, D = dia	gnosis]
	5 =	Short text		[max. 5	4 ASCII charac	ters]
	6 =	Reference text		[x= exis	ts, = does not e	exist]
	7 =	2 bytes of addition information for the message			red to resolve tl tion "@" (see A	



Read the current system error messages of all defined devices of the Example ASM1 MWSX device group.

Assumption: The following three devices are defined:

- Device address 01, •
- Device address 07 and
- Device address 10.

FI comma	and	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
2	1	07
	2	Drill station 1
	3	74
	4	F
	5	SLM time monitoring
	6	x
	7	0
3	1	10
	2	Drill station 2
	3	1
	4	D
	5	Error has been corrected.
	6	X
	7	0

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)

The following table shows the general structure of the response to the FI **Response Structure** 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 error message.

		Line 1n	Column 1		Column 7
Value Range/Meaning of Columns	1 = 2 =	device address Device name	[0015 [max_3] 2 ASCII charac	tersl
	3 =	Message number	-		, long]
	4 =	Type of message	[F = fau	ılt/error, D = dia	ignosis]



- 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 information "@" (see ASM5) for the message number

Example ASM2 Read the current system error messages of device address 01.

Assumption:

5 =

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	х
	7	0

FI command Output the current system error messages 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 m devices	ax. of 10 MWSX	[00_01_0215]

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 1n		Column 1		Column 7
Value Range/Meaning	1 =	device address	[0015]		
of Columns	2 =	Device name	[max. 32 ASCII characters] [0150]		acters]
	3 =	Message number			
4 =		Type of message	[F = fault/er	ror, D = d	iagnosis]
	5 = Short text [max. 54 ASC		CII chara	acters]	
6 = F		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) ber		



Example ASM3 Read the current system error messages 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	Х		
	7	0		
2	1	10		
	2	Drill center 2		
	3	1		
	4	D		
	5	Error has been corrected.		
	6	X		
	7	0		

FI command Output the current system error messages 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 error message.

Line 1n		Column 1		Column 7	
1 =	device address	[0015]			
2 = Device name		[max. 32 ASCII characters]			
3 =	Message number	[0150]			
 4 = Type of message 5 = Short text 6 = Reference text 		[F = fault/error, D = diagnosis]			
		[max. 54 ASCII characters]			
		[x= exists, = does not exist]			
7 =	2 byte additional information for the message number	is required to resolve the inform "@" (see ASM5) per		e information	
	2 = 3 = 4 = 5 = 6 =	 1 = device address 2 = Device name 3 = Message number 4 = Type of message 5 = Short text 6 = Reference text 7 = 2 byte additional information 	1 =device address[0015]2 =Device name[max. 32 ASCI3 =Message number[0150]4 =Type of message[F = fault/error5 =Short text[max. 54 ASCI6 =Reference text[x= exists, = d7 =2 byte additionalis required to r"@" (see ASM:	1 =device address[0015]2 =Device name[max. 32 ASCII charact3 =Message number[0150]4 =Type of message[F = fault/error, D = diag5 =Short text[max. 54 ASCII charact6 =Reference text[x= exists, = does not error7 =2 byte additionalis required to resolve th"@" (see ASM5)	



Value

Example ASM4	Read the current system error messages of all defined devices of the
	MWSX device group.

Assumption:

The following devices are defined:

- Device address 01 and
- Device address 10.

FI comm	and	01_BR_ASM4_MWSX		
Line	Column	Answer		
1	1	01		
	2	Drill center		
	3	71		
	4	F		
	5	PLC battery voltage too low.		
	6	Х		
	7	0		
2	1	10		
	2	Drill center 2		
	3	1		
	4	D		
	5	Error has been corrected.		
	6	х		
	7	0		

FI command Output the reference text for the currently pending error message, related to the device and the message number.

BR_ASM5_(1)_(2)	(Single Read)
(1) = Message number	[0150]

(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 1n		Column 1	Column 1		
Value Range/Meaning of Columns	1 =	device address	[0015]	[0015]		
	2 =	Device name	[max. 32 ASCII characters]			
	3 =	Message number	[0150]			
	4 =	= Type of message [F = fault/error, D = diagno		sis]		
	6 =	Reference text	[max. 14 lines with a max. 78 characters/line]		78	



MWMX and MWSX device groups

Example	ASM5
---------	------

SM5	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].

Trigger Control Reset: CRT

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. Note: Carrying out a reset completely re-initializes the device. During initialization, communication is temporarily interrupted (inherent to design). CW_CRT (Single Write) **FI command** Value to be written Trigger reset 0 Note: The value to be written is passed to the "acValue" parameter in the "DataTransfer" routine. The return value of the "DataTransfer" routine is [0] if the write procedure **Response Structure** 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** 00 CW CRT 0 Value to be written **Reference to Literature** See chapter entitled "Literature" [26].



MWMX and MWSX device groups

Setting the Communication Timeout Time DCT

Designation	DCT		ommunicatio	n Timoquit		
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) = 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 de	evice 00, th	e timeout time	e is set 1500 ms.		
	FI comm	and	00_BW_DCT	1_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_DC	Г2		(Single Write)		
Response Structure	The response to the "DCT2" FI command consists of one line with one column.			mmand consists of one line with one		
		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 comma	and	00_BW_DCT	2		
	Line	Column	Answer			
	1	1	(P_ACK)			

Long ID of PLC Data Block: DIS

Designation	DIS	Data Identification String
Explanation	directory en length and	ong ID (directory entries) of the PLC program. Included in the tries are the number of the entry in the directory, the name, date and time of creation and/or details of the last time the ata record was changed.
FI command	BR_DIS2	(Single Read)
	BC_DIS2	(Cyclic Read)
	BB_DIS2	(Break Cyclic Read)
Response Structure	•	g table shows the general structure of the response to the "DIS2". . The response consists of a line with six columns.



	Line 1			Column 1		Column 6
Value Range/Meaning	1 =	Number in F	[0199]			
of Columns	2 =	Name of the	e PLC program	n	[max. 8 ASC characters]	
	3 =	Length of th	e PLC progra	im	[byte]	
		Date of crea program	ation/last char	nge to PLC	[DD.MM.YY	l
		Time of crea PLC progra	ation/last chai m	nge to the	[HH:MM:SS]]
		Date of crea program	ation/last char	nge to PLC	[DD.MM.YY	YY]
Example DIS2	Note: If there is no valid NC package in the selected NC mem then all columns contain []. Read the directory entries of the PLC program at address 00. Assumption: There is a valid PLC program in the selected device.					
	FI comm	nand	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].

Reading the Device Status Information: DSI

Designation	DSI	Device Status Inform	ation			
Explanation	This allows the most important device status information to be read out. The following information is returned:					
	Туре о	of information	Status	Statement		
	System	error information				
	Mechar	nism error information				
	Machin	e key information	valid	Yes/No		
	Machin	e key information				
	Machin	e status information				
	Sercan	s information				
	Parame	eter download	running	Yes/No		
	PLC do	wnload	running	Yes/No		
	Firmwa	re download	running	Yes/No		
	Offline/	Online information				
	Device	simulation	switched on	Yes/No		
	Device	status information		ON/OFF		

FI command	Read ou	ut device status inf	formation fo	or ALL	defined devices	5.	
	BR_DS	l1 (Sin	(Single Read)				
	BC_DS	l1 (Cy	(Cyclic Read)				
	BB_DS	l1 (Bre	ak Cyclic	Read)			
	Note:	group. Therefor the command	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		owing table show FI command.	s the gene	eral str	ucture of the r	esponse to the	
		Line 1n	Colum	n 1		Column 11	
Value Range/Meaning	1 =	device address		[006	3]		
of Columns	2 =	System error infor	System error information		[0 = there is no system error 1 = there is a system error]		
	3 =	Mechanism error		[0 = there is no mechanism error			
		information	0 = there is a mechanism error				
	4 =	Machine key infor		[4 byte in HEX coding]			
	5 =	Machine key infor valid?	mation	[0 = not valid, 1=valid]			
	6 =	Machine status in	formation	[4 byte in HEX coding]			
	7 =	Sercans informati	on	[4 byte in HEX coding]			
	8 =	Is parameter down active?	nload	[0 = parameter download not running 1 = parameter download running]			
	9 =	Is PLC download	active?	[0 = PLC download not running 1 = PLC download running]			
-		Is firmware downle active?	oad	[0 = PLC download not running 1 = PLC download running]			
	11 = Offli		rmation	[0 = device connection interrupted 1 = device connection O.K.]			
	12 =	Device simulation on?	n switched	[0 = NO Simulation mode 1 = simulation mode]			
	13 =	13 = Current device status information)evice-Status=C)evice-Status=C		

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)

and	01_BR_DSI1
Column	Answer
1	01
2	0
3	0
4	0000000
5	0
6	0000000
7	0000000
8	0
9	0
10	0
11	1
1	03
2	1
3	0
4	0000000
5	0
6	0000000
7	0000000
8	0
9	0
10	0
11	1
	Column 1 2 3 4 5 6 7 8 9 10 11 2 3 4 5 6 7 8 9 10 11 2 3 4 5 6 7 8 9 10

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 1n	Column 1		Column 11
Value Range/Meaning of Columns	1 =	device address	[00	-	
2 = System error information $[0 = there is]$		here is no systen here is a system			
	3 =	Mechanism error information	1 =	here is no nechanism error here is a mechar error]	nism
	4 =	Machine key infor	mation [4 by	te 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.]

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	0000000
	5	0
	6	0000000
	7	0000000
	8	0
	9	0
	10	0
	11	1

Device Type and Accompanying Components: DTY

Designation	DTY	Device T	Ype			
Explanation		The device type and the accompanying components of the selected device address are output.				
FI command	BR_C	DTY1	(Single Read	I)		
Response Structure	"DTY1 type, a	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 FI Command"		
	2 =	Component ty	•	ND_DEV.INI-E		
	3 =	Component ty	•	ND_DEV.INI-E	•	



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				

Diagnosis Window Data: DWD

Designation	DWD Diagnosis Winde	ow D ata			
Explanation	Diagnostic messages are o they can be output direc applicable, different types report, are returned simultar	tly in the	e diagno	sis overviev	w, i.e., where
FI command	Output all diagnostic message	ges.			
	BR_DWD1_(1){_(2)}	(Single Read)			
	BC_DWD1_(1){_(2)}	(Cyclic	Read)		
	(1) = Type of diagnosis window	3 = gen 10 = sta	eral errors	= sequence s, 4 = messa ditions, t = setup diag	ages,
	(2) = Module number	[199] !	only for v	vindow type	1 -4 !
	Output first diagnostic mess	ages.			
	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		19] ! only for window type 1 -4 !		
Response Structure	The following table shows "DWD2" FI commands. The messages pending. Differen diagnosis.	e number	of lines of	depends on	the number of
	If there are no messages, th	e numbei	r of lines is	s 0.	
	Line 1n	Col	umn 1		Column 12
Meaning of the Columns	1 = Message text		[ASCII cł	naracters]	
	2 = Time stamp day	[mm.dd.yyyy]			
	3 = Time stamp hour		[hh:mm:ss]		
	4 = Reference text avai	lable	le [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	r	[031] (N	ITC-NC) [0]	(MTA-NC)
	9 = 2 byte additional infe	ormation	[ASCII cl	naracters] (N	ITC NC)
	10 = Message group		[19999] (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 comma	and	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						



	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		
		10	120		
		17			
Reference to Literature	See chapter entitled "Literature" [13].				
Existing errors: EDE					
				MMMAX and MMAX device groups	
				MWMX and MWSX device groups	
Designation	EDE	Existing	Diagnosis Error		
Explanation				ntrol unit or in a module is queried. IC errors, MTA200 errors or ProVi	
FI command	Query wh	ether there	are errors in this	control unit.	
	BR_EDE	1	(1	Single Read)	
	BC_EDE	1	(Cyclic Read)	
Response Structure	The follo		shows the ge	eneral structure of the "EDE1" FI	
		Line	1	Column 1	
Meaning of the Columns	1 = Error	exists	[YES	S, NO]	
Example EDE1	Do errors	exist in cor	ntrol unit 0?		
	FI comm	and	00_BR_EDE1		
	Line	Column	Answer		
	1	1		YES	
F I commente		other or pe	t orrors ovict in a	specific module	
FI command				specific module.	
	BR_EDE BC_EDE		-	Single Read) Cyclic Read)	
			-		
	(1) = 100	dule num		199]	

Example DWD2 First diagnostic message from module 3 in control unit 0. There are two messages.

FI command

00_BR_DWD2_4_3

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	S, NO]	
Example EDE2	Do errors	exist in mo	odule 1 on	contro	l unit 0?	
	FI comma	and	00_BR_E	DE2_2		
	Line	Column	Answer			
	1 1 NO					
Existing Diagnosis Wi	ndow: E	DW				
					MWMX and	I MWSX device groups
Designation	EDW	Existing	Diagnosis	Windo	W	
Explanation	Which typ	es of diagr	nosis wind	ow exis	st 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 0n		C	Column 1	Column 2
Meaning of the Columns	1 = Type of diagnosis [1 = CNC error, 2 = sequence errors, window 3 = general errors, 4 = messages, 10 = start requirements, 11 = warnings, 12 = setup diagnosis]				its,	
	2 = Modu	ule number		0 = Dia	characters] gnosis windov to any module	v type does not
Example EDW1	All types of	of diagnosis	s window i	n contr	ol unit 0.	
		three diag				
	FI comm	and	00_BR_E	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 ty	pes for	a module.	
	BR_EDV	V2_(1)		(Single Read)	
	(1) = Mo	dule numbe	er	[199]	



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 0n	(Column 1	Column 2
Meaning of the Columns	1 = Type of diagnosis [1 = CNC error, 2 = sequence errors, window 3 = general errors, 4 = messages]				
	2 = Modu	ıle number	0 = Dia	characters] gnosis window to any module	v type does not
Example EDW2	All types of	of diagnosis	s window in Modu	ule 3, Control u	ınit 0.
	There are	two diagno	osis windows.		
	FI comma	and	00_BR_EDW2_3	}	
	Line	Column	Answer		
	1	1	1		
		2	3		
	2	1	2		
		2	3		
FI command	Query a s	pecific type	e of diagnosis wir	ndow.	
	BR_EDV	V3_(1){_(2)	} (Single I	Read)	
	 (1) = Type of diagnosis (1) = CNC error, 2 = sequence errors, 3 = general errors, 4 = messages, 10 = start requirements, 11 = warnings, 12 = setup diagnosis] 				nessages,
	(2) = Mo	dule numbe	er [199]!	only for window	w type 1 -4 !
Response Structure	The following table shows the general structure of the "EDW3" command.			e of the "EDW3" FI	
		Line '	1	(Column 1
Meaning of the Columns	1 = Type of diagnosis window exists [YES, NO]				10]
Example EDW3	Query whether or not an NC error window exists in module 3, control unit 0.				
	FI command 00_BR_EDW3			_3	
	Line	Column	Answer		
	1	1	YES		
Reference to Literature	See chap	ter literature	e [13].		

Existing SPS Diagnoses: EPD

Designation	EPD	Existing PLC Diagnosis
Explanation	command,	C diagnostic types exist is queried. Depending on the FI specific types are queried or else the diagnostic types for a module are output together.
FI command	Query whice BR_EPD1	n PLC diagnostic types are available on a control unit. (Single Read)

Response Structure	The follo	-	e shows the	general	structure of the "EPD1" FI		
		Line	1		Column 1-3		
Meaning of the Columns	1 = Start	requireme	nt exists		[YES, NO]		
U U		ning exists			[YES, NO]		
	3 = Setu	p diagnosis	exists		[YES, NO]		
Example EPD1	Querv PL	C diagnost	ic types in con	trol unit ().		
	FI comm	-	00_BR_EPD1				
	Line	Column	Answer				
	1	1	YES				
		2	NO				
		3	YES				
FI command	Querv wh	ich PLC dia	agnostic types	are avai	lable in a module.		
	BR_EPD		.9		e Read)		
	(1) = Mo	dule numbe	er	[199]	·		
Response Structure	The follo	-	e shows the	general	structure of the "EPD2" FI		
		Line	1		Column 1-3		
Meaning of the Columns	1 – Mess	sages exist			[YES, NO]		
meaning of the columns	2 = Error	•			[YES, NO]		
		chains exis	st		[YES, NO]		
	Oversethe		antin turnan in	Madula			
Example EPD2	FI comm		00_BR_EPD2		2 on Control unit 0.		
	Line	Column	Answer	_ Z			
	1	1	NO				
		2	YES				
		3	YES				
F I							
FI command	Query a specific PLC diagnostic type. BR_EPD3_(1){_(2)} (Single Read)						
		ssage type		-	essages, 3 = SFC,		
	10 = warnings, 11 = start requirements, 12 = setup diagnosis]						
	(2) = Module number [199] ! 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 = Diag	nosis type e	exists [Y	(ES, NO			
Example EPD3	Are there	any messa	ages in module	e 4 in cor	trol unit 0?		
	FI comm	•	00_BR_EPD3				
	Line	Column	Answer				
	1	1	YES				

Existing ProVi Types: EPT

MWMX and MWSX device groups

Designation	EPT Existing ProVi Types					
	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 type	s.			
	BR_EPT	1		(Single Read)		
-	The following table shows the general structure of the "EPT1" command. The number of lines depends on the number of ProVi typ existing.					
		Line 0n		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.					
				D_BR_EPT1		
	Line 1	Column 1	Answer 20			
		2	0			
	2	1	11			
		2	3			
	3	1	12			

3

1

Error Status: EST

Designation	EST	Error STate						
Explanation	Queries th	e error state of a	e error state of a variable.					
FI command	Query the	ery the frozen error state of a variable.						
	BR_EST1	l !(1)!(2)	(Single Read)					
	BC_EST1	l !(1)!(2)	(Cyclic Read)					
	(1) = Erro	r ID	[ASCII characters] (DWORD, decimal)					
	(2) = Variable name		[ASCII characters]					
	Note:	The separator "	!" is used in this command.					

Response Structure	The follo command		shows the ge	eneral structure of the "EXD1" FI	
		Line	1	Column 1	
Meaning of the Columns	1 = Error	state			
WinPcl - Example EST		value of W address 00		B_EXT24" in WinPcl program "Prog",	
	Exception The Win "Prog" as	Pcl variable	e "IB_EXT24" i	s declared in the WinPcl Program	
	FI comma	and	00_BR_EST1!58	392855!:Prog.IB_EXT24	
	Line	Column	Answer		
	1	1	1		
Execution Display: EX	D				
				MWMX and MWSX device groups	
Designation	EXD	EX ecutio	on 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) = SF0	C entity nan	ne [ASCII characters]	
	(2) = Ste	p or action	name [ASCII - characters]	
	(3) = Beł	naviour of m		1 = all modes, 2 = manual mode]	
	Note:	The sepa	rator "!" is used	in this command.	
Response Structure	The follo		shows the ge	eneral structure of the "EXD1" FI	
		Line	1	Column 1	
Meaning of the Columns	1 = Exec	ution	[1 = execu	can be executed, 0 = cannot be ited]	
Example EXD1	Query the execution of the step "open" for the chain "clamp" in control unit 0 for all modes.				
	FI comma	and	00_BR_EXD1!St	tation03A.Clamp!Open!1	
	Line	Column	Answer		
	1	1	1		
FI command	Query wh enabled.	ether the o	condition analysi	is (control image) of a step chain is	
	BR_EXD	02!(1)	(Single Read)	
	(1) = SF0	C entity nan	ne [ASCII characters]	
	Note:	The sepa	rator "!" is used	in this command.	



Response Structure	The follo command	-	shows the ge	eneral structure of the "EXD2" FI	
		Line	1	Column 1	
Meaning of the Columns	1 = Enab	led	[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			ation03A.Clamp	
	Line	Column	Answer		
	1	1	1		

Read Reference Name of a PLC Variable: MAR

			MWMX and MWSX device groups			
Designation	MAR Map Absolute PCL-Reference					
PLC Explanation	The abso	lute referen	ice name of a symbolic PLC variable is read out.			
FI command	Read the	absolute re	ference name of a PLC variable.			
	BR_MAR	_(1)	(Single Read)			
	(1) = Iden	tifier of the	PLC variable			
PLC – Example MAR		absolute re device add	eference name of the PLC variable with the identifier ress 00.			
	<u>Assumption</u> The PLC		h the identifier "abref" is of the type "INTEGER".			
	FI comma	and	00_BR_MAR_abref			
	Line	Column	Answer			
	1	1	%M100.0			
WinPlc Explanation		olute refere entity is rea	nce name of a symbolic WinPlc PLC variable with d out.			
FI command	Read the	absolute re	ference name of a WinPlc PLC variable.			
	BR_MAR	1_(1)	(Single Read)			
	(1) = Iden	tifier of the	PLC variable			
Win PLC - Example MAR1			reference name of the Win PLC variable with the "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 "Prog".					
	FI comma	and	00_BR_MAR1_:Prog.abref			
	Line	Column	Answer			
	1	1	%M100.0			
Reference to Literature	See chap	ter entitled	"Literature" [30].			



Device Data of the Module Configuration: MCD

MWMX and MWSX device groups

Designation	MCD Module	e C onfiguratio	on: Device Infor	mation			
Explanation	All device data for module configuration is read out from the "Moduldef.ini" file which is stored in the "[LW]:\Programme\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 Re	ead)				
	BC_MCD1	(Cyclic Re	ead)				
	BB_MCD1	(Break Cy	clic Read)				
	MWSX	device group	mmand refers . Therefore, an ommand line (s	y valid device	address can		
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 addres	S		[015]			
	2 = PLC-FB name	for the setup	diagnostics	[max. 9 ASC characters]			
	3 = PLC-FB name	for the warni	ng messages	[max. 9 AS0 characters]			
	4 = PLC-FB name for the start requirements [max. 9 ASCII characters]						
Example MCD1	Read all device da	ta of the mod	ule configuratio	n			
	Assumption: The following devices have been configured in the MWSX device group:						
	Device address 01 (ISP200-P)						
	Device address	03 (ISP200-I	२)				
	FI command	03_BR_MCD)1				
		1	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].



Module Data of the Module Configuration: MCM

MWMX and MWSX device groups

Designation	MCM Module Configuration: Modul 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]:\Programme\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			ta from the module con SX device group.	nfiguration wit	h respect to a	
	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.					
	l	_ine 1	Column 1		Column 4	
Value Range of the Columns	1 = Module	e number		[099]		
	2 = Module name [max. 28 ASCII characters]					
	2 = Module	e name		-		
			or general module errors	characters	s] SCII	
	3 = PLC-F	B name fo	or general module errors or module messages	characters [max. 9 A	5] SCII 5] SCII	
Example MCM1	3 = PLC-F 4 = PLC-F	B name fo B name fo		characters [max. 9 A characters [max. 9 A characters	5] SCII 5] SCII 5]	
Example MCM1	3 = PLC-F 4 = PLC-F Read the m <u>Assumption</u>	B name fo B name fo nodule dat <u>n:</u>	or module messages	characters [max. 9 A characters [max. 9 A characters	5] SCII 5] SCII 5]	
Example MCM1	3 = PLC-F 4 = PLC-F Read the m <u>Assumption</u> The followin	B name fo B name fo nodule dat <u>n:</u>	or module messages a of device 03 from the r	characters [max. 9 A characters [max. 9 A characters	5] SCII 5] SCII 5]	
Example MCM1	3 = PLC-F 4 = PLC-F Read the m <u>Assumption</u> The followin	B name fo B name fo nodule dat <u>n:</u> ng module number 5	or module messages a of device 03 from the r es have been defined:	characters [max. 9 A characters [max. 9 A characters	5] SCII 5] SCII 5]	
Example MCM1	3 = PLC-F 4 = PLC-F Read the m <u>Assumption</u> The followin • Module	B name fo B name fo nodule dat ng module number 5 number 7	or module messages a of device 03 from the r es have been defined:	characters [max. 9 A characters [max. 9 A characters	5] SCII 5] SCII 5]	
Example MCM1	3 = PLC-F 4 = PLC-F Read the m <u>Assumption</u> The followin • Module • Module	B name fo B name fo nodule dat ng module number 5 number 7	or module messages a of device 03 from the r es have been defined:	characters [max. 9 A characters [max. 9 A characters	5] SCII 5] SCII 5]	
Example MCM1	3 = PLC-F 4 = PLC-F Read the m <u>Assumption</u> The followin • Module • Module	B name fo B name fo nodule dat ng module number 5 number 7	or module messages a of device 03 from the r es have been defined: 03_BR_MCM1 Answer	characters [max. 9 A characters [max. 9 A characters	5] SCII 5] SCII 5]	
Example MCM1	3 = PLC-F 4 = PLC-F Read the m <u>Assumption</u> The followin • Module • Module	B name fo B name fo nodule dat ng module number 5 number 7 nd 0	or module messages a of device 03 from the r es have been defined: 03_BR_MCM1 Answer	characters [max. 9 A characters [max. 9 A characters module config	s] SCII SCII SCII s] uration:	

Reference to Literature

See chapter entitled "Literature" [36].



SFC Data of the Module Configuration: MCS

Designation	MCS Module Configuration: SFC 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]:\Programme\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	[099]				
Response Structure Value Range of the Column	chains for a device. E Indrastep step chains.	epends on the number of configured Indrastep step Each line contains a column for the name of the tep step chain [format W.X.Y.Z]				
	Format W.X.Y.Z	Value Range				
	W	Max. 9 ASCII characters				
	Х	Max. 9 ASCII characters ! OPTIONAL !				
	Y	Max. 9 ASCII characters ! OPTIONAL !				
	Z	Max. 9 ASCII characters ! OPTIONAL !				
Example MCS1	the module configuration	Indrastep step chain of module 5 from device 03 of on.				
	Assumption: The following Indraster	step chains have been defined:				
	 ISFB_1 					
	 FB_US.ISFB_3 					

- FB_US.ISFB_3.SW1
- FB_US.ISFB_3.SW1.ABBA

FI comma	and	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			



Downloading Message Texts: MFD

MWMX and MWSX device groups

Designation	MFD Message Files Download						
FI command	message	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 	n error mes	sages				
	 transm 	nission erro	r messag	jes, and/or			
	mecha	anism mess	sages.				
	Note:	This FI co	ommand	is an FI job!			
	BW_MFD	1		(Single Write)			
Response Structure				FI command consists of three lines, each with e elements is as follows:			
	Line 1 = Job ID			[0120] (refer to chapter entitled "FI Commands for the MPCX Device Group", IFJ).			
	Line 2 =	FI comman	ld	[string, in accordance to chapter entitled "Elements of the FI Command"]			
	Line 3 = FI job error code (see chapter entitled "Error Codes")			(see chapter entitled "Error Codes")			
Example MFD1	Load message texts into the device with device address 00.						
	FI comma	and	00_BW_	MFD1			
	Line	Column	Answer				
	1	1	01				
	2	1	00_BW_	MFD1			

Reading Machine Key Information : MKS

3

1

0

Designation	MKS	Machine Key Status			
Explanation	Current machine key information can be read for the selected device.				
FI command	Read machine key information for selected device.				
	BR_MK	S (Single Read)			
	BC_MK	S (Cyclic Read)			
	BB_MK	S (Break Cyclic	Read)		
Response Structure		owing table shows the gener nd "MKS".	al structure of the	response to the FI	
		Line 1	Column 1	Column 2	
Value Range/Meaning	1 =	Information of machine key	[4 byte in	HEX coding]	
of Columns	2 =	Information valid?	[0 = not v	alid, 1=valid]	

Example MKS

IKS Read the current machine key information for device 0.

FI command		00_BR_MKS		
Line	Column	Answer		
1	1	0000000		
	2	0		

Writing the GUI-SK Block: MKT

Designation	МКТ	Machine	Key Table		
Explanation	Writes the GUI-SK16 block in the PLC.				
FI command	BW_MK (1) = Lis var	I-SK16 bloc T1_(1) t of the 48 F iables for w II-SK16 bloc	PLC vriting the	 (Single Write) 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)		following suc	ccessful transmission.	
Value Range/Meaning of the Columns	Line 1 1 = Successfully completed			Column 1 (P_ACK)	
1. Example MKT1	1.Clear G	UI-SK16 bl		τ1	
	FI comm	and	00_BW_MK Value to be	written: \$EMPTY	
	Line	Column	Answer		
	1	1	(P_ACK)		
2. Example MKT1	Write GU	I-SK16 bloc	sk:		
	FI comm	and		T1 written: \$EMPTY PSVAR2,\$SPACE,	
	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)				
		t of the 48 F		nction is made between the following	
	var wri	iables for ting the GU 16 block.	cases: I- 1. Clea BW 2. Writ varia BW	ar GUI-SK16 block: _MKT2 \$EMPTY e the GUI-SK16 block with the 48 PLC ables, filling gaps with \$SPACE: _MKT1 SPSVAR1,SPSVAR2, ACE,\$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 = S	tatus report	t	current PL	st 1 PLC variable in the C program is NOT = ALL PLC variables could	
	2 = List of the NON-defined [= ALL PLC variables could be PLC variables in the current PLC program viriten, 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 comm	and	00_BW_MKT Value to be v SPSVAR1,SF			
	Line	Column	Answer			
	1	1	(P_ACK)			
Extended information	The varia following		vided into 3 gr	oups of 16 va	ariables each and have the	
	Variables	1 – 16:	Machine f	unction keys		
	Variables	17 – 32:	Status pre	essed		
	Variables 33 -48: Status shining					
	telegram will contain variables need not be			nly these 8 P defined in the g. of M keys,	8 M keys are used, the LC variables. The other 40 transmission parameter. are left unused, they must	
		be filled u	ip with '\$SPA	CE' up to the	next variable.	

Read System Messages: MSG

Designation	MSG	MeSsaGe	
Explanation	Reading of	system messages	
FI command	Message CC_MSG (1) = SYS	_ (1) -Message number	(Cyclic Read)
	Note:	Exists only as a cy	clic command
Response Structure	The respor data.	nse of the FI comm	and 'MSG' consists of the system message
Example MSG	00_CC_MS	664 =	MSG_SYSERRGEN)



			1				
	FI comma		_	_MSG_64/3			
	Line	Column	Answe	r			
	1	1	00				
Restriction	The follow	ving system	n messa	ges:			
	SYS Mes	sage		SYS Message number			
	MSG_PC	LUPDBEG		52			
	MSG_PA	RUPDBEG		24			
	MSG_FW	AUPDBEG	ì	82			
	These commands cannot be used with the following programs:						
	Indramat OPC server						
	Indramat DDE server						
Reading the Firmware	Identifi	cation:	мтс				
5				MWMX and MWSX device groups			
Designation	МТС	MT-CNC	Slot So	oftware Version			
FI command	This com	mand is us	ed to rea	ad the firmware identification from the various			
	control components (slot numbers).						
	Note:			s FI command is executed, the internal FI			
				interlocks (fast timeout monitoring, offline re switched off.			
FI command	BR_MTC	C_(1)		(Single Read)			
	(1) = Sloc	t number		[1=CNC, 2=SIO, 3=PLC, 4=APR1			
				5=APR2, 6=APR3, 7=APR4]			
Response Structure				ne general structure of the response to the FI			
	command	3 "MTC". A	line of 1	column is output.			
		Line 1		Column 1			
Value Range/Meaning	1 = Firm	ware identif	ication s	string [max. 16 ASCII characters]			
of Columns							
Example MTC				tion of slot number 1 (CPU) of device 00.			
	FI comma		00_BR	_MTC_1			
	Line	Column	Answe				
	1	1	CPU01	/0004-20V00			
FI command	This com	mand is us	ed to rea	ad the firmware identification from the various			
		omponents	(slot nur	nbers).			
	CR_MTC			(Single Read)			
	(1) = Slo	t number		[1=CNC, 2=SIO, 3=PLC, 4=APR1 5=APR2, 6=APR3, 7=APR4]			
				$\mathbf{U} = \mathbf{A} \mathbf{U} \mathbf{U} \mathbf{U}, \mathbf{U} = \mathbf{A} \mathbf{U} \mathbf{U} \mathbf{U} \mathbf{U} \mathbf{U} \mathbf{U} \mathbf{U} U$			
Response Structure				ne general structure of the response to the FI			
	command		iine of 1	column is output.			
		Line 1		Column 1			
Value Range/Meaning	1 = Firm	ware Identif	ication s	string [max. 16 ASCII characters]			
of Columns							



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

FI command		00_CR_MTC_1				
Line	Column	Answer				
1	1	CPU01/0004-20V00				

ProVi Diagnosis Data: PDD

Designation	PDD	Provi Dia	agnosis D ata						
Explanation	Data for ProVi criteria analysis is output.								
FI command	Output of files to indicate the detail BR_PDD1_(1)_(2){_(3)} (1) = Message ID (2) = Message type			il in the editor. (Single Read) [ASCII characters] [1 = error, 2 = messages, 10 = warnings, 11 = start requirements, 12 = setup diagnosis]					
	. ,	dule numbe		[199] ! only for message type 1 -2!					
Response Structure	The following table shows the general structure of the PDD1 command.							FI	
	Line 1		Column 1	nn 1		Col	umn 5		
Meaning of the Columns	3 = Error 4 = POE	il morphem	e	[ASCII ch [ASCII ch [ASCII ch	[ASCII characters] [ASCII characters] (DWORD, decimal) [ASCII characters] (DWORD, decimal) [ASCII characters] [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 1	Column 1	Answer	2					
		2	STATION_1_2 98243823						
	3		34985304						
		4 Station2.Modu		lule3	ıle3				
		5	43493454						
FI command	sses to display a detail. (Single Read) [ASCII characters] [1 = error, 2 = messages, 10 = warnings, 11 = start requirements,								
	(3) = Module number				12 = setup diagnosis] [199] ! only for message type 1 -2!				
Response Structure		wing table			general structure of the PDD2 FI				

	L	ine 1-n		Co	lumn 1	Column 2	
Maaning of the Oslam							
Meaning of the Columns	1 = Varia2 = I/O a	able morphe ddress	eme	-	ASCII characters] (DWORD, decimal) ASCII characters]		
Example PDD2		the I/O ac in control u			ProVi error	with ID 43923028 from	
	Three var	riables have	e an l	/O addres	s.		
	FI comm	FI command 00_BR_PDD2_43923028_1_1					
	Line	Column	Ans	wer			
	1	1	9824	43823			
		2	%I3.	.2.0			
	2	1	4092	23423			
		2	%Q2	23.21.7			
	3	1	3498	35304			
		2	%110	00.3.5			
FI command	Determine the multilingual comments for displaying a detail.						
	BR_PDD3_(1)_(2){_(3)} (Single Read)						
	(1) = Message ID [ASCII characters]					ters]	
	(2) = Message type [1 = error, 2 = messages, 10 = warnings, 11 = start requirements,					s, juirements,	
	(2) Ma	12 = setup diagnosis]					
	(3) = 100	dule numbe	1		[199] ! Only	for message type 1 -2!	
Response Structure	The following table shows the general structure of the PDD3 FI command.						
	command	J.					
		l. .ine 1-n		Co	lumn 1	Column 2	
Meaning of the Columns	L 1 = Com	-	heme	e [A		s] (DWORD, decimal)	
Meaning of the Columns Example PDD3	L 1 = Com 2 = New Query of	ine 1-n ment morpl comment	ents f	e [A [A	SCII character	s] (DWORD, decimal)	
-	L 1 = Com 2 = New Query of from mod	ine 1-n ment morpl comment the comme	ents f	e [A [A or indicati unit 0.	SCII character SCII character	s] (DWORD, decimal) s]	
-	L 1 = Com 2 = New Query of from mod	ine 1-n ment morpl comment the comme lule 3 in cor ments are r	ents f ntrol u replac	e [A [A or indicati unit 0. ced by and	SCII character SCII character	s] (DWORD, decimal) s]	
-	L 1 = Com 2 = New Query of from mod Two com	ine 1-n ment morpl comment the comme lule 3 in cor ments are r	ents f ntrol u replac	e [A [A or indicati unit 0. ced by and BR_PDD3	SCII character SCII character on of a ProVi other text.	s] (DWORD, decimal) s]	
-	L 1 = Com 2 = New Query of from mod Two com FI comm	ine 1-n ment morpl comment the comme lule 3 in cor ments are r and	ents f ntrol u replac 00_l Ans	e [A [A or indicati unit 0. ced by and BR_PDD3	SCII character SCII character on of a ProVi other text.	s] (DWORD, decimal) s]	
-	L 1 = Com 2 = New Query of from mod Two com FI comm Line	ine 1-n ment morpl comment the comme lule 3 in cor ments are r and Column	ents f ntrol u eplac 00_l Ans 9824	e [A [A or indicati unit 0. ced by and BR_PDD3_ wer	SCII character SCII character on of a ProVi other text.	s] (DWORD, decimal) s]	
-	L 1 = Com 2 = New Query of from mod Two com FI comm Line	ine 1-n ment morpl comment the comme lule 3 in cor ments are r and Column	ents f ntrol u replac 00_1 Ans 9824 Clan	e [A [A or indicati unit 0. ced by and BR_PDD3_ wer 43823	SCII character SCII character on of a ProVi other text.	s] (DWORD, decimal) s]	
-	Line	ine 1-n ment morpl comment the comme lule 3 in cor ments are r and Column 1 2	ents f trol u replace 00_1 Ans 9824 Clan 4092	e [A [A or indication unit 0. ced by and BR_PDD3 wer 43823 np open	SCII character SCII character on of a ProVi other text.	s] (DWORD, decimal) s]	
-	Line 2 2 2 2 2 2 2 2	ine 1-n ment morpl comment the comme lule 3 in cor ments are r and Column 1 2 1	ents f trol u eplac 00_I Ans 982 ² Clan 4092 Clan	e [A [A [A or indication unit 0. ced by and BR_PDD3 wer 43823 np open 23423 np closed	SCII character SCII character on of a ProVi other text. _43923028_1_1	s] (DWORD, decimal) s]	
Example PDD3	L 1 = Com 2 = New Query of from mod Two com FI comm Line 1 2 Query of	ine 1-n ment morpl comment the comme lule 3 in cor ments are r and Column 1 2 1 2	ents f trol u eplac 00_1 Ans 9824 Clan 4092 Clan	e [A [A [A or indication unit 0. ced by and BR_PDD3 wer 43823 np open 23423 np closed	SCII character SCII character on of a ProVi other text. _43923028_1_1	s] (DWORD, decimal) s] error with ID 43923028	
Example PDD3	Line 1 = Com 2 = New Query of from mod Two com FI comm Line 1 2 Query of BR_PDE	ine 1-n ment morpl comment the comme lule 3 in cor ments are r and Column 1 2 1 2 the status c	ents f trol u replac 00_1 Ans 9824 Clan 4092 Clan of a co _(3)}	e [A [A [A or indication unit 0. ced by and BR_PDD3 wer 43823 np open 23423 np closed	SCII character SCII character on of a ProVi other text. _43923028_1_1	s] (DWORD, decimal) s] error with ID 43923028	
Example PDD3	L 1 = Com 2 = New Query of from mod Two com FI comm Line 1 2 Query of BR_PDE (1) = Me	ine 1-n ment morpl comment the comme lule 3 in cor ments are r and Column 1 2 1 2 the status co 04_(1)_(2){	ents f trol u replac 00_1 Ans 9824 Clan 4092 Clan of a co _(3)}	e [A [A [A or indication unit 0. ced by and BR_PDD3 wer 43823 np open 23423 np closed	SCII character SCII character on of a ProVi other text. 43923028_1_1 ssage (Single Read [ASCII charac [1 = error, 2 = 10 = warning 11 = start rec	s] (DWORD, decimal) s] error with ID 43923028	
Example PDD3	L 1 = Com 2 = New Query of from mod Two com FI comm Line 1 2 Query of BR_PDDE (1) = Me (2) = Me	ine 1-n ment morph comment the comment lule 3 in cor ments are r and Column 1 2 1 2 the status c 04_(1)_(2){ ssage number	ents f trol u replac 00_1 Ans 9824 Clan 4092 Clan of a co _(3)} ber	e [A [A [A or indication unit 0. ced by and BR_PDD3 wer 43823 np open 23423 np closed	SCII characters SCII characters on of a ProVi other text. _43923028_1_1 	s] (DWORD, decimal) s] error with ID 43923028	



Response Structure	The follo		e sho	ws the	genera	al structure o	f the	PDD4	FI
	Line	1-n	Co	umn 1		Colum	າn 2		
Meaning of the Columns		1 = Message is present[YES, NO]2 = Criteria analysis exists[YES, NO]							
Example PDD4	Query of control 0.	Query of the status of a ProVi error, number 1001 from module 3 in control 0.						in	
	This mes analysis.	his message is not present at the moment, and there is a criterianalysis.						ria	
	FI comma	and	00_B	R_PDD4	_1001_1	_1			
	Line	Column	Ansv	ver					
	1	1	NO						
		2	YES						
FI command	Determina	ation of the	e Mess	ageID o	f a certa	in message			
) 5!(1)!(2)!(-		e Read)			
	(1) = PO	U entity na	ame		[ASCI	characters]			
	(2) = Nw	ID			[ASCI	characters]			
	(3) = Me	ssage num	nber		[ASCI	characters]			
	(4) = Message type [1 = error, 2 = messages, 10 = warnings, 11 = start requirements,								
		12 = setup diagnosis]							
	(5) = Mo	dule numb	er		[199] ! only for mes	sage ty	/pe 1 -2	!
	Note: The separator "!" is used in this command.					<u> </u>			
Response Structure	The follo	-	e sho	ws the	genera	al structure o	f the	PDD5	FI
		Line 1-n		Colu	mn 1		Co	lumn 3	
Meaning of the Columns	1 = Mess	sage ID			[ASCII decima	characters	;] (E	OWORI	D,
	2 = Mess	sage is pro	esent		[YES, N				
		ria analysi		6	[YES, N	-			
Example PDD5	Determination of the MessageID of a ProVi error, number 1001 from module 25.40 mm control 0. <u>Assumption:</u> This message is not present at the moment, and there is a criteria analysis.								
	FI comm	and	00_BR	_PDD5!\$	Station2.	Modul3!434934	54!1001	!1!1	
	Line	Column	Answe	r					
	1	1	240872	2342					
		2	NO						
		3	YES						
	L	1 I							



Generating Physical Directory Names: PHD

Explanation Generates physical directory names according to the BDI data written. Note: This is based on BDI philosophy. FI command Generate physical directory names. BR_PHD1_(1)_(2)_(3)_(4)_(5)_(6) (Single Write) (1) = Project ID [-1= PROJECT_NEUTRAL -2= PROJECT_DEFAULT] (2) = Section ID [0 = SECT_DETAULT] (2) = Section ID [0 = SECT_BASIC_DATA 3=SECT_OWD_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 F command "PHD1". Value Range/Meaning of Columns 1 = Physical directory name [Complete physical directory name in accordance with the BDI data written]	Designation	PHD	PH ysical	Directory		0.1	
FI command Generate physical directory names. 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_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)] The following table shows the general structure of the response to the F command "PHD1". Value Range/Meaning of Columns 1 = Physical directory name [complete physical directory name in accordance with	Explanation	Generate	s physical o	directory names	according to the BDI data	written.	
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_OEM_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 The following table shows the general structure of the response to the F command "PHD1". Value Range/Meaning of Columns 1 = Physical directory name [complete physical directory name in accordance with		Note: Thi	s is based	on BDI philosop	ny.		
(1) = Project ID [-1= PROJECT_NEUTRAL -2= PROJECT_DEFAULT] (2) = Section ID [0= SECT_NEUTRAL 1= SECT_BIN 2= SECT_OEM_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 F command "PHD1". Value Range/Meaning of Columns 1 = Physical directory name [complete physical directory name in accordance with	FI command	Generate	physical di	rectory names.			
-2= PROJECT_DEFAULT] (2) = Section ID [0= SECT_NEUTRAL 1= SECT_BIN 2= SECT_BASIC_DATA 3= SECT_CUSTOM_DATA 3= SECT_CUSTOM_DATA 4=SECT_CUSTOM_DATA] (3) = Device address (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)] The following table shows the general structure of the response to the F command "PHD1". Value Range/Meaning of Columns 1 = Physical directory name [complete physical directory name in accordance with		BR_PHD	01_(1)_(2)_	(3)_(4)_(5)_(6)	(Single Write)		
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)] (6) = Language ID [possible write values see BDI documentation (WINNT.H)] (6) = Language ID [possible write values see BDI documentation (WINNT.H)] (6) = Language ID [possible write values see BDI documentation (WINNT.H)] Value Range/Meaning 1 = Physical directory name [complete physical directory name in accordance with		(1) = Pro	ject ID				
(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 F command "PHD1". Line 1 Column 1 Value Range/Meaning of Columns 1 = Physical directory name [complete physical directory name in accordance with		(2) = Sec	ction ID		1= SECT_BIN 2= SECT_BASIC_DATA 3=SECT_OEM_DATA 4=SECT_CUSTOM_DA	ТА	
(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 F command "PHD1". Value Range/Meaning of Columns 1 = Physical directory name 1 = Physical directory name		(3) = Dev	vice addres	s	otherwise the required	AL	
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 Ficormand "PHD1". Line 1 Column 1 Value Range/Meaning of Columns 1 = Physical directory name (complete physical directory name in accordance with		otherwise the required					
BDI documentation (WINNT.H)] Response Structure The following table shows the general structure of the response to the F command "PHD1". Line 1 Column 1 Value Range/Meaning of Columns 1 = Physical directory name [complete physical directory name in accordance with		(5) =Data type ID			BDI documentation		
command "PHD1". Line 1 Column 1 Value Range/Meaning of Columns 1 = Physical directory name [complete physical directory name in accordance with		(6) = Lar	iguage ID				
Value Range/Meaning of Columns 1 = Physical directory name directory name [complete physical directory name in accordance with	Response Structure			shows the gener	al structure of the respons	se to the FI	
of Columns directory name in accordance with			Lir	ne 1	Column 1		
	u u	1 = F	Physical dire	directory name in accordance w			
Example PHD1 Requesting the physical directory name for:	Example PHD1	Requestir	ng the phys	ical directory na	me for:		
PROJECT_NEUTRAL SECT_BIN	·	PROJECT_NEUTRAL					
DEVADDR_NEUTRAL PROCESS_NEUTRAL DATATYPE_NEUTRAL LANG_NEUTRAL		DEVADDR_NEUTRAL PROCESS_NEUTRAL DATATYPE_NEUTRAL					
FI command XX_BR_PHD11_011_0_0		FI comma	and	XX_BR_PHD1_	·1_011_0_0		
Line Column Answer		Line	Column	Answer			
1 1 D:\Programme\Indramat\Mtgui\Bin		1	1	D:\Programme\I	ndramat\Mtgui\Bin		



Issuing SYS Messages Specific to the PCL: PSM

Designation	PSM	PCL Sys Mes	ssage				
Explanation		the most imp nming interface –					PCL
		propriate device ac	ldress is	passed as	the write value		
	It allows	the following SY	S messa	iges to be	initiated:		
	 Start 	t of PCL download	d,				
		of PCL download,					
		of PLC online ed	it,				
		PLC online edit,					
			laration change, and				
	• end	of PCL declaration	n change	Э.			
FI command	Issue th	e most important	PCL SY	S messag	es.		
	BW_P	SM1_(1)	(Single	e Write)			
	· · /	Requested SYS		rt of PCL o d of PCL d			
	I	nessage		rt of PCL of			
				d of PCL o			
					leclaration chai eclaration chan	•	
	Value t	o be written					
	device	address					
Response Structure	The foll	owing table show	s the ae	neral stru	cture of the res	ponse to	the FI
		nd "PSM1".	- -				
		Line 1	Col	umn 1		Colum	n 8
Value Range/Meaning of Columns	1 =	Status report		[READY=SYS message has been correctly acknowledged by the WIN3 applications] [ERROR=SYS message has NOT been acknowledged by a WIN32 application within the pre-set time]			
	2 =	Task name (LogInIf name)		[Task na SYS mes	me that has trig ssage]	gered the	•
	3 =	SYS message n	umber	[contains number]	the issued SY	S messag	е
	4 =	Acknowledgement time [contains the pre-set acknowledgement time]					
	5 =	Reference inforr	Reference information [contains, where applicable, the additional information transferred a write value]				as a
	6 =	Length of referent information	nce		NO reference i transferred]	informatio	n
	7 =	Where applicabl channel of the F has NOT acknow	FI that completed in time or the LOG channel				

- 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	

Edit PROVI Message Files: PVA

			MWMX and MWSX device groups			
Designation	PVA	P RO V I-Me	ssages Access			
Explanation	it is possit	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	l	(Single Write)			
	Note:	This comma	and is an FI job command.			
Value to be written	No write	alue exists/	PROVI message files according to the current PLC project.			
	Write val	ue 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:		o be written is passed to the "acValue" parameter I value in the "DataTransfer" routine.			
Response Structure			'BW_PVA1" FI command consists of three lines, The meaning of the elements is as follows:			
	 Line 1 = Job ID [0120] (see Chapter "FI Commands for the MPCX Device Group", IFJ). 					
		= FI comma g, in accordar	nd nce with Chapter "Elements of the FI Command"]			
	• Line 3	= FI Job Erro	or Code (see Chapter "Error Codes")			



	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_PV	'A1	(Sing	gle Read)		
Response Structure	The following table shows the general construction of the answer of th command BR_PVA1. For each available PROVI message file, 1 line 10 columns each is created.						
		Line 1n	Colu	ımn 1		Column 10	
Value Range/Meaning	1 =	PROVI diagno	osis type	[120]			
of Columns	2 =	PROVI diagno designation		[The foll returned	ndition, Error, M		
	3 =	Module numb	er	[199]			
		PROVI diagno and module n			diagnosis type: , see write value /A2]		
	5 =	Complete nar PROVI messa		[max. 20	00 ASCII charac	cters]	
		Memory requi PROVI messa control		[figure in ASCII format]			
		Complete name of the PROVI index file		[max. 200 ASCII characters]			
	8 = Memory required for [figure in PROVI index files in the control			e in ASCII format]			
		Total memory (text+index) required in the control		[figure in ASCII format]			
		Total memory PROVI files (t required in the	ext+index)	[figure ir	n ASCII format]		

Example PVA1 No write value is passed, i.e. the PROVI message files are generated according to the current PLC project.

	returned.						
	FI comm	and	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:\Programme\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	device. T	te command transmits PROVI message files into the select Through the write value, it is possible to chose whether AL PROVI messages selected via the write value are to red.					
FI command	BW_PVA	2	(Single Write)				
	Note:	This com	imand is an FI job command.				
/alue to be written	No write	value exist	s All PROVI message files are transmitted into the selected device				
	Write va	lue 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:		e to be written is passed to the "acValue" parameter CII value in the "DataTransfer" routine.				

Example PVA1 The most significant information of 2 available PROVI message files are returned.



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

Formatted Input / Output of PLC Variables: PVF

MWMX and MWSX device groups

Designation	PVF	PLC Variable Formatte	ed			
Explanation	Formatted	Formatted reading and writing of PLC variables, arrays and structures.				
FI command	Read PLC	variables.				
	CR_PVF_	_(1)	(Single Read)			
	CC_PVF_	<u>(</u> 1)	(Cyclic Read)			
	CB_PVF_	_(1)	(Break Cyclic Read)			
	(1) = Iden	tifier 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 1n:	Column 1			
	n = numbe	er 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	read. The	following table indicates	depends on the data type of the variable the range in which the results string is 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	[-128127]	char
INT	[-3276832767]	short
DINT	[21474836482147483647]	long
USINT	[0255]	unsigned char
UINT	[065535]	unsigned short
UDINT	[04294967295]	unsigned long
BYTE	[0x000xFF]	unsigned char
WORD	[0x00000xFFFF]	unsigned short
DWORD;	[0x00000000xFFFFFFF]	unsigned long
TIME	[04294967295]	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</string></string>	Char[xx+1]] +1 i.e. room for the zero byte
REAL	[-3.402823567E+383.402823567E+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. –128127	1
INT	[8000 н (-32768)7FFF н (32767)]	2
DINT	[80000000 _н (-2147483648) 7FFFFFF _н (2147483647)]	4
USINT	[00 н (0)FF н (255)]	1
UINT	[00 _н (0)FFFF _н (65535)]	2
UDINT	[04294967295]	4
BYTE	[0x000xFF]	1
WORD	[0x00000xFFFF]	2
DWORD;	[0x00000000xFFFFFFF]	4
TIME	[04294967295]	4
CHAR	[\$00\$20,!~,\$7F\$FF]	1



	Data Typ	e Value	Range	Length (bytes)
	STRING	<strin where maxin</strin 	<string> whereby <string> string is a character string with a maximum of as many characters as are declared for the string in the PLC</string></string>	
	REAL	[-3.40	2823567E+383.402823567E+38]	4
	Note:		rray and structure elements are joine ny spaces between (1-byte alignment).	d together
PLC - Example 1 PVF	Read the address 0		e PLC variable "STK_TXT" in ASCII format	from device
	<u>Assumpti</u> The "STK		able is declared as STRING in the PLC pro	ogram.
	FI comma	and	00_CR_PVF_STK_TXT/1	
	Line	Column	Answer	
	1	1	Repeat counter	
WinPcl - Example 1 PVF			/inPcl variable "STK_TXT" in ASCII forma evice address 00.	t in WinPcl
	Assumption The WinF STRING.		"STK_TXT" is declared in WinPcl program	n "Prog" as
	FI comma	and	00_CR_PVF_:Prog.STK_TXT/1	
			00_0K_1 VI1109.01K_1X1/1	
	Line	Column	Answer	
	Line 1			
PLC - Example 2 PVF	1	Column 1 value of th	Answer	from device
PLC - Example 2 PVF	1 Read the address 0 <u>Assumpti</u>	Column 1 value of th 00. on:	Answer Repeat counter	
PLC - Example 2 PVF	1 Read the address 0 <u>Assumpti</u> The "BEG	Column 1 value of th 00. on: 6_END" var	Answer Repeat counter e PLC array "BEG_END" in ANSI format f	
PLC - Example 2 PVF	1 Read the address 0 <u>Assumpti</u> The "BEG program.	Column 1 value of th 00. on: 6_END" var	Answer Repeat counter e PLC array "BEG_END" in ANSI format f iable is declared as BYTE with 2 elements	
PLC - Example 2 PVF	1 Read the address 0 <u>Assumpti</u> The "BEO program. FI comm	Column 1 value of th 00. <u>on:</u> 6_END" var	Answer Repeat counter e PLC array "BEG_END" in ANSI format f iable is declared as BYTE with 2 elements 00_CR_PVF_BEG_END/3	
PLC - Example 2 PVF	1 Read the address 0 <u>Assumpti</u> The "BEG program. FI comma Line	Column 1 value of th 0. <u>on:</u> B_END" var and Column	Answer Repeat counter e PLC array "BEG_END" in ANSI format f iable is declared as BYTE with 2 elements 00_CR_PVF_BEG_END/3 Answer	
PLC - Example 2 PVF WinPcl - Example 2 PVF	1Read the address 0Assumpti The "BEC program.FI commaLine12Read the program.	Column 1 value of th 00. on: S_END" var and Column 1 value of Var	Answer Repeat counter e PLC array "BEG_END" in ANSI format f iable is declared as BYTE with 2 elements 00_CR_PVF_BEG_END/3 Answer 0x00	in the PLC
	1 Read the address O Assumpting The "BEO program. FI command 1 2 Read the program. Assumpting 1 2 Read the program. The WinF	Column 1 value of th 0. on: b_END" var and Column 1 value of \ 'Prog" at de on:	Answer Repeat counter e PLC array "BEG_END" in ANSI format f iable is declared as BYTE with 2 elements 00_CR_PVF_BEG_END/3 Answer 0x00 0x1F WinPcl array "BEG_END" in ANSI formated evice address 00. "BEG_END" is declared in WinPcl program	t in WinPcl
	1 Read the address O Assumpting The "BEO program. FI command 1 2 Read the program. Assumpting 1 2 Read the program. The WinF	Column 1 value of th 00. on: S_END" var and Column 1 value of \ value of \ Prog" at demonstrations on: Pcl variable h two elemonstrations	Answer Repeat counter e PLC array "BEG_END" in ANSI format f iable is declared as BYTE with 2 elements 00_CR_PVF_BEG_END/3 Answer 0x00 0x1F WinPcl array "BEG_END" in ANSI formated evice address 00. "BEG_END" is declared in WinPcl program	t in WinPcl
	1Read the address (1)Assumpti The "BEC program.FI comma Line12Read the program ' Assumpti The WinF BYTE with	Column 1 value of th 00. on: S_END" var and Column 1 value of \ value of \ Prog" at demonstrations on: Pcl variable h two elemonstrations	Answer Repeat counter e PLC array "BEG_END" in ANSI format f iable is declared as BYTE with 2 elements 00_CR_PVF_BEG_END/3 Answer 0x00 0x1F WinPcl array "BEG_END" in ANSI formate formate address 00. "BEG_END" is declared in WinPcl programents.	t in WinPcl
	1 Read the address O Assumpting The "BEO program. FI command Line 1 2 Read the program ' Assumpting The WinF BYTE with FI command	Column 1 value of th 0. on: D_END" var and Column 1 value of V 'Prog" at de on: Pcl variable h two eleme and	Answer Repeat counter e PLC array "BEG_END" in ANSI format f iable is declared as BYTE with 2 elements 00_CR_PVF_BEG_END/3 Answer 0x00 0x1F WinPcI array "BEG_END" in ANSI formate evice address 00. "BEG_END" is declared in WinPcI programents. 00_CR_PVF_:Prog.BEG_END/3	t in WinPcl

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
Т3	STRING[16]
T4	TIME

END

FI comma	and	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
- Т3 STRING[16]
- Τ4 TIME

END

FI comma	and	00_CR_PVF_:Prog.MSTRCT/1
Line	Column	Answer
1	1	0
2	1	Α
3	1	ROBOT AXIS X
4	1	2000

FI command Write PLC variable.

	CW_PVF_(1) (1) = Identifier of the PLC variable		(Single Write) [acc. to declaration part of the PLC]
Value to be written	Value of data element		[see value ranges]
	Note:		s passed to the "acValue" parameter itine. The data code of the value is ValType".
Response Structure	One line is output with a column for the FI command has been executed s		
	(P_ACK) =	= Positive ACKnowledge	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 / ASCI value ranges during read access. ANSI umlauts are thereby co into ASCII umlauts. Only ASCII umlauts are stored in the control of 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 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 PLC - Example 4 PVF The value ranges agree with the binary result-value range during read access. For deviations to this, please refer to the following note:

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 comma	and	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.



	E 1 1				
	FI comma		00_CR_PVF_BEG_END/3		
	Line	Column	Answer		
	1	1	(P_ACK)		
	Value to b	e written:			
	Value of c	lata elemer	nt 0x20 0x3f		
	Data code	9	/3		
WinPcI - Example 5 PVF	Write into the WinPcl byte array "BEG_END" in WinPcl program "Prog" a device address 00. The value is passed in ANSI format. <u>Assumption:</u> The WinPcl variable "BEG_END" is declared in WinPcl program "Prog" as BYTE with two elements.				
	FI comma	and	00_CW_PVF_:Prog.BEG_END/3		
	Line	Column	Answer		
	1	1	(P_ACK)		
	Value to t	o writton:	·		
		lata elemer	nt 0x20 0x3f		
	Data code		/3		
PLC - Example 6 PVF			lement T3 of the PLC structure "MSTRCT" at device ng "COUNTER" is transferred in binary format.		
	Assumpting The "MST follows:		able is declared as a structure in the PLC program as		
	TYP STR				
	Т		. —		
	T2 CHAR T3 STRING[16]				
	1	० ०१८	T4 TIME		
	Т				
	END	4 TIME	E · · ·		
	T END FI comma	4 TIME	00_CW_PVF_MSTRCT.T3/2		
	T END FI comma Line	4 TIME and Column	00_CW_PVF_MSTRCT.T3/2 Answer		
	T END FI comma	4 TIME	00_CW_PVF_MSTRCT.T3/2		
	T END FI comma Line 1 Value to b	4 TIME and Column 1 be written:	00_CW_PVF_MSTRCT.T3/2 Answer (P_ACK)		
	T END FI comma Line 1 Value to b	4 TIME and Column 1	00_CW_PVF_MSTRCT.T3/2 Answer (P_ACK)		
	T END FI comma Line 1 Value to b	4 TIME and Column 1 be written: data eleme	00_CW_PVF_MSTRCT.T3/2 Answer (P_ACK) ent Binary sequence: 43 4F 55 4E 54 45		
WinPcl - Example 6 PVF	T END FI comma Line 1 Value to b Value to b Value of Data cod	4 TIME and Column 1 be written: data eleme le value of	00_CW_PVF_MSTRCT.T3/2 Answer (P_ACK) ent Binary sequence: 43 4F 55 4E 54 45 52 00		
WinPcl - Example 6 PVF	T END Fl comma Line 1 Value to b Value to b Value of Data cod Write the device ad Assumptin The Winl program	4 TIME and Column 1 be written: data eleme dress 00. T on: Pcl variable 'Prog" as fo	00_CW_PVF_MSTRCT.T3/2 Answer (P_ACK) ent Binary sequence: 43 4F 55 4E 54 45 52 00 /2 element T3 of the WinPcl structure "MSTRCT" at The string "COUNTER" is transferred in binary format. e "MSTRCT" is declared as a structure in WinPcl		
WinPcl - Example 6 PVF	T END Fl comma Line 1 Value to b Value to b Value of Data cod Write the device ad Assumptin The Wind program ''	4 TIME and Column 1 be written: data eleme de value of dress 00. T <u>on:</u> Pcl variable 'Prog" as fo UCT	00_CW_PVF_MSTRCT.T3/2 Answer (P_ACK) ent Binary sequence: 43 4F 55 4E 54 45 52 00 /2 element T3 of the WinPcl structure "MSTRCT" at The string "COUNTER" is transferred in binary format. e "MSTRCT" is declared as a structure in WinPcl billows:		
WinPcl - Example 6 PVF	T END FI comma Line 1 Value to b Value to b Value of Data cod Write the device ad Assumptio The Wind program ' TYP STR T	4 TIME and Column 1 be written: data eleme data eleme data eleme e value of dress 00. T on: Pcl variable 'Prog" as fo UCT 1 BOC 2 CHA	00_CW_PVF_MSTRCT.T3/2 Answer (P_ACK) ent Binary sequence: 43 4F 55 4E 54 45 52 00 /2 v/2 element T3 of the WinPcl structure "MSTRCT" at The string "COUNTER" is transferred in binary format. e "MSTRCT" is declared as a structure in WinPcl billows: DL		
WinPcI - Example 6 PVF	T END Fl comma Line 1 Value to b Value to b Value of Data cod Write the device ad Assumptio The Wind program ' TYP STR T	4 TIME and Column 1 be written: data eleme data eleme e value of dress 00. T on: Pcl variable 'Prog" as fo UCT 1 BOC 2 CHA 3 STR	00_CW_PVF_MSTRCT.T3/2 Answer (P_ACK) ent Binary sequence: 43 4F 55 4E 54 45 52 00 /2 element T3 of the WinPcl structure "MSTRCT" at The string "COUNTER" is transferred in binary format. e "MSTRCT" is declared as a structure in WinPcl billows: DL ING[16]		



	FI command		00_CW_PVF_:Prog.MSTRCT.T3/2	
	Line	Column	Answer	
	1	1	(P_ACK)	
	Value to b	e written:		
	Value of o	data eleme	nt Binary sequence: 43 4F 55 4E 54 45 52 00	
	Data code	e	/2	
PLC - Example 7 PVF	Write the value of the PLC structure "MSTRCT" from the "matrix" previously stored in the C program at device address of			
	Assumption The "MST follows:		able is declared as a structure in the PLC program as	
	TYP STR	UCT		
	T		· —	
	= =	2 CHA 3 STR	ING[16]	
	T			
	END			
	To excha be used:	nge binary	data in a C program, the following "C" data type can	
	#pragma	pack(1)	//Write all elements	
			//without spaces next to each other.	
	typeder	struct		
	{	ngignod	abar T1.	
		nsigned (har	T2;	
		har	T3[17]; //Space for zero byte	
	u	nsigned		
	} Tymst	rct;	// Declare structure	
	Tymstrc	t 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.			
		data eleme		
	Data cod	е	/2	
WinPcl - Example 7 PVF			the WinPcl structure "MSTRCT" from the structure tored in the C program at device address 00.	
			e "MSTRCT" is declared as a structure in WinPcl ollows:	
	TYP STR			
	T T			
		-		
	T2 CHAR T3 STRING[16] T4 TIME END			



MWMX and MWSX device groups

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 comma	and	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 /2

Data code

ProVi Messages: PVM

PVM Designation ProVi Messages ProVi messages are output. These messages are assigned to a particular Explanation message type or module. **FI** command Output all ProVi messages. BR_PVM1_(1){_(2)} (Single Read) BC_PVM1_(1){_(2)} (Cyclic Read) [1 = error, 2 = messages, 10 = warnings, (1) = Message type 11 = start requirements, 12 = setup diagnosis] [1...99] ! only for message type 1 -2! (2) = Module number Output first ProVi messages. BR_PVM2_(1){_(2)} (Single Read) BC_PVM2_(1){_(2)} (Cyclic Read) [1 = error, 2 = messages, 10 = warnings,(1) = Message type 11 = start requirements, 12 = setup diagnosis] [1...99] ! only for message type 1 -2! (2) = Module number **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. Column 1 Column 6 Line 1...n ... 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)} (1) = Message ID

(Single Read) [ASCII characters]

(2) = Message type [1 = error, 2 = messages,



Deemense Structure	10 = warnings, 11 = start requirements, 12 = setup diagnosis] (3) = Module number [199] ! only for message type 1 -2! The following table shows the general structure of the "PVM3" FI				
Response Structure	comma	0	general structure of		
		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] (Pl	LC, CNC)	
	10 =	POE name	[ASCII characters]		
	11 =	Detail name	[ASCII characters] (er implementation)	npty	
	12 =	= Detail type [1 = action b 3 = transitio 4 = implem			
	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 PV/M3	Reference text of a ProVi error with ID 43923028 from module 3 in control				

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			



FI command		00_BR_PVM3_43923028_1_3
Line	Column	Answer
	10	MODULE3
	11	
	12	4
	13	34
	14	EschutzT
	15	Station2.Module3
	16	3
	17	NO
	18	
	19	D:\Programme\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(3) = Clock frequency	[199] ! only for message type 1 -2! [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 chara	[ASCII characters] [ASCII characters]				
	2 = Message number	[ASCII chara					
	3 = Time stamp day	[mm.dd.yyy	[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 chara	acters]				
Example PVM1	ProVi errors from module 3 in cor						
	The 2 nd message is being output. The clock frequency is to be 2 seconds.						



FI comma	and	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	

Download of PLC Retain Variables: PVR

Designation	PVR PLC Variable Retain Backup				
Explanation	Download of PLC retain variables.				
FI command	BW_PVF (1) = Dov	.,	(Single Write) 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 [0120] (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 comma	and	00_BW_PVR1!"D:\Program Files\Indramat\Mtgui\ Temp\download.ini"/3		
	Line	Column	Answer		
	1	1	01		
	2	2 1 00_BW_PVR1!"D:\Program Files\Indramat\Mtgui\ Temp\download.ini" /3			
	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.				



Upload of PLC Retain Variables: PVR

			MWMX and MWSX device groups	
Designation	PVR PLC Variable Retain Backup			
Explanation	PLC retain variables are uploaded via all active processes.			
FI command	BR_PVR1!(1) (Single Read)			
	(1) = Up	load file wit	h path details	
	Note:		file and path details in inverted commas.	
		The sepa	arator "!" is used in this command.	
Response Structure			e "PVR1" FI command consists of three lines, each e meaning of the elements is as follows:	
	-	1 = Job ID Chapter "I	[0120] FI Commands for the MPCX Device Group", IFJ).	
		2 = FI comr g, in accord	mand dance with Chapter "Elements of the FI Command"]	
	-	-	Error Code (see Chapter "Error Codes")	
Example PVR	00_BR_F	vR1_"D:\P	rogram Files\Indramat\Mtgui\Temp\Upload.ini"/3	
	FI comm	and	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 mu	st be taken in the use of upper and lower case letters.	
ing the PLC Varia	ble Dec	laration	n: PVT	
			MWMX and MWSX device groups	

Designation	Ρ٧Τ	PLC Variable Type					
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 1n:	Column 1	Column 2			
	n = number of elements.						

n = number of elements.



Reading

Value Range/Meaning	1 = Identifier of the I	on part of the PLC]				
of Columns	2 = Type	[s	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 (N	ame)	Name		
	1	TEST		WORD		
WinPcI: Reading a Variable	Assumption: The WinPcl variable "TEST" is declared as WORD in WinPcl prog "Prog".			in WinPcl progra		
	FI command	FI command 00_BR_PVT_:Prog.TEST				
	Answer					
	Line	Column 1 (N	ame)	Name		
	1	TEST		WORD		
PLC: Reading a Structure	Assumption: The "TEST1" variable is declared as STRUCT in the PLC progra STRUCT E1 BOOL E2 INT E3 SINT END			PLC program.		
	FI command	00_BR_PVT_TES	ST1			
		Ansv	wer			
	Line	Column	1	Column 2		
	1	TEST1.E	1	BOOL		
	2	TEST1.E	2	INT		
	3	SINT				
VinPel: Peading a Structure	Assumption:		1			

WinPcl: Reading a Structure

Assumption:

The WinPcl variable "TEST1" is declared as STRUCT in WinPcl program "Prog".

ST	RU	ICT	
ST	RU	ICT	

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 As

<u>Assumption:</u> The "TEST2" variable is declared as ARRAY in the PLC program.

0..3]OF BOOL

FI command	00_BR_PVT_TEST2	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				
	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 IN			
	3	SINT		
	1 TEST3[1].E1 2 TEST3[1].E2		BOOL	
			INT	
	3	TEST3[1].E3	SINT	

Assumption:

The data types are output according to IEC1131.

See also command PVF.

Set the Device Status Information: SDS

MWMX and MWSX device groups

 Designation
 SDS
 Set Device Status

 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 t "SDS1" FI command.			
	Line 1	Column 1		
Value Range/Meaning of Columns	1 = Status report	[(P_ACK)]		



•			1		
	FI command		00_BW_SDS1_	D	
	Line	Column	Answer		
	1	1	(P_ACK)		
FI command	With this command, the device status information for a selected devic can be set.				n for a selected device
	BW_SD	S2_(1)	(Sing	gle Write)	
		vice status mation to b			nformation OFF nformation 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 devic	e status info	ormation to OFF	for the select	ed device 00.
	FI comm	and	00_BW_SDS2_0	0	
	Line	Column	Answer		
	1	1	(P_ACK)		
Sequencer Data: SFD	SFD	SFC Dat	а	MWMX and	d MWSX device groups
Designation	310		a		
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	e data for a	step chain.		
	BR_SFD	01!(1)!(2)		(Single Read)	
	(1) = Mo	dule numbe	er	[199]	
	(2) = SF0	C entity nan	ne	ASCII charact	ers]
	Note:	The sepa	rator "!" is used	in this comma	nd.
Response Structure	The follo	-	shows the g	eneral structu	ire of the "SFD1" FI
		Line 1	Colu	umn 1	Column 2
Meaning of the Columns	1 = Step	chain com	ment [AS	CII characters]
	2 = POE	name	[AS	CII characters]

Example SDS1 Set device status information to OFF for ALL defined devices.



Example SFD1	Query data of the "clamp" chain in module 3 in control unit 0.			III U.	
	FI comma	and	00_BR_SFD1!3!	Station03A.Clamp	
	Line	Column	Answer		
	1	1	Clamping device		
		2	CLAMP		
FI command	Querv the	e data of a s	step		
	BR_SFD2!(1)!(2)!(3) (Single Read)				
		dule numbe	, ,	199]	
	()	C entity nar	-	ASCII characters]	
	(3) = Step name [ASCII characters]				
			•	-	
	Note:	The sepa	arator "!" is used i	n this command.	
Response Structure		d. The num		eneral structure of ends on the numb	
	If there ar	e no detail:	s the line number	is 1.	
	Lin	e 1	Column 1		Column 3
	Line	2n:	Column 1		Column 6
Meaning of the Columns	Line 1				
U	1 = Step comment [ASCII characters]				
	2 = Maximum time [ASCII characters]				
	3 = Minimum time [ASCII characters]				
	Line 2n:				
	1 = Detail type [1 = action block, 3 = transition]				ansition
	2 = Nam		-	CII characters]	anshorij
	2 = Nam 3 = Com		-	CII characters]	
		ean variable	-	S, NO]	
	5 = Qual		-	CII characters]	
	6 = Time			CII characters]	
Example SFD2			-	' chain in module 3	on control unit 0.
·	FI comm		-	Station03A.Clamp!C	
	Line	Column	Answer		-
	1	1	Open clamping d	evice	
		2	T#5s		
		3			
	2	1	1		
		2	aOpen		
		3	Clamp open		
		4	NO		
		5	D		
		6	T#3s		

Example SFD1 Query data of the "clamp" chain in module 3 in control unit 0.

3

1

3



-

Г

	FI command		00_BR_SFD2!3!Station03A.Clamp!Open	
	Line	Column	Answer	
		2	tOpen	
		3	Clamping device is open	
		4	NO	
		5		
		6		
FI command	Output the	e data for a	detail.	
	•	3!(1)!(2)!(3		
	(1) = Moo	dule numbe	er [199]	
	(2) = SFC	C entity nam	ne [ASCII characters]	
	(3) = Detail type [1 = action block, 2 = action n 3 = transition]			
	(4) = Detail name [ASCII characters]			
	Note:	The sepa	rator "!" 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 = Com	ment	[ASCII characters]	
	2 = Boole	an variable	e [YES, NO]	
	Data for the action "aOpen" in the "clamp" chain in module 3 in controunit 0.			
	FI command 0		00_BR_SFD3!3!Station03A.Clamp!aOpen	
	Line	Column	Answer	
	1	1	Clamp open	
		2	NO	

Т

Sequencer Messages: SFE

Designation	SFE	SFC Error	
Explanation	The seque	encer messages of a mo	dule are output.
FI command	Output all	SFC messages.	
	BR_SFE	1_(1)	(Single Read)
	BC_SFE	1_(1)	(Cyclic Read)
	(1) = Moo	dule number	[199]
	Output fire	st SFC messages.	
	BR_SFE	2_(1)	(Single Read)
	BC_SFE	2_(1)	(Cyclic Read)
	(1) = Moo	dule number	[199]

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 a	e no messa	ages, the num	ber of lines is	0.		
		Line 1	n:	Column 1		Column 7	
Meaning of the Columns	1 = Mess	sage text	[4	SCII characte	ers]		
-	2 = SFC	entity name	e [A	[ASCII characters]			
	3 = Step	name	[4	[ASCII characters]			
	4 = Time	stamp day	′ [n	nm.dd.yyyy]			
	5 = Time	stamp time	e [h	h:mm:ss]			
	6 = Type	of error		= time error, B = monitor ev		error,	
	7 = analysis?	Is there	condition [Y	(ES, NO]			
Example SFD1	All SFC n	nessages fr	om module 2	in control unit	0.		
	There are	e two messa	ages:				
	FI comm	and	00_BR_SFE1	_2			
	Line	Column	Answer				
	1	1	TIME ERROR	: Chain: chucki	ng Step: up m	nalfunction	
		2	Station03A.Cl	amp			
		3	Open				
		4	01.27.2000				
		5	11:56:32 AM				
		6	1				
		7	YES				
	2	1	ASSY ERROF	R: Chain: drilling	g Step: down	malfunction	
		2	Station02A.Dr	ill			
		3	Down				
		4 01.27.200					
		5	5 13:03:12				
		6 2					
		7 NO					
Example SFE2		message f	from module 2	in control uni	t 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



Sequencer Mode: SFM

Designation	SFM SFC Mode						
Explanation	Queries s	tep chain n	node.				
FI command	Query the mode of a step chain. BR_SFM1!(1)!(2) BC_SFM1!(1)!(2) (1) = Module number (2) = SFC entity name			(Single Read) (Cyclic Read) [199] [ASCII characters]			
	Note:	Note: The separator "!" is used in this command.					
Response Structure	The following table shows the general structure of the "SFM1" F command.					"SFM1" FI	
		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_SF			M1!3!Station03A.Clamp			
	Line	Line Column Answer					
	1	1	1				
Software Installation D	ata: SII)					
		-			and MWSX c		
Designation	SID	S oftware	e Installation			evice groups	
Explanation	installatio		ne software	g installation. version used,			
FI command	Read-in tl	ne installati	on data.				
	BR_SID1		(Si	ngle Read)			
	BC_SID1		(C)	clic Read)			
Response Structure	One line v	with 8 colur	nns is output	for the returne	ed values.		
		Line 1		Column 1		Column 8	
Meaning of the Columns	1 = Basic	c directory		[EXE files of th	ne BOF]		
		stallation di	-	[FI directory]			
	3 = Data directory			[in accordance	-		
	4 = GBO			[from INDRAN	-		
	5 = IF-DI			[from INDRAN	-		
	6 = IF ve	rsion		[from INDRAN 400]	ia i .ini - from	DEE MODE	
	7 = Servi	ce package		[from INDRAN 420]	IAT.ini - from	DLL mode	

8 = Release info

[from INDRAMAT.ini - from DLL mode 420]

Example SID1	Return information on the current installation.	
--------------	---	--

FI command		00_BR_SID1
Line	Column	Answer
1	1	
	2	D:\Programme\Indramat\MTGUI\Bin
	3	
	4	005-22Vxx
	5	07.00
	6	07\/00
	7	
	8	

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

PLC Long Identification: SLI

-			MWM>	and MWSX	device groups	
Designation	SLI	SPS Long Identification	on			
Explanation	Returns	the unit data from the PL	C long identif	ication.		
FI command	Read P	LC long identification.				
	BR_SLI		(Single Rea	d)		
Response Structure	One line	e with 15 columns is outpu	it for the retur	ned values.		
		Line 1	Column 1	Column	Column 15	
Value Range/Meaning of the	1 =	device address	[00	15]		
Columns	2 =	program number	[01	[0199]		
	3 =	Project name	[max	[max. 8 ASCII characters]		
	4 =	Program name	[max	. 8 ASCII cha	aracters]	
	5 =	User name	[acc.	to password	entry]	
	6 =	Program length	[byte	s]		
	7 =	Compilation time	[LON	IG] (coded in	long value)	
	8 =	Compilation date	[8 AS	SCII characte	rs]	
	9 =	Compilation time	[8 AS	SCII characte	rs]	
	10 =	Download time	[LON	IG] (coded in	long value)	
	11 =	Download date	[8 AS	SCII characte	rs]1	
	12 =	Download time	[8 AS	SCII characte	rs]	
	13 =	Version of PLC long identification	[LON	IG]		
	14 =	RUN flags	[HEX	(value]		
	15 =	Compiler info	[LON	IG]		

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

Read the unit data from the PLC long identification.

Example SLI

FI

Reference to Literature see chapter entitled "Literature" [30].

Requesting Watch List Allocations: WLA

Designation	WLA	Watch List Al	location					
Explanation	•	Requests free watch list allocations. A maximum of ten free watch list allocations can be requested with one FI command.						
	BR_W	BR_WLA1_(1) (Single Read)						
	· · /	umber of requeste atch list allocations		ed number of free is identified here e: 110.				
Response Structure		owing table shows nd "WLA1".	s the general stru	icture of the res	ponse to the FI			
		Line 1	Column 1		Column n			
Value Range/Meaning	1 =	1. free watch list	allocation	Value range: 1?	16			
of Columns	2 = 2. free watch list allocation Value ran			Value range: 1?	16			
	3 = 3. free watch list allocation			Value range: 116				
	n =	nth free watch list	allocation	Value range: 1?	16			



Example WLA1

1 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

Freeing Watch List Allocations: WFL

Designation	WLF	Watch L	ist F re	ee					
Explanation	Previously	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:					es ALL assig her WIN32 app	ned watch list lications.		
Response Structure	The following table shows the general structure of the response to the FI command "WLF1".								
		Line 1		Col	umn 1		Column n		
Value Range/Meaning	1 =	1. freed wat	ch lis	t allocat	ion	Value ra	Value range: 116		
of Columns	2 = 2	2. freed wat	ch lis	t allocat	ion	Value ra	ange: 116		
	3 = 3	3. freed wat	ch lis	t allocat	ion	Value range: 116			
	n = 1	nth freed wat	tch list	allocatio	on	Value ra	ange: 116		
Example WLF1	Free Al I	assigned v	vatch	list alloc	ations				
	<u>Assumpti</u>	on:				allocated: 1, 2	, 3, 4.		
	FI comm	and	00_E	BR_WLF	1				
	Line	Column	Ans	wer					
	1	1	1						
		2	2						
		3	3						
		4	4						
FI command	Free the	required wa	tch lis	st alloca	tions for a	selected devic	e.		
	BR_WLI	F2_(1)_{(2)	(10)	}	(Single I	Read)			
		= List of wa ns to be rele				um of 10 watch ns can be trans 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	1 =	1. freed watch lis	t allocation	Value range: 116		
of Columns	2 =	2. freed watch lis	Value range: 116			
	3 =	3. freed watch lis	t allocation	Value range: 116		
	n =	nth freed watch list	t allocation	Value ra	ange: 116	

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



7.6 FI Commands for the MWAX Device Group

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 Add		Address
	MWAX MTA200-P (MTA200-controller)		[0063]
1		Please note that the device address must be se respective FI command, e.g. "00_BR_ASM1" (rel to the Chapter entitled "Elements of the FI Comma	fer also here

MTA200 Messages: ADM

MWAX device group

Designation	ADM MTA200 Messag	Jes		
Explanation	MTA200 NC messages are output. These messages are assigned to a specific module and message type.			
FI command	Output all MTA200 messages	S.		
	BR_ADM1_(1)_(2)	(Single Read	(k	
	BC_ADM1_(1)_(2)	(Cyclic Read	I)	
	(1) = Message type	[1 = error, 2	= messages]
	(2) = Module number	[199]		
	Output of first MTA200 mess	age.		
	BR_ADM2_(1)_(2)	(Single Read	(k	
	BC_ADM2_(1)_(2) (Cyclic Read)			
	(1) = Message type [1 = error, 2 = messages]]	
	(2) = Module number	[199]		
Response Structure		the general structure of the FI commands number of lines depends on the number of		
	If there are no messages, the	e number of lines is (0.	
	Line 1n	Column 1 Column 6		
Meaning of the Columns	1 = Message text	[ASCII chara	acters]	
	2 = Message number	[032768]		
	3 = Time stamp day	[mm.dd.yyyy]	
	4 = Time stamp time	[hh:mm:ss]		
	5 = Message group	[19999]		
	6 = Reference text exists	[YES, NO]		

Example ADM1

All MTA200 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 MTA200 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

[YES, NO]

[ASCII characters]

FI command Output the additional information of a MTA200 message.

BR_ADM3_(1)_(2)_(3)	(Single Read)
(1) = Module number	[199]
(2) = Message number	[032768]
(3) = Message group	[19999]

Response Structure	The following table shows the command.	general structure of th	e "ADM3" FI
	Line 1	Column 1	Column 7
Meaning of the Columns	1 = Message text	[ASCII characters]	
	2 = Message number	[032768]	
	3 = Time stamp day	[mm.dd.yyyy]	
	4 = Time stamp time	[hh:mm:ss]	
	5 = Message group	[19999]	

6 = Additional text exists

7 = Additional text

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

Example ADM3

Additional text of an MTA200 error in module 3 in control unit 0.

Active Mechanism Messages

Active Mechanism Messages: AMM

Designation

AMM

MWAX device group

Designation			lessayes		
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_AMI	M7	(Single Rea	ld)	
	BC_AMI	М7	(Cyclic Rea	d)	
	BB_AMI	M7	(Break Cyc	lic Read)	
Response Structure	Note: The "AMM7" FI command refers <u>only</u> to devices within the MWAX device group. You should therefore make sure that only MTA devices are addressed via the system address. 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.			ake sure that address. onse to the FI maximum of ndividual error est (triggering)	
		Line 1n:	Column 1	Column	Column 11
Value Range/Meaning of the Columns	1 = 2 = 3 = 4 = 5 = 6 = 7 = 8 = 9 = 9 = 5 = 5 = 5 = 5 = 5 = 5 = 5 = 5	Device address Device name Mechanism number Mechanism name Type of message Message source Message group Message number Message text	 [0015] [max. 32 ASCII characters] [0, default value always 0] [max. 28 ASCII characters, default value always the MTA process] [F = Fault/Error, D = Diagnosis] [CNC, SPS, default value always "CNC"] [19999] [032768] [max. 1024 ASCII characters] 		
	5 -	message lext	[11ax. 1024 A		515]



10 =	Additional text	[X = exists, = does not exist, Default value does not exist (compatibility with Rexroth Indramat control units)]
11 =	2 bytes of additional information for the message number	[is required to resolve the information "@", default value "0" (compatibility with Rexroth Indramat control units)]
12 =	File name for additional information	e.g. in HTML format

Example AMM7	Read the current mechanism messages of device address 3 (MTA200).
--------------	---

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
	12	
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
	12	
3	1	03
	2	Crankshaft grinding machine
	3	0
	4	MTA process
	5	F
	6	CNC



FI command		00_BR_AMM7
Line	Column	Answer
	7	1
	8	19
	9	Emergency-OFF with immediate stop
	10	
	11	0
	12	

Actual (Current) Position of an Axis: APO

Designation	ΑΡΟ	Actual Axis POsition						
Explanation	"APO1" ret meaning. C	al position of a selected axis is read out. The FI command returns the position of an axis, related to the code of the axis . On the other hand, the FI command "APO2" returns the position s, 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.						
		ptional fourth parameter it is tinto mm or inches.	possible to pre-select conversion					
	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 p	rocess number	[0]					
	(2) = Axis ı	meaning	[116, the axis meaning corresponds to the physical axis number]					
	(3) = Syste	em of coordinates	[1 = machine coordinates2 = program coordinates3 = relative coordinates]					
	(4) = Req (opt.)	uired measurement system	[mm, inch]					
FI command		position of the selected axis I axis number.	of the device specified, related to					
	Using the of the result	pptional third parameter it is tinto mm or inches.	possible to pre-select conversion					
	CR_APO2	_(1)_(2){_(3)}	(Single Read)					
	CC_APO2	_(1)_(2){_(3)}	(Cyclic Read)					
	CB_APO2	_(1)_(2){_(3)}	(Break Cyclic Read)					
	(1) = Physi	ical axis number	[10.16, according to settings of the system parameters]					
	(2) = Syste	em of coordinates	[1 = machine coordinates2 = program coordinates3 = relative coordinates]					
	(3) = Req (opt.)	uired measurement system	[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".					
	Lir	ne 1	Column 1	Colum	n 2 Column 3	3 Column 4
Value Range/Meaning of Columns	1 = Axis 2 = Posit 3 = Unit 4 = Posit	ion	[accord [accord mm, in	ling to se ling to se	ttings of axis pa ttings of proces ttings of proces	s parameters]
	Note:	If the sel columns i		s not de	fined then the	response in all
Example APO1	process		e address (ordinates in NC ed in the basic
	FI command 00_CR_APO1_0_3_1					
	1	0.1		Answer	0	O alterna d
	Line 1	Column Z		1 mn 2 2345	Column 3 [mm]	Column 4 -1.2345
Example APO1	Read the	current po	sition of the	e Z axis		ordinates in NC
	FI comma	and	00_CR_APO	01_0_3_1_	inch	
				Answer		
	Line	Column	1 Colu	ımn 2	Column 3	Column 4
	1	Z	-0.0	0486	[inch]	-0.0486
Example APO2	machine		for the devi			number = 3) in es are indicated
	FI comma	and	00_CR_APO	2_3_1		
				Answer		
	Line	Column		ımn 2	Column 3	Column 4
	1	Z1	-1.2	2345	[mm]	-1.2345
Reference to Literature	See chap	ter entitled	"Literature" [16].		



Active System Error Messages: ASM

Designation	ASM	Active System	n M essages	S		
Explanation	The active system error messages 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 system error messages only refers to the PLC part (ISP200).					
FI command		he system error n MWAX device gr		urrently	y pending for a	Il active devices
	BR_ASM	И1	(Single R	ead)		
	BC_ASM	И1	(Cyclic Re	ead)		
	BB_ASN	/ 1	(Break Cy	clic R	ead)	
	Note:	device group.	Therefore	, any		ices within this ddress can be SM1").
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 error message.					
		Line 1n	Colum	n 1		Column 7
Value Range/Meaning	1 =	Line 1n Device address	Colum	n 1 [00?		Column 7
Value Range/Meaning of Columns		-	Colum	[00?		
	1 =	Device address		[00?	15] 32 ASCII char	
	1 = 2 =	Device address Device name	r	[007 [max. [018	15] 32 ASCII char	acters]
	1 = 2 = 3 =	Device address Device name Message numbe	r	[007 [max. [015 [F = fa	15] 32 ASCII char 50]	acters] diagnosis]
	1 = 2 = 3 = 4 =	Device address Device name Message numbe Type of message	r	[007 [max. [015 [F = fa [max.	15] 32 ASCII char 50] ault/error, D = c	acters] liagnosis] acters]
	1 = 2 = 3 = 4 = 5 =	Device address Device name Message numbe Type of message Short text	r e onal	[007 [max. [015 [F = fa [max. [x= ex is req	15] 32 ASCII char 50] ault/error, D = c 54 ASCII char	acters] diagnosis] acters] ot exist] e the
	1 = 2 = 3 = 4 = 5 = 6 =	Device address Device name Message numbe Type of message Short text Reference text 2 bytes of addition information	r Ə ənal number	[00 ² [max. [018 [F = fa [max. [x= ex is req inform	15] 32 ASCII chara 50] ault/error, D = c 54 ASCII chara ists, = does no uired to resolve	acters] diagnosis] acters] ot exist] e the
	1 = 2 = 3 = 4 = 5 = 6 = 7 = 8 =	Device address Device name Message numbe Type of message Short text Reference text 2 bytes of addition information for the message File name for add	r e onal number ditional	[00? [max. [01 [F = fa [max. [x= ex is req inforn e.g. ir	15] 32 ASCII char 50] ault/error, D = c 54 ASCII char ists, = does no uired to resolve nation "@" (see n HTML format	acters] diagnosis] acters] ot exist] e the e ASM5)
of Columns	1 = 2 = 3 = 4 = 5 = 6 = 7 = 8 = Read the MWAX of Assump	Device address Device name Message numbe Type of message Short text Reference text 2 bytes of addition for the message File name for add information e current system device group. tion:	r e number ditional error mess	[00 ² [max. [018 [F = fa [max. [x= ex is req inforn e.g. ir	15] 32 ASCII char 50] ault/error, D = c 54 ASCII char ists, = does no uired to resolve nation "@" (see n HTML format	acters] diagnosis] acters] ot exist] e the e ASM5)
of Columns	1 = 2 = 3 = 4 = 5 = 6 = 7 = 8 = Read the MWAX of <u>Assump</u> The follo	Device address Device name Message numbe Type of message Short text Reference text 2 bytes of addition information for the message File name for add information e current system device group. tion: powing three device	r e number ditional error mess	[00 ² [max. [018 [F = fa [max. [x= ex is req inforn e.g. ir	15] 32 ASCII char 50] ault/error, D = c 54 ASCII char ists, = does no uired to resolve nation "@" (see n HTML format	acters] diagnosis] acters] ot exist] e the e ASM5)
of Columns	1 = 2 = 3 = 4 = 5 = 6 = 7 = 8 = Read the MWAX of Assump The follo	Device address Device name Message numbe Type of message Short text Reference text 2 bytes of addition for the message File name for add information e current system device group. tion: wing three device ce address 01,	r e onal number ditional error mess es are defin	[00 ² [max. [018 [F = fa [max. [x= ex is req inforn e.g. ir	15] 32 ASCII char 50] ault/error, D = c 54 ASCII char ists, = does no uired to resolve nation "@" (see n HTML format	acters] diagnosis] acters] ot exist] e the e ASM5)
of Columns	1 = 2 = 3 = 4 = 5 = 6 = 7 = 8 = Read the MWAX of <u>Assump</u> The follo • Devic	Device address Device name Message numbe Type of message Short text Reference text 2 bytes of addition information for the message File name for add information e current system device group. tion: powing three device	r e onal number ditional error mess es are defin	[00 ² [max. [018 [F = fa [max. [x= ex is req inforn e.g. ir	15] 32 ASCII char 50] ault/error, D = c 54 ASCII char ists, = does no uired to resolve nation "@" (see n HTML format	acters] diagnosis] acters] ot exist] e the e ASM5)



	FI comm	and	07_BR_ASM1			
	Line	Column	Answer			
	1	1	01			
		2	Drill center			
		3	71			
		4	F			
		5	PLC battery volt	age too	low.	
		6	x			
		7	0			
		8				
	2	1	07			
		2	Milling center 1			
		3	74			
		4	F			
		5	SLM time monit	oring		
		6	X			
		7	0			
		8	10			
	3	1				
	Ŭ	2 Milling center 2				
		3	1			
		4	D			
		5	Error has been	correcter	4	
		6	X	Conceller	J.	
		7	^ 0			
		-	0			
		8				
FI command			pending system	error m	nessage of the	selected device
	BR_ASN	MWAX dev	- .	ood)		
	BC_ASN		(Single R (Cyclic R			
	BB_ASN		(Break C		ad)	
Response Structure	The follow command device ac	wing table s d "ASM2". ddress, dev indication	shows the gener The response c vice name, mes of whether ther	ral struc onsists sage nu	ture of the res of a line of 7 o umber, messag	columns for the ge status, short
	L	ine 1n	Colum	n 1		Column 7
Value Range/Meaning	1 =	Device add	dress	[00	15]	
of Columns	2 =	Device nar	ne	[max	. 32 ASCII chai	racters]
	3 =	Message r	number	[01	-	
		Type of me	essage	[F = f	ault/error, D =	diagnosis]
	5 =	Short text		-	. 54 ASCII chai	-
	6 -	6 – Additional text [x- exists does not exist]			viete does n	ot oviet]

6 = Additional text



[x= exists, -- = does not exist]

- 7 = 2 bytes of additional is required to resolve the information for the message number (see ASM5)
- 8 = File name for additional e.g. in HTML format information

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	Answer
1	1	01
	2	Drill center
	3	71
	4	F
	5	PLC battery voltage too low.
	6	Х
	7	0
	8	

FI command Output current system error messages 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 $[00_01_02_\dots_15]$ devices

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 error message.

		Line 1n	Column 1		Column 7
Value Range/Meaning	1 =	Device address	[0015] [max. 32 ASCII chara		
of Columns	2 =	Device name			acters]
	3 =	Message number	[0150]	[0150] [F = fault/error, D = diagnosis] [max. 54 ASCII characters] [x= exists, = does not exist] is required to resolve the information "@" (see ASM5) e.g. in HTML format	
	4 =	Type of message	[F = fault/er		
	5 =	Short text	[max. 54 AS		
	6 =	Reference text	[x= exists,		
	7 =	2 bytes of additional information for the message number	•		
	8 =	File name for additional information	e.g. in HTM		



Example ASM3

M3 Read the current system error messages for the selected MWAX devices.

Assumption:

The following devices addresses are defined:

- Device address 01,
- Device address 07 and
- Device address 10.

FI comm	and	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	Х
	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

mand Output current system error messages 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 error message.

	Line 1n		Column 1		Column 7		
Value Range/Meaning of Columns	1 =	Device address	[0015]				
	2 =	Device name	[max. 32 ASCII characters]				
	3 = Message number [0150]		[0150]				
	4 =	Type of message	[F = fault/err	[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	is required to resolve the information "@" (see ASM5)				



for the message number

- 8 = File name for additional e.g. in HTML format information
- Example ASM4 Read the current system error messages of all defined devices within the MWAX device group. <u>Assumption:</u>

The following devices are defined:

- Device address 01 and
- Device address 10.

	FI com	mand	01_BR_ASM4_	MWAX		
	Line	Column	Answer			
	1	1	01			
		2	Drill center			
		3	71			
		4	F			
		5	PLC battery vol	tage too	low.	
		6	Х			
		7	0			
		8				
	2	1	10			
		2	Milling center 2			
		3	1			
		4	D			
		5	Error has been	correcte	d.	
		6	х			
		7	0			
		8				
FI command			l text for the cur e message num		ending error m	essage, related
	BR_ASI	M5_(1)_(2)	(\$	Single I	Read)	
	(1) = Me	ssage numb	er [()150]		
	(2) = 2 number	bytes of a	dditional inform	nation f	or the messag	je
Response Structure	commar	nd "ASM5". T	shows the gene he response cor e, message num	nsists of	a line with 5 col	
		Line 1n	Colum	n 1		Column 5
Value Range/Meaning	1 =	Device addre	ess	[001	5]	
of Columns	2 =	Device name	Э			cters]
	3 =	Message nu	mber	[015	0]	
	4 =	Type of mes	sage	[F = fa	ault/error, D = di	agnosis]
	5 =	Reference te	ext		14 lines with a m ters/line]	ax. 78
		File name fo information	r additional	e.g. in	HTML format	



Example ASM5	Read the additional 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].

Reading and Writing CMOS RAM ASCII Parameters: CMA

					MWAX device group
Designation	СМА	CMOS F	AM A S	SCII Parameter	
Explanation	CMOS RA	M ASCII pa	aramete	ers can be read and	written.
FI command	Read CM	OS RAM A	SCII pa	arameters.	
	CR_CMA_	_(1)			(Single Read)
	(1) = CMC	S RAM AS	CII pa	ameter numbers	[079]
Response Structure		with one co II parameto		s output for the va	alue of the selected CMOS
Example Read CMA Parameter	Read the value of the CMOS RAM ASCII parameter with the number 0 at device address 00.				ameter with the number 0 at
	FI comma	and	00_CI	R_CMA_0	
	Line	Column	Answ	er	
	1	1	Waitir	g for tool change	
FI command	Write CM	OS RAM A	SCII pa	arameters.	
	CW_CMA_(1) (Single Write)				
	(1) = CMC	S RAM AS	CII pa	ameter numbers	[079]
Value to be written	Value of th	ne parameto	er	[ASCII characters]]
	Note:			written is passed sfer" routine.	to the "acValue" parameter
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		aiting for t 0 at device			OS RAM ASCII parameter
	FI comma	and		00_CW_CMA_0	
	Value to b	e written		Waiting for tool cha	nge



Reading and Writing CMOS RAM Floating Point Parameters: CMF

MWAX device group

Designation	CMF CMOS RAM Floatingpoint Parameter						
Explanation	CMOS RAM Floating Point parameters can be read and written.						
FI command	CR_CMF	Read CMOS RAM Floating Point parameters.CR_CMF_(1)(Single Read)(1) = CMOS RAM Floating Point parameter [079]numbers					
Response Structure			column is output for the value of the selected CMOS parameter.				
Example Read CMF Parameters		value of th address 00	he CMOS RAM Floating Point parameter numbered 1 0.				
	FI comma	and	00_CR_CMF_1				
	Line	Column	Answer				
	1	1	4711.0123				
FI command	Write CM	OS RAM F	Floating Point parameters.				
	CW_CMF_(1) (Single Write)						
	(1) = CMC	DS RAM Flo	loating Point parameter numbers [079]				
Value to be written	Value of t	he paramet	eter [Type: floating point]				
	Note:		ue to be written is passed to the "acValue" parameter pataTransfer" 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 comma	and	00_CW_CMF_1				
	Value to b	e written	4711.0123				

Read and Write CMOS RAM Integer Parameters: CMI

Designation	СМІ	CMOS RAM Integer Parameter				
Explanation	CMOS RAM Integer parameters can be read and written.					
FI command	Read CMOS RAM Integer parameters.(Single Read)CR_CMI_(1)(Single Read)(1) = CMOS RAM integer parameter numbers[079]					
Response Structure		ith one column is output for the va	lue of the selected CM	10S		





Example Read CMI Parameters	Read the value of the CMOS RAM Integer parameter numbered 2 at device address 00.					
	FI comma	and	00_CF	00_CR_CMI_2		
	Line	Column	Answ	er		
	1	1	12027	0		
FI command	Write CM	OS RAM Ir	nteger p	parameters.		
	CW_CMI_	<u>(</u> 1)			(Single Write)	
	(1) = CM0	OS RAM int	teger pa	arameter numbers	[079]	
Value to be written	Value of the parameter [Type: in				[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 comma	and		00_CW_CMI_2		
	Value to b	e written		120270		

Trigger Control Reset: CRT

Designation	CRT	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.					
		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 res	et 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	00_CW_CRT
Value to be written	0

Reference to Literature See chapter entitled "Literature" [26].

Device Configuration Parameter: DCP

Designation	DCP Device Configuration Parameter							
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 param	eters of all defined devic	ces.				
	BR_DO	CP1	(Single Read)					
	Note:		ommand refers to all d device address can b example DCP1).					
Response Structure	"DCP1"	'Fl command. The res	e general structure of th sponse consists of a m anisms = 512), each with	aximum of n=512				
		Line 1n:	Column 1	Column 7				
	Note: If no active machine parameter record exists in the device, then the columns [17] for the respective device are not applicable.							
Value Range/Meaning	1 =	Device address	[0063]					
of Columns	2 =	Device name	[max. 32 ASCII chara	acters]				
	3 =	Device Type	[MTC200-P-G2, MTC200-R-G2, MTVNC, MTRA-P, MTRA-R]					
	4 =	Mechanism number	[031]					
	5 =	Mechanism name	[max. 28 ASCII chara	acters]				
	6 =	Process type	[1= internal, 2 = exter	nal process]				
	7 =	Process type	[1 = NC process, 2 =	PLC process]				
Example DCP1	Read th Assum		parameters of all defined	devices.				
		devices have been defin	ed					
	• Dev	ice address 00 (MTC20	0-R-G2)					
	• Dev	ice address 01 (MTC20	0-P-G2) and					
	Device address 02 (MTC200-P-G2)							



	FI command		00_BR_DCP1	1		
	Line	Column	Answer			
	1	1	00			
		2	Rotary transfe	er machine		
		3	MC200-R-G2			
		4	1			
		5	Master			
		6	1			
		7	2			
	2	1	01			
	_	2	0			
		3	MTC200-P-G	2		
		4	0			
		5	Milling machir	ne 01		
		6	1			
		7	1			
	3	1	02			
	Ŭ	2	0			
		3	MTC200-P-G)		
		4	1	-		
		5	Milling machir	ne 02		
		6	1	10 02		
		7	1			
		1	1			
FI command	•	-	ation paramete		ted device.	
	BR_DC	P2	(Si	ngle Read)		
Response Structure			shows the ge d. The respons			esponse to the columns.
		Line 1	1	Column 1		Column 7
	Note:		columns [1.			in the device, levice are not
Value Range/Meaning	1 =	Device add	rass	[0015]		
of Columns	2 =	Device add		[max. 32 AS0	II characte	arel
	3 =	Device Typ	-	[MTC200-P-0 MTVNC, MT	G2, MTC20	0-R-G2,
	4 =	Mechanism	number	[031]		
	5 =	Mechanism		[max. 28 AS0	CII characte	ers]
	6 =	Process typ	be	[1= internal, 2		-
	_	_				

7 = Process type [1 = NC process, 2 = PLC process]

Rexroth Indramat

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].

Setting the Communication Timeout Time DCT

Designation	DCT	DCT Device Communication Timeout			
Explanation			nmand, the tir time in ms).	meout time for the selected device is set	
FI command	BW_DC	Г1_(1)		(Single Write)	
	(1) = req	uested time	out time in m	S	
Response Structure	The response column.	onse to the	"DCT1" FI co	mmand consists of one line with one	
		Line 1		Column 1	
Value Range/Meaning of Columns	1 = 5	1 = Status message (P_ACK) (P_ACK)			
Example DCT1	For the de	evice 00, th	e timeout time	e is set 1500 ms.	
	FI comma	and	00_BW_DCT	1_1500	
	Line	Column	Answer		
	1	1	(P_ACK)		
FI command	With this to default		the timeout ti	me for the selected device can be reset	
	BW_DC	Г2		(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 = 5	Status mess	age (P_ACK)	(P_ACK)	



Example DCP2 For the device 00, the timeout time is reset to the default value.

ſ	FI command		00_BW_DCT2
Ī	Line	Column	Answer
Ī	1	1	(P_ACK)

Long ID of PLC Data Block: DIS

Designation	DIS Data Identification String						
Explanation	 Reads the long identification (directory entries) of MTA200/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 MTA200/PLC data records are output: MTA200 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_DIS						
	BC_DIS						
			, neady				
Response Structure		lowing table shows the ge					
	"DIS1" FI command. The response consists of a line with five columns. Line 1 Column 1 Column 5						
		Line 1	Column 1		Column 5		
Value Range/Meaning of Columns	1 =	Line 1 Number in MTA200 param directory		 [0199]	Column 5		
	1 = 2 =	Number in MTA200 param	neter				
	-	Number in MTA200 param directory	neter meter record	[0199] [max. 32 AS			
	2 =	Number in MTA200 param directory Name of the MTA200para Length of the MTA200 par	neter meter record rameter ge to	[0199] [max. 32 ASt characters]	CII		
	2 = 3 =	Number in MTA200 param directory Name of the MTA200para Length of the MTA200 par record Date of creation/last chang	neter meter record rameter ge to d ge to	[0199] [max. 32 AS characters] [byte]	CII]		



Example DIS1	Read the directory entries of the MTA200 parameter record at device address 00. Assumption: There is a valid MTA200 parameter record in the selected device.						
	FI comm	nand	00_BR_DIS1				
	Line	Column	Answer				
	1	1	01				
		2	KEY1				
		3	3 3579				
		4	16.05.99				
		5	10:41:08				
Reference to Literature	See cha	pter entitled	"Literature" [2	29].			
FI command	BR_DIS	2	(Single Read)			
	BC_DIS:	2	(Cyclic Read)			
	BB_DIS	2	(Break Cyclic	c Read)			
Response Structure				eral structure of is of a line with		to the "DIS2"	
		Line 1		Column 1		Column 6	
Value Range/Meaning	1 =	Number in I	PLC directory		[0199]		
of Columns	2 =	Name of the	e PLC progra	m	[max. 8 AS0 characters]	CII	
	3 =	Length of th	ne PLC progra	am	[byte]		
	4 =	Date of creat program	ation/last cha	nge to PLC	[DD.MM.YY]	
	5 =]	
		PLC progra	m			1	
	6 =		m ation/last cha	nge to PLC	[DD.MM.YY	-	
	6 = Note:	Date of creat program	ation/last cha	C package in	-	YY]	
Example DIS2	Note: Read the Assump	Date of crea program If there is then all c e directory en tion:	ation/last chan s no valid No olumns conta tries of the PL	C package in	the selected	YY]	

and	00_BR_DIS2
Column	Answer
1	01
2	KEY1
3	20018
4	10.05.99
5	12:42:00
6	10.05.1999
	Column 1 2 3 4 5

Reference to Literature see chapter entitled "Literature" [30].



Reading the Device Status Information: DSI

Designation	DSI	Device Status	formatic	on		
Explanation		ables the most imp g information is retu		rice status information to be read. The		
	Туре с	of information		Sta	atus	Statement
	System	error information				
	Informa	ation on mechanism	n error			
	Machin	e key information		val	id	Yes/No
	Machin	e key information				
	Machin	e status informatior	า			
	Sercan	s information				
	Param	eter download		run	ning	Yes/No
	PLC do	ownload		run	ning	Yes/No
	Firmwa	are download		run	ning	Yes/No
		Online information				
		simulation		SW	tched on	Yes/No
	Device	status information				ON/ OFF
FI command	Read o	ut device status inf	formation for	or ALL	defined device	S.
	BR_DS	I1 (Sin	gle Read)			
	BC_DS	I1 (Cy	clic Read)			
	BB_DS	l1 (Bre	ak Cyclic	Read)		
	Note:	group. Therefore the command	ore, any val line (see e	lid devi exampl	ce address car	vithin this device be indicated in device polling configurator)!
Response Structure		lowing table show FI command.	rs the gene	eral str	ucture of the r	esponse to the
		Line 1n	Colum	n 1		Column 11
Value Range/Meaning	1 =	Device address		[006	53]	
of Columns	2 =	System error infor	mation	[0 = there is no system error 1 = there is a system error]		
	3 =	Information on m error	echanism	e 1 = th	nere is no mecha rror nere is a mecha	
	4 =	Machina kay infor	mation	[4 by#	a in HEX coding	1
	4 – 5 =	Machine key information		[4 byte in HEX coding] [0 = not valid, 1=valid]		
		Is machine key information valid?		-		
	6 =	Machine status int			e in HEX coding	-
	7 = 8 -	Sercans informati			e in HEX coding	-
	8 =	Is parameter down		1 = p	arameter downl	
	9 =	Is PLC download	active?		LC download no LC download ru	

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]

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	0000000
	5	0
	6	0000000
	7	0000000
	8	0
	9	0
	10	0
	11	1
	12	0
	13	1
2	1	03
	2	1
	3	0
	4	0000000
	5	0
	6	0000000
	7	0000000
	8	0
	9	0
	10	0
	11	1
	12	0
	13	1



FI command		ut device status inf		or a sel	ected device.	
	BR_DSI	·	igle Read)			
	BC_DSI		clic Read)			
	BB_DSI	2 (Bre	eak Cyclic	Read)		
Response Structure		owing table show -I command.	rs the gene	eral stru	ucture of the re	esponse to the
		Line 1n	Colum	า 1		Column 11
Value Range/Meaning	1 =	Device address		[006	3]	
of Columns	2 =	System error infor	mation		ere is no system ere is a system	
	3 =	Information on me error	 [0 = there is no mechanism error 1 = there is a mechanism error] 			
	4 =	Machine key infor	mation	[4 byte in HEX coding]		
	5 =	Machine key infor valid?	mation	[0 = not valid, 1=valid]		
	6 =	Machine status in	formation	[4 byte in HEX coding]		
	7 =	Sercans informati	on	[4 byte in HEX coding]		
	8 =	Is parameter dow active?	nload	[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 info12 = Device simulationon?		rmation		evice connection evice connection	
			n switched	-	O Simulation m mulation mode	
	13 =	Current device st information	tatus	-	evice status=O evice status=O	

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	0000000
	5	0
	6	0000000
	7	0000000
	8	0
	9	0
	10	0
	11	1
	12	0
	13	1



Distance to Go of Axis Movement: DTG

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 other hand, returns the distance to go of an axis, related to the other hand, returns the distance to go of the selected axis of the device specified, related to the code of the axis meaning. 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)} (Break Cyclic Read) (1) = NC process number [0] (2) = Axis meaning [116, 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 [mm, inch] (opt.) (C_DTG2_(1)_(2){_(3)} (Single Read) CL_DTG2_(1)_(2){_(3)} (Single Read) (C_DTG2_C1)_(2){_(3)} (Single Read) (C_DTG2_C1)_(2){_(3)} (4) = Required measurement system [mm, inch] (opt.) (Dupt) (C_DTG2_C1)_(2){_(3)} (Cyclic Read) (1) = Physical axis number [10.16, according to setings of the system p	Designation	DTG	Distance To	Go				
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)} (Break Cyclic Read) CB_DTG1_(1)_(2)_(3){_(4)} (Break Cyclic Read) (1) = NC process number [0] (2) = Axis meaning [116, the axis meaning corresponds to the physical axis number] (3) = System of coordinates [1 = machine coordinates 2 = program coordinates 3 = relative coordinates 3 = relative 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)} CB_DTG2_(1)_(2){_(3)} (Single Read) (Cyclic Read) CB_DTG2_(1)_(2){_(3)} [Break Cyclic Read] (1) = Physical axis number (1) = Physical axis number [10.16, according to settings of the system parameters] (2) = System of coordinates 2 = program coordinates (2) = System of coordinates [1 = machine coordinates 2 = program coordinates 3 = relative coordinates	Explanation	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,						
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 [116, the axis meaning corresponds to the physical axis number] (3) = System of coordinates [1 = machine coordinates 3 = relative coordinates 3 = relative coordinates] (4) = Required measurement system [mm, inch] (opt.) 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)} (Gyclic Read) (1) = Physical axis number [10.16, according to settings of the system parameters] (2) = System of coordinates [1 = machine coordinates 2 = program coordinates 3 = relative coordinates 3 = relative coordinates 3 = relative coordinates (2) = System of coordinates [1 = machine coordinates 2 = program coordinates 3 = relative coordinates 1 = program coordinates 3 = relative coordinates 3 = relative coordinates 3 = re	FI command					of the dev	ice specified,	
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 [116, 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 [mm, inch] (opt.) 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)} CC_DTG2_(1)_(2){_(3)} (Single Read) CC_DTG2_(1)_(2){_(3)} (Break Cyclic Read) (1) = Physical axis number [10.16, according to settings of the system parameters] (2) = System of coordinates 2 = program coordinates 2 = program coordinates 2 = program coordinates 3 = relative coordinates 4 = program coordinates 3 = relative coordinates 3 = relative coordinates 3 = relative coordinates 4 = program coordinates 3 = relative co					s possible	to pre-sele	ct conversion	
CB_DTG1_(1)_(2)_(3){_(4)} (Break Cyclic Read) (1) = NC process number [0] (2) = Axis meaning [116, the axis meaning corresponds to the physical axis number] (3) = System of coordinates [1 = machine coordinates 2 = program coordinates 3 = relative coordinates 3 = relative coordinates] (4) = Required measurement system [mm, inch] (pt.) 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)} CR_DTG2_(1)_(2){_(3)} (Single Read) CC_DTG2_(1)_(2){_(3)} (Break Cyclic Read) (1) = Physical axis number [116, the axis meaning correspondence of the system parameters] (2) = System of coordinates [1 = machine coordinates 2 = program coordinates 3 = relative coordinates 1 = mochine coordinates 3 = relative coordinates 3 = relative coordinates 3 = relative coordinates 1 = mochine coordinates 3 = relative coordin		CR_DTG1	_(1)_(2)_(3){	_(4)}	(Single	Read)		
(1) = NC process number [0] (2) = Axis meaning [116, the axis meaning corresponds to the physical axis number] (3) = System of coordinates [1 = machine coordinates 2 = program coordinates 3 = relative coordinates 3 = relative coordinates] (4) = Required measurement system [mm, inch] (pt.) 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)} (Break Cyclic Read) (1) = Physical axis number [10.16, according to settings of the system parameters] (2) = System of coordinates [1 = machine coordinates 3 = relative coordinates 1 = mochine coordinates 3 = relative coordinates 3 = relative coordinates 1 = mochine coordinates 1 = mochine coordinates 3 = relative coordinates 3 = relative coordinates 3 = relative coordinates 1 = mochine coordinates 2 = program coordinates 3 = relative coordinates 1 = mochine coordinates 2 = program coordinates 3 = relative coordinates 3 = relative coordinates 3 = relative coordinates 1 = mochine coordinates 2 = program coordinates 3 = relative coordinates 3 = relative coordinates 3 = relative coordinates 1 = mochine coordinates 1 = mochine coordinates 3 = relative coordinates 3 = relative coord		CC_DTG1	_(1)_(2)_(3){	_(4)}	(Cyclic Read)			
(2) = Axis meaning [116, 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 [mm, inch] (opt.) 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) (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 1 = machine coordinates 3 = relative coordinates 3 = relative coordinates 3 = relative coordinates 1 = mochine coordinates 3 = relative coordinates 1 = mochine coordinates 2 = program coordinates 2 = program coordinates 3 = relative coordinates 1 = mochine coordinates 3 = relative coordinates 1 = mochine coordinates 2 = program coordinates 2 = program coordinates 3 = relative coordinates 2 = program coordinates 2 = rotex coordinates 2 = program		CB_DTG1	_(1)_(2)_(3){	_(4)}	(Break (Cyclic Read	d)	
(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. (Single Read) CC_DTG2_(1)_(2){_(3)} (Single Read) CB_DTG2_(1)_(2){_(3)} (Break Cyclic Read) (1) = Physical axis number [10.16, according to settings of the system parameters] (2) = System of coordinates [1 = machine coordinates 2 = program coordinates 3 = relative 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".		(1) = NC p	rocess numb	er	[0]			
2 = program coordinates 3 = relative coordinates 3 = relative coordinates 3 = relative coordinates (opt.) 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)} Cyclic Read) CC_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) = Required measurement system (mm, inch] (opt.) 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".		(2) = Axis	meaning		correspo	onds to the		
(opt.) 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 [10.16, according to settings of the system parameters] (2) = System of coordinates [1 = machine coordinates 2 = program coordinates 3 = relative coordinates 3 = relative coordinates] (3) = Required measurement system [mm, inch] (opt.) 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".		(3) = System of coordinates			2 = proę	gram coordi	nates	
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 [10.16, according to settings of the system parameters] (2) = System of coordinates [1 = machine coordinates 2 = program coordinates 3 = relative 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".		• •	ired measure	ement system	[mm, inc	:h]		
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 [10.16, 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".	FI command						d axis of the	
CC_DTG2_(1)_(2){_(3)} (Cyclic Read) CB_DTG2_(1)_(2){_(3)} (Break Cyclic Read) (1) = Physical axis number [10.16, 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".					s possible	to pre-sele	ct conversion	
CB_DTG2_(1)_(2){_(3)} (Break Cyclic Read) (1) = Physical axis number [10.16, 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".		CR_DTG2	2_(1)_(2){_(3)	}	(Single	Read)		
(1) = Physical axis number [10.16, 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".		CC_DTG2	2_(1)_(2){_(3)	}	(Cyclic	Read)		
(2) = System of coordinates [1 = machine coordinates (2) = System of coordinates [1 = machine coordinates (2) = System of coordinates [1 = machine coordinates (3) = Required measurement system [mm, inch] (3) = Required measurement system [mm, inch] (opt.) 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".		CB_DTG2	2_(1)_(2){_(3)	}	(Break (Cyclic Read	d)	
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 "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".		(1) = Phys	ical axis num	ber				
(opt.) 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".		(2) = Syste	em of coordina	ates	2 = program coordinates			
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".		• •	iired measure	ement system	[mm, inc	:h]		
Line 1 Column 1 Column 4	Response Structure	commands axis desigr	"DTG1" and nation, distan	"DTG2". One ce to go, unit	ine is outp	ut with 4 co	lumns for the	
			Line 1	0	Column 1		Column 4	
Value Range/Meaning 1 = Axis name [according to settings of axis parameters]	Value Range/Meaning	1 = Axis na	ame	[according to	settings of	faxis paran	neters]	
of Columns 2 = Distance to go [according to settings of process parameters]		2 Distan	ce to go		-		-	
3 = Unit [mm, inch]		z = Distan	• • •			ennigo or proceso parametero]		
4 = Distance to go [as column 2]					Ū			
of Columns2 = Distance to go 3 = Unit[according to settings of process]3 = Unit[mm, inch]	Value Range/Meaning				-			

	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 comma	and 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 comma	and 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			go of the moveme coordinates at the			
	FI comma	and 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 chap	ter entitled "Lit	terature" [16].			
Device Type and Accor	mpanyir	ng Compo	onents: DTY			

Designation	DTY	Device TYpe					
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 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 = 2 =	Device Type Component type1	(see Chapter "I IND_DEV.INI-E Componenttype	intry:			
	3 =	Component type 2	IND_DEV.INI-E Componenttype				
Example DTY1 Output the device type and the accompanying components of address 00.					s of device		

FI command	00_BR_DTY1				
Answer					
Line	Column 1	Column 2	Column 3		
1	MTA200-P	MTS-P	MTC-P		

Diagnosis Window Data: DWD

Designation	DWD	Diagnosis Wine	dow D ata			
Explanation	they ca applicab	tic messages are n be output dire le, different types message, are retu	ctly in th of diagnos	ie diagno sis, such a	sis overviev as a ProVi m	v, i.e., where
FI command	Output a	all diagnostic mess	ages.			
	BR_DW	VD1_(1){_(2)}	(Single	Read)		
	BC_DW	VD1_(1){_(2)}	(Cyclic	Read)		
	(1) = Ty window	/pe of diagnosis	3 = gen 10 = sta	eral errors	sequence ei s, 4 = messa ditions, t = setup diag	ges,
	(2) = Mo	odule number	[199] !	only for v	vindow type ?	1 -4 !
	Output fi	irst diagnostic mes	sages.			
	BR_DW	VD2_(1){_(2)}	(Single	Read)		
	BC_DW	VD2_(1){_(2)}	(Cyclic I	Read)		
	(1) = Ty window	/pe of diagnosis	3 = gene 10 = star	 [1 = NC error, 2 = sequence errors, 3 = general errors, 4 = messages, 10 = start preconditions, 11 = warnings, 12 = setup diagnosis] 		
	(2) = Mo	odule number	[199] !	only for w	indow type 1	-4 !
Response Structure	"DWD2" message	owing table show FI commands. Thes pending. Differe	ne number	of lines	depends on	the number of
	diagnosi	s.				
	•	is. are no messages, f		r of lines i	s 0.	
	•		he numbe	r of lines i I umn 1	s 0. 	Column 12
Meaning of the Columns	If there a	are no messages, 1	he numbe	umn 1		
Meaning of the Columns	If there a	are no messages, t Line 1n	he numbe	umn 1	 haracters]	
Meaning of the Columns	If there a 1 = 2 =	are no messages, t Line 1n Message text	he numbe	lumn 1 [ASCII c	 haracters] yyyy]	
Meaning of the Columns	If there a 1 = 2 = 3 =	are no messages, t Line 1n Message text Time stamp day	he numbe	lumn 1 [ASCII c [mm.dd.]	 haracters] yyyy] ss]	
Meaning of the Columns	If there a 1 = 2 = 3 = 4 =	are no messages, t Line 1n Message text Time stamp day Time stamp hour	he numbe	[ASCII cl [Mm.dd.] [hh:mm: [YES, No [1 = Pro	 haracters] yyyy] ss]	Column 12
Meaning of the Columns	If there a 1 = 2 = 3 = 4 = 5 =	are no messages, t Line 1n Message text Time stamp day Time stamp hour Reference text ava	he numbe	[ASCII cl [Mm.dd.] [hh:mm: [YES, No [1 = Pro 3 = MTC	 haracters] yyyy] ss] D] Vi, 2 = SFC,	Column 12
Meaning of the Columns	If there a 1 = 2 = 3 = 4 = 5 = 6 =	Are no messages, t Line 1n Message text Time stamp day Time stamp hour Reference text ava Type of diagnosis	he numbe	[ASCII cl [mm.dd.] [hh:mm:: [YES, NG [1 = Prov 3 = MTC [ASCII cl [ASCII cl	 haracters] yyyy] ss] O] Vi, 2 = SFC, Vi, 2 = MT	Column 12
Meaning of the Columns	If there a 1 = 2 = 3 = 4 = 5 = 6 = 7 =	Are no messages, t Line 1n Message text Time stamp day Time stamp hour Reference text ava Type of diagnosis Message number	he numbe Co	[ASCII c] [ASCII c] [mm.dd.] [hh:mm:[hh:mm:[YES, NG] [1 = Provolution 3 = MTC] [ASCII c] [AS	 haracters] yyyy] ss] D] Vi, 2 = SFC, Vi, 2 = MT haracters] haracters]	Column 12 [A-NC] ProVi)
Meaning of the Columns	If there a 1 = 2 = 3 = 4 = 5 = 6 = 7 = 8 =	Are no messages, the Line 1n Message text Time stamp day Time stamp hour Reference text ava Type of diagnosis Message number Message ID	he numbe Co ailable	[ASCII cl [mm.dd.] [hh:mm:: [YES, NG [1 = Prov 3 = MTC [ASCII cl [ASCII cl (DWORI [031] (N	 haracters] yyyy] ss] D] Vi, 2 = SFC, C-NC, 4 = MT haracters] haracters] D, decimal) (MTC-NC) [0]	Column 12 [A-NC] ProVi) (MTA-NC)
Meaning of the Columns	If there a 1 = 2 = 3 = 4 = 5 = 6 = 7 = 8 = 9 = 10 =	Are no messages, the Line 1n Message text Time stamp day Time stamp hour Reference text ava Type of diagnosis Message number Message ID Mechanism number	he numbe Co ailable	[ASCII cl [mm.dd.] [hh:mm:: [YES, NG [1 = Prov 3 = MTC [ASCII cl [ASCII cl [ASCII cl [031] (N [ASCII cl [19999]	 haracters] yyyy] ss] D] Vi, 2 = SFC, C-NC, 4 = MT haracters] haracters] D, decimal) (MTC-NC) [0]	Column 12 [A-NC] ProVi) (MTA-NC)



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 comma	and	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 comma	and	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].

Existing MTA200 Diagnoses: EAD

Designation	EAD Existing MTA200 Diagnosis				
	Which MTA200 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 wh	ich MTA200	0 diagnostic type	s are available in a module.	
	BR_EAD	1_(1)	(\$	Single Read)	
	(1) = Moo	dule numbe	er [1	199]	
-	The following table shows the general structure of the "EAD1" FI command.				
	Line 1			Column 1-2	
Meaning of the Columns	1 = Mess	ages exist		[YES, NO]	
	2 = Error	s exist		[YES, NO]	
Example EAD1	Query the MTA200 diagnostic types in Module 2 on Control unit 0.				
	FI command 00_BR_EAD1_2				
	Line Column Answer				
	1	1	NO		
	2 YES		YES		



FI command	Query a sp	pecific MT	A200 diagnostic	c type.	
	BR_EAD	2_(1)_(2)	-	(Single Read)	
	(1) = Mes	sage type		[1 = error, 2 = messages]	
	(2) = Mod	lule numbe	er	[199]	
Response Structure	The follow command.	-	shows the	general structure of the "EAD2" FI	
		Line	1	Column 1	
Meaning of the Columns	1 = Diagn	iosis type e	exists [YI	ES, NO]	
Example EAD2	Are there a	any messa	iges in module	4 in control unit 0?	
	FI comma	Ind	00_BR_EAD2_	2_4	
	Line	Column	Answer		
	1	1	YES		
Existing errors: EDE					
				MWAX device group	
Designation	EDE	Existing	Diagnosis Erro	r	
Explanation				ontrol unit or in a module is queried. NC errors, MTA200 errors or ProVi	
	Query whether there are errors in this control unit.				
FI command	Query whe	ether there	are errors in th	is control unit.	
FI command	Query whe		are errors in th	is control unit. (Single Read)	
FI command		1	are errors in th		
FI command Response Structure	BR_EDE ² BC_EDE ²	1 1 wing table		(Single Read)	
	BR_EDE BC_EDE	1 1 wing table	shows the g	(Single Read) (Cyclic Read)	
	BR_EDE BC_EDE	1 1 wing table Line	shows the g	(Single Read) (Cyclic Read) general structure of the "EDE1" FI	
Response Structure	BR_EDE ² BC_EDE ² The follow command. 1 = Error	1 1 wing table Line exists	shows the g	(Single Read) (Cyclic Read) general structure of the "EDE1" FI Column 1	
Response Structure Meaning of the Columns	BR_EDE ² BC_EDE ² The follow command. 1 = Error	1 wing table <u>Line</u> exists exist in cor	shows the g 1 [YI	(Single Read) (Cyclic Read) general structure of the "EDE1" FI Column 1	
Response Structure Meaning of the Columns	BR_EDE BC_EDE The follow command. 1 = Error Do errors of FI comma	1 wing table Line exists exist in cor nd Column	shows the g	(Single Read) (Cyclic Read) general structure of the "EDE1" FI Column 1	
Response Structure Meaning of the Columns	BR_EDE ² BC_EDE ² The follow command. 1 = Error Do errors of FI comma	1 wing table Line exists exist in cor nd	e shows the g 1 [YI ntrol unit 0? [00_BR_EDE1	(Single Read) (Cyclic Read) general structure of the "EDE1" FI Column 1	
Response Structure Meaning of the Columns	BR_EDE BC_EDE The follow command. 1 = Error Do errors of FI comma Line 1	1 wing table Line exists exist in cor nd Column 1	e shows the g 1 [YI ntrol unit 0? 00_BR_EDE1 Answer YES	(Single Read) (Cyclic Read) general structure of the "EDE1" FI Column 1	
Response Structure Meaning of the Columns Example EDE1	BR_EDE BC_EDE The follow command 1 = Error Do errors of FI comma Line 1 Query whe BR_EDE	1 wing table Line exists exist in cor nd Column 1 ether or nor 2_(1)	e shows the g 1 [YI ntrol unit 0? 00_BR_EDE1 Answer YES	(Single Read) (Cyclic Read) general structure of the "EDE1" FI Column 1 ES, NO]	
Response Structure Meaning of the Columns Example EDE1	BR_EDE BC_EDE The follow command 1 = Error Do errors FI comma Line 1 Query whe BR_EDE BC_EDE	1 wing table Line exists exist in cor nd Column 1 ether or no 2_(1) 2_(1)	e shows the g 1 [YI ntrol unit 0? 00_BR_EDE1 Answer YES t errors exist in	(Single Read) (Cyclic Read) general structure of the "EDE1" FI Column 1 ES, NO] a specific module. (Single Read) (Cyclic Read)	
Response Structure Meaning of the Columns Example EDE1	BR_EDE BC_EDE The follow command 1 = Error Do errors FI comma Line 1 Query whe BR_EDE BC_EDE	1 wing table Line exists exist in cor nd Column 1 ether or nor 2_(1)	e shows the g 1 [YI ntrol unit 0? 00_BR_EDE1 Answer YES t errors exist in	(Single Read) (Cyclic Read) general structure of the "EDE1" FI Column 1 ES, NO] a specific module. (Single Read)	
Response Structure Meaning of the Columns Example EDE1	BR_EDE ² BC_EDE ² The follow command. 1 = Error Do errors of FI comma Line 1 Query whe BR_EDE ² BC_EDE ² (1) = Mod	1 wing table Line exists exist in cor nd Column 1 ether or nor 2_(1) 2_(1) lule number wing table	e shows the g 1 [YI ntrol unit 0? 00_BR_EDE1 Answer YES t errors exist in er	(Single Read) (Cyclic Read) general structure of the "EDE1" FI Column 1 ES, NO] a specific module. (Single Read) (Cyclic Read)	
Response Structure Meaning of the Columns Example EDE1 Fl command	BR_EDE BC_EDE The follow command. 1 = Error Do errors of FI comma Line 1 Query whe BR_EDE (1) = Mod The follow	1 wing table Line exists exist in cor nd Column 1 ether or nor 2_(1) 2_(1) lule number wing table	e shows the g 1 [YI ntrol unit 0? 00_BR_EDE1 Answer YES t errors exist in er e shows the g	(Single Read) (Cyclic Read) general structure of the "EDE1" FI Column 1 ES, NO] a specific module. (Single Read) (Cyclic Read) [199]	



Example EDE2 Do errors exist in Module 1 on Control unit 0?

FI comma	and	00_BR_EDE2_2
Line	Column	Answer
1	1	NO

Existing Diagnosis Window: EDW

Designation	EDW	Existing	Diagnosis	Window	
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 0n		Column 1	Column 2
Meaning of the Columns	1 = Type window	of diagnos	3 = 10	NC error, 2 = seque general errors, 4 = r start preconditions, warnings, 12 = setu	nessages,
	2 = Mode	ule number	0 =	CII characters] Diagnosis window ty module	pe does not belong to
Example EDW1	All types	of diagnosis	s window i	n control unit 0.	
	There are	e three diag	nosis wind	dows:	
	FI comm	and	00_BR_E	DW1	
	Line	Column	Answer		
	1	1	10		
		2	0		
	2	1	1		
		2	3		
	3	1	2		
		2	3		
FI command	Output al	diagnosis	window ty	pes for a module.	
	BR_EDV	V2_(1)		(Single Read)	
	(1) = Mo	dule numbe	er	[199]	
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 0n		Column 1	Column 2
Meaning of the Columns	window	of diagnos	3 -	= NC error, 2 = seque = general errors, 4 =	
	2 = Mod	ule number	0 =	SCII characters] Diagnosis window ty / module	vpe does not belong to



	FI command		00_BR_EDW2_3	3
	Line	Column	Answer	
	1	1	1	
		2	3	
	2	1	2	
		2	3	
FI command	Query a s	pecific type	of diagnosis wir	ndow.
	BR_EDV	V3_(1){_(2)	} (Single I	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 [199] ! 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 comma	and	00_BR_EDW3_1	_3
	Line	Column	Answer	
	1 1 YES			

Example EDW2 All types of diagnosis window in Module 3, Control unit 0. There are two diagnosis windows.

Existing ProVi Types: EPT

Designation	EPT Existing ProVi Typ	Des			
Explanation	Which ProVi types are prog queried. The data is returned the small control panels. T Moduldef.ini.	in a suitable form for	the message texts of		
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 0n	Column 1	Column 2		
Meaning of the Columns		1 = error, 12 = messa 0 = start requirements	•		
	2	1 = warnings, 22 = se	tup diagnosis]		



Example EPT1 All ProVi types in control unit 0.

There are three diagnosis windows.

FI comm	and	00_BR_EPT1
Line	Column	Answer
1	1	20
	2	0
2	1	11
	2	3
3	1	12
	1	3

Error Status: EST

Designation	EST	Error ST	ate	
Explanation	Queries the error state of a variable.			
FI command	Query the	frozen erro	or state of a varia	ble.
	BR_EST	1!(1)!(2)	(Single R	ead)
	BC_EST	1!(1)!(2)	(Cyclic Re	ead)
	(1) = Erro	or ID	[ASCII cha	aracters] (DWORD, decimal)
	(2) = Var	iable name	[ASCII cha	aracters]
	Note:	The sepa	rator "!" is used i	n 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.			
	<u>Suggestion:</u> The WinPcI variable "IB_EXT24" is declared as BOOL in the WinPcI program "Prog".			
	FI command 00_BR_EST1!58			92855!:Prog.IB_EXT24
	Line	Column	Answer	



Execution Display: EXD

Designation	EXD	EX ecutio	on D isplay	
Explanation	Information for displaying the execution of a movement is output.			
FI command	Query the execution of a step or of BR_EXD1!(1)!(2)!(3) BC_EXD1!(1)!(2)!(3) (1) = SFC entity name (2) = Step or action name (3) = Behaviour of mode Note: The separator "!" is used			an action. (Single Read) (Cyclic Read) [ASCII characters] [ASCII - characters] [1 – all modes, 2 – manual mode] d in this command.
Response Structure	command	-	snows the g	general structure of the "EXD1" FI
		Line	1	Column 1
Meaning of the Columns	1 = Exec	ution	-	– can be executed, 0 – cannot be ecuted]
Example EXD1	Query the 0 for all m		of the step "op	en" for the chain "clamp" in control unit
	FI comma	FI command 00_BR_EXD1!Station03A.Clamp!Open!1		
	Line	Column	Answer	
	Line 1	Column 1	Answer 1	
FI command	1 Query wh enabled.	1 nether the c	1	sis (control image) of a step chain is
FI command	1 Query wh enabled. BR_EXD	1 nether the c	1 condition analy	(Single Read)
FI command	1 Query wh enabled. BR_EXD	1 nether the c	1 condition analy	
FI command	1 Query wh enabled. BR_EXD	1 nether the c 2 !(1) C entity nan	1 condition analy ne	(Single Read)
FI command	1 Query wh enabled. BR_EXD (1) = SF(Note:	1 Dether the of D 2!(1) C entity nan The sepa wing table	1 condition analy ne rator "!" is used	(Single Read) [ASCII characters]
	1 Query whenabled. BR_EXD (1) = SFC Note: The follo	1 Dether the of D 2!(1) C entity nan The sepa wing table	1 condition analy ne rator "!" is used shows the g	(Single Read) [ASCII characters]
	1 Query whenabled. BR_EXD (1) = SFC Note: The follo	1 Dether the of D2!(1) C entity nan The sepa wing table I. Line	1 condition analy ne rator "!" is used shows the (1 1	(Single Read) [ASCII characters] d in this command. general structure of the "EXD2" FI
Response Structure	1 Query whenabled. BR_EXD (1) = SF0 Note: The follocommand 1 = Enable	1 ether the o 2!(1) C entity nan The sepa wing table I. Line	1 condition analy ne rator "!" is used shows the (1 [1 0	(Single Read) [ASCII characters] d in this command. general structure of the "EXD2" FI Column 1 - enabled,
Response Structure Meaning of the Columns	1 Query whenabled. BR_EXD (1) = SF0 Note: The follo command 1 = Enable Query when	1 nether the o 2!(1) C entity nan The sepa wing table t. Line led	1 condition analy ne rator "!" is used shows the (1 [1 [1 0 condition anal	(Single Read) [ASCII characters] d in this command. general structure of the "EXD2" FI Column 1 - enabled, – not enabled]
Response Structure Meaning of the Columns	1 Query whenabled. BR_EXD (1) = SFC Note: The follocommand 1 = Enable Query whenabled.	1 nether the o 2!(1) C entity nan The sepa wing table t. Line led	1 condition analy ne rator "!" is used shows the (1 [1 [1 0 condition anal	(Single Read) [ASCII characters] d in this command. general structure of the "EXD2" FI Column 1 - enabled, - not enabled] ysis of the "clamp" chain has been
Response Structure Meaning of the Columns	1 Query whenabled. BR_EXD (1) = SF0 Note: The follo command 1 = Enable Query whenabled. FI command	1 ether the o 2!(1) C entity nan The sepa wing table I. Line Ied	1 condition analy ne rator "!" is used shows the (1 [1 [1 0 condition anal 00_BR_EXD2!	(Single Read) [ASCII characters] d in this command. general structure of the "EXD2" FI Column 1 - enabled, - not enabled] ysis of the "clamp" chain has been

Module Assignment of a Process: MAP

MWAX device group

				5 1		
Designation	MAP	Modul As	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]:\Programme\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 out from the module configuration of the MWAX device group.					
	BR_MAF		(Single Read)			
	BC_MAF	,	(Cyclic Read)			
				n.		
	BB_MAF	_, ,	(Break Cyclic Read			
	1 = Mech	nanism num	ber	[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.					
		Lin	e 1	Column 1		
Value Range of the Column	1 = Mod	ule number		[099]		
Example MAP1	Read the module number which is assigned to NC process number 0 from the module configuration. <u>Assumption:</u> The module to which NC process 0 is assigned has module number 5.					
	FI comm		00_BR_MAP1_0	<u> </u>		
	Line	Column		Answer		
	1	1	5			
	1	I	5			
Reference to Literature	See chap	oter entitled	"Literature" [36].			
ad Reference Name	of a PL	.C Varia	ble: MAR			
				MWAX device group		
Designation	MAR	Map Abs	olute PCL- R eference	9		
PLC Explanation	The abso	lute referen	ce name of a symbol	ic PLC variable is read out.		
FI command	Read the absolute reference name of a PLC variable.BR_MAR_(1)(Single Read)					
Response Structure	command	d "MAR". O		ucture of the response to the FI umn is output for the reference		
		Lin	e 1	Column 1		
Meaning of the Column	1 – Ideni	tifier of the I	PLC variable			
meaning of the colulin						

Read



PLC – Example MAR	Read the absolute reference name of the PLC variable with the identifier "abref" at device address 00.			
	Assumpti The PLC		th the identifier "abref" is of the type "INTEGER".	
	FI comma	and	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 re	ference name of a WinPlc PLC variable.	
	BR_MAR1_(1) (Single Read)			
	(1) = Identifier of the PLC variable			
WinPLC - Example MAR1	Read the absolute reference name of the WinPLC variable with the identifier "Prog.abref" at device address 00.			
	Assumption: The WinPLC variable with the identifier "Prog.abref" is of the type "INTEGER" and is present in WinPLC program "Prog".			
	FI comma	and	00_BR_MAR1_:Prog.abref	
	Line	Column	Answer	
	1	1	%M100.0	
	0		"I iteratura" [20]	

Reference to Literature See chapter entitled "Literature" [30].

Device Data of the Module Configuration: MCD

Designation	MCD	Modul Configuration: Device Information						
Explanation	"Modulde "[LW]:\Pro standard	f.ini" fi ogramme\Ind installation.	le wh dramat\MT The devic		stored ita\Resource ne sections	out from the in the ce" directory on [DeviceAddrX],		
FI command	Read-out device data within the module configuration of the MWAX device group. BR_MCD1 (Single Read)							
	BC_MC)1	(Cyclic Re	ead)				
	BB_MC)1	(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	C	olumn 1		Column 4	
Value Range of the Columns	1 = Device addres	S		[015]		
	2 = PLC-FB name	for the setup	diagnostics	[max. 9 AS	CII characters]	
	3 = PLC-FB name	for the warni	ng messages	[max. 9 AS	CII characters]	
	4 = PLC-FB name	for the start	requirements	[max. 9 AS	CII characters]	
Example MCD1	Read all device dat	a of the mod	ule configurat	ion		
	Assumption: The following devic	es have beer	n configured ir	n 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	3 PVStart_3	
Poforonco to Litoraturo	See chanter entitle	d "Literature"	[36]			

Reference to Literature

See chapter entitled "Literature" [36].

Module Data of the Module Configuration: MCM

MWAX device group

- **Designation MCM** Modul Configuration: Modul Information
- Explanation All module data for module configuration is read out from the "Moduldef.ini" file which is stored in the "[LW]:\Programme\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 2 = Module name 3 = PLC-FB name for g errors	jeneral module	•	CII characters] CII 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:

<u>Assumption:</u> The following modules have been defined:

- Module number 5
- Module number 7

FI commar	nd	03_E	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].

Process Data of the Module Configuration: MCP

MWAX device group

Designation	МСР	Modul C	onfiguration: P roc	cess Information	
Explanation	All process data of a certain module is read out from the "Moduldef.ini" file which is stored in the "[LW]:\Programme\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_MCP	P1_(1)	(Single	Read)	
	BC_MCP	1_(1)	(Cyclic I	Read)	
	BB_MCP	1_(1)	(Break C	Cyclic Read)	
	(1) = Moo	dule numbe	r [099]		
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 = Mech	nanism num	ber [0]		
Example MCP1	Read the configurat		ss number of mo	odule 5 of device 03 of the module	
	Assumption		cesses are defin	ed:	
	NC process number 0				
	FI command 00_BR_MCP1_				
	Line	Column	Answer		
	1	1	1		
	2	1	0		
	•		"Litoroturo" [26]		

Reference to Literature See chapter entitled "Literature" [36].

SFC Data of the Module Configuration: MCS

MWAX device group

Designation	MCS Modul Configuration: SFC Information			
Explanation	All SFC data of a certain module is read out from the "Moduldef.ini" file which is stored in the "[LW]:\Programme\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				espect to the module of a device from the MWAX device group.
	BR_MCS	•		Single Read)
	BC_MCS	• •		Cyclic Read)
	BB_MCS	• •	•	Break Cyclic Read)
		ule number	•)99]
Response Structure Value Range of the Column	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. 1 = Name of the Indrastep step chain [format W.X.Y.Z]			
		mat W.X.Y.	•	Value Range
	101	W	2	Max. 9 ASCII characters
		X		Max. 9 ASCII characters ! OPTIONAL !
		Y		Max. 9 ASCII characters ! OPTIONAL !
		Z		Max. 9 ASCII characters ! OPTIONAL !
Example MCS1	the modul	name of th e configura		step step chain of module 5 from device 03 of
	Assumption		ep step	chains have been defined:
	 ISFB_ 	•		
		S.ISFB_3		
		S.ISFB_3.S	W1	
	• FB_US	S.ISFB_3.S	W1.AB	3A
	FI comma	and	03_BR	_MCS1_5
	Line	Column	Answe	r
	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 chap	ter entitled	"Literatu	

Reference to Literature See chapter entitled "Literature" [36].

Downloading Message Texts: MFD

MWAX device group

Designation	MFD	Message Files Download				
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 	n error mes	sages			
	 transm 	nission erro	r messages	5		
	mecha	anism mess	sages			
	Note:	This FI co	ommand is	an FI job!		
	BW_MFD	1		(Single Write)		
Response Structure				command consists of three lines, each with elements is as follows:		
	Line 1 = Job ID			[0120] (refer to chapter entitled "FI Commands for the MPCX Device Group", IFJ).		
	Line 2 =	FI comman	ld	[string, in accordance to chapter entitled "Elements of the FI Command"]		
	Line 3 =	Line 3 = FI job error code (see chapter entitled "Error Codes")				
Example MFD1	Load mes	sage texts	into the dev	vice with device address 00.		
	FI comma	and	00_BW_M	FD1		
	Line	Column	Answer			
	1	1	01			
	2	1	00_BW_M	FD1		

Reading Machine Key Information : MKS

3

1

0

MKS	Machine Key Status				
Currei	nt machine key information can b	e read for the selec	ted device.		
Read	machine key information for se	nformation for selected device.			
BR_M	IKS (Single Read)	(Single Read)			
BC_N	IKS (Cyclic Read)	(Cyclic Read)			
BB_N	IKS (Break Cyclic	Read)			
The following table shows the general structure of the response to the FI command "MKS".					
Line 1		Column 1	Column 2		
1 = 2 =	Machine key information Information valid?	[4 byte in HEX coding] [0 = not valid, 1=valid]			
	Curren Read BR_W BC_W BB_W The for comm	Current machine key information can be Read machine key information for set BR_MKS (Single Read) BC_MKS (Cyclic Read) BB_MKS (Break Cyclic The following table shows the generic command "MKS". 1 = Machine key information	Current machine key information can be read for the select Read machine key information for selected device. BR_MKS (Single Read) BC_MKS (Cyclic Read) BB_MKS (Break Cyclic Read) The following table shows the general structure of the command "MKS". Line 1 Column 1 1 = Machine key information		



Value

Example MKS

KS Read the current machine key information for device 0.

FI command		00_BR_MKS
Line	Column	Answer
1	1	0000000
	2	0

Writing the GUI-SK Block: MKT

Designation	MKT Machine Key Table					
Explanation	Writes the GUI-SK16 block in the PLC.					
FI command	Write GUI-SK16 block.					
	BW_MK	T1_(1)		(Single Write)		
	 (1) = List of the 48 PLC variables for writing the GUI-SK16 block. 		writing the	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		
Range/Meaning of the Columns	1 = Successfully completed (P_ACK)					
1. Example MKT1	1.Clear GUI-SK16 block: 00_BW_MKT1 FI command Value to be written: \$EMPTY					
	Line	Column	Answer			
	1	1	(P_ACK)			
2. Example MKT1	Write GUI-SK16 block: 00_BW_MKT1 Value to be written: \$EMPTY SPSVAR1,SPSVAR2,\$SPACE,					
				written: \$EMPTY		
	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 resul (column 2).					
	BW_MK	T2_(1)	(Single	e Write)		
PLC variables for writing the GUI-			s for 1. Dele JI- BW_ 2.Write the 4 \$SPA BW_	lowing cases are to be differentiated: te the GUI-SK-16 block: MKT2 \$EMPTY the GUI-SK16 block with 8 PLC variables, filling gaps with ACE: MKT1 SPSVAR1,SPSVAR2, ACE,\$SPACE,		



Response Structure	Afte	rsuco	cessful tran	smission, or	ne line with two co	olumns is returned.
			Line 1		Column 1	Column 2
Value Range/Meaning of Columns	1 = Status report		PLC	[0 = at least 1 PLC variable in the current PLC program is NOT defined 1 = ALL PLC variables could be written]		
	2 =	d va ci	st of the NC efined PLC ariables in t urrent PLC rogram	else he not	e list of the PLC	les could be written, or variables that could ndividual PLC variables omma.
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 Line Column		00_BW_MK Value to be SPSVAR1,S		AR48	
			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.	Varia	bles 1 - 16:	Machine	function keys	
	2.	Varia	bles 17 - 32	: Status p	ressed	
	3. Variables 33 - 48: Status shining					
	t		telegram	will contain	only these 8 PLC	M keys are used, the variables. The other 40
						ansmission parameter. left unused, they must be

Read System Messages: MSG

Designation	MSG	MeSsaGe
Explanation	Reading c	f system messages
FI command	Message CC_MSG (1) = SYS	G-(1) (Cyclic Read) G-Message number
	Note:	Exists only as a cyclic command
Response Structure	The respo data.	onse of the FI command 'MSG' consists of the system message



Example MSG	00_CC_N	/ISG_64	(64 = N	ISG_SYSERRGEN)
	FI comm	and	00_CC_M	SG_64/3
	Line	Column	Answer	
	1	1	00	
Restriction	• The f	ollowing sys	stem mess	ages:
	SYS Mes	sage		SYS Message number
	MSG_PC			52
	MSG_PA	RUPDBEG		24
	MSG_FW	AUPDBEG	3	82
	These co	mmands ca	annot be us	ed with the following programs:
		nat OPC se		
		nat DDE se		
Dooding the Firmwore	Idontifi	ootion	мтс	
Reading the Firmware	identin	cation.		
				MWCX device group
Designation	МТС	MT-CNC	Slot Softw	vare Version
FI command		mand is us omponents		the firmware identification from the various ers).
FI command		For the communi	(slot numbe time this F cation inte	
FI command FI command	control co	For the communi	(slot numbe time this F cation inte , etc.) are s	ers). FI command is executed, the internal FI erlocks (fast timeout monitoring, offline switched off.
	Control co Note: BR_MTC	For the communi	(slot numbe time this F cation inte a, etc.) are s	ers). FI command is executed, the internal FI erlocks (fast timeout monitoring, offline
	Control co Note: BR_MTC (1) = Slo The follow	For the	(slot number time this F cation inter , etc.) are s	ers). FI command is executed, the internal FI erlocks (fast timeout monitoring, offline switched off. (Single Read) [1=CNC, 2=SIO, 3=PLC, 4=APR1
FI command	Control co Note: BR_MTC (1) = Slo The follow	For the	(slot number time this F cation inter , etc.) are s	FI command is executed, the internal FI erlocks (fast timeout monitoring, offline switched off. (Single Read) [1=CNC, 2=SIO, 3=PLC, 4=APR1 5=APR2, 6=APR3, 7=APR4] general structure of the response to the FI

of Columns 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

Parameter Download: PAA

Designation	PAA	PA ramet	ter Access		
Explanation	Complete parameter records are downloaded by means of a download file.				
FI command	Parameter download command whereby the parameter download file i directly indicated. BW_PAA2_(1) (Single Write)				
	(1) = Cor	nplete para	ameter download file name		
Response Structure	The response to the "PAA2" FI command consists of three lines, e with one column. The meaning of the elements is as follows:				
	(see	•	[0120] FI Commands for the MPCX Device Group", IFJ).		
		2 = FI comn g, in accord	nand Jance with Chapter "Elements of the FI Command"]		
		8 = FI Job E Chapter "Er	Error Code rror Codes")		
	Note:	File and p	bath details must be enclosed in inverted commas.		
Example PAA2	00_BW_F	PAA2_"D:\D	DOWNLOAD.DAT"		
	FI comma	and	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	file. Indra V20_Para	mat's own c am_08_Defi nded for a r	download file corresponds to that of a Windows Ini description in initions_Parameter_Download_01.doc is more detailed account of the structure of the		
	Summary	<u>:</u>			
	-	ID_PARAM	IETER] ng parameter identification.		
	-	ID_SYSTE	M] ng system parameter identification.		
		DATA_SYS	STEM] rameter data.		
		ID_PROCE	ESSX] ng process parameter identification.		
	-	DATA_PRO process pa	OCESSX] Irameter data.		
	-	ID_AXISX]	ng axis parameter identification.		
	-	DATA_AXI axis param	-		



ProVi Diagnosis Data: PDD

Designation	PDD	Provi D	iagnosis D ata	l					
Explanation	Data for ProVi criteria analysis is output.								
Explanation FI command Response Structure Meaning of the Columns	Output of BR_PDE (1) = Mes (2) = Mes (3) = Mod	files to inc o1_(1)_(2) ssage ID ssage type dule numb owing tab d. e 1	dicate the deta	ail in th (S [A [1 10 12 [1. e gen	he edito ingle R SCII cha = error, 0 = warr 1 = start 2 = setu 99] ! o neral s	ead) aracters] 2 = mess nings, t requirem p diagnos only for me structure	nents, sis] essage of the		FI
J	2 = Deta 3 = Error 4 = POU	il morpher	ne	[AS [AS [AS	CII char	acters] (E acters] (E acters]			,
Example PDD1	Indication of data of a ProVi error with ID 43923028 from module control unit 0.			3 in					
	FI comm	and		1 430	023028	1 1			
	FI comma		00_BR_PDD	01_439	923028_	1_1			
	FI comma Line	and Column 1	Answer		923028_	1_1			
	Line	Column			923028_	1_1			
	Line	Column 1 2	Answer STATION_1 98243823		923028_	.1_1			
	Line	Column 1 2 3	Answer STATION_1 98243823 34985304	_2	923028_	.1_1			
	Line	Column 1 2 3 4	Answer STATION_1 98243823 34985304 Station2.Mod	_2	923028_	.1_1			
	Line 1	Column 1 2 3 4 5	Answer STATION_1 98243823 34985304 Station2.Mod 43493454	_2 dule3					
FI command	Line 1 Output the	Column 1 2 3 4 5 e I/O addr	Answer STATION_1 98243823 34985304 Station2.Mod 43493454	_2 dule3 ay a c	detail.				
FI command	Line 1 Output the BR_PDD	Column 1 2 3 4 5 e I/O addr 02_(1)_(2)	Answer STATION_1 98243823 34985304 Station2.Mod 43493454	_2 dule3 ay a c (S	detail. ingle R	ead)			
FI command	Line 1 Output the BR_PDD (1) = Mes	Column 1 2 3 4 5 e I/O addr 02_(1)_(2)	Answer STATION_1 98243823 34985304 Station2.Mod 43493454 esses to displ {_(3)}	_2 dule3 ay a c (S [A: [1 1(1)	detail. ingle R SCII cha = error, 0 = warr 1 = start	ead) aracters] 2 = mess	nents,		
FI command	Line 1 0utput the BR_PDE (1) = Mes (2) = Mes	Column 1 2 3 4 5 e I/O addr 02_(1)_(2) ssage ID	Answer STATION_1 98243823 34985304 Station2.Mod 43493454 esses to displ {_(3)}	_2 dule3 ay a c (S [A [1 10 11	detail. ingle R SCII cha = error, 0 = warr 1 = start 2 = setu	ead) aracters] 2 = mess nings, t requirem	nents, sis]		2!
FI command	Line 1 0utput the BR_PDD (1) = Mes (2) = Mes (3) = Mod	Column 1 2 3 4 5 e I/O addr 02_(1)_(2) ssage ID ssage type dule numb	Answer STATION_1 98243823 34985304 Station2.Mod 43493454 esses to displ {_(3)}	_2 dule3 ay a c [A [1 10 12 12 [1.	detail. ingle R SCII cha = error, 0 = warr 1 = start 2 = setu 99] ! o	ead) aracters] 2 = mess nings, t requiren p diagnos	nents, sis] essage		
	Line 1 Output the BR_PDE (1) = Mes (2) = Mes (3) = Moo The follo command	Column 1 2 3 4 5 e I/O addr 02_(1)_(2) ssage ID ssage type dule numb	Answer STATION_1 98243823 34985304 Station2.Mod 43493454 esses to displ {_(3)} e ber le shows th	_2 dule3 ay a c [A [1 10 12 12 [1.	detail. ingle R SCII cha = error, 0 = warr 1 = start 2 = setu 99] ! o neral s	ead) aracters] 2 = mess nings, t requiren p diagnos	nents, sis] essage	PDD2	



Example PDD2	Query of the I/O addresses of a ProVi error with ID 43923028 from module 3 in control unit 0.				
	Three va	riables have	e an I/O addre	SS.	
	FI comm	and	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	%1100.3.5		
FI command	Determin	e the multil	ingual comme	nts for displayin	a a detail.
		D3_(1)_(2){	-	(Single Read	-
		ssage ID	_(//	[ASCII charac	
	(1) = Message type (2) = Message type [1 = error, 2 = messages, 10 = warnings, 11 = start requirements, 12 = setup diagnosis]				s, juirements,
	(3) = Mo	dule numbe	er	[199] ! only 1	for message type 1 -2!
Response Structure	The follo		e shows the	general struc	ture of the PDD3 FI
	L	ine 1-n	C	olumn 1	Column 2
Meaning of the Columns		ment morp comment	-	ASCII characters	s] (DWORD, decimal) s]
Example PDD3		the comme lule 3 in cor		tion of a ProVi	error with ID 43923028
	Two com	ments are i	replaced by an	other text.	
	FI comm	and	00_BR_PDD3	43923028_1_1	
	Line	Column	Answer		
	1	1	98243823		
		2	Clamp open		
	2	1	40923423		
		2	Clamp closed		
FI command	Querv of	the status o	of a certain me	ssage	
	•	D4_(1)_(2){		(Single Read)
		ssage num		[ASCII charac	
	(2) = Me	ssage type		[1 = error, 2 = 10 = warnings 11 = start pred 12 = setup dia	s, conditions,
	(3) = Mo	dule numbe	er	[199] ! only 1	for message type 1 -2!
Response Structure	(3) = Module number [199] ! only for messagThe following table shows the general structure of the				
	The follo	-	e shows the	general struc	ture of the PDD4 FI

Line 1-n Column 1	Column 2
-------------------	----------

2 = Criteria analysis exists [YES, NO] Example PDD4 Query of the status of a ProVi error, number 1001 from module control 0. This message is not present at the moment, and there is a crianalysis. FI command 00_BR_PDD4_1001_1_1 Line Column 1 1 2 YES
control 0. This message is not present at the moment, and there is a crianalysis. FI command 00_BR_PDD4_1001_1_1 Line Column 1 1 NO
Image: state
Line Column Answer 1 1 NO
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 [199] ! only for message type 1
Note: The separator "!" is used in this command.
Response Structure The following table shows the general structure of the PDDS command.
Line 1-n Column 1 Column
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 module 25.40 mm control 0.
<u>Assumption:</u> This message is not present at the moment, and there is a cri analysis.
FI command 00_BR_PDD5!Station2.Modul3!43493454!1001!1!1
Line Column Answer
1 1 240872342
2 NO
3 YES

Generating Physical Directory Names: PHD

					0 1		
Designation	PHD PHysical Directory						
Explanation	Generate	s physical c	directory names	according to th	e BDI data written.		
	Note: This is based on BDI philosophy.						
FI command	Generate	Generate physical directory names.					
	BR_PHD	01_(1)_(2)_	(3)_(4)_(5)_(6)	(Single Writ	e)		
	(1) = Pro	ject ID			CT_NEUTRAL CT_DEFAULT]		
	(2) = Section ID			3=SECT_O 4=SECT_C	IN ASIC_DATA		
	(3) = Device address			[-1= DEVAD otherwise th device addre			
	(4) = Process ID			[-1= PROCESS_NEUTRAL otherwise the required process number]			
	(5) =Data	a 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 follow		shows the gener	al structure of	the response to the FI		
		Lir	ne 1		Column 1		
Value Range/Meaning of Columns	1 = Physical directory [complete physical directory name name accordance with the BDI data writt						
Example PHD1	Requestir	ng the physi	ical directory nar	ne for:			
·	Requesting the physical directory name for: PROJECT_NEUTRAL SECT_BIN DEVADDR_NEUTRAL PROCESS_NEUTRAL DATATYPE_NEUTRAL LANG_NEUTRAL						
	FI comma	and	XX_BR_PHD1	1_011_0_0			
	Line	Column	Answer				
	1	1	D:\Programme\Ir	\Indramat\Mtgui\Bin			



Line 16

Actual (Current) Information on Position of All Axes: POI

MWAX device group

Designation	POI	POsition Information					
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_	<u>(</u> 1){_(2)}	(Single Read)				
	BC_POI1_	_(1){_(2)}	(Cyclic Read)				
	BB_POI1_	<u>(</u> 1){_(2)}	(Break Cyclic Read)				
	(1) = updat	ted position information	[031, 1 = axis has been homed 2 = machine coordinates 4 = program coordinates 8 =relative coordinates 16 = distance to go] all combinations are possible!				
	(2) = Req (opt.)	uired measurement system	[mm, inch]				
Response Structure			ructure of the response to the FI is 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 1Column 1Column 2Column 3Column 4Column 5Column 6Column 7Column 8Line 2..............................

Value range/Meaning of lines	Line = axis meaning	$\begin{bmatrix} 1 = A \text{ axis, } 2 = X \text{ axis} \\ 3 = Z \text{ axis, } 4 = Y \text{ axis} \\ 5 = B \text{ axis, } 6 = C \text{ axis} \\ 7 = D \text{ axis, } 8 = E \text{ axis} \\ 9 = X' \text{ axis, } 10 = Y' \text{ axis} \\ 11 = P \text{ axis, } 12 = Q \text{ axis} \\ 13 = R \text{ axis, } 14 = U \text{ axis} \\ 15 = V \text{ axis, } 16 = W \text{ axis} \end{bmatrix}$
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.

	FI command	0	0_BR_POI1_	_22				
	Answer							
Line	Column 1	Column 2	Column 3	Column 4	Column 5	Column 6	Column 7	Column 8
1	0							
2	1	Х		148.0000	123.0000		0.0000	[mm]
3	1	Z		23.4548	56.0000		0.0000	[mm]
4	1	Y		0.0000	0.0000		0.0000	[mm]
5	2	В		180.0000	180.0000		16.0000	[deg]
6	2	С		270.0000	90.0000		0.0000	[deg]
7	0							
8	0							
9	4	X'		0.0000	0.0000		0.0000	[%]
10	0							
11	0							
12	0							
13	0							
14	0							
15	0							
16	0							

Issuing SYS Messages Specific to the PCL: PSM

MWAX device group

Designation PSM PCL Sys Message

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.
- 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	1		Column 8
Value Range/Meaning of Columns	1 =	Status report		orrec /IN32 ERRC een a	PY=SYS messa tly acknowledge 2 applications] DR=SYS messa acknowledged b ation within the	ed by the lige has NOT by a WIN32
	2 =	Task name (LogIn			name that has t nessage]	riggered the
	3 =	SYS message nu	-		ins the issued nessage number	er]
	4 =	Acknowledgeme	-		ins the pre-set wledgement tim	e]
	5 =	Reference inform	a	dditio	ins, where appli nal information rite value]	
	6 =	Length of the reference information	-		re NO additionant	
	7 =	Where applicable channel of the FI NOT acknowledg	that has co ged ch ap	omple hanne pplica	cknowledgemer eted in time or t el number of the ation that has <u>N</u> wledged in time	he LOG e WIN32 <u>OT</u>
	8 =	Where applicable name that has N acknowledged in	OT continue. th	omple	cknowledgemer eted in time or t as NOT acknow	he task name

Example PSM1 Issue the SYS message Beginning PCL Download. The additional information, device address 00, is also transferred as a write value.

FI comma	and	XX_BW_PSM1_1 – value to be written: 00
Line	Column	Answer
1	1	READY
	2	WINPCL.EXE
	3	14
	4	30000
	5	00
	6	2
	7	
	8	



Edit PROVI Message Files: PVA

Designation	PVA	P RO V I-N	/lessa	ges Access		
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 Writ	e)	
	Note:	This com	mand	is an FI job comr	mand.	
Value to be written	No write	value exists	5	PROVI mes current PLC	sage files acco project.	rding to the
	Write val	ue exists		files (separa to the forma [PROVI-Dia	equested PROV ited by a comm t: g-type: module I:01,01:02,02:0	a) according number]
	Note:			e written is passe ue in the "DataTi		
Response Structure	each with • Line ²	one colum I = Job ID	n. The [01	2_PVA1" FI com meaning of the 20]	elements is as i	follows:
	 (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") 			I Command"]		
Example PVA1		value is pa		i.e. the PROVI C project.	message files	are generated
	FI comma	and	00_B	W_PVA1		
	Line	Column	Answ	ver		
	1	1	01			
	2	1	00_B	W_PVA1		
	3	1	0			
Explanation		l command ROVI mess		rns the most s les.	ignificant infor	mation on the
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.					
	10 columi	ns each is c	created	d.		-, -



Value Range/Meaning of Columns	1 = 2 =	PROVI diagnostic type PROVI diagnosis type designation	[120] [The following designations can be returned: StartCondition, Error, Message, War- ning, Setup]
	3 =	Module number	[199]
	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 comma	and	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:\Programme\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



	FI comm		00_BR_PVA1_1
	Line	Column	Answer
		7	D:\Programs\indramat\Mtgui\Project_000\ Programmdata\Device_000\Diag\De\ERROR 1.TXT
		8	456
		9	2912
		10	4491
Explanation	device. T	hrough the PROVI m	d transmits PROVI message files into the selected write value, it is possible to chose whether ALL or nessages selected via the write value are to be
FI command	BW_PVA	2	(Single Write)
	Note:	This com	mand is an FI job command.
Value to be written	No write	value exists	s 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:		e to be written is passed to the "acValue" parameter CII value in the "DataTransfer" routine.
Response Structure	The resp	onse to th	e "BW_PVA2" FI command consists of three lines,
	each with	1 = Job ID	n. The meaning of the elements is as follows: [0120]
	each with • Line (see	1 = Job ID Chapter "F	n. The meaning of the elements is as follows: [0120] FI Commands for the MPCX Device Group", IFJ).
	each with • Line (see • Line 2	1 = Job ID Chapter "F 2 = FI comr	n. The meaning of the elements is as follows: [0120] FI Commands for the MPCX Device Group", IFJ). nand
	each with Line (see) Line 2 [Strin 	1 = Job ID Chapter "F 2 = FI comr g, in accord	n. The meaning of the elements is as follows: [0120] FI Commands for the MPCX Device Group", IFJ). nand dance with Chapter "Elements of the FI Command"]
	each with Line (see) Line 2 [Strin 	1 = Job ID Chapter "F 2 = FI comr g, in accord	n. The meaning of the elements is as follows: [0120] FI Commands for the MPCX Device Group", IFJ). nand
Example PVA2	each with Line (see) Line 2 [Strin Line 3 	1 = Job ID Chapter "F 2 = FI comr g, in accord 3 = FI Job E value is	n. The meaning of the elements is as follows: [0120] FI Commands for the MPCX Device Group", IFJ). nand dance with Chapter "Elements of the FI Command"]
	each with • Line (see • Line 2 [Strin • Line 3 No write	1 = Job ID Chapter "F 2 = FI comr g, in accord 3 = FI Job E value is ed.	n. The meaning of the elements is as follows: [0120] FI Commands for the MPCX Device Group", IFJ). mand dance with Chapter "Elements of the FI Command"] Error Code (see Chapter "Error Codes")
	each with • Line (see • Line 2 [Strin • Line 3 No write transmitte	1 = Job ID Chapter "F 2 = FI comr g, in accord 3 = FI Job E value is ed.	 n. The meaning of the elements is as follows: [0120] FI Commands for the MPCX Device Group", IFJ). nand dance with Chapter "Elements of the FI Command"] Error Code (see Chapter "Error Codes") passed, i.e. all PROVI message files should be
	 each with Line 2 (see Line 2 [Strin Line 3 Mo write transmitte FI comments 	1 = Job ID Chapter "F 2 = FI comr g, in accord 3 = FI Job E value is ed. and	n. The meaning of the elements is as follows: [0120] FI Commands for the MPCX Device Group", IFJ). mand dance with Chapter "Elements of the FI Command"] Error Code (see Chapter "Error Codes") passed, i.e. all PROVI message files should be 00_BW_PVA2
	each with Line '	1 = Job ID Chapter "F 2 = FI comr g, in accord 3 = FI Job E value is ed. and Column	 n. The meaning of the elements is as follows: [0120] FI Commands for the MPCX Device Group", IFJ). nand dance with Chapter "Elements of the FI Command"] Error Code (see Chapter "Error Codes") passed, i.e. all PROVI message files should be 00_BW_PVA2 Answer



Formatted Input / Output of PLC Variables: PVF

Designation	PVF	PLC Variable Formattee	d	
Explanation	Formatted	reading and writing of PL	C variables, arrays	and structures.
FI command	Read PLC variables.			
	CR_PVF_(1)	(Single Read)	
	CC_PVF_(1)	(Cyclic Read)	
	CB_PVF_(1)	(Break Cyclic Re	ad)
	(1) = Identi	fier of the PLC variable	[acc. to declaratio	n 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 1n:	Colur	nn 1
	n = number	r of elements.		
		Only defined PLC va Addressing a non-declar A PLC variable can on exceed 240 byte. (Refer "Guidelines").	ed variable results i ly be read if its da	n an error message. ata length does not
Value Ranges ANSI / ASCII	The value r read. The f	ollowing table indicates		
Value Ranges ANSI / ASCI	read. The f to be expective type this str	ollowing table indicates cted when reading out a ring can be converted wit	the range in which single variable an	the results string is d into which C-data ation: Can be converted
Value Ranges ANSI / ASCI	read. The f to be expective this str Data Type	ollowing table indicates cted when reading out a ring can be converted wit Value Range	the range in which single variable an	the results string is d into which C-data ation: Can be converted to C-data type
Value Ranges ANSI / ASCI	read. The f to be expect type this str Data Type BOOL	ollowing table indicates cted when reading out a ring can be converted wit Value Range [0;1]	the range in which single variable an	the results string is d into which C-data ation: Can be converted to C-data type unsigned char
Value Ranges ANSI / ASCI	read. The f to be expective this str Data Type	ollowing table indicates cted when reading out a ring can be converted win Value Range [0;1] [-128127]	the range in which single variable an	the results string is d into which C-data ation: Can be converted to C-data type
Value Ranges ANSI / ASCI	read. The f to be expert type this str Data Type BOOL SINT	Value Range [0;1] [-128127] [-3276832767]	the range in which single variable an thout loss of inform	the results string is d into which C-data ation: Can be converted to C-data type unsigned char char short
Value Ranges ANSI / ASCI	read. The f to be expert type this str Data Type BOOL SINT INT	Value Range[0;1][-128127][-3276832767][2147483648214748364	the range in which single variable an thout loss of inform	the results string is d into which C-data ation: Can be converted to C-data type unsigned char char short long
Value Ranges ANSI / ASCI	read. The f to be expert type this str Data Type BOOL SINT INT DINT	iollowing table indicates cted when reading out a ring can be converted with Value Range [0;1] [-128127] [-3276832767] [2147483648214748364 [0255]	the range in which single variable an thout loss of inform	the results string is d into which C-data ation: Can be converted to C-data type unsigned char char short long unsigned char
Value Ranges ANSI / ASCI	read. The f to be expert type this str Data Type BOOL SINT INT DINT USINT	iollowing table indicates cted when reading out a ring can be converted with Value Range [0;1] [-128127] [-3276832767] [2147483648214748364 [0255] [065535]	the range in which single variable an thout loss of inform	the results string is d into which C-data ation: Can be converted to C-data type unsigned char char short long unsigned char unsigned short
Value Ranges ANSI / ASCI	read. The f to be expert type this str Data Type BOOL SINT INT DINT USINT UINT	iollowing table indicates cted when reading out a ring can be converted wit Value Range [0;1] [-128127] [-3276832767] [2147483648214748364 [0255] [04294967295]	the range in which single variable an thout loss of inform	the results string is d into which C-data ation: Can be converted to C-data type unsigned char char short long unsigned char unsigned short unsigned long
Value Ranges ANSI / ASCI	read. The f to be expert type this str Data Type BOOL SINT INT DINT USINT UINT UDINT	iollowing table indicates cted when reading out a ring can be converted with Value Range [0;1] [-128127] [-3276832767] [2147483648214748364 [0255] [065535]	the range in which single variable an thout loss of inform	the results string is d into which C-data ation: Can be converted to C-data type unsigned char char short long unsigned char unsigned short
Value Ranges ANSI / ASCI	read. The f to be expert type this str Data Type BOOL SINT INT DINT USINT USINT UDINT BYTE	Value Range [0;1] [-128127] [-3276832767] [2147483648214748364 [0255] [065535] [00xFF]	the range in which single variable an thout loss of inform 47]	the results string is d into which C-data ation: Can be converted to C-data type unsigned char char short long unsigned char unsigned short unsigned long unsigned char
Value Ranges ANSI / ASCI	read. The f to be expective this still Data Type BOOL SINT INT DINT USINT UINT UDINT BYTE WORD	iollowing table indicates cted when reading out a ring can be converted wit Value Range [0;1] [-128127] [-3276832767] [2147483648214748364 [0255] [065535] [04294967295] [0x000xFF] [0x00000xFFF]	the range in which single variable an thout loss of inform 47]	the results string is d into which C-data ation: Can be converted to C-data type unsigned char char short long unsigned char unsigned short unsigned long unsigned char unsigned char unsigned short
Value Ranges ANSI / ASCI	read. The f to be expert type this still Data Type BOOL SINT INT USINT USINT UDINT UDINT BYTE WORD DWORD;	iollowing table indicates cted when reading out a ring can be converted wit Value Range [0;1] [-128127] [-3276832767] [2147483648214748364 [0255] [04294967295] [0x000xFF] [0x00000000xFFFFFF	the range in which single variable an thout loss of inform 47]	the results string is d into which C-data ation: Can be converted to C-data type unsigned char char short long unsigned char unsigned short unsigned long unsigned short unsigned long unsigned long unsigned long unsigned long
	read. The f to be expective this str Data Type BOOL SINT INT DINT USINT UINT UDINT BYTE WORD DWORD; TIME	iollowing table indicates cted when reading out a ring can be converted wit Value Range [0;1] [-128127] [-3276832767] [2147483648214748364 [0255] [065535] [04294967295] [0x0000xFF] [0x0000000xFFFFF [04294967295]	the range in which single variable an thout loss of inform 47] FF] s a character string ny characters as	the results string is d into which C-data ation: Can be converted to C-data type unsigned char char short long unsigned char unsigned short unsigned long unsigned long unsigned long unsigned long unsigned long (msec)



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 н7F н] i.e. –128127	1
INT	[8000 _H (-32768)7FFF _H (32767)]	2
DINT	[80000000 н (-2147483648) 7FFFFFF н (2147483647)]	4
USINT	[00 _H (0)FF _H (255)]	1
UINT	[00 н (0)FFFF н (65535)]	2
UDINT	[04294967295]	4
BYTE	[0x000xFF]	1
WORD	[0x00000xFFFF]	2
DWORD;	[0x00000000xFFFFFF]	4
TIME	[04294967295]	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</string></string>	XX+1
REAL	[-3.402823567E+383.402823567E+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 comma	and	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 comma	and	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



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 Line Column		00_CR_PVF_:Pr	og.MSTRCT/1
			Answer	
	1	1	0	
	2	1	A	
	3	1	ROBOT AXIS X	
	4	1	2000	
FI command	Write PLC	C variable.		
	CW_PVF	_(1)		(Single Write)
	(1) = Iden	tifier of the	PLC variable	[acc. to declaration part of the PLC]
Value to be written	Value of d	lata elemer	nt	[see value ranges]
	Note:	in the "D		s passed to the "acValue" parameter titine. The data code of the value is 'ValType".
Response Structure			ith a column for been executed	acknowledgement of whether or not successfully.
	(P_ACK) =	= P ositive /	ACKnowledge	Data element has been set
ANSI / ASCII Format into ASCII umlauts. Only ASCII umla deviations to this, please refer to the		NSI umlauts are thereby converted uts are stored in the control unit. For		
	110101	'drill'.		, e.g.
		Special of 1131 by a		e indicated in accordance with DIN-
		The follow	wing are used:	
		• \$'		
		• \$\$	\$	
		• \$R	\r (Car	riage Return)
		• \$L	\n (Line	feed)
		• \$P	\f (Forr	n feed)
		• \$T	∖t (Tab)
		• \$xx	hexa	s to a character written as a decimal value. e.g. \$20 (space)
		Array and	d structure eleme	ents 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	is passed	in ANSI fo		XT" at device address 00. The value
	<u>Assumption</u> The "STK		able 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	FI command		00_CW_PVF_:Prog.STK_TXT/3	
L	_ine	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.

I	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
	Т3	STRING[16]
	T4	TIME
h		

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
Т3	STRING[16]
Τ4	TIME

END

		00_CW_PVF_:Prog.MSTRCT.T3/2
		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

52 00 /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
Т3	STRING[16]
T4	TIME

END



To exchange binary data in a C program, the following "C" data type can be used:

I	Fl command Line Column		00_CW_PVF_MSTRCT/2
			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:

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



MWAX device group

ProVi Messages: PVM

Designation	PVM	ProVi Messages	6			
Explanation	ProVi messages are output. These messages are assigned to a particular message type or module.			o a particular		
FI command	Output all F	ProVi messages.				
	BR_PVM1	L_(1){_(2)}	(Singl	e Read)		
	BC_PVM1	L_(1){_(2)}	(Cycli	c Read)		
	(1) = Message type [1 = error, 2 = messages, 10 = warnings, 11 = start requirements, 12 = setup diagnosis]					
	(2) = Mod	ule number	[199] ! only for me	essage type 1	-2!
	Output first	ProVi messages	i.			
	BR_PVM2	2_(1){_(2)}	(Singl	e Read)		
	BC_PVM2	2_(1){_(2)}	(Cycli	c Read)		
	(1) = Mess	sage type	10 = v 11 = s	rror, 2 = mess warnings, start requirem setup diagnos	ents,	
	(2) = Mod	ule number	[199] ! only for me	essage 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, th	e numl	per of lines is	0.	
		Line 1n		Column 1	•••	Column 6
Meaning of the Columns	1 = Messa	ige text	[ASCII charac	ters]	
	2 = Messa	ige number	[ASCII charac	ters]	
	3 = Time s	stamp day	I	[mm.dd.yyyy]		
	4 = Time s	stamp time	[[hh:mm:ss]		
	5 = Messa	ige ID	I	ASCII charac	ters] (DWOF	RD, decimal)
		ence text exists	-	[YES, NO]		
		a analysis exists	-	[YES, NO]		
	8 = Messa	ge HTML file	[ASCII charac	ters]	



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

I 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	[199] ! only for message type 1 -2!
Response Structure	The following table shows command.	s the general structure of the "PVM3"

F١

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 comma	and	00_BR_PVM3	_43923028_1	_3		
	Line	Column	Answer				
		16	3				
		17	NO				
		18					
		19	D:\Programme\Indramat\MtGui\Project_000\ ProgramData\HMTL\DE\Error34.html				
FI command	one line e returned. value car	ach is retur The clock o only be s	, all active ProVi messages are output. In the result, irned. After expiry of the set time, the next message is frequency can be set via the last parameter. This set once for the PC. The value transmitted last is i.e. Default setting is 1 second.				
	BR_PVN	l4_(1){_(2)	_(3)}	(Single Rea	ad)		
	BC_PVN	l4_(1){_(2)	_(3)}	(Cyclic Rea	ıd)		
	(1) = Mes	ssage type	e [1 = error, 2 = messages, 10 = warnings, 11 = start requirements, 12 = setup diagnosis]			,	
	(2) = Moo	dule numbe	er	[199] ! onl	y for messag	e type 1 -2!	
	(3) = Clo	ck frequenc	су	[ASCII char	acters] Time	in ms	
Response Structure	The following table shows the general structure of the "PVM4" FI command.					"PVM4" FI	
	If there ar	e no messa	ages, the num	per of lines is	0.		
		Line 1		Column 1		Column 8	
Meaning of the Columns	1 = Mess	age text		[ASCII characters]			
	2 = Mess	age numbe	er	[ASCII characters]			
	3 = Time	stamp day		[mm.dd.yyyy]			
	4 = Time	4 = Time stamp time			[hh:mm:ss]		
	5 = Message ID [ASCII characters] (DWORD, decimal)			DRD,			
	6 = Addit	ional text a	vailable	[YES, NO]			
	7 = Criter	ria analysis	exists	[YES, NO]			
		age index I. message)	[ASCII char	acters]		
	9 = Mess	age HTML	file	[ASCII characters]			

Example	e PVM1
---------	--------

1 ProVi errors from module 3 in control unit 0.

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			

Download of PLC Retain Variables: PVR

Designation	PVR	PLC Var	iable R etain Backup		
Explanation	Download of PLC retain variables.				
FI command		V_PVR1!(1) (Single Write) = 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 [0120] (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 00_BW_PVR1!"D:\Program Files\Indramat\Mtgui\				
	FI comma	and	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		ture of the ile in Windc	"download.ini" file used in this example corresponds		
	Note:	Care mus	t be taken in the use of upper and lower case letters.		

MWAX device group

Upload of PLC Retain Variables: PVR

Evolonetics						
Explanation	PLC retain variables are uploaded via all active processes.					
FI command	BR_PVR1!(1)(Single Read)(1) = Upload file with path details					
	Note:		ile and path details in inverted commas.			
		The sepa	arator "!" is used in this command.			
Response Structure			e "PVR1" FI command consists of three lines, eac e meaning of the elements is as follows:			
	 Line 1 = Job ID [0120] (see Chapter "FI Commands for the MPCX Device Group", IFJ 					
	(see	Chapter "F	FI Commands for the MPCX Device Group", IFJ).			
	• Line 2	2 = FI comn				
	 Line : [Strin 	2 = FI comr ig, in accord	nand			
Example PVR	Line : [StrinLine :	2 = FI comn ig, in accord 3 = FI Job E	nand dance with Chapter "Elements of the FI Command"]			
Example PVR	Line : [StrinLine :	2 = FI comm Ig, in accord 3 = FI Job E PVR1_"D:\P	nand dance with Chapter "Elements of the FI Command"] Error Code (see Chapter "Error Codes")			
Example PVR	 Line : [Strin Line : 00_BR_F 	2 = FI comm Ig, in accord 3 = FI Job E PVR1_"D:\P	nand dance with Chapter "Elements of the FI Command"] Error Code (see Chapter "Error Codes") rogram Files\Indramat\Mtgui\Temp\Upload.ini"/3 00_BR_PVR1!"D:\Program Files\Indramat\Mtgui\			
Example PVR	Line :: [Strin Line :: 00_BR_F FI comm	2 = FI comn ig, in accord 3 = FI Job E PVR1_"D:\P and	nand dance with Chapter "Elements of the FI Command"] Error Code (see Chapter "Error Codes") rogram Files\Indramat\Mtgui\Temp\Upload.ini"/3 00_BR_PVR1!"D:\Program Files\Indramat\Mtgui\ Temp\upload.ini" /3			
Example PVR	Line :: [Strin . Line :: 00_BR_F FI comm Line	2 = FI comm ig, in accord 3 = FI Job E 2VR1_"D:\P and Column	nand dance with Chapter "Elements of the FI Command"] Error Code (see Chapter "Error Codes") rogram Files\Indramat\Mtgui\Temp\Upload.ini"/3 00_BR_PVR1!"D:\Program Files\Indramat\Mtgui\ Temp\upload.ini" /3 Answer			
Example PVR	Line :: [Strin] Line :: 00_BR_F Fl comm Line 1	2 = FI comn ig, in accord 3 = FI Job E PVR1_"D:\P and Column 1	nand dance with Chapter "Elements of the FI Command"] Error Code (see Chapter "Error Codes") rogram Files\Indramat\Mtgui\Temp\Upload.ini"/3 00_BR_PVR1!"D:\Program Files\Indramat\Mtgui\ Temp\upload.ini" /3 Answer 01 00_BR_PVR1!"D:\Program Files\Indramat\Mtgui\			
Example PVR	 Line : [Strin Line : 00_BR_F FI comm Line 1 2 3 	2 = FI comn ig, in accord 3 = FI Job E PVR1_"D:\P and 1 1 1	nand dance with Chapter "Elements of the FI Command"] Error Code (see Chapter "Error Codes") rogram Files\Indramat\Mtgui\Temp\Upload.ini"/3 00_BR_PVR1!"D:\Program Files\Indramat\Mtgui\ Temp\upload.ini" /3 01 00_BR_PVR1!"D:\Program Files\Indramat\Mtgui\ Temp\upload.ini" /3			

Reading the PLC Variable Declaration: PVT

Designation	Ρ٧Τ	PLC Variable Type					
Explanation	as structur		e. To evaluate complex variables such ponents and types must be read out. rred PLC Variables.				
FI command	BR_PVT_	PLC variable type. (1) ifier of the PLC variable	(Single Read) [acc. to declaration part of the PLC]				



Response Structure	One line with 2 colum	ns is output for ea	ch eleme	ent of the v	ariables.		
	Line 1	n:	Colum	n 1	Column 2		
	n = number of eleme	nts.					
Value Range/Meaning of Columns	(1) = Identifier of the PLC variable [acc. to declaration part of the PL2 = 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	FI command 00_BR_PVT_TEST					
	Answer						
	Line	Column 1 (Na	ime)		Name		
	1	TEST			WORD		
WinPcI: Reading a Variable	<u>Assumption:</u> The WinPcl variable "Prog".	"TEST" is decla	red as \	WORD in	WinPcl program		
	FI command	00_BR_PVT_:Pro	g.TEST				
		Answ	ver				
	Line	Column 1 (Na	ime)		Name		
	1	TEST			WORD		
	ding a Structure Assumption: The "TEST1" variable is declared as STRUCT in the PLC STRUCT E1 BOOL E2 INT E3 SINT END						
	FI command	00_BR_PVT_TES	T1				
		Answ	/er				
	Line	Column 1		C	olumn 2		
	1	TEST1.E1			BOOL		
	2	TEST1.E2			INT		
	3	TEST1.E3			SINT		
WinPcI: Reading a Structure	g 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_:Pro	g.TEST1				
		Answ	ver				
	Line	Column 1		C	olumn 2		
	1	TEST1.E1			BOOL		

2

3

TEST1.E2

TEST1.E3



INT

SINT

PLC: Reading an Array

Assumption:

The "TEST2" variable is declared as ARRAY in the PLC program.

ARRAY]	
	0	 3

	00
] 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".



] 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:

СТ	RI	ICT	
31	κι		

E1	BOOL
	· · · —

- E2 INT
- E3 SINT

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	



WinPcI: Reading an Array of a Structure	Assumption: The WinPcl variable "Prog". ARRAY [0 1] OF STRUCT1, where STRUCT1 is of STRUCT E1 BOC E2 INT E3 SINT END	declared as follows: DL	ARRAY in WinPcI program
	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

Assumption:

The data types are output according to IEC1131.

TEST3[1].E2

TEST3[1].E3

See also command PVF.

2

3

SFC Diagnosis Data: SDD

MWAX device group

INT

SINT

Designation	SDD SFC Diagnosis Data			
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 disrupte	d step of a step ch	ain.	
	BR_SDD1!(1)!(2)	(Single Read)	
	(1) = Module num	ber ['	199]	
	(2) = SFC entity na	ame [/	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	[AS	CII characters]	
	4 = POU ID	[AS	CII 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		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	[199]
(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		action block, action network, 3 =	transition]
	2 = Detail name	[ASC	CII characters]	
	3 = POU ID	[ASC	CII characters]	
	4 = Detail morphe	me [AS0	CII characters] (DW	/ORD, decimal)
	5 = Error ID	[ASC	CII characters] (DW	/ORD, decimal)
	6 = POE entity na	me [ASC	CII characters]	

Example SDD2 Query faulty action of the disrupted step "open" of the "clamp" chain in module 3 in control unit 0.

FI comma	and	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 th	Output the definite ID to display the action, monitor error or transition.					
	BR_SDD	03!(1)!(2)!(3	s)!(4)	(Sing	le Read)		
	(1) = Mo	dule numbe	er	[19	9]		
	(2) = SF	C entity nan	ne	[ASCII characters]			
	(3) = Det	ail type			ction block,		
					ction network, ansition]		
	(4) = Det	ail name		[ASC	II characters]		
	Note:	The sepa	rator	"!" is used in thi	s command.		
Response Structure	The following table shows the general structure of the FI "SDD3".				he FI command		
		Line 1		Column 1		Column 4	
Meaning of the Columns	1 = POU	ID		[ASCII cl	naracters]		
	2 = Deta	il morphem	е	[ASCII cl	naracters] (DW	ORD, decimal)	
	3 = Erroi	· ID		[ASCII cl	naracters] (DW	ORD, decimal)	
	4 = POE	entity name	е	[ASCII cl	naracters]		
Example SDD3	Query ID to display the action "aOpen" of the "clamp" chain in mod control unit 0.				ain in module 3 in		
	FI command 00_			00_BR_SDD3!3!Station03A.Clamp!1!aOpen			
	Line	Column	Ans	wer			
	1	1	SFC_1_2				
		2	982	43823			
	3 3498530			85304			
		3	349	85304			
		4		85304 ion2.Module3			
FI command	Output th	4	Stat		ail.		
FI command	•	4	Stat sses	ion2.Module3 to display a deta	ail. e Read)		
FI command	BR_SDI	4 e I/O addre	Stat sses 3)!(4)	ion2.Module3 to display a deta	e Read)		
FI command	BR_SDE (1) = Mo	4 e I/O addre 04!(1)!(2)!(3	Stat sses 3)!(4) er	ion2.Module3 to display a deta (Sing l [199	e Read)		
FI command	BR_SDE (1) = Mo	4 e I/O addre 04!(1)!(2)!(3 dule numbe C entity nan	Stat sses 3)!(4) er	ion2.Module3 to display a deta (Sing l [199 [ASCI [1 = at 2 = at	e Read)]		
FI command	BR_SDE (1) = Mo (2) = SF (3) = Det	4 e I/O addre 04!(1)!(2)!(3 dule numbe C entity nan	Stat sses 3)!(4) er	ion2.Module3 to display a deta (Singl [199 [ASCI [1 = ac 2 = ac 3 = tr	e Read)] I characters] ction block, ction network,		
FI command	BR_SDE (1) = Mo (2) = SF (3) = Det	4 e I/O addre 04!(1)!(2)!(3 dule numbe C entity nan tail type tail name	Stat sses s)!(4) er ne	ion2.Module3 to display a deta (Singl [199 [ASCI [1 = ac 2 = ac 3 = tr	e Read)] I characters] ction block, ction network, ansition] I characters]		
FI command	BR_SDE (1) = Mo (2) = SF (3) = Det (4) = Det Note:	4 e I/O addre D4!(1)!(2)!(3 dule numbe C entity nan cail type cail name The sepa	Stat sses s)!(4) er ne	ion2.Module3 to display a deta (Singl [199 [ASCI [1 = ac 2 = ac 3 = tr [ASCI "!" is used in thi	e Read)] I characters] ction block, ction network, ansition] I characters] s command.	he FI command	
	BR_SDE (1) = Mo (2) = SF (3) = Def (4) = Def Note: The follo "SDD4".	4 e I/O addre D4!(1)!(2)!(3 dule numbe C entity nan cail type cail name The sepa	Stat sses s)!(4) er ne	ion2.Module3 to display a deta (Singl [199 [ASCI [1 = ac 2 = ac 3 = tr [ASCI "!" is used in thi	e Read)] I characters] ction block, ction network, ansition] I characters] s command. structure of 1	he FI command	
	BR_SDE (1) = Mo (2) = SF (3) = Det (4) = Det (4) = Det Note: The follo "SDD4".	4 e I/O addre D4!(1)!(2)!(3 dule numbe C entity nan cail type cail name The sepa wing table	Stat sses s)!(4) er ne rator sho	ion2.Module3 to display a deta (Singl [199 [ASCI [1 = ac 2 = ac 3 = tr [ASCI "!" is used in thi ws the general Column 1	e Read)] I characters] ction block, ction network, ansition] I characters] s command. structure of 1		

Example SDD4 Query I/O addresses to display the action "aOpen" of the "clamp" chain in module 3 in control unit 0.

			e an I/O addres			
	FI comma	and	00_BR_SDD4	3!Station03A.C	lamp!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	e the multili	ingual comments for displaying a detail.			
	BR_SDD) 5!(1)!(2)!(3	(3)!(4) (Single Read)			
	(1) = Mo	dule numbe	ber [199]			
	(2) = SF0	C entity nan	ne	[ASCII charac	cters]	
	(3) = Det	ail type		[1 = action blocket]	-	
			2 = action network, 3 = transition]			
	(4) = Det	ail name	-			
	Note: The separator "!" is used in this command.					
Response Structure	The follo "SDD5".	wing table	shows the g	eneral structu	re of the FI command	
	L	ine 1-n	Co	lumn 1	Column 2	
Meaning of the Columns	1 = Com	ment morpl	heme [A	SCII character	s] (DWORD, decimal)	
	1 = Comment morpheme[ASCII characters] (DWORD, decimal)2 = New comment[ASCII characters]					
	2 = New	comment	[A	SCII character	s]	
Example SDD5	Query co		display the a		s] of the "clamp" chain in	
Example SDD5	Query co module 3	mments to in control u	display the a	ction "aOpen"	-	
Example SDD5	Query co module 3	mments to in control u ments are r	display the a init 0. replaced by and	ction "aOpen"	of the "clamp" chain in	
Example SDD5	Query co module 3 Two com	mments to in control u ments are r	display the a init 0. replaced by and	ction "aOpen" other text.	of the "clamp" chain in	
Example SDD5	Query co module 3 Two comm	mments to in control u ments are r and	display the a init 0. eplaced by and 00_BR_SDD5!	ction "aOpen" other text.	of the "clamp" chain in	
Example SDD5	Query co module 3 Two com FI comma Line	mments to in control u ments are r and Column	display the a init 0. replaced by and 00_BR_SDD5! Answer	ction "aOpen" other text.	of the "clamp" chain in	
Example SDD5	Query co module 3 Two com FI comma Line	mments to in control u ments are r and Column	display the a init 0. replaced by and 00_BR_SDD5! Answer 98243823	ction "aOpen" other text.	of the "clamp" chain in	
Example SDD5	Query co module 3 Two comi Fl comma Line 1	mments to in control u ments are r and Column 1 2	display the a init 0. eplaced by and 00_BR_SDD5 Answer 98243823 Clamp open	ction "aOpen" other text.	of the "clamp" chain in	
Example SDD5	Query co module 3 Two come FI comma Line 1 2 Output th	mments to in control u ments are r and Column 1 2 1 2 e action tha	display the a init 0. replaced by and 00_BR_SDD5! Answer 98243823 Clamp open 40923423 Clamp closed	ction "aOpen" other text. 3!Station03A.C	of the "clamp" chain in	
	Query co module 3 Two com FI comma Line 1 2 Output th calculated	mments to in control u ments are r and Column 1 2 1 2 e action tha	display the a init 0. replaced by and 00_BR_SDD5 Answer 98243823 Clamp open 40923423 Clamp closed at has not been the online state	ction "aOpen" other text. 3!Station03A.C	of the "clamp" chain in	
	Query co module 3 Two come FI comma Line 1 2 Output th calculated BR_SDE	mments to in control u ments are r and Column 1 2 1 2 e action that based on	display the a init 0. replaced by and 00_BR_SDD5 Answer 98243823 Clamp open 40923423 Clamp closed at has not been the online state	ction "aOpen" other text. 3!Station03A.C	of the "clamp" chain in	
	Query co module 3 Two come FI comma Line 1 2 Output th calculated BR_SDE (1) = Mod	mments to in control u ments are r and Column 1 2 1 2 e action that based on 06!(1)!(2)!(3	display the a init 0. replaced by and 00_BR_SDD5 Answer 98243823 Clamp open 40923423 Clamp closed at has not been the online state 3)	ction "aOpen" other text. 3!Station03A.C	of the "clamp" chain in	
	Query co module 3 Two come FI comma Line 1 2 Output th calculated BR_SDE (1) = Mod	mments to in control u ments are r and Column 1 2 1 2 e action that based on 06!(1)!(2)!(3) dule number C entity nan	display the a init 0. replaced by and 00_BR_SDD5 Answer 98243823 Clamp open 40923423 Clamp closed at has not been the online state 3)	ction "aOpen" other text. 3!Station03A.C Single Read [199]	of the "clamp" chain in clamp!1!aOpen r the transition of a step) cters]	

Three variables have an I/O address.



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 morpher	4 = Detail morpheme [ASCII characters] (DWORD, d				
	5 = Error ID	5 = Error ID [ASCII characters] (DWORD, of				
	6 = POU entity nar	ne	[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 comma	and	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	

Setting of Device Status Information: SDS

Designation	SDS Set Device Status				
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 o be set.	command, th	ne device status i	nformation of ALL defined devices can	
	BW_SDS	61_(1)	(Singl	e Write)	
	· · ·	vice status mation to b	• - •	evice status information OFF evice status information ON	
Response Structure		ving table : I command		eral structure of the response to the	
		Line [·]	1	Column 1	
Value Range/Meaning of Columns	1 = Status report			[(P_ACK)]	
Example SDS1	Set device	e status info	ormation to OFF	for ALL defined devices.	
	FI comma	and	00_BW_SDS1_0)	
	Line	Column	Answer		
	1	1	(P_ACK)		



FI command	With this can be se		the device s	status informatio	n for a selected device
	BW_SD	S2 (1)	(5	ingle Write)	
		vice status	•	= Device status i	nformation OFF
	· · ·	rmation to b		= Device status i	
Response Structure		wing table		eneral structure	of the response to the
	Line 1				Column 1
Value Range/Meaning of Columns	1 =	Status repo	rt	[(P_ACK)]	
Example: SDS2	Set devic	e status info	ormation to O	FF for the select	ted device 00.
	FI comm	and	00_BW_SDS	2_0	
	Line	Column	Answer		
	1	1	(P_ACK)		
Sequencer Data: SFC	C				
	MWAX devi				MWAX device group
Designation	SFD	SFC Dat	a		
Explanation	concern	a sequence	er comment,	POU name, ste	he FI command this can ep comment, maximum ment), qualifier and time
FI command	Query the	e data for a	step chain.		
	BR_SFC	D1!(1)!(2)		(Single Read)
	(1) = Mo	dule numbe	er	[199]	
	(2) = SF	C entity nan	ne	[ASCII charac	ters]
	Note:	The sepa	rator "!" is us	ed in this comma	and.
Response Structure	The follo		shows the	general struct	ure of the "SFD1" FI
		Line 1	C	olumn 1	Column 2
Meaning of the Columns	1 = Step	chain comr	ment [ASCII characters	sl
	2 = POU		-	ASCII characters	-
Europeite OED4	O	4 f 4	-		-
Example SFD1	-			module 3 in cor	
	FI comm		_	1!3!Station03A.C	amp
	Line	Column	Answer		
	1	1	Clamping dev	/ICE	
		2	CLAMP		
FI command	Query the data of a step.				

FI command Query the data of a step. BR_SFD2!(1)!(2)!(3)

(1) = Module number

(Single Read) [1...99]

		SFC entity name [ASCII cha Step name [ASCII cha				
	Note:	The sepa	arator "!" is used in	n this command.		
Response Structure	The following table shows the general structure of the "SFD2" F command. The number of lines depends on the number of actions an transitions. If there are no details the line number is 1.					
	Lin		Column 1		Column 3	
	Line	-	Column 1		Column 6	
Meaning of the Columns	Line 1	I				
		comment	IASC	CII characters]		
	-	mum time	-	CII characters]		
	3 = Minir	num time	-	CII characters]		
	Line 2r	1:				
	1 = Deta		[1 =	action block, 3 = tra	ansition]	
	2 = Nam		-	CII characters]		
	3 = Com	ment	[ASC	CII characters]		
	4 = Bool	ean variabl	le [YES, NO]			
	5 = Qual	ifier	[ASCII characters]			
	6 = Time	e value	[ASC	CII characters]		
Example SFD2	Data for t	he step "Op	pen" in the "clamp"	chain in module 3	on control unit 0.	
	FI comm	and	00_BR_SFD2!3!	Station03A.Clamp!C)pen	
	Line	Column	Answer			
	1	1	Open clamping d	evice		
		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) (1) = Module number

(Single Read) [1...99]



	(2) = SFC entity name			e [ASCII characters]		
	(3) = Detail type			[1 = action block, 2 = action network, 3 = transition]		
	(4) = Detail name			[ASCII charac	cters]	
	Note:	The sepa	rato	"!" is used in this comm	and.	
Response Structure	The following table shows the general structure of the "SFI command.				ure of the "SFD3" FI	
		Line 1		Column 1	Column 2	
Meaning of the Columns	1 = Com	ment		[ASCII characters	s]	
	2 = Boole	ean variable	e	[YES, NO]		
Example SFD3	Data for the action "aOpen" in the "clamp" chain in module 3 on control unit 0.				module 3 on control unit	
	FI comma	and	00_	BR_SFD3!3!Station03A.C	lamp!aOpen	
	Line	Column	Ans	swer		
	1	1	Cla	mp open		
	2 NC					

Sequencer Messages: SFE

Designation	SFE	SFC Error				
Explanation	The sequencer messages of a module are output.					
FI command	Output all SFC messages.					
	BR_SFE1_(1)		(Single Read)			
	BC_SFE1_(1)		(Cyclic Read)			
	(1) = Module number		[199]			
	Output first SFC messages.					
	BR_SFE2_(1)			(Single Read)		
	BC_SFE2_(1)		(Cyclic Read)			
	(1) = Module number		[199]			
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 1n:		Column 1		Column 7
Meaning of the Columns	1 = Messa	ige text	[A	SCII characte	ers]	
	2 = SFC e	ntity name	[ASCII characters]			
	3 = Step n	ame	[A	[ASCII characters]		
	4 = Time s	stamp day	[m	[mm.dd.yyyy]		
	5 = Time s	stamp time	[hł	[hh:mm:ss]		
	6 = Туре с	of error	[1 = time error, 2 = monitor error, 3 = monitor event]		error,	



7 = Is there condition	[YES, NO]
analysis?	

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	

Sequencer Mode: SFM

Designation	SFM	SFC Mode			
Explanation	Queries step chain mode.				
FI command	Query the mode of a step chain. BR_SFM1!(1)!(2) BC_SFM1!(1)!(2) (1) = Module number (2) = SFC entity name		(Single Read) (Cyclic Read) [199] [ASCII characters]		

	Note:	Note: The separator "!" is used in this command.				
Response Structure	The following table shows the general structure of the "SFM1" F 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 mo	de of the "d	clamp" chain in m	nodule 3 in control unit 0.		
	FI comma	and	00_BR_SFM1!3!	Station03A.Clamp		
	Line	Column	Answer			
	1	1	1			
ware Installation Da	ata: SID)				
				MWCX device group		
Designation	SID	S oftware	Installation Data	l		
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 BR_SID1	ne installatio		Single Read)		

Softv

Designation	SID Software Installation Data						
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 th	ne installatio	on data	a.			
	BR_SID1				(Single Rea	d)	
	BC_SID1				(Cyclic Rea	d)	
Response Structure	One line v	with 8 colun	nns is	output fo	or the returned	d values.	
		Line 1			Column 1		Column 8
Meaning of the Columns	1 = Basic	c directory		[EXE fil	les of the BO	F]	
	2 = FI ins	tallation dire	ctory	[FI dire	ctory]		
	3 = Data	directory		[in acco	ordance with	BOF]	
	4 = GBO	version		[from IN	NDRAMAT.in	i]	
	5 = IF-DL	L mode		[from IN	NDRAMAT.in	i]	
	6 = IF version [from INI			NDRAMAT.ini from DLL mode 400]			
	7 = Servi	ce package	e info	[from IN	NDRAMAT.ini from DLL mode 420]		
	8 = Relea	ase info		[from IN	NDRAMAT.in	i from DLL m	ode 420]
Example SID1	Return inf	ormation o	n the c	urrent in	stallation.		
	FI comma	and	00_BI	R_SID1			
	Line	Column	Answ	er			
	1	1					
		2	D:\Pro	gramme	Indramat\MTG	€UI\Bin	
		3					
	4 005-22Vxx						
		5 07.00					
		6	07V00)			
		7					
		8					
			-				

MWAX device group

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

PLC Long Identification: SLI

Designation	SLI SPS Long Identification					
Explanation	Returns	the unit data from the PL	C long identif	ication.		
FI command	Read P BR_SL	LC long identification. I	(Single Re	ead)		
Response Structure	One line	e with 15 columns is outpu	t for the retur	ned values.		
		Line 1	Column 1	Column	Column 15	
Value Range/Meaning of the	1 =	Device address	[00	15]		
Columns	2 =	Program number	[01	99]		
	3 =	Project name	[max	. 8 ASCII cha	aracters]	
	4 =	Program name	[max	[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	[LON	IG]		
Example SLI	Read th	ne unit data from the PLC le	ong identifica	ition.		
	El command 00 PD SI I					

FI comma	and	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



	FI comm	and	00 BB S			
	Line	Column	00_BR_SLI Answer			
	Line	13	2			
		13	2 0x0000			
		14	13			
		15	13			
Reference to Literature	see chap	oter entitled	"Literature	" [30].		
Requesting Watch List	Alloca	tions: W	VLA			
					MWA	X device group
Designation	WLA	Watch L	ist A llocati	on		
Explanation		Requests free watch list allocations. A maximum of ten free watch list allocations can be requested with one FI command.				
	BR_WL	.A1_(1)		(Single Rea	ıd)	
	re	umber of the equested fre st numbers				ee watch list e. The allowed
Response Structure		wing table s d "WLA1".	shows the	general struc	cture of the res	ponse to the FI
		Line 1	(Column 1		Column n
Value Range/Meaning	1 =	1. free watc	h list alloc	ation	Value range:	1 16
of Columns	-	2. free watc			Value range:	
		3. free watc			Value range:	
	n =	nth free wate	ch list alloc	ation	Value range:	116
Example WLA1	Request	four free wa	atch list all	ocations.		
	<u>Assumpt</u>	tion:		are already as	ssigned!	
	FI comm	nand	00_BR_W	/LA1_4		
	Line Column Answer					
	1	1	1			
		2	2			
		3	4			
		4	6			
Freeing Watch List Allo	ocation	s: WFL				

MWAX device group

Designation	WLF	Watch List Free					
Explanation	Previously	Previously requested watch list allocations are freed again.					
FI command	Free ALL a BR_WLF1	ssigned watch list allocations for the selected device. (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 th command "WLF1".					ponse to the FI	
		Line 1		Column 1		Column n	
Value Range/Meaning	1 =	1. freed wat	ch lis	t allocation	Value ra	ange: 116	
of Columns	2 = 2	2. freed wat	ch lis	t allocation	Value ra	ange: 116	
	3 = 3	3. freed wat	ch lis	t allocation	Value ra	ange: 116	
	n = r	nth freed wa	tch lis	st allocation	Value ra	ange: 116	
Example WLF1	Free ALL	assigned w	/atch	list allocations.			
	Assumption: The following watch list numbers have been allocated: 1, 2, 3, 4.					, 3, 4.	
	FI comm	and	00_1	BR_WLF1			
	Line	Column	Ans	wer			
	1	1	1				
		2	2				
		3	3				
		4	4				
FI command	Free the r	equired wat	ch list	allocations for a se	elected device.		
		F2_(1)_{(2).					
Response Structure	allocation	= List of wa ns to be rele wing table s	eased		n of 10 watch li isferred here to cture of the res	be freed	
		d "WLF2".		<u>j</u>			
		Line 1		Column 1		Column n	
Value Range/Meaning	1 =	1. enabled v	vatch	list allocation	Value ra	ange: 116	
of Columns	2 = 2	2. enabled v	vatch	list allocation	Value range: 116		
	3 = 3	3. enabled v	vatch	list allocation	Value ra	ange: 116	
	n = r	nth freed wat	ch lis	t allocation	Value ra	ange: 116	
Example WLF2	Free required watch list allocations: <u>Assumption:</u> Watch list allocations 1,3,4, and 8 have first been requested using the FI command "WLA1".						
	FI comm	and	00_l	BR_WLF2_1_3_4_8	3		
	Line	Column	Ans	wer			
	1	1	1				
		2	3				
		3	4				
	L						



7.7 FI Commands for the MSYX Device Group

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	[0063]
Note:	Please note that the device address must b respective FI command, e.g. 00_CR_AAC_0 to the chapter "Elements of the FI Command") (refer also here
	grouped together ailability of the Fl	

Determining the Actual (Current) System Error: ASE

MSYX device group

Designation	ASE	Actual S	Actual System Error				
Explanation			error is read out is functioning co		e answer 0x0	000 indicates	
FI command	CR_ASE		(Single Rea	d)			
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.					rive is output errors can be	
		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 t	he current	system error re	turns LWL rin	ng interrupted		
	FI comma	and	00_CR_ASE				
	Line	Column	Answer				
	1	1	0x0000				
	2 0x0000						
	3 0x8009 4 0x0000						
Reference to Literature	See chan	ter entitled	"I iterature" [42	1			

Reference to Literature

See chapter entitled "Literature" [42].



Deleting the Actual (Current) System Error: CSE

MSYX device group

Designation	CSE	Clear System Error					
Explanation	An error r	eported by	the Synax devi	ce is deleted	again.		
FI command	CW_CSE		(Single V	Vrite)			
	Value to b	e 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.					is output that rrors can be	
		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 t	he actual (c	current) system	error:			
	FI comma	and	00_CW_CSE	_CW_CSE			
	Line	Column	Answer				
	1	1	0x0000				
		2	0x0000				
		3 0x0000					
		4	0x0000				
Poforonao to Litoroturo	ana ahant	or optitlad ("Litoroturo" [45"				

Reference to Literature see chapter entitled "Literature" [45].

Setting the Communication Timeout Time DCT

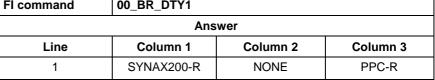
Designation	DCT Device Communication Timeout					
Explanation	By means of this command, the timeout time for the selected device is set dynamically (timeout time in ms).					
FI command		CT1_(1) equested timeout time in m	(Single Write)			
Response Structure	The response to the "DCT1" FI command consists 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 comma	and	00_BW_DCT1_1500			
	Line	Column	Answer	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 respo column.	onse to the	"DCT2" FI co	mmand consists of one line with one		
	Line 1			Column 1		
Value Range/Meaning of Columns	1 = Status message (P_ACK			(P_ACK)		
Example DCP2	For the de	evice 00, th	e timeout time	e is reset to the default value.		
	FI comma	and	00_BW_DCT	2		
	Line	Column	Answer			
	1	1	(P_ACK)			

Device Type and Accompanying Components: DTY

Designation	DTY	Device	TYpe				
Explanation		The device type and the accompanying components of the selected device address are output.					
FI command	BR_D	TY1	(Sin	gle Re	ad)		
Response Structure	"DTY1 output	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		Co	umn 1		Column 3
Value Range/Meaning of Columns	1 =	1 = Device Type			(see chapter entitled "Elements of the FI Command", and "Identifier")		
	2 =	Component	Component type1 IND_DEV.INI entry: Component type1=				
	3 =	3 = Component type 2				V.INI entry: ent type2=	
Example DTY1	Outpu addres		type a	nd the	accomp	anying compor	nents of device
	FI cor	nmand	00_BR_	DTY1			
		Answer					



Read System Messages: MSG

MSYX device group

MSYX device group

Designation	MSG	MeSsaG	e		
Explanation	Reading	of system m	nessages		
FI command	Message CC_MSG_(1) (Cyclic Read) (1) = SYS message numbers				
	Note:	Exists on	y as a cyclic command		
Response Structure	The respo data.	onse of the	FI command 'MSG' consists of th	e system message	
Example MSG	00_CC_N	ISG_64	(64 = MSG_SYSERRGEN)		
	FI comma	and	00_CC_MSG_64/3		
	Line	Column	Answer		
	1	1	00		
Restriction	The follov	ving system	messages:		
	SYS Mes	sage	SYS Message nu	umber	
	MSG_PC	LUPDBEG	52		
	MSG_PA	RUPDBEG	24		
	MSG_FW	AUPDBEG	82		
	Indramat	mmands ca OPC serve DDE serve	-	ograms:	

Generating Physical Directory Names: PHD

			U 1
Designation	PHD	PHysical Directory	
Explanation		physical directory names is based on BDI philosoph	according to the BDI data written. y.
FI command		ohysical directory names. _(1)_(2)_(3)_(4)_(5)_(6) ect ID	(Single Write) [-1= PROJECT_NEUTRAL -2= PROJECT_DEFAULT]
	(2) = Sect	ion 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(4) = Process ID	[-1= DEVADDR_NEUTRAL otherwise the required device address] [-1= PROCESS_NEUTRAL otherwise the required
	(5) =Data type ID	process number] [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 gener	al structure of the response to the FI

e 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_PHD11_011_0_0	
Line	Column	Answer	
1	1	D:\Programme\Indramat\Mtgui\Bin	

Setting of Device Status Information: SDS

Designation	SDS	Set Device Status	6		
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 comn messages are ger MSG_DEVICEOF	nerated		
FI command	With this command, the device status information of ALL defined devices can be set.				
	BW_SDS	61_(1)	(Singl	e 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 = 5	Status report		[(P_ACK)]	



Example SDS1

1 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_SD	S2_(1)	(Sing	le Write)		
	(1) = Device status information to be set0 = 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)			

Software Installation Data: SID

				MSYX	device group		
Designation	SID Soft	ware Installation	Data				
Explanation	Information is returned regarding installation. This information includes the installation paths, the software version used, DLL mode, plus service pack and release information.						
FI command	Read-in the inst	allation data.					
	BR_SID1		(Single Rea	ıd)			
	BC_SID1		(Cyclic Rea	d)			
Response Structure	One line with 8	columns is output	or the returne	d values.			
	L	Line 1 Column 1 Column 8					
Meaning of the Columns	1 = Basic direc	tory [EXE	files of the DO	S-BOF]			
	2 = FI installatio	n directory [FI dir	ectory]				
	3 = Data direct	ory [in ac	cordance with	DOS-BOF]			
	4 = GBO versio	4 = GBO version [from INDRAMAT.ini]					
	5 = IF-DLL mode [from INDRAMAT.ini]						
	6 = IF version	[from	INDRAMAT.in	i from DLL m	ode 400]		
	7 = Service page	k info [from	INDRAMAT.in	i from DLL m	ode 420]		
	8 = Release inf	o [from	INDRAMAT.in	i from DLL m	ode 420]		



FI comma	and	00_BR_SID1
Line	Column	Answer
1	1	
	2	D:\Programme\Indramat\MTGUI\Bin
	3	
	4	005-22Vxx
	5	07.00
	6	07\/00
	7	
	8	

Example SID1 Return information on the current installation.

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

SERCOS Parameters: SPA

Designation	SPA	SERCOS PAran	neter			
Explanation		S drive parameter is output or written. Each parameter consists nts, whereby any combination of elements can be selected by ding.				
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	[0254]			
	(2) = Parar	meter No.	in format X-Y-ZZZZ			
	(3) = Elem	ent coding	[standard or advanced format]			
Parameter No.	Form	nat X-Y-ZZZZ	Value Range			
		х	S = standard data			
			P = product data Y = SERCANS parameter			
		Y	[000.15] = parameter record			
		Z	[04095] = data block no.			
Element Coding	operating d request, th operating d	t coding in standard format allows individual elements, such as the g date, to be requested. If several elements are to be read out in one then the element coding can be OR'd in advanced format, e.g. g date (0x40) and unit (0x08) produces OR'd (0x48) \rightarrow 48 vanced 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	Н	20H	Decimal word	20000
Operating date	D	40H	(see Displaying the Operat	ing 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
Х	S = standard data P = product data
Y	[00.7] = parameter record
Z	[04095] = 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:				
		er S-0-0030, operation date			
	Standard				
	Advanced	format: "16\n16\nDKC2.1-SSE-01V09"			
	Note:	When requesting SERCANS parameters the drive address can be anywhere within the range [0254].			
Response Structure	command	wing table shows the general structure of the response to the FI d "SPA1". Line 1 is output both when reading and when writing. I lines are only output when reading depending on the element			
	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 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=""></sercos>	<drive no.<br="">SERCOS error></drive>	<global sercans<br="">error></global>	<drive no.<br="">Global SERCANS error></drive>
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	2_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.



00_BW_SPA1_3_S-0-0305_40FI commandValue to be written: : P-0-0037						
			Answ	/er		
Line	Colu	umn 1	Co	lumn 2	Column 3	Column 4
1	0x0	0000	0	x0003	0x0000	0x0000
Reference to	Literature	See chapt	er entitled	"Literature" [4	41].	
		See chapt	er entitled	"Literature" [4	46].	
		Owitab	0			
Active SERCO	5 Phase	Switch	-Over:	SPH		
						MSYX device grou
De	esignation	SPH	SERCO:	S PH ase		
Ex	xplanation				are in the same co	
		The phase	e status ca	n be read-out	t or changed by this	command.
FI	command	CR_SPH			(Single Read)	
		CW_SPH (Single Write)				
Value to	be written Phase				[2, 4]	
		Note: The value to be written is passed to the "acValue" para				
		Note:		ataTransfer" r		acvalue paramet
E.v.		Deed the				
Read SERC	ample SPH COS Phase	FI comma		-	ax control at device a	
		Line	Column	00_CR_SPH Answer		
		1	1	2		
Fxa	Example SPH Switch-over the synax control (write) after phase 4; phase 2 is active.					ase 2 is active
Write SERC	•			00_CW_SPH		
		FI command		Value to be		
		Line Column		Answer		
		1	1	52		
			2	1		
		Note:	Switching	over from pl	hase 2 to phase 4 re	eturns the value [5
		11010.	as the re	sult in colum	n 1. On switching o	ver from phase 4
			phase 2,	column 1 d	contains the value	[50]. The result

Reference to Literature

See chapter entitled "Literature" [42].



8 Error codes

8.1 General Error Result Line

If the "DataTransfer" routine returns an error code, then the requested data is not returned by the "ReadGroupItem" routine, but a general error result line is returned instead. This general error result line contains additional information regarding the possible causes of the error.

Note: How the routines work, as well as tips and tricks for working with the Rexroth Indramat function interface, is described in Chapter "Programming".

The following table shows the general structure of the error result line. One line is output consisting of 5 columns for the class of error, error code, expanded additional information, error text and additional text.

	Line 1	Column 1		Column 5
--	--------	----------	--	----------

Meaning of the Columns 1

1 = Error class

2 = NACK code or return error code (depends on error classes)

3 = Expanded additional information [hexadecimal LONG value]

4 = Error text [ASCII characters]

5 = Additional text [x= exists, -- = does not exist]

The following error classes are contained in the file "INDIF000.h" or "INDIF000.BAS":

Error class	Meaning
FI_ERROR_CLASS_NACK	NACK control messages
FI_ERROR_CLASS_FCT	Protocol function error codes

8.2 Error Codes 200 to 999

Code	Error Text	Name and Meaning of Error
201	BOF_INVALID_MTCNC_NUMBER	Invalid device address
202	BOF_NO_INST_PATH	No installation path found (Indramat.INI).
203	BOF_NO_MAP_FILE_FOUND	MAP file "PLCMAP.SPS" not found.
204	BOF_NO_MAP_FILE_NAME	No entry in the "PLCMAP.DAT" file has been found for the device address.
205	BOF_FILE_ERROR	File processing error.
206	BOF_VERSION_ERROR	More than the MAXIMUM NUMBER OF LINES contained in the "VERSION.DAT" file. Remedy: Delete "VERSION.DAT" file
207	BOF_MUTEX_ERROR	Error generating a MUTEX object.
208	BOF_FILE_MAPPING_ERROR	Error generating file mapping.
209	BOF_MEMORY_ERROR	Memory allocation error.

Code	Error Text	Name and Meaning of Error
210	BOF_DATA_MAP_ERROR	DATA MAP access error.
211	BOF_MUTEX_TIMEOUT	MAP file access error.
212	BOF_DATA_LENGTH_ERROR	Data buffer is too small.
213	BOF_FILE_NOT_FOUND	File not found.
214	BOF_SYS_MAP_ERROR	SYSTEM MAP access error.
215	BOF_MAP_ELEMENT_ERROR	No valid MAP structure element.
216	BOF_INVALID_CHANNEL_ERROR	LOG channel number invalid.
217	BOF_TIMEOUT_ERROR	Pre-set timeout has expired.
218	BOF_SHMEM_ALREADY_EXIST	SHARED MEM already exists.
219	BOF_PROCESS_NOT_EXIST	Process addressed does not exist.
220	BOF_FILE_EOF	End of file reached.
221	BOF_EVENT_ERROR	Error generating an event object.
222	BOF_PROCESS_ALREADY_EXIST	Process to be started already running.
223	BOF_COMM_ADDRESS_ERROR	No valid communication address.
224	BOF_DEVICE_TYP_ERROR	No valid device type.
225	BOF_DEVICE_ERROR	No valid device address defined.
226	BOF_DEVICE_NAME_ERROR	Invalid device name.
227	BOF_DEVICE_STATUS_ERROR	No valid device status.
228	BOF_DEVICE_PLC_ERROR	No valid PLC information.
229	BOF_TASK_ID_ERROR	Invalid or false task ID.
230	BOF_TASK_ADM_ERROR	Task administration error.
231	BOF_TASK_TRIGGER_ERROR	Task trigger-event error.
232	BOF_EVENT_NOT_FOUND	Event object does not exist.
233	BOF_TASK_NAME_ERROR	Task name is too long.
234	BOF_SYS_STACK_INDEX_ERROR	Invalid SYS-MSG STACK INDEX
235	BOF_SYS_STACK_FULL_ERROR	SYS-MSG STACK is full.
236	BOF_SYS_STACK_MSG_ERROR	SYS-MSG message is not known in SYS-MSG STACK.
237	BOF_SYS_STACK_ACCEPT_ERROR	SYS-MSG message could not be accepted by the SYS- MSG STACK within the pre-set time.
238	BOF_SYS_MSG_SET_ERROR	Access to SYS-MSG channel not possible in the pre- set time. (SYS-Message is issued).
239	BOF_SYS_MSG_GET_ERROR	Access to SYS-MSG channel not possible in the pre- set time (SYS-Message is fetched).
240	BOF_DATA_TIME_ERROR	A data element in the shared memory area was not released in the pre-set time.
241	BOF_DATA_ACCESS_ERROR	Access to a data element in the shared memory area is locked.
242	BOF_FCT_PAR_ERROR	An incorrect parameter value has been passed within the function.
243	BOF_SYS_STACK_QUIT_ERROR	SYS-MSG acknowledgement event has not been released in the pre-set time.
244	BOF_NO_SYS_MSG_RDY	No SYS-MSG message.



Code	Error Text	Name and Meaning of Error
245	BOF_FORMAT_ERROR	Format error
246	BOF_SYS_MSG_LIST_ERROR	Error in the SYS-MSG list for manual acknowledgement.
247	BOF_NO_IFDLL_MODE_ERROR	Mode details missing in "IfDIIMode=" entry in "INDRAMAT.INI" file.
248	BOF_LOG_GRP_COUNT_ERROR	Invalid group number for the LOG channel.
249	BOF_NO_SYS_MSG_CONF_ERROR	No SYS-MSG acknowledgement message.
250	BOF_NO_PATH_ENV_ERROR	No path environment variable.
251	BOF_LOGIN_IF_EVENT_ERROR	LOGINIF event could not be created.
252	BOF_LOGIN_SYS_MSG_ERROR	LOGINIF could not be carried out in the pre-set time due to pending SYS-MSGs.
253	BOF_LOGIN_EVENT_TIME_ERROR	No SYS-MSG could be issued because the login event took too long.
254	BOF_DLL_MUTEX_TIMEOUT_ERROR	Access to the DLL-lock Mutex was not possible in the pre-set time.
255	BOF_DLL_ALREADY_INSTALLED	DLL already installed.
256	BOF_DLL_LOAD_ERROR	DLL could not be loaded by the load library.
257	BOF_DLL_LIST_FULL_ERROR	DLL list already full.
258	BOF_DLL_LIST_DELETE_ERROR	DLL to be deleted is not in the list.
259	BOF_DOS_NT_SYS_MSG_ERROR	Invalid SYS-MSG message number in DOS \rightarrow NT job processing.
260	BOF_DOS_NT_JOB_STR_ERROR	Invalid DOS \rightarrow NT command string.
261	BOF_SYS_MSG_RANGE_ERROR	SYS-MSG message number is outside the valid number range.
262	BOF_DOS_NT_JOB_INFO_ERROR	$DOS \rightarrow NT$ command information is too long
263	BOF_DOS_NT_SYS_MSG_Q_ERROR	An odd SYS-MSG message number (acknowledgement) was passed by the DOS \rightarrow NT command SYSC_xxx.
264	BOF_DOS_NT_FKT_NOT_FOUND_ERROR	$DOS \rightarrow NT$ command issued for which there is no processing function in the "BOFINTFC.DAT" file.
265	BOF_DOS_NT_DLL_NAME_NOT_FOUND_ERROR	No DLL name exists for DOS \rightarrow NT commands in the "BOFINTFC.DAT" file.
266	BOF_DOS_NT_DLL_NOT_FOUND_ERROR	DLL for the DOS \rightarrow NT commands not found.
267	BOF_DOS_NT_FKT_NOT_IN_DLL_ERROR	$DOS \rightarrow NT$ processing function not found in the specified DLL.
268	BOF_DOS_NT_BOF_DAT_NOT_FOUND_ERROR	The "BOFINTFC.DAT" file could not be found.
269	BOF_TASK_NAME_NOT_FOUND_ERROR	Task name is not in the task list.
270	BOF_TASK_ID_NOT_FOUND_ERROR	No task exists for the task ID.
271	BOF_NT_CODE_ERROR	WIN-32 API error has occurred.
272	BOF_DOS_NT_PROCESS_PRIORITY_ERROR	Invalid process priority class.
273	BOF_DOS_TASK_NAME_ERROR	Error in generating the DOS-BOF task name.
274	BOF_PARENT_WIN_NAME_LEN_ERROR	Name of the parent window is too long.
275	BOF_TERMINATE_EVENT_NAME_LEN_ERROR	Name of the terminate event is too long.
276	BOF_PARENT_WIN_NOT_EXIST_ERROR	Registered task does not have a parent window.
277	BOF_DLL_NOT_EXIST_ERROR	DLL sought does not exist.
278	BOF_DLL_FUNCTION_NOT_FOUND_ERROR	Function sought does not exist in the specified DLL.



Code	Error Text	Name and Meaning of Error
279	BOF_PROCESS_NOT_LOGIN_ERROR	An FI command has been called although the client is not yet logged in. The "LogInIf" routine has not yet been run.
280	BOF_DEVICE_HANDLE_ERROR	Device handle could not be generated.
281	BOF_DEVICE_ASSIGN_ERROR	There is no "DeviceAssign" entry in the "IND_DEV.INI" file, or the entry is invalid.
282	BOF_MEMORY_CLASS_ERROR	No valid memory class for DOS \rightarrow NT command RDNT/WRNT.
283	BOF_MEMORY_DOS_NT_DATA_LEN_ERROR	Data length of RDNT/WRNT command is too long.
284	BOF_SHMEM_INDEX_ERROR	No valid SHARED MEMORY INDEX.
285	BOF_NO_PORT_ADDR_ERROR	No port address in the communication address.
286	BOF_NO_PORT_VALUE_ERROR	No port value in the communication address.
287	BOF_VRT_MANAGER_MODE_ERROR	Invalid VRT manager mode
288	BOF_VRT_START_MODE_ERROR	There is no MTVNC mode entry in the "IND_DEV.INI" file, or the entry is invalid.
289	BOF_PAR_MIN_ERROR	No PARMIN value in the "BOFINTFC.DAT" file for the FI command.
290	BOF_PAR_MAX_NUMBER_ERROR	Too many pass parameters in the FI command.
291	BOF_PAR_MAX_ERROR	No PARMAX value in the "BOFINTFC.DAT" file for the FI command.
292	BOF_PAR_NUMBER_ERROR	Number of FI parameters does not agree with the data in the DAT files (e.g.: MTCX.DAT, BOFINTFC.DAT, etc.)
293	BOF_PAR_DESCRIPT_ERROR	No parameter description.
294	BOF_PAR_TYPE_ERROR	Invalid FI parameter type.
295	BOF_PAR_TYPE_NOT_FOUND_ERROR	No FI parameter type description found.
296	BOF_PAR_DATA_ERROR	Invalid FI parameter data, i.e. FI data not defined in FI data type.
297	BOF_PAR_TYPE_DESCRIPT_ERROR	Invalid FI parameter type description.
298	BOF_PAR_INDEX_ERROR	FI parameter index is too large.
299	BOF_PAR_NO_CYCLIC_ERROR	Either there is no CYCLIC entry in the "BOFINTFC.DAT" file or the CYCLIC entry is invalid.
300	BOF_PAR_NO_CYCLIC_FI_COMMAND_ERROR	No cyclic FI command released.
301	BOF_PAR_NO_BINAER_ERROR	Either there is no binary entry, or an invalid binary entry in the "BOFINTFC.DAT" file.
302	BOF_PAR_NO_BINAER_FI_COMMAND_ERROR	No binary operation of FI command released.
303	BOF_NT_DOS_CHANNEL_ACCESS_ERROR	Access to NT \rightarrow DOS job channel not possible in the pre-set time.
304	BOF_NT_DOS_COMMAND_LENGTH_ERROR	$NT \rightarrow DOS$ command string is too long.
305	BOF_NT_DOS_COMMAND_INFO_LENGTH_ERRO R	$NT \rightarrow DOS$ command info string is too long.
306	BOF_NT_DOS_TIMEOUT_ERROR	$NT \rightarrow DOS$ job could not be executed in the pre-set time.
307	BOF_NT_DOS_FKT_NOT_FOUND_ERROR	An NT \rightarrow DOS command was issued that had not been declared in the "BOFINTFC.DAT" file.
308	BOF_NT_DOS_DLL_NAME_NOT_FOUND_ERROR	No DLL is declared in the "BOFINTFC.DAT" file for the NT→DOS command issued.

Code	Error Text	Name and Meaning of Error
309	BOF_NT_DOS_DLL_NOT_FOUND_ERROR	DLL for the NT \rightarrow DOS commands not found.
310	BOF_NT_DOS_FKT_NOT_IN_DLL_ERROR	$NT \rightarrow DOS$ processing function not found in the specified DLL.
311	BOF_NT_DOS_JOB_STR_ERROR	Invalid NT \rightarrow DOS command string.
312	BOF_NT_DOS_JOB_INFO_ERROR	$NT \rightarrow DOS$ command information is too long
313	BOF_NT_DOS_DPR_TIMEOUT_ERROR	Access to NT-DOS-DPR memory not possible in the pre-set time.
314	BOF_NT_DOS_NO_COMMAND_ERROR	No NT \rightarrow DOS command string.
315	BOF_NT_DOS_BOF_INDEX_ERROR	Invalid DOS-BOF INDEX when issuing an NT \rightarrow DOS command.
316	BOF_PAR_INVALID_VALUE_ERROR	Pass parameter to the function has an invalid value.
317	BOF_DOS_BOF_EXE_PATH_ERROR	DOS-BOF EXE file must not contain details of path.
318	BOF_LOG_IN_LOG_OUT_TIMEOUT_ERROR	Login/Logout not possible in the pre-set time.
319	BOF_DEVICE_TYP_GROUP_ERROR	Selected device address does not exist in this device group.
320	BOF_INVALID_PROCESS_NUMBER_ERROR	Invalid NC process number
321	BOF_PROCESS_NAME_LENGTH_ERROR	Process name is too long or invalid.
322	BOF_PARAM_IDENT_REQUEST_ERROR	Invalid data was returned by the interface on requesting the parameters.
323	BOF_SWITCH_DEVICE_ERROR	An attempt was made to switch to a virtual MTC that is assigned to a real MTC.
324	BOF_DEVICE_TYPE_REQUEST_ERROR	Invalid data was returned on requesting the device type ID.
325	BOF_DEVICE_SPS_IDENT_ERROR	Invalid data was returned by the interface on requesting the long ID of the PLC MAP file.
326	BOF_INVALID_AXIS_NUMBER_ERROR	Invalid axis number received [132].
327	BOF_NO_GBOVERSION_ERROR	There is no "GBOVERSION=" entry in the "INDRAMAT.INI" file, or the entry is invalid.
328	BOF_NO_ACHSREF_TABLE_ERROR	Axis reference table error.
329	BOF_DEVICE_GROUP_ERROR	The device group for this job is invalid.
330	BOF_PROCESS_NOT_DEFINED	Process is not defined in the current parameters.
331	BOF_INVALID_DEVICE_GROUP_VALUE_ERRO R	Invalid device group number.
332	BOF_INVALID_DEVICE_ID_STR_ERROR	Invalid device ID string.
333	BOF_INVALID_DEVICE_GROUP_STR_ERROR	Invalid device group string.
334	BOF_FI_JOB_CLASS_ALREADY_RUN_ERROR	FI JOB already running.
335	BOF_FI_JOB_REQUEST_ERROR	No more FI JOBs possible.
336	BOF_FI_JOB_ID_ERROR	No valid FI JOB ID.
337	BOF_FI_JOB_NO_ID_FOUND_ERROR	No FI JOB ID found in the management structure.
338	BOF_FI_JOB_PROGRESS_TYPE_ERROR	Invalid request for the progress of an FI JOB.
339	BOF_FI_JOB_EXECUTE_FKT_NOT_FOUND_ ERROR	Execute function for the FI JOB was not found in the specified DLL.
340	BOF_FI_JOB_ERROR_STRING_TO_LONG	FI JOB ERROR STRING is too long.
341	BOF_FI_JOB_TIMEOUT_ERROR	FI JOB could not be executed in the pre-set time.

Code	Error Text	Name and Meaning of Error
342	BOF_FI_ERROR_STRING_TO_LONG	String for the general FI ERROR RESPONSE TELEGRAM (general error result line) is too long.
343	BOF_DOS_MANAGERPROG_NOT_READY_ERRO R	DOS-NT manager program not running.
344	BOF_NT_DOS_ORDER_TO_LONG	$\text{NT} \rightarrow \text{DOS}$ job description is too long.
345	BOF_FILE_CLASS_OBJECT_INSTALL_ERROR	File class object for access to the BOF files could not be created.
346	BOF_FILE_DIAGOFF_NOT_FOUND_ERROR	BOF file "DIAGOFF.XXX" not found.
347	BOF_FILE_DIAGOFF_OPEN_ERROR	Error opening the "DIAGOFF.XXX" file.
348	BOF_SH_MEM_DIAGOFF_NOT_FOUND_ERROR	No SHARED MEMORY for DIAGOFFxxx found.
349	BOF_FILE_DIAGTAB_NOT_FOUND_ERROR	BOF file "DIAGTAB.XXX" not found.
350	BOF_FILE_READ_WITH_FS_CLASS_ERROR	Read error with FS classes.
351	BOF_FILE_DIAGTEXT_NOT_FOUND_ERROR	Diagnostic text file "STERRxx.YYY" not found.
352	BOF_FILE_STERR_FILE_CLOSE_ERROR	Diagnostic file "STERRxx.YYY" could not be closed.
353	BOF_FILE_STERR_FILE_OPEN_ERROR	Diagnostic text file "STERRxx.YYY" could not be opened.
354	BOF_FILE_STERR_FILE_POSITION_ERROR	File positioning in diagnostic text file "STERRxx.YYY" could not be carried out.
355	BOF_FILE_STERR_FILE_READ_ERROR	Read function of diagnostic text file "STERRxx.YYY" could not be carried out.
356	BOF_FILE_STERR_FILE_NOT_FOUND_ERROR	Diagnostic text file "STERRxx.YYY" not found.
357	BOF_FILE_DIAGTAB_POSITION_ERROR	File positioning in "DIAGTAB.xxx" could not be carried out.
358	BOF_FILE_STERR_FILE_TIMEOUT_ERROR	TIMEOUT when waiting for the MUTEX release for access to the STERR files.
359	BOF_TASK_THREAD_TRIGGER_INFO_TO_LON G	Additional information passed for the TASK THREAD triggering is too long.
360	BOF_TASK_THREAD_TRIGGER_TIMEOUT_ERRO R	TIMEOUT of MUTEX release for access to the TASK-THREAD triggering.
361	BOF_FILE_SPRACHE_FILE_OPEN_ERROR	"SPRACHE.DAT" file could not be opened.
362	BOF_COMMAND_RESULT_DATA_TYPE_ERRO R	A result data type that is not permissible (e.g. 00_BR_AMM1/2) was requested for an FI-command (BR).
363	BOF_FILE_TEXT_FILE_NOT_FOUND_ ERROR	Relevant TEXTxx.YY file does not exist.
364	BOF_FILE_TIND_FILE_NOT_FOUND_ERROR	Relevant TINDxx.YY file does not exist.
365	BOF_FILE_TIND_FILE_OPEN_ERROR	TINDxx.YY could not be opened.
366	BOF_TEXT_NUMBER_TO_LARGE_ERROR	Text number to be read from BOF text file is too large.
367	BOF_FILE_TEXT_FILE_OPEN_ERROR	TEXTxx.YY could not be opened.
368	BOF_FILE_TEXT_FILE_POSITION_ERROR	File positioning in the text file "TEXTxx.YY" could not be carried out.
369	BOF_FILE_TEXT_FILE_READ_ERROR	Read function in the text file "TEXTxx.YY" could not be carried out.
370	BOF_DIAGNOSTIC_NUMBER_TO_LARGE_ERROR	Message number for CNC/PLC message system is too large.
371	BOF_FILE_SYSERI_NOT_FOUND_ERROR	BOF file "SYSERI.XXX" not found.
372	BOF_FILE_SYSERI_OPEN_ERROR	Error opening the "SYSERI.XXX" file.

Code	Error Text	Name and Meaning of Error
373	BOF_FILE_SYSERI_POSITION_ERROR	File positioning in SYSERI.xxx could not be carried out.
374	BOF_SH_MEM_SYSERI_NOT_FOUND_ERROR	No SHARED MEMORY for SYSERI.xxx found.
375	BOF_FILE_SYSANW_NOT_FOUND_ERROR	Diagnosis text file SYSANW.YY is not available.
376	BOF_FILE_SYSANW_FILE_CLOSE_ERROR	Diagnostic text file SYSANW.YY could not be closed.
377	BOF_FILE_SYSANW_FILE_OPEN_ERROR	Diagnostic text file SYSANW.YY could not be opened.
378	BOF_FILE_SYSANW_POSITION_ERROR	File positioning in SYSANW.YY could not be carried out.
379	BOF_FILE_SYSANW_READ_ERROR	Read function in the diagnosis text file "SYSANW.YY" could not be carried out.
380	BOF_FILE_SYSANW_FILE_TIMEOUT_ERROR	TIMEOUT when waiting for the MUTEX release for access to the SYSANW.YY files.
381	BOF_FILE_TXERR_FILE_NOT_FOUND_ERROR	Relevant TXERR.YY file not found.
382	BOF_FILE_TXERI_FILE_NOT_FOUND_ERROR	Relevant TXERI.YY file not found.
383	BOF_FILE_TXERI_FILE_OPEN_ERROR	TXERI.YY could not be opened.
384	BOF_FILE_TXERR_FILE_OPEN_ERROR	TXERR.YY could not be opened.
385	BOF_FILE_TXERR_FILE_POSITION_ERROR	File positioning in the text file "TXERR.YY" could not be carried out.
386	BOF_FILE_TXERR_FILE_READ_ERROR	Read function in the text file "TXERR.YY" could not be carried out.
387	BOF_COMMAND_NOT_AVAILABLE_DLL_MODE	The requested FI command not available for the "IfDIIMode=" set in the "INDRAMAT.INI" file.
388	BOF_NO_PARAMETER_SET_IN_CONTROL	No valid parameter record in the control unit.
389	BOF_MTA200_COMMANDLINE_ERROR	No valid command line for the MTA200 DRIVER.
390	BOF_FAR_DEVICE_STATUS_ERROR	No, or invalid, FARDEVICE entry.
391	BOF_DEVICE_PATH_ERROR	No, or invalid, device path entry.
392	BOF_DEVICE_PROTOCOL_ERROR	No, or invalid, device protocol entry.
393	BOF_DEVICE_IP_ERROR	No, or invalid, DEVICEIP entry.
394	BOF_DOS_NT_TASK_CHANNEL_TIMEOUT_ERRO R	Access to DOS \rightarrow NT job channel not possible in the pre-set time.
395	BOF_PROCESS_NAME_ERROR	A syntax error has been detected in the process name.
396	BOF_NETINTFC_MANAGER_MODE_ERROR	Invalid NETINTFC MANAGER MODE.
397	BOF_NET_MANAGER_STATUS_ERROR	Invalid NET MANAGER STATUS entered in the "IND_DEV.INI" file.
398	BOF_TERMINATE_EVENT_NOT_FOUND_ERRO R	No terminate event found for the registered TASK.
399	BOF_PARENT_WIN_ALREADY_EXIST_ERROR	PARENT WINDOW name already exists in the task management file.
400	BOF_NO_IFVERSION_ERROR	No "IfVersion=" entry exists in the "Indramat.INI" file.
401	BOF_NO_IFVERSION_ERROR	No IFVERSION entry in INDRAMAT.INI
402	BOF_NO_MTA200_INST_PATH	No MTA200 installation path found.
403	BOF_SYSANW_FILTER_FILE_CREATE_ERROR	Filter file SYSSTW.XX for SYSANW.XX (SHORT MESSAGES only!) could not be created.
404	BOF_FILTER_FILE_DIRECTORY_CREATE_ERRO R	The temporary sub-directory TEMPDATA could not be created for the data files of the small devices.



Code	Error Text	Name and Meaning of Error
405	BOF_DELETE_FILE_ERROR	Data file (small devices) cannot be deleted.
406	BOF_TXERR_FILTER_FILE_CREATE_ERROR	Filter file TXEST.XX for TXERR.XX (SHORT MESSAGES only!) could not be created.
407	BOF_STERR_FILTER_FILE_CREATE_ERROR	Filter file STESTyy.XX for STERRyy.XX (SHORT MESSAGES only!) could not be created.
408	BOF_TXERR_FILTER_FILE_NOT_FOUND_ERR OR	Filter file TXEST.XX for TXERR.XX (SHORT MESSAGES only!) does not exist in the temporary sub-directory TEMPDATA.
409	BOF_TXERR_FILTER_FILE_OPEN_ERROR	Filter file TXEST.XX for TXERR.XX (SHORT MESSAGES only!) could not be opened in the temporary sub-directory TEMPDATA.
410	BOF_TXEST_INDEX_FILE_CREATE_ERROR	INDEX file TXEST.XX (SHORT MESSAGES only!) could not be created.
411	BOF_BUFFER_LENGTH_ERROR	The PROCESSING BUFFER is too small for the data to be processed.
412	BOF_MSG_NUMBER_0_NOT_EXIST_ERROR	NO message number 0 exists in the message file.
413	BOF_MSG_NUMBER_TO_BIG_ERROR	Message number in message file is too big.
414	BOF_WRITE_FILE_ERROR	File could not be written.
415	BOF_SYSANW_FILTER_FILE_NOT_FOUND_ERRO R	Filter file SYSSTW.XX for SYSANW.XX (SHORT MESSAGES only!) does not exist in the temporary sub-directory TEMPDATA.
416	BOF_SYSSTW_INDEX_FILE_CREATE_ERROR	Index file SYSSTW.XX (SHORT MESSAGES!) could not be created.
417	BOF_SYSANW_FILTER_FILE_OPEN_ERROR	Filter file SYSSTW.XX for SYSANW.XX (SHORT MESSAGES only!) could not be opened in the temporary directory TEMPDATA.
418	BOF_STERR_FILTER_FILE_NOT_FOUND_ERR OR	Filter file STESTyy.XX for STERRyy.XX (SHORT MESSAGES only!) does not exist in the temporary directory TEMPDATA.
419	BOF_STESTYY_INDEX_FILE_CREATE_ERROR	The index file for STESTyy.XX (SHORT MESSAGES only!) could not be created.
420	BOF_STERR_FILTER_FILE_OPEN_ERROR	Filter file STESTYY.XX for STERRYY.XX (SHORT MESSAGES only!) could not be opened in the temporary sub-directory TEMPDATA.
421	BOF_WRONG_TELEGRAMM_CODE_ERROR	An INCORRECT TELEGRAM CODE has been returned by the control unit.
422	BOF_TXEST_INDEX_FILE_NOT_FOUND_ERRO R	Index file TXESI.XX could not be found.
423	BOF_TXEST_INDEX_FILE_OPEN_ERROR	Index file TXESI.XX could not be opened.
424	BOF_TXEST_INDEX_FILE_READ_ERROR	Index file TXESI.XX could not be read.
425	BOF_SYSSTW_INDEX_FILE_NOT_FOUND_ERR OR	Index file SYSSIW.XX could not be found.
426	BOF_SYSSTW_INDEX_FILE_OPEN_ERROR	Index file SYSSIW.XX is not open.
427	BOF_SYSSTW_INDEX_FILE_READ_ERROR	Index file SYSSIW.XX could not be read.
428	BOF_STESTXX_INDEX_FILE_NOT_FOUND_ERRO R	Index file STESIYY.XX could not be found.
429	BOF_STESTXX_INDEX_FILE_OPEN_ERROR	Index file STESIYY.XX could not be opened.
430	BOF_STESTXX_INDEX_FILE_READ_ERROR	Index file STESIYY.XX cannot be read.
431	BOF_DEVICE_TYPE_VALUE_TO_LARGE	DEVICE TYPE number is too large.

Code	Error Text	Name and Meaning of Error
432	BOF_NOT_ENOUGH_MEMORY_IN_CONTROL	The required memory is not available in the selected slot number.
433	BOF_TXEST_KENNUNG_FILE_CREATE_ERROR	The ID FILE for TXEST.XX (SHORT MESSAGES only!) could not be created (TXESK.XX).
434	BOF_TXEST_KENNUNG_FILE_OPEN_ERROR	The ID FILE for TXEST.XX (SHORT MESSAGES only!) could not be opened (TXESK.XX).
435	BOF_TXEST_KENNUNG_FILE_READ_ERROR	The ID FILE for TXEST.XX (SHORT MESSAGES only!) could not be read (TXESK.XX).
436	BOF_SYSSTW_KENNUNG_FILE_CREATE_ ERROR	The ID FILE for SYSSTW.XX (SHORT MESSAGES only!) could not be created (SYSSKW.XX).
437	BOF_SYSSTW_KENNUNG_FILE_OPEN_ERROR	The ID FILE for SYSSTW.XX (SHORT MESSAGES only!) could not be opened (SYSSKW.XX).
438	BOF_SYSSTW_KENNUNG_FILE_READ_ERROR	The ID FILE for SYSSTW.XX (SHORT MESSAGES only!) could not be read (SYSSKW.XX).
439	BOF_STESK_KENNUNG_FILE_CREATE_ ERROR	The ID FILE for STESTxx.YY (SHORT MESSAGES only!) could not be created (STESKxx.YY).
440	BOF_STESK_KENNUNG_FILE_OPEN_ERROR	The ID FILE for STESTxx.YY (SHORT MESSAGES only!) could not be opened (STESKxx.YY).
441	BOF_STESK_KENNUNG_FILE_READ_ERROR	The ID FILE for STESTxx.YY (SHORT MESSAGES only!) could not be read (STESKxx.YY).
442	BOF_COMPONENT_TYPE_STR_TO_LARGE	The component string in IND_DEV.INI is too large.
443	BOF_INVALID_COMPONENT_NUMBER_ERROR	Invalid component number.
444	BOF_DEVICE_COMPONENT_TYPE_REQUEST_ ERROR	INVALID DATA was returned by the interface on requesting the DEVICE COMPONENT TYPES.
445	BOF_DEVICE_DAT_FILE_NOT_FOUND_ERROR	Relevant DEVICE-DAT file not found for the BOF configuration.
446	BOF_MAIN_MENU_ITEM_ERROR	Invalid GUI main menu item.
447	BOF_MAIN_DEF_FILE_CONTENT_ERROR	BOF configuration file \MT_TEXTE\MAIN_DEF.INI not entered in sought device type.
448	BOF_DEVICE_INI_FILE_NOT_FOUND_ERROR	Relevant DEVICE-INI file not found for the BOF configuration.
449	BOF_DEVICE_INI_FILE_SYNTAX_ERROR	Format error in DEVICE-INI file for the BOF configuration.
450	BOF_DEVICE_POLLING_STATUS_ERROR	NO, or invalid, PollDeviceStatus in IND_DEV.INI.
451	BOF_DEVICE_POLLING_RATE_ERROR	NO, or invalid, PollDeviceStatusRate in IND_DEV.INI.
452	BOF_DEVICE_POLLING_CHECK_FACTOR_ERR OR	NO, or invalid, PollStatusCheckFactor in IND_DEV.INI.
453	BOF_DOS_BOF_EXE_SYNTAX_ERROR	NO "_" character may be included in DOS-BOF-EXE file names (WITH TSR connection).
454	BOF_DOS_BOF_EXE_CMDLINE_SYNTAX_ERROR	NO "_" character may be included in the call parameters for the DOS-BOF-EXE (WITH TSR connection).
455	BOF_SYS_MSG_LENGTH_ERROR	The additional information for the SYS message is too long.
456	BOF_DEVICE_STATUS_INFO_ERROR	More than one "critical" condition is managed in the DEVICE-STATUS INFO (SYSTEM MAP) e.g.: parameter download.
457	BOF_SYS_MSG_HOOK_LIST_TIMEOUT_ERROR	The SYS-MSG HOOK LIST cannot be accessed within the pre-set time.

Code	Error Text	Name and Meaning of Error
458	BOF_PROCESS_LOGOUT_TIMEOUT_NETINTFC	NETINTFC has not logged out from the TASK MANAGEMENT LIST within the pre-set DELAY TIME.
459	BOF_PROCESS_LOGOUT_TIMEOUT_DESKTOP	DESKTOP has not logged out from the TASK MANAGEMENT LIST within the pre-set DELAY TIME.
460	BOF_PROCESS_LOGOUT_TIMEOUT_ CONTROLDATA	CONTROL DATA has not logged out from the TASK MANAGEMENT LIST within the pre-set DELAY TIME.
461	BOF_PROCESS_LOGOUT_TIMEOUT_LOGDBCO	LOGDBCOM has not logged out from the TASK MANAGEMENT LIST within the pre-set DELAY TIME.
462	BOF_PROCESS_LOGOUT_TIMEOUT_MPI	MPI has not logged out from the TASK MANAGEMENT LIST within the pre-set DELAY TIME.
463	BOF_PROCESS_LOGOUT_TIMEOUT_BOFINTFC	BOFINTFC has not logged out from the TASK MANAGEMENT LIST within the pre-set DELAY TIME.
464	BOF_IF_DLL_MODE_TO_SMALL	IF-DLL MODE set is too small for the function to be executed.
465	BOF_WATCH_LIST_OVERRUN_ERROR	NO WATCH LIST available (overrun) for the selected device.
466	BOF_INVALID_WATCH_LIST_NUMBER_ERROR	INVALID WATCH LIST NUMBER for the selected DEVICE.
467	BOF_NO_SYSTEM_ERRORTEXT_ADM	There is NO management system for access to the SYSTEM ERROR TEXTS (SYSANW.XX)
468	BOF_NO_TX_ERRORTEXT_ADM	There is NO management system for access to the TRANSMISSION ERROR TEXTS (TXERR.XX)
469	BOF_NO_MECH_ERRORTEXT_ADM	There is NO management system for access to the MECHANISM ERROR TEXTS (STERRyy.XX)
470	BOF_INVALID_PLC_TYPE	An INVALID PLC type has been detected for the selected device.
471	BOF_SHMEM_INTERN_INDEX_ERROR	Maximum INTERNAL SHM INDEX is too big.
472	BOF_SHMEM_INDEX_TO_BIG_ERROR	SHM INDEX is too big.
473	BOF_INTERNAL_PROCESS_NUMBER_ERROR	NO INTERNAL PROCESS NUMBER (MT- NCPROZESSE : 06).
474	BOF_AXIS_NUMBER_NOT_DEFINED_ERROR	Requested AXIS NUMBER is not defined in the ACTUAL PARAMETER RECORD.
475	BOF_NO_PARAMETER_DOWNLOAD_FILE_EXI ST	NO PARAMETER DOWNLOAD FILE available.
476	BOF_PARAMETER_DOWNLOAD_FILE_LOAD_ ERROR	The PARAMETER DOWNLOAD FILE CANNOT be loaded.
477	BOF_PARAMETER_DOWNLOAD_FILE_ID_ERROR	A LENGTH ERROR has occurred in the various parameter IDs OR necessary entries are missing.
478	BOF_ALLOCATE_MEMORY_ERROR	NO memory could be allocated.
479	BOF_PARAMETER_DOWNLOAD_FILE_INDEX_E RROR	Max. DATA INDEX ENTRY in the PARAMETER DOWNLOAD FILE is NOT available.
480	BOF_PARAMETER_DOWNLOAD_DATA_TO_LON G	An individual PARAMETER DATA STRING to be passed is too long for the telegram.
481	BOF_PARAMETER_IDENTIFICATION_NOT_EXIS T	NO [ID_PARAMETER] section exists in the PARAMETER DOWNLOAD FILE.

Code	Error Text	Name and Meaning of Error
482	BOF_SYSTEM_PARAMETER_IDENTIFICATION_ NOT_EXIST	NO [ID_SYSTEM] section exists in the PARAMETER DOWNLOAD FILE.
483	BOF_SYSTEM_PARAMETER_DATA_NOT_EXIST	NO [DATA_SYSTEM] section exists in the PARAMETER DOWNLOAD FILE.
484	BOF_PROCESS_PARAMETER_DATA_NOT_EXI ST	NO [DATA_PROCESSx] section exists in the PARAMETER DOWNLOAD FILE.
485	BOF_AXIS_PARAMETER_DATA_NOT_EXIST	NO [DATA_AXISx] section exists in the PARAMETER DOWNLOAD FILE.
486	BOF_INVALID_PARAMETER_DATA	The control unit has detected INVALID PARAMETER DATA during PARAMETER DOWNLOAD.
487	BOF_PARAMETER_DATA_NOT_COMPLETE	INCOMPLETE PARAMETER DATA has been passed during a PARAMETER DOWNLOAD.
488	BOF_NECESSARY_FUNCTION_NOT_FOUND	The necessary FUNCTIONS are NOT contained in the selected DLL, or the DLL is not found.
489	BOF_PARAMETER_DATA_LINE_TO_MUCH	The NUMBER of DATA LINES in the PARAMETER DOWNLOAD FILE is too great.
490	BOF_PARAMETER_MAX_ALLOCATE_MEMORY_ ERROR	MEMORY REQUIREMENT for the INTERNAL DATA STRUCTURE during PARAMETER DOWNLOAD is too great.
491	BOF_PARAMETER_DOWNLOAD_BREAK_ERRO R	A PARAMETER DOWNLOAD PROCEDURE has been interrupted by BREAK-INFO.
492	BOF_PARAMETER_UPLOAD_FILE_ALREADY_EXIS T	PARAMETER UPLOAD FILE already exists.
493	BOF_PARAMETER_IDENT_SECTION_CREATE_ ERROR	NO [ID_PARAMETER] section could be created in the PARAMETER UPLOAD FILE.
494	BOF_PARAMETER_UPLOAD_DATA_WRITE_ ERROR	NO UPLOAD DATA could be written in the PARAMETER UPLOAD FILE.
495	BOF_SYSTEM_PARAMETER_IDENT_SECTION_ CREATE_ERROR	NO [ID_SYSTEM] section could be created in the PARAMETER UPLOAD FILE.
496	BOF_DATA_SYSTEM_SECTION_CREATE_ERROR	NO [DATA_SYSTEM] section could be created in the PARAMETER UPLOAD FILE.
497	BOF_MAX_INDEX_DATA_SECTION_CREATE_ ERROR	NO [MAX_INDEX_DATA] section could be created in the PARAMETER UPLOAD FILE.
498	BOF_ID_PROCESS_SECTION_CREATE_ERROR	NO [ID_PROCESSx] section could be created in the PARAMETER UPLOAD FILE.
499	BOF_DATA_PROCESS_SECTION_CREATE_ ERROR	NO [DATA_PROCESSx] section could be created in the PARAMETER UPLOAD FILE.
500	BOF_ID_AXIS_SECTION_CREATE_ERROR	NO [ID_AXISx] section could be created in the PARAMETER UPLOAD FILE.
501	BOF_SAVE_ARRAY_PROCESSING_ERROR	An error occurred when processing SAVE ARRAYS.
502	BOF_COM_INTERFACE_REQUEST_ERROR	Requested COM INTERFACE could NOT be returned.
503	BOF_COM_DIAG_SERVER_INIT_ERROR	An error occurred when INITIALIZING THE COM DIAG TEXT SERVER.
504	BOF_CO_INITIALIZE_ERROR	A co-initialize procedure has NOT been carried out in the user process.
505	BOF_COM_DIAG_TEXT_ACCESS_ERROR	An error occurred fetching a message text.
506	BOF_COM_INTERFACE_DIAG_SERVER_NULL_ ERROR	COM INTERFACE POINTER for accessing the DIAG SERVER is ZERO.
507	BOF_OBJECT_CREATE_ERROR	NO OBJECT could be generated – GENERAL CREATE ERROR.

Code	Error Text	Name and Meaning of Error
508	BOF_LANGUAGE_CONVERT_ERROR	A CONVERSION ERROR has occurred during the various conversions regarding language management.
509	BOF_NO_WRITE_DATA_AVAILABLE_ERROR	NO write value is available when calling the BW command.
510	BOF_TO_MANY_SPS_VARIABLES_ERROR	Too many PLC variables were transferred when calling up the BW-MKT command.
511	BOF_COM_INTERFACE_LANGUAGE_SERVER_ NULL_ERROR	COM INTERFACE POINTER for accessing the LANGUAGE SERVER is ZERO.
512	BOF_COM_CURRENT_LANGUAGE_ACCESS_ ERROR	Language extension CURRENTLY set could not be determined via LANGSUPP.
513	BOF_COM_CURRENT_CONTEXT_NAME_ERROR	The CURRENT CONTEXT NAME could NOT be determined.
514	BOF_SYS_MSG_NUMBER_RANGE_ERROR	SYS-MESSAGE number is NOT within the permissible value range – ONLY EVEN NUMBERS are permitted.
515	BOF_IF_STARTUP_TIMEOUT_ERROR	STARTUP of the BOFINTFC.EXE exceeds the max. permissible startup time.
516	BOF_TCP_IP_OBJECT_CREATE_ERROR	No further TCP-IP communication object can be created. The maximum number has already been opened.
517	BOF_TCP_IP_OBJECT_NOT_EXIST_ERROR	The TCP-IP communication object addressed does NOT exist.
518	BOF_SYSTEM_PARAMETER_AXIS_INFO_ERROR	The AXIS INFORMATION (AXIS TYPE, APR NUMBER) could NOT be converted in the parameter download file – FORMATTING ERROR!
519	BOF_NO_MACHINE_DATA_DOWNLOAD_FILE_ EXIST	MACHINE DATA DOWNLOAD FILE does not exist.
520	BOF_MACHINE_DATA_DOWNLOAD_FILE_LOAD_ ERROR	The MACHINE DATA DOWNLOAD FILE CANNOT be loaded.
521	BOF_MACHINE_DATA_IDENTIFICATION_NOT_ EXIST	NO [ID_MACHINE_DATA] section exists in the MACHINE DATA DOWNLOAD FILE
522	BOF_MACHINE_DATA_DOWNLOAD_FILE_ID_ ERROR	A LENGTH ERROR has occurred in the various machine data IDs OR necessary entries are missing.
523	BOF_MACHINE_DATA_TYPEDEF_INFORMATION _NOT_EXIST	[TYPE_DEFINITION_INFO] section does not exist in the MACHINE DATA DOWNLOAD FILE.
524	BOF_MACHINE_DATA_TYPEDEF_INFORMATION _NOT_EXIST	[TYPE_DEFINITION_INFO] section does not exist in the MACHINE DATA DOWNLOAD FILE.
525	BOF_MACHINE_DATA_TYPEDEF_INFORMATION _ERROR	A FORMATTING ERROR has occurred in the [TYPE_DEFINITION_INFO] section in the MACHINE DATA DOWNLOAD FILE.
526	BOF_MACHINE_DATA_TO_MUCH_TYPEDEF_ ERROR	Too many TYPE DEFINITIONS in the MACHINE DATA DOWNLOAF FILE.
527	BOF_MACHINE_DATA_TO_MUCH_TYPEDEF_ ERROR	Too many TYPE DEFINITIONS in the MACHINE DATA DOWNLOAF FILE.
528	BOF_MACHINE_DATA_PAGE_INFO_ERROR	NO [PAGE_INFO] section present in the MACHINE DATA DOWNLOAD FILE, or there is a formatting error.
529	BOF_MACHINE_DATA_TO_MUCH_PAGEDEF_ ERROR	Too many PAGE DEFINITIONS present in the MACHINE DATA DOWNLOAD FILE, OR the PAGE NUMBER is too high.
530	BOF_MACHINE_DATA_PAGE_DEFINITION_NOT_ EXIST	NO [PAGE_DEFINITION_XXX] section exists in the MACHINE DATA DOWNLOAD FILE, although a relevant [ID_PAGE_DEFINITION_XXX] section has been defined.



Code	Error Text	Name and Meaning of Error
531	BOF_MACHINE_DATA_PAGE_DEFINITION_ERROR	A FORMATTING ERROR has occurred in the [PAGE_DEFINITION_XXX] section in the MACHINE DATA DOWNLOAD FILE.
532	BOF_MACHINE_DATA_ELEMENT_DESCRIPTION_ NOT_EXIST	The [PAGE_DESCRIPTION_XXX_YYY] section which should be present based on the PAGE DEFINITION does NOT exist in the MACHINE DATA DOWNLOAD FILE – NO DATA ELEMENT DESCRIPTION exists.
533	BOF_MACHINE_DATA_ELEMENT_DESCRIPTION_ ERROR	A FORMATTING ERROR has occurred in the [PAGE_DESCRIPTION_XXX_YYY] section in the MACHINE DATA DOWNLOAD FILE.
534	BOF_MACHINE_DATA_PAGE_INFORMATION_ ERROR	A FORMATTING ERROR has occurred in the [PAGE_DATA_INFO] section in the MACHINE DATA DOWNLOAD FILE or necessary entries are missing.
535	BOF_MACHINE_DATA_PAGE_NOT_EXIST	NO [PAGE_DATA_XXX] section exists in the MACHINE DATA DOWNLOAD FILE
536	BOF_MACHINE_DATA_ELEMENT_INFORMATION _ERROR	A FORMATTING ERROR or logic error has occurred in the [PAGE_DATA_ELEMENTS_XXX] section in the MACHINE DATA DOWNLOAD FILE.
537	BOF_MACHINE_DATA_PAGE_LINE_ERROR	Either the requested data line does NOT exist in the [PAGE_DATA_XXX] section in the MACHINE DATA DOWNLOAD FILE, OR there is a formatting error in the data line.
538	BOF_MACHINE_DATA_VALUE_STRING_ CONVERT_ERROR	A MACHINE DATA STRING that is to be written CANNOT be converted – FORMATTING ERROR, or a formatting error has occurred in the parameter string of the MDS command.
539	BOF_MACHINE_DATA_VALUE_RANGE_ERROR	A VALUE RANGE ERROR has occurred when converting the MACHINE DATA STRING.
540	BOF_MACHINE_DATA_STRUCT_TO_LARGE_ERR OR	A DATA STRUCTURE that is too large is present in the MACHINE DATA that is to be written – this CANNOT be written as a COMPLETE DATA STRUCTURE.
541	BOF_MACHINE_DATA_PAGE_INFO_NOT_EXIST	NO [PAGE_INFO] section exists in the MACHINE DATA DOWNLOAD FILE
542	BOF_MACHINE_DATA_VALUE_STRING_TO_LONG	A data string that is TOO LARGE exists in the [PAGE_DATA_XXX] section in the MACHINE DATA DOWNLOAD FILE (max. 50 characters possible).
543	BOF_MACHINE_DATA_DOWNLOAD_BREAK_ERRO R	A MACHINE DATA DOWNLOAD PROCEDURE has been interrupted by BREAK-INFO.
544	BOF_MACHINE_DATA_VALUE_TO_MUCH_ERROR	Too many MACHINE DATA VALUES to be written are entered in the download file, or too many MACHINE DATA VALUES have been indicated in the MDS command.
545	BOF_GLOBAL_DATA_BUFFER_INDEX_TO_LARGE	The GLOBAL DATA BUFFER INDEX is outside the permissible range.
546	BOF_GLOBAL_DATA_BUFFER_CONTENT_TO_ LARGE	Too many BYTES are to be copied into the GLOBAL DATA BUFFER.
547	BOF_MACHINE_DATA_UPLOAD_FILE_ALREADY_ EXIST	PARAMETER UPLOAD FILE already exists.
548	BOF_NO_MACHINE_DATA_SET_IN_CONTROL	NO MACHINE DATA RECORD exists in the CONTROL.
549	BOF_MACHINE_DATA_UPLOAD_BREAK_ERROR	A MACHINE DATA UPLOAD PROCEDURE has been interrupted by BREAK-INFO.



Code	Error Text	Name and Meaning of Error
550	BOF_MACHINE_DATA_ELEMENT_NUMBER_TO_ LARGE	The pre-set MACHINE DATA ELEMENT NUMMER is outside the valid range [1110].
551	BOF_MACHINE_DATA_ELEMENT_COMPUTING_ ERROR	A computation error has occurred in computing the necessary number of telegrams for reading the machine data.
552	BOF_NO_TOOL_MANAGEMENT_EXIST	NO tool management has been switched on in the system parameters or process parameters.
553	BOF_NO_DIAG_SERVER_AVAILABLE	NO DIAG SERVER available.
554	BOF_UNKNOWN_MESSAGE_TYPE	An unknown message type has occurred during message download for the small devices via the MFD command.
555	BOF_MESSAGE_DOWNLOAD_BREAK_ERROR	A MESSAGE DOWNLOAD PROCEDURE has been interrupted by BREAK-INFO.
556	BOF_MACHINE_DATA_IDENT_SECTION_CREATE_ ERROR	NO [ID_MACHINE_DATA] section could be created in the MACHINE DATA UPLOAD FILE.
557	BOF_MACHINE_DATA_UPLOAD_DATA_WRITE_ ERROR	NO UPLOAD DATA could be written in the MACHINE DATA UPLOAD FILE.
558	BOF_MACHINE_DATA_ID_TYPE_SECTION_ CREATE_ERROR	NO [ID_TYPE_DEFINITION] section could be created in the MACHINE DATA UPLOAD FILE.
559	BOF_MACHINE_DATA_TYPE_INFO_SECTION_ CREATE_ERROR	NO [TYPE_DEFINITION_INFO] section could be created in the MACHINE DATA UPLOAD FILE.
560	BOF_MACHINE_DATA_TYPE_DEF_SECTION_ CREATE_ERROR	NO [TYPE_DEFINITION_XXX] section could be created in the MACHINE DATA UPLOAD FILE.
561	BOF_MACHINE_DATA_PAGE_INFO_SECTION_ CREATE_ERROR	NO [PAGE_INFO] section could be created in the MACHINE DATA UPLOAD FILE.
562	BOF_MACHINE_DATA_PAGE_ID_SECTION_ CREATE_ERROR	NO [ID_PAGE_DEFINITION_XXX] section could be created in the MACHINE DATA UPLOAD FILE.
563	BOF_MACHINE_DATA_PAGE_DEF_SECTION_ CREATE_ERROR	NO [PAGE_DEFINITION_XXX] section could be created in the MACHINE DATA UPLOAD FILE.
564	BOF_MACHINE_DATA_PAGE_DESCRIPTION_ SECTION_CREATE_ERROR	NO [PAGE_DESCRIPTION_XXX_YYY] section could be created in the MACHINE DATA UPLOAD FILE.
565	BOF_MACHINE_DATA_PAGE_DATA_INFO_ SECTION_CREATE_ERROR	NO [PAGE_DATA_INFO] section could be created in the MACHINE DATA UPLOAD FILE.
566	BOF_MACHINE_DATA_PAGE_DATA_ELEMENT_ SECTION_CREATE_ERROR	NO [PAGE_DATA_ELEMENTS_XXX] section could be created in the MACHINE DATA UPLOAD FILE.
567	BOF_MACHINE_DATA_PAGE_DATA_SECTION_ CREATE_ERROR	NO [PAGE_DATA_XXX] section could be created in the MACHINE DATA UPLOAD FILE.
568	BOF_LOGIN_COMINTFC_ERROR	An attempt has been made to log in the COMINTFC PROZESS ROOM.
569	BOF_LOGIN_LOGINTFC_ERROR	An attempt has been made to log in the LOGINTFC PROZESS ROOM.
570	BOF_PARAMETER_UPLOAD_BREAK_ERROR	A PARAMETER UPLOAD PROCEDURE has been interrupted by BREAK-INFO.
571	BOF_HOMATIC_DRIVER_DLL_NOT_FOUND_ ERROR	The HOMATIC DRIVER DLL (INDIFY00.DLL) could NOT be found.
572	BOF_HOMATIC_DRIVER_DLL_COULD_NOT_BE_ LOAD_ERROR	The HOMATIC DRIVER DLL (INDIFY00.DLL) could NOT be loaded.
573	BOF_HOMATIC_DRIVER_DLL_FUNCTION_LOAD_ ERROR	The HOMATIC DRIVER DLL (INDIFY00.DLL) does NOT contain ALL THE NECESSARY functions.
574	BOF_TIME_DATE_SET_STATUS_ERROR	NO, or invalid, TimeDateSetStatus entry in IND_DEV.INI.



Code	Error Text	Name and Meaning of Error
575	BOF_TIME_DATE_SET_STATUS_RATE_ERROR	NO, or invalid, TimeDateSetStatusRate entry in IND_DEV.INI.
576	BOF_NO_PCL_PROGRAMM_IN_CONTROL	NO valid PLC program exists in the control.
577	BOF_FI_START_DISPLAY_ERROR	Invalid FIStartDisplay entry in IND_DEV.INI.
578	BOF_FI_START_DISPLAY_MODE_ERROR	NO, or invalid, FIStartDisplayExtendedMode entry in IND_DEV.INI.
579	BOF_INVALID_DEVICE_SIMULATION_ERROR	Invalid DeviceSimulation entry in IND_DEV.INI.
580	BOF_NO_SIMULATION_DEVICE_TYPE_EXIST_ ERROR	NO SIMULATION is possible for the pre-selected device address.
581	BOF_SOFT_MONITOR_START_MODE_ERROR	Invalid SoftMonitorMode entry in IND_DEV.INI.
582	BOF_SIMISP_START_MODE_ERROR	Invalid SimIspMode entry in IND_DEV.INI.
583	BOF_SEKTION_NOT_FOUND_ERROR	The desired SECTION NAME is not AVAILABLE in the file to be modified
584	BOF_SIMTRA_START_MODE_ERROR	Invalid SimTraMode entry in IND_DEV.INI.
585	BOF_DESTINATION_FILE_ALREADY_EXIST_ ERROR	The destination file already exists on file copying
586	BOF_PARAMETER_OFFLINE_FILE_LOAD_ERROR	The File ParamOff.dat cannot be loaded
587	BOF_PARAMETER_OFFLINE_FILE_ID_ERROR	A LENGTH ERROR has occurred in the various parameter IDs OR necessary entries are missing.
588	BOF_PARAMETER_ONLINE_FILE_LOAD_ERROR	The File ParamOn.dat cannot be loaded
589	BOF_PARAMETER_ONLINE_FILE_ID_ERROR	A LENGTH ERROR has occurred in the various parameter IDs OR necessary entries are missing.
590	BOF_PROCESS_TYPE_ERROR	Invalid process type
591	BOF_PARAMETER_DATA_NOT_FOUND_ERROR	In the parameter data file, necessary parameter data are not available or not correct
592	BOF_DEBUG_DEVICE_MODE_ERROR	Invalid DeviceDebugMode entry in IND_DEV.INI.
593	BOF_FI_COMMAND_LENGTH_ERROR	The FI command string is too long
594	BOF_GENERAL_FILE_NOT_FILE_ERROR	General: file NOT found
595	BOF_FI_COMMAND_DEVICE_STATUS_ERROR	There is an invalid DeviceStatus for the FI command
596	BOF_OPERATING_SYSTEM_NOT_SUPPORTED_ ERROR	The operating system is NOT supported by FI.
597	BOF_SOFTMONITOR_SHUTDOWN_ERROR	Invalid SoftMonitorShutDown entry in IND_DEV.INI.
598	BOF_STARTED_PROCESS_TERMINATE_ERROR	The PROCESS to be started from FI BEFORE the READY MESSAGE
599	BOF_COMM_ADDRESS_TYPE_ERROR	The communication type is invalid
600	BOF_INVALID_AXIS_MEANING_INFO	The axis meaning is invalid
601	BOF_PAR_MIN_NUMBER_ERROR	Necessary PARAMETERS in the FI requirement command are not available
602	BOF_IND_DEV_READ_ERROR	IND_DEV.INI can NOT be read in with the PROFILE CLASS
603	BOF_PROFILE_SECTION_NOT_FOUND_ERROR	The searched SECTION is NOT available in the profile
604	BOF_FI_ERROR_TEXT_INFO_ERROR	Invalid FiErrorTextInfo entry in IND_DEV.INI.
605	BOF_DEVICE_ADDRESS_CONSISTENT_ERROR	The selected DEVICE ADDRESS in the FI command does NOT correspond to the DEVICE ADDRESS in the binary telegram.



Code	Error Text	Name and Meaning of Error
606	BOF_WRONG_DEVICE_PROTOCOL_ERROR	A false data protocol was chosen for the selected DEVICE ADDRESS
607	BOF_TOOL_POSITION_TO_LARGE_ERROR	The information on max. tool storage is too large according to the process parameters.
608	BOF_FI_COMMAND_STACK_FULL_ERROR	There is NO more free space in the FI command stack
609	BOF_FI_COMMAND_STACK_SOURCE_DATA_ LENGTH_ERROR	The SOURCE data is too long for the FI command stack
610	BOF_FI_COMMAND_STACK_RESULT_DATA_ LENGTH_ERROR	The RESULT data is too long for the FI command stack
611	BOF_FI_COMMAND_STACK_INDEX_OUT_OF_ RANGE_ERROR	Invalid FI command stack index – out of range
612	BOF_FI_COMMAND_STACK_ERROR	Invalid FiCommandStack entry in IND_DEV.INI.
613	BOF_FI_COMMAND_STACK_RATE_ERROR	NO, or invalid, FiCommandStackRate entry in IND_DEV.INI.
614	BOF_DEVICE_POLLING_OFF_ERROR	NO device polling is switched on in IND-DEV.INI
615	BOF_TIMEOUT_FOR_FURTHER_INFO_FROM_ DEVICE_ERROR	In the selected delay time, the controller did NOT supply the additional information
616	BOF_ERROR_TEXT_NUMBER_CONVERT_ERROR	The error number can NOT be converted through sscanf()
617	BOF_ERROR_TEXT_NOT_FOUND_ERROR	The error number can NOT be resolved in an error text
618	BOF_MACHINE_DATA_PAGE_SIZE_ERROR	A defined PAGE is larger than 64Kbyte
619	BOF_RECEIVED_TELEGRAMM_LENGTH_TO_ LARGE_ERROR	The received telegram data is too long
620	BOF_SPS_VARIABLE_NAME_LENGTH_ERROR	The names of the PLC variables are too long
621	BOF_LOGDBCOM_TIMEOUT_ERROR	LOGDBCOM could NOT be started in the preselected delay time
622	BOF_SSCANF_CONVERT_ERROR	Conversion through sscanf() is NOT possible
623	BOF_PROVI_ADM_FILE_LOAD_ERROR	The PROVI administration file can NOT be read
624	BOF_PROVI_ADM_FILE_ALREADY_EXIST	The PROVI administration file already exists
625	BOF_PROFILE_SECTION_CREATE_ERROR	The SECTION to be written could NOT be generated through the profile class
626	BOF_PROVI_TEXT_FILE_NOT_FOUND_ERROR	The PROVI TEXT FILE does NOT exist
627	BOF_PROVI_INDEX_FILE_NOT_FOUND_ERROR	The PROVI INDEX FILE does NOT exist
628	BOF_PROVI_TEXT_FILE_OPEN_ERROR	The PROVI TEXT FILE can NOT be opened
629	BOF_PROVI_INDEX_FILE_OPEN_ERROR	The PROVI INDEX FILE can NOT be opened
630	BOF_PROVI_MESSAGE_FILE_CREATE_BREAK_ ERROR	The generation of the PROVI MESSAGE FILES was interrupted by BREAK INFO
631	BOF_PROVI_MESSAGE_ACCESS_ERROR	General access error with PROVI MESSAGES via the DIAG SERVER
632	BOF_PROVI_MESSAGE_TYPE_ERROR	There is an Invalid PROVI MESSAGE TYPE
633	BOF_PROVI_ADM_FILE_DATA_ERROR	Necessary data is NOT available in the PROVI administration file
634	BOF_PROVI_ADM_FILE_SIZE_ERROR	There are too many data entries in the PROVI administration file
635	BOF_PROVI_TEXT_FILE_NOT_EXIST_ERROR	The selected PROVI message text file is NOT available



Code	Error Text	Name and Meaning of Error
636	BOF_NO_MEMORY_EXIST_IN_SLOT_NUMBER_ ERROR	There is NO free memory in the requested SLOT NUMBER
637	BOF_PROVI_MESSAGE_FILE_DOWNLOAD_ BREAK_ERROR	The download of the PROVI MESSAGE FILES was interrupted by BREAK INFO
638	BOF_M_KEY_ADM_TIMEOUT_ERROR	The interlocking mutex for the M key administration could NOT be assigned in the preselected time
639	BOF_WRITE_DATA_TO_LONG_ERROR	On calling the BW command. the write value is too long
640	BOF_SERCOS_CHANNEL_ERROR	Access via the SERCOS channel NOT possible
641	BOF_DATE_TIME_READ_ERROR	The information on date and time could NOT be read from the controller
642	BOF_DATE_TIME_STRING_ERROR	There is a FORMAT error in the transferred date/time string
643	BOF_SYNAX_IN_SLAVE_MODE_ERROR	The addresses SYNAX controller is in SLAVE mode
644	BOF_PC_TIME_DATE_SET_STATUS_ERROR	Invalid PCTimeDateSetStatusFromDeviceAddr entry in IND_DEV.INI.
645	BOF_PC_TIME_DATE_SET_RATE_ERROR	NO or invalid PCTimeDateSetStatusRateFromDeviceAddr entry in IND_DEV.INI
646	BOF_PC_TIME_DATE_SET_DEVICE_ADDR_ERROR	NO or invalid PCTimeDateSetFromDeviceAddr entry in IND_DEV.INI
647	BOF_PCI_DRIVER_DLL_NOT_FOUND_ERROR	The PCI DRIVER DLL (PCIDP_IF32.DLL) could NOT be found
648	BOF_PCI_DRIVER_DLL_COULD_NOT_BE_LOAD_ ERROR	The PCI DRIVER DLL (PCIDP_IF32.DLL) could NOT be loaded
649	BOF_PCI_DRIVER_DLL_FUNCTION_LOAD_ERROR	The PCI DRIVER DLL (PCIDP_IF32.DLL) does NOT CONTAIN ALL NECESSARY functions
650	BOF_PCI_OBJECT_NOT_EXIST_ERROR	The addresses PCI communication object does NOT exist
651	BOF_TRIGGER_EVENT_COUNTER_TO_MUCH_ ERROR	A maximum of 1280 event handles are admissible in the trigger list
652	BOF_INVALID_VISUAL_MOTION_ASCII_ PROTOCOL_ERROR	There is an invalid VISUAL-MOTION ASCII request
653	BOF_INVALID_VISUAL_MOTION_ERROR_TEXT	There is an invalid VISUAL-MOTION ERROR TEXT answer
654	BOF_VISUAL_MOTION_ERROR	There is a VISUAL MOTION ERROR

8.3 Error Codes 1000 to 1999

Code	Error Text	Name and Meaning of Error
1001	BOF_FAULT_FCT	Invalid function code passed (e.g. "CW" for a read function).
1002	BOF_DATA_FAULT	Data is invalid.
1003	BOF_FAULT_PIPE_NR	Incorrect pipe number
1004	BOF_NO_CREATED_PIPE	Pipe not created.
1005	BOF_PIPE_NOT_RUN	Pipe not running.
1006	BOF_NO_DATA_CREATED	Data not created.
1007	BOF_PIPE_NOT_BREAK	Pipe not running.



Code	Error Text	Name and Meaning of Error
1008	BOF_NO_VALUE	No value string.
1009	BOF_BUFFER_SIZE_TO_SMALL	Buffer is too small.
1010	BOF_NO_INDEX_DATA	No index data.
1011	BOF_FAULT_INDEX_NR	No index number.
1012	BOF_DATA_NO_FOUND	Data not found.
1013	BOF_FUNC_LOCK	Function blocked; repeat access.
1014	BOF_NEGATIVE_ACKNOWLEDGE	Negative acknowledge for the FI command executed.
1015	BOF_PARAMETER_INVALID	Invalid parameter details.
1016	BOF_FUNCTION_INVALID	Invalid FI command.
1017	BOF_DEVICE_TIMEOUT	Timeout of NC-Task
1018	BOF_INDEX_DATA_ERROR	Index data from the resultbuf is corrupt.
1019	BOF_UNKNOWN_TOOL_STORE	Unknown type of memory (tool store)!= magazine, spindle, gripper.
1020	BOF_MAX_COUNT_ERROR_FOR_TOOL_DATA	Maximum count error for tool data.
1021	BOF_NO_TOOLMANAGMENT	No tool management.
1022	BOF_NO_TOOLMANAGMENT_FOR_PROCESS	Tool management not available for process.
1025	BOF_RESULT_BUF_TYPE_ERROR	Error result type is incorrect or not supported.
1030	BOF_NC_PACKET_IS_PRESENT	NC package already present in control.
1031	BOF_NC_PARTPROGRAM_IS_NOT_PRESENT	NC program is not present.
1032	BOF_NC_PROGRAM_DIRECTORY_IS_EMPTY	Part-directory or program directory is empty.
1033	BOF_NC_PROGRAM_COMPILER_ERROR	Error flag set by program.
1034	BOF_NC_DAT_FILE_NO_PRESENT	NC-DAT file does not exist or cannot be opened.
1035	BOF_NC_PACKET_DIR_NOT_PRESENT	Package directory does not exist.
1036	BOF_NC_PACKET_DIR_READ_ERROR	Package directory can not be read in.
1037	BOF_NC_PARTPROGRAM_DIR_NOT_PRESENT	Program directory does not exist.
1038	BOF_NC_PARTPROGRAM_DIR_READ_ERROR	Program directory can not be read in.
1039	BOF_PIPE_CYCLE_LIST_EMPTY	Pipe request list is empty.
1040	BOF_PIPE_RUN	Pipe already running.
1041	BOF_ITEM_DATA_INVALID	Partial result is invalid.
1042	BOF_FUNC_INVALID_PARAM	Invalid parameter for function
1043	BOF_PIPE_NO_FREE_PIPE	All pipes already assigned.
1044		Communication channel is already running.
1301	Exception	
1302	No "Common" information section present	
1303	No "Common" information key present	
1304	FI job interrupted	
1305	Download control file cannot be read	
1310	No "Package" information section present	
1311	No "Package" information key present	
1312	"Package" key value not permitted	
1315	No ListOfPrograms section present	

Code	Error Text	Name and Meaning of Error
1321	No NC program section present	
1322	Program key value not permitted	
1323	No NC program info present	
1324	Process number does not conform	
1325	Data file cannot be read	
1331	Parameter record not active	
1332	No memory for creating object	
1333	Invalid value passing	
1334	Invalid access mode	
1340	No valid function call	
1341	No COMPILER info section present	
1342	No COMPILER info key present	
1350	No variable/events/D-correction section present	
1351	Variable/event/D-correction data file cannot be read	
1355	No events section present	
1360	No D-correction section present	
1361	No D-correction parameter record present	
1370	Invalid transfer of parameters	
1371	Invalid key	
1372	Invalid PLC variables type	
1373	Telegram delimiter reached	
1374	Configuration file does not exist	
1375	Section "project" does not exist	
1376	Key "project" does not exist	
1377	Section "variables" does not exist	
1378	Key "variables" does not exist	
1379	Section "buffer" does not exist	
1380	Key "buffer" does not exist	
1381	Invalid parameter transfer at "buffer" section	
1382	Invalid parameter transfer at "buffer" section	
1383	Section "trigger\condition" does not exist	
1384	Key "trigger\condition" does not exist	
1390	No I/O tables section	
1391	I/O tables value do not exist	
1392	Short identification is invalid	
1501	BOF_FUNC_NAME_LIMIT150	Name of interface 'B' functions is too large.
1502	EXCEPTION	Internal error.
1503	EXCEPTION	Internal error.
1504	EXCEPTION	Internal error.
1505	EXCEPTION Internal error.	



Code	Error Text	Name and Meaning of Error
1506	EXCEPTION	Internal error.
1507	EXCEPTION	Internal error.
1508	EXCEPTION	Internal error.
1509	EXCEPTION	Internal error.
1510	EXCEPTION	Internal error.
1511	EXCEPTION	Internal error.
1512	BOF_FUNC_EOF_STRING_150	FI command incomplete.
1513		Maximum number of lines has been reached
1514		Maximum number of columns has been reached

8.4 Error Codes 2000 to 2999

Code	Meaning
2001	No channel free.
2002	Channel already open.
2003	Channel cannot be closed.
2004	Channel not open.
2005	Re-initialization error.
2006	Channel cannot be opened.
2007	Version is incompatible to file "LOGINTFC.EXE".
2008	Channel flags are blocked.
2009	Access to controls temporarily blocked due to download.
2010	Receive request timeout.
2011	No request active.
2012	Invalid event in receive.
2013	Status request still active.
2014	Cyclic request still active.
2015	No cyclic request active.
2016	Single request still active.
2017	Pass format of routine "GetSysMsg" is faulty.
2018	System message (SysMsg) cannot be issued.
2019	DMA request is still active.
2020	Invalid FI command code.
2021	Invalid result type.
2022	Result too long for receive buffer.
2023	Invalid FI command during group request.
2024	Empty result buffer
2025	Request too long for request buffer.
2026	Faulty input format.
2050	"LOGINTFC.DAT" file cannot be opened.
2051	No channel free.



Code	Meaning	
2052	Communication process (COM task) not responding.	
2053	"LOGINTFC.EXE" file not found.	
2059	Error message from the LOG process.	
2060	False syntax in the command string	
2081	The addressed controller is off line (DeviceStatus=OFF)	
2082	The addresses controller is in monitor mode	
2126	Faulty result type	
2127	Faulty request on response	
2150	"LOGINTFC.DAT" file cannot be opened.	
2154	File Version Mismatch.	
2155	"LOGINTFC.DAT" file is too large.	
2156	Internal configuration error.	
2157	Faulty ChFreeList from GetDeviceCommAddrExtend()	
2160	Invalid command string.	
2161	Telegram code not implemented.	
2162	Parameter outside the limit value.	
2163	Invalid parameter syntax.	
2164	Unknown PLC variable.	
2165	Not enough parameters transmitted.	
2166	PLC map file cannot be opened.	
2167	PLC variable type not implemented.	
2168	PLC variable reference error.	
2169	Date cannot be edited.	
2170	Checksum error.	
2171	Undefined telegram code.	
2172	Missing processing rule.	
2173	Too much data for the response telegram.	
2174	Unknown additional diagnostics information.	
2175	Unknown unit.	
2176	PLC variable is larger than 240 byte.	
2177	Device has no PLC	
2178	NDC parameter too large	
2179	Result buffer too small	
2180	No axis defined	
2181	The addressed controller is off line (DeviceStatus=OFF)	
2182	The addresses controller is in monitor mode	
2201	Input string "Date-Time" not in format: "DD.MM.YY hh:mm:ss".	
2202	Effective data length of SIS telegram is too large.	
2304	Specified file not found.	



8.5 Error Codes 4000 to 4999

Code	Error Text	Name and Meaning of Error
4000		An error has been detected in checking the composition of the request of the "BR_NPA1" and ff command. (see also FI command: "NPA1_/?") The following error messages in the error window provide additional information regarding the error.
	ERROR : invalid ParNo/value	An incorrect parameter number has been transmitted.
	ERROR : invalid ParNo/value	An error has been detected in checking transmission of the parameter. The possible cause of this is an invalid parameter name or an error in the order in which the entry was made. The first parameter number must be smaller than the second parameter number. (see also FI command: "NPA1_/?")
	invalid Parametervalue or No.:[<parnr>]</parnr>	An error has been detected in checking the command. Either a directory number has been selected that is outside of the range of validity or a parameter name is invalid.
	[No.] missing Startparameter	The command has not been passed on in its entirety.
	Illegal start parameter value [wrong value passed]	An incorrect value has been detected for the parameter number.
	ERROR : different Parametertypes	Requesting different types of parameter within one request command is not possible.
	ERROR : Second ParNo before First ParNo	The parameter request must be made starting from the lower number and moving to the higher number. (see also FI command: "NPA1_/?")
	ERROR : Invalid startparameter - ProcNo out of Range	When requesting one or more process parameters, an invalid definition range has been detected. A request is only possible within the NC process numbers [06].
	ERROR : Invalid startparameter - AxesNo out of Range	When requesting one or more axis parameters, an invalid definition range has been detected. A request is only possible within 1 to 20 or 32.
4001	ERROR : invalid function	The FI command contains an invalid parameter.
4002	NO_PARAMETER_DATA_FOUND	The requested parameter(s) do not exist. Either parameters have been requested that have not already been defined or the appropriate parameter has been removed. Check all entries and make sure that the corresponding data exists in BOF menu item <f5> (Parameters).</f5>
4003	Verz_No_Out_of_Range	An invalid range has been detected when checking the command passed. Check the directory number entries.
4004	BR_NPA_No_Data_File_exists	The attempt to read data from a file could not be executed. Re-check your entries for possible processes or axes on the definition range. Otherwise, try to view the data using BOF menu item <f5> (Parameters). The data may not exist or the installation has not been made correctly . In this case, please contact our customer service department.</f5>

Code	Error Text	Name and Meaning of Error
4005	BR_NPA_No_INI_File_exists	Parameter data could not be read from an initialization file. Possible causes are:
		The file does not exist. There has been an installation error or the file has been deleted accidentally. \Rightarrow Execute Update/Installation
		• There is an error in the file. The file has been accidentally edited or illegally copied. Data recognition has thereby been rendered invalid.
		\Rightarrow Carry out an update installation or contact our customer service department.
		• The file has been damaged, either by a system crash or by a defect on the storage media.
		\Rightarrow Contact our customer service department.
4006	Device Address out of Range	A system outside the definition range has been selected in the command.
4007	Buffer error detected =[Error Code]	Internal error. The data range set for provision of the results is not large enough. This problem can be remedied as follows:
		Request fewer data. Use a group request.
		• Increase the memory made available for the data range when creating the application yourself.
		\Rightarrow Contact our customer service department.
4012	Create_DLL_Error detected!	The result buffer could not be initialized. Contact our customer service department.
4013	Function will not run for DLL-Version- Mode:[DLL-Version]	An attempt has been made to execute a command that is not available in the existing DLL version.
4014	Corrupted Parameter Identification = [Parameterident.]	Initialization of the required data memory is not possible die to an error in parameter recognition. Check to make sure that there is a valid parameter record for all devices. If necessary, re-transmit the parameter(s) to the controls. If the error remains, or the parameter(s) cannot be transmitted, then please contact our customer service department.
4015	Wrong version installed	This error message appears always appears on starting the GUI when the memory could not be initialized based on the version being used. Up to and including version 18, error code 4109 is returned. From version 19, the corrected error code 4015 is returned.
4017	**OK** (none Parameterset in CNC) – finished function FillParamDataInCncDataMap	This text message only appears in the starting-up phase with the setting "/U0" of the start parameter (for TSRPG25I.EXE) if an empty parameter name has been transmitted. This means that no parameter record is as yet in the control. No error is returned.
4100	Couldn't open ParameterIndexFile: [File Error=xxxx]	 An error has been detected when attempting to open the parameter directory file. Any of the following could trigger this error: Versions do not conform
		⇒ Parameters need to be converted. (refer to Parameters)
		• The parameter directory file has been accidentally destroyed.
		The disk drive is faulty.
4102	ParameterIndexFile has wrong structure	An error has been detected when reading the parameter directory file indicating that the data in the file is not in the correct format. Check this by running Converting Parameters. If the error continues to occur after this then you must contact our customer service department.



Code	Error Text	Name and Meaning of Error
4103	to much Indizies found – File has wrong structure ?:	An error has been detected when creating the directory data. More directories have been recognized than allowed by the definition range. Probably the parameter directory data is from an earlier version. Execute Converting Parameters. If the error persists, please contact our customer service department.
4104	Invalid parameter value detected	An invalid range was detected when initializing (booting up the GUI). Contact our customer service department.
4105	Can't create Parameterindexbuffer: [filename]	No data could be provided in the memory. Close other applications to free up enough memory for the compilation of the data.
4108	Don't found the Parameter: [Parameternummer]	This message text only appears in the starting phase with the setting "/U0" (in case of TSRPG25I.EXE). This error code is only returned when an attempt has been made to request a non-defined parameter.
4109	Didn't get BOF-Version – BOF installed? [error code]	The attempt to determine the GUI version has failed. Contact our customer service department.
4110	Couldn't load Parameters in shared Memory – Error= [ErrorCode]	Initialization failed when starting the GUI. Contact our customer service department.
4111	Invalid parameter value Cxx.053 [Cxx.053 <value>]</value>	Initialization failed when starting the GUI. An invalid axis meaning has been detected in the current parameter record of a device. Switch the corresponding system to offline and correct the appropriate parameter record. After you have done this, the system should be brought back online and the altered parameter record should then once more be transmitted to the controls. If the problem persists, please contact our customer service department.
4200	Invalid start parameter	This message text only appears in the starting phase with the setting "/U0" (in case of TSRPG25I.EXE). This error code is always returned when a parameter request has been made outside the definition range. Otherwise, please contact our customer service department.
4201	Invalid parameter type	A parameter request has been made with a non-defined parameter type. Check the entry and/or request
4202	Buffer size not enough	The result of the parameter request cannot be transmitted as the transmission range is not large enough. For applications that you have created yourself, increase the size of the transmission range. Otherwise, please contact our customer service department.
4203	Error detect by ReadPar_Value – can't read Data [Error number or directory number]	The requested parameter could not be formed or found. Re- check your request or contact our customer service department.
4204	Could not find direct. entry	No error message is emitted. The error code is always returned when, after a request for a particular parameter directory entry, the parameter number has not been found.
4205	Function will not running by InterFace-Ver- sion: [Version]	During the command request, the program has detected that it cannot be run on this version. Contact our customer service department.
4220	Invalid Save Order by Save function please test the ParType by Save_Begin;	The "writing parameters" function has been repeatedly started before the previously started command has been completed.
4221	invalid IndexNo by Save[ParameterNumber]	The parameter number is outside the definition range.
4222	co_str_ConWData_Buffer_Size_to_small [defined size 2000]	This message text only appears in the starting phase with the setting "/U0" (for TSRPG25I.EXE). The error code is always returned if the defined memory range in the program is too small. In this case, please contact our customer service department.



Code	Error Text	Name and Meaning of Error
4223	WriteError by Config-SCR-File = [error number]	An error has been detected when writing the configuration parameters. The function has been cancelled. Contact our customer service department.
4224	SaveError – Couldn't rename DAT->old [file name]	This message text only appears in the starting phase with the setting "/U0" (for TSRPG25I.EXE). The attempt to rename the original file could not be executed. Check the properties for the relevant parameter file and also the remaining free space on the storage medium.
4225	SaveError – Couldn't rename tmp \rightarrow dat \Rightarrow copy old to DAT[Dateiname]	This message text only appears in the starting phase with the setting "/U0" (for TSRPG25I.EXE). The attempt to recopy the newly created file could not be executed. Check the properties for the relevant parameter file and also the remaining free space on the storage medium.
4226	Missing file = [file name]	The previously created file could not be found or opened. Check the free memory on the storage medium.
4227	Create instance failed - can't save Data	An internal error has occurred. Contact our customer service department.
4228	Can't create file = [file name <additional info="">]</additional>	The specified file could not be created acc. to the additional info. Check the amount of free memory on the storage medium and the access properties of the directory concerned. Otherwise, please contact our customer service department.
4229	Can't create file = [file name]	Specified file could not be found. Check your entry. Perhaps an incorrect directory number has been entered.
4230	ConWData_Error_by_WPar_Begin	This error code is reported after an error has occurred in transmitting an error code to the function interface.
4231	ConWData_Error_by_WPar_End	This error code is reported after an error has occurred in transmitting an error code to the function interface.
4232	Error detect by WritePar_Value - can't save Data:[file name or parameter value]	This error message is only displayed internally when in debug mode. The value of a parameter could not be transferred to the file specified. Contact our customer service department.
4233	Attention - Return value of process definition undefined	This error message is only displayed internally when in debug mode. An error has been detected in the generation of the process definition. Check the process definitions within the processing of the parameter. Otherwise, please contact our customer service department.
4234	Can't actualize direct.line [Parameter directory line]	The specified parameter directory line could not be updated.
4235	Can't actualize date or length in directory line	The date or length could not be updated when the parameter directory line was being updated. Contact our customer service department.
4236	CreateFiErrorResult_DLL failed	This error message is only displayed internally when in debug mode. The error message could not be transmitted to the FI. Contact our customer service department.
4237	Can't write by undefined parameter number [Parameter number]	An attempt has been made to write a non-defined parameter for this type of parameter. Check your entry. Check, e.g. that parameters exist for the various axis types.
4238	Cxx.083 : more as defined Elements for Cxx.083 found:	An attempt has been made to transmit a larger number of compensation values than is listed as the max. range of a compensation list. Re-check your entry. A maximum of 1000 values can be included in a list.
4239	Installation error - missing file: [File name]	Specified file not found. Re-run the update installation. If the error persists after this then you must contact our customer service department.



Code	Error Text	Name and Meaning of Error
4240		An invalid range has been detected in the specified parameter line. Re-check your entry.

Converting Parameters

An update installation of the Rexroth Indramat GUI automatically results in a parameter conversion from version "xx" to the next version "yy". Parameters can be converted by calling the "COPAxxyy.EXE" conversion program in standard installation directory "C:\Programme\Indramat\MTGUI\Bin". Both wildcards "xx" and "yy" represent the directory ID from which version and into which version the conversion is carried out.

Note: In case of an error, you can start the conversion program "COPAxxyy.EXE" with the starting parameter "/?" to receive additional messages.

8.6 Error Codes 5000 to 5999

Code	Name and Meaning of Error
5001	Alias used is not defined (is not yet used)
5002	Invalid device address
5003	Syntax error in the device address
5007	Invalid device address
5010	No request active
5011	Error in request string
5012	No response buffer specified
5013	Wrong mode during cyclic login (internal error)
5014	Invalid receive telegram
5015	Invalid receive telegram
5050	No access to remote PC possible
5051	Remote connection has been canceled
5052	Network interface cannot be initialized
5101	Unexpected general error (internal error)
5102	Memory error
5151	No memory for data to be sent
5152	No memory for data to be received
5153	No memory for telegram to be received
5401	Unspecific error
5402	Invalid parameter passed to function
5403	Transfer timeout, remote PC not ready, or network connection down
5404	Send failed; error sending to a remote PC
5405	Memory shortage; in remote access of the interface
5406	Invalid connection to a remote PC
5407	Service disabled



Code	Name and Meaning of Error	
5408	Connection to remote partner aborted	
5409	Invalid parameter hook ID; Sys Message Handling	
5410	Invalid device number	

8.7 Error Codes 6000 to 6999

Code	Error Text	Name and Meaning of Error
6001	BOF_C_TYP_FAULT	Transmitted data type not OK.
6002	BOF_C_LEN_FAULT	Transmitted data length not OK.
6003	BOF_C_DEV_FAULT	Transmitted system number not OK.
6004	BOF_C_PAKNR_FAULT	Transmitted package number not OK.
6005	BOF_C_PROZ_FAULT	Transmitted process number not OK.
6006	BOF_C_PROG_FAULT	Transmitted program number not OK.
6007	BOF_C_FILE_NOT_DEL	File cannot be deleted
6008	BOF_C_NO_NCPROG_CREATED	No NC program in part-program directory
6009	(BOF_C_NCPROG_CREATED)	NC program exists (where check =1)
6010	BOF_C_DESCR_FAULT	Identifier, e.g. data length not OK
6011	BOF_C_FILE_WRITE_CLOSE_ERROR	Error writing or closing a file.
6012	BOF_C_PACK_EXIST	NC package already available
6013	BOF_C_INVALID_MTCNC_NUMBER	Invalid system number
6014	BOF_C_FILE_NOT_FOUND	File not found
6015	BOF_C_PAR4_FAULT	Parameter 4 not OK
6016	BOF_C_NO_NC_SEEK_SET	NC program cannot be positioned to N0000
6017	BOF_C_NCPROG_NOT_READ	File cannot be opened
6018	BOF_C_PART_PROGR_DIRECTORY_ERROR	Part-program directory could not be read.
6019	BOF_C_PACKET_DIRECTORY_ERROR	Package program directory could not be read.
6020	BOF_C_PAR5_FAULT	Parameter 5 not OK
6021	BOF_C_PAR6_FAULT	Parameter 6 not OK
6022	BOF_C_COMP_ERROR	Test error after commands to be compiled.
6023	BOF_C_CURS_FILE_ERROR	Handling error in NCCPxx.DAT file.
6024	BOF_C_TOOL_SETUP_LIST_NOT_READ	Error in setup list
6025	BOF_C_TOO_MUCH_TOOLS_IN_LIST	More tools in the setup list than in the parameters.



8.8 Error Codes 7000 to 7999

All error codes – except for error code 7000, which shows a syntax error in the compiled NC program – normally require you to contact Rexroth Indramat for further clarification of their cause. Either this is a software error or files for the GUI have been deleted or corrupted.

Note: As for all error codes, additional information regarding an error that has occurred can be requested via the "General error result line" (see chapter entitled "Error Codes"). The error information informs the user in plain text about the cause of the error.

Code	Meaning and Notes Regarding Diagnosis and Troubleshooting	
7000	Syntax error in NC program. The "General Error Result Line" contains further information.	
7002	File with incorrect information. The "General Error Result Line" contains the file name and the line.	
7005	File not found. The "General Error Result Line" contains the file name.	
7006	File cannot be created. The "General Error Result Line" contains the file namen.	
7008	File cannot be read. The "General Error Result Line" contains the file name.	
7009	Error in connecting the function interface. No connection can be made to the device (control unit) via the function interface.	
7015	Too many axes defined. More than 9 Axes used in NC process.	
7016	Invalid number of parameters. The number of parameters in the "NCPRG.CFG" file has been exceeded.	
7017	Axis name is invalid. The axis name in axis parameter "CXX.001" or "CXX.075" is invalid.	
7018	Axis meaning is invalid. The axis meaning in axis parameter "CXX.053" is invalid.	
7019	Maximum axis speed is invalid. The value of axis parameter "CXX.016" is invalid.	
7020	Maximum axis acceleration is invalid. The value of axis parameter "CXX.018" is invalid.	
7021	Lowest run time of an NC record = [2.530ms]. The counter value of the parameter "METB" in the NC options of the BOF/GBO is outside the allowed range.	
7022	Lowest run time of an NC record is invalid. The counter value of the parameter "METB" in the NC options of the BOF/GBO is invalid.	
7023	Only 4 or 5 decimal places are allowed. Process parameter "BXX.002" is invalid.	
7024	Invalid counter value. The counter value of parameter "VFBT" or "BBTC" in the NC options of the BOF/GBO is invalid.	
7025	Only 0 (mm) or 1 (inch) permitted! The process parameter "BXX.001" is invalid.	
7026	Counter value outside the allowed range. Axis parameter "CXX.006" is smaller than 0.1.	



Code	Meaning and Notes Regarding Diagnosis and Troubleshooting	
7027	Internal block number is invalid. The block numbers of the NC program file are in the wrong order.	
7028	Block number in the file is invalid. The "General Error Result Line" (p. 8-1) contains the names of the file in which the block numbers are incorrect.	
7070	Counter value outside the permitted range (110). The counter value of parameter "BBTC" in the NC options of the BOF/GBO is outside the permitted range.	
7077	Counter value outside the permitted range (1.0.25). The counter value of parameter "VFBT" in the NC options of the BOF/GBO is outside the permitted range.	
7083	Invalid parameter. The "General Error Result Line" contains an invalid control parameter.	

8.9 Error Codes 8000 to 8999

Code	Error Text	Name and Meaning of Error
8000	OUTOFMEMORY	Heap memory is full
8001	PARAMETER_FAILURE	Error in transmitting parameter (response telegram)
8002	INVALIDARG	Incorrect request string
8003	REQUEST_NOT_FILLED	Internal run error
8004	GET_ATTRIBUT_FAILED	Incorrect attribute contained in response telegram
8005	WALK	Internal run error
8006	EXTRACT_COMMON_INFO_FAILED	Error in transmitting parameter (response telegram)
8007	WRONG_DATA_SIZE	Undefined data length in the response
8008	ELEMENT_UNEXPECTED	Unexpected coding in BW_SPA1
8009	SERCOS_LONG_TO_ASCII	Result conversion error.
8010	VERSION_MISMATCH	Command did not yet exist for set IfDIIMode.
8011	ERROR_BYTE_INFO	Error reading error byte information
8012	CANT_OPEN_MODULDEF_INI	The "Moduldef.ini" file cannot be opened.
8013	WRONG_PROFILE_FILENAME	Wrong profile file name
8014	WRONG_SECTION_INFORMATION	Wrong section information in profile
8015	ERROR_IN_LAST_LINE	Error in the last profile line
8016	Reserved	Reserved
8017	Reserved	Reserved
8018	SECTION_NOT_FOUND	Section not found; (e.g., incorrect device or module parameter).
8019	LANGUAGE_NOT_FOUND	Language not supported
8020	Reserved	Reserved
8021	MODUL_NOT_FOUND	Module not found; (e.g., missing keyword module name).
8022	DEVICE_ADDR_GENERAL_NOT_FOUND	No device entry found.
8023	FB_NOT_FOUND	No function component found; (e.g., error or message keyword missing).
8024	DEVICE_ADR_FALSE	Device address not in the valid range.
8025	MODULE_NO_FALSE	Module number not within valid range (0-99)

Code	Error Text	Name and Meaning of Error
8026	KEY_WORD_FALSE	Wrong keyword; (e.g., no ModulY in section name [DeviceAddrX\ModulY])
8027	MODULE_ASSIGN_PROCESS	No module can be found for the specified process.
8028	PROCESS_NO_FALSE	Process number not within valid range (0-31)
8031	RESULT_TYPE_INVALID	Invalid result type.
8032	E_COM_SIS_TEL_TOO_LONG	Transmitted length of telegram exceeds maximum SIS telegram length.
8033	E_COM_SIS_TEL_POS	Telegram position addressed is outside the SIS telegram range.
8034	E_COM_SIS_TEL_NO_LEN	SIS telegram length is "0"
8035	E_COM_OPERATING_SYSTEM_NOT_ SUPPORTED	Operating system is not supported.
8036	SERCOS_ASCII_TO_LONG	Error in converting the value to be written.
8038	PROCESS_NOT_DEFINED	The process addressed does not exist
8039	NO_TOOLMANAGEMENT	The tool management is not activated for the process
8040	WRONG_TOOL_NUMBER	Invalid tool number
8041	WRONG_SPINDLE_NUMBER	Wrong spindle number
8042	WRONG_GRIPPER_NUMBER	Wrong gripper number
8043	UNKNOWN_TOOL_STORE	Unknown tool store (memory)
8044	INVALID_VALUE	Value or element of the value list not correctly formatted
8045	MUTEX_TIMEOUT	The command access control was not quit in time
8046	UNKNOWN_DEVICETYPE	An unknown device type has been detected

8.10 Error Codes 10000 and above

Code	Meaning	
10001	The WinHMI component is not installed.	
10002	Incorrect WinHMI version	
10021	DDS not installed	
10022	Incorrect DDS version	
10101	Incorrect version of the function interface.	
10102	The "CreateGroup" routine has failed.	
10103	Error in command string.	
10104	Unknown variable requested.	
10105	Error in determining the status.	
10107	"HMI_Data.DLL" file not found.	
10110	WinHMI has not been started in the same process	
10111	Error on determination of the ProVi message.	
10112	ProVi message type does not exist.	
10201	DDS not ready yet.	
10210	Unknown error	
10211	Unknown function has been requested	
10212	Function of this controller is not available.	
10221	Internal error.	
10222	Invalid module number.	
10223	Faulty message ID.	
10224	Invalid diagnosis type	
10225	Incorrect detail type	
10226	Unknown detail	
10227	Unknown step.	
10228	Unknown sequencer.	
10229	Sequencer is not disturbed.	
10230	Step is not disturbed.	
10231	Unknown variable requested.	
10232	Unknown error ID.	
10233	Unknown message number.	
10234	Retain variable up or download error	
10235	Retain variable up or download message	

8.11 Error Codes 35000 and above

Code	Error Text	Name and Meaning of Error
35500	PARA_NOT_DEFINED	CMOS parameter not yet defined. Recovery: Describe CMOS parameter.
35501	VALUE_TYPE_INVALID	Invalid coding type.
35502	ERROR_VERSION_MISMATCH	Command does not yet exist for set IfDIIMode.

8.12 Error Codes 100000 and above

Code	Error Text	Name and Meaning of Error	
100101	FS_NO_TEXT_FILE_ACCESS	Record file is opened in text mode.	
100102	FS_REC_SIZE_TO_SMALL	Invalid record size	
100103	FS_REC_FILE_BOUND_ERROR	Invalid file position	
100104	FS_NO_CREATE_OBJECT	An interface object could not be created.	
100105	FS_ERROR_SIM5	Without "iMTc" ID	
100106	FS_ERROR_FILETYP	Different file type	
100107	FS_ERROR_FILEVERSION	Current file version is larger than file version	
100108	FS_ERROR_FILELENGTH	Current file length != ID length	
100109	FS_ERROR_FILEDATE	File date != ID date	
100110	FS_ERROR_FILETIME	File time != ID time	
100111	FS_ERROR_FILENAME	File name != ID name	
100112	FS_ERROR_CHECKSUM	Checksum is incorrect	
100113	FS_ERROR_FILE_NOT_EXIST	File does not exist	
100114	FS_ERROR_FILE_MIN_LENGTH	File with ID must be at least 65 byte.	
100115	FS_ERROR_T04	Without "iT04" ID	
100116	FS_ERROR_FILE_NOT_OPEN	File cannot be opened.	
100117	FS_ERROR_NO_SIGN	File has no ID (sign)	
100118	FS_ERROR_MMIVERSION	GUI version is smaller than file version.	



8.13 Error Codes 110000 and above

Code	Error Text	Name and Meaning of Error	
110001	BOF_MAP_VERSION_FUNC_ERROR	Incorrect DII mode set	
110002	BOF_MAP_FILE_VERSION_ERROR	Incorrect file version number	
110003	BOF_MAP_LONGID_VERSION_ERROR	If the long ID version is invalid	
110004	BOF_MAP_LONGID_INVALID_ERROR	If the long ID is invalid	
110005	BOF_MAP_LONGID_PARAM_ERROR	Missing parameter in split long ID	
110006	BOF_MAP_COMMON_ERROR	Error not clearly defined	
110007	BOF_MAP_FILE_NOT_OPEN	File could not be opened.	
110008	BOF_MAP_FILE_IS_OPEN	File is already open.	
110009	BOF_MAP_PLAUSIBLE_TEST_ERROR	Plausibility test of map file long ID is negative.	
110010	BOF_MAP_ID_ERROR	Long ID comparison is negative.	
110011	BOF_MAP_TO_MANY_IMPORT_TAB_ENTRIES	Too many import table entries (>65535).	
110012	BOF_MAP_INVALID_DATA	MAP file contains invalid data.	
110013	BOF_MAP_PARAMETER_INVALID	Missing parameters for a function.	
110014	BOF_MAP_INVALID_DEVICE_NO	Transmitted device number does not agree with the system number in the MAP.	
110015	BOF_MAP_INVALID_STATUS	Invalid access status	
110016	BOF_MAP_ACCESS_ERROR	Access to a MAP when MAP has not been loaded, incorrect DeviceNo	
110017	BOF_MAP_NO_LOAD_ERROR	MAP file is not loaded internally	
110018	BOF_MAP_NO_LOAD_MAPFILE_ERROR52	MAP file is not loaded with error 52	
110019	BOF_MAP_NO_LOAD_MAPFILE_ERROR52	MAP file is not loaded with error 55	
110020	BOF_MAP_MAPFILE_INVALID_VERSION_ERROR56	MAP file has invalid version 56	
110021	BOF_MAP_VARIABLE_NO_FOUND_ERROR	1346 \rightarrow variable not found.	
110022	BOF_MAP_LONGID_DIFFERENT_TO_MAP12	MAP file long ID is different from PLC long ID	
110023	BOF_MAP_INVALID_ARRAY_INDEX50	Invalid array index	
110024	BOF_MAP_INVALID_STRING_INDEX51	Invalid string index	
110025	BOF_MAP_NO_CREATED_MAP_ACCESS	MAP access has not been generated	
110026	BOF_MAP_LONGID_INVALID_NO_MAP_ERROR	Long ID is not valid and no PLC MAP access has yet been initialized.	
110027	BOF_MAP_OUTOFMEMORY	No more memory available for creating object	
110028	BOF_MAP_STRUCT_ELEMENT_NO_FOUND	Structure element does not exist	
110029	BOF_MAP_STRUCT_ELEMENT_NO_FOUND	Global management information has not been created	
110030	BOF_MAP_DOWNLOAD_STATUS	Access to MAP during a download	
110031	BOF_MAP_NO_DOS_PCL	System is not DOS - PLC	
110032	BOF_MAP_NO_WIN_PCL	System is not WIN - PLC	
110033	BOF_MAP_EXCEPTION	Exception has been triggered while determining PLC address	
110034	BOF_WINPCL_INSTANCE	WinPcl object instance could not be created	
110035	BOF_WINPCL_CREATE	Error creating access to WinPcl	



Code	Error Text	Name and Meaning of Error	
110036	BOF_WINPCL_INIT	Error initializing WinPcl; current PLC long ID differs from database	
110037	BOF_WINPCL_ACCESS	WinPcl access object has not been created	
110038		Parameter too small	
110040		An administration error occurred	
110041		Initialization of the WinPCL address server not possible	
110050	BOF_MAP_COMMON_FILETOOL_ERROR	Basic number cErrorGroup_filetool	
110100	BOF_MAP_COMMON_MAP_BAS_C_ERROR	Basic number ErrorGroup_map_bas_c	
110150 BOF_MAP_COMMON_LKENN_ERROR Basic numbe		Basic number cErrorGroup_Ikenn	
110200	BOF_MAP_COMMON_GROUP_ERROR	Basic number general error	
110263	BOF_MAP_VARIABLE_NO_FOUND_ERROR_ BASE+13	PLC variable does not exist; to clearly identify the error, the error number is added to the BASE.	
 110296	 BOF_MAP_VARIABLE_NO_FOUND_ERROR_ BASE+46		

8.14 Error Codes 210000 and above

Code	Meaning		
210917	String is too long		
210920	String does not begin with '		
210921	String does not end with '		
210923	Counter value has been exceeded		
210924	Minimum counter value not achieved		
210925	Incorrect counter format		



8.15 SERCOS Error

Code	Error Messages in Serial Protocol		
0x0000	No error in NC/MMI service channel.		
0x0001	NC/MMI service channel not opened.		
0x0009	Incorrect access to Element 0.		
0x0090	The control is currently busy. The request is not possible at the moment. Please try again later.		
0x00A0	"invalid request" e.g. access to S-/P parameter in initialization mode.		
0x00B0	"invalid element" Only the operating date element is valid for write access.		
0x00C0	"invalid drive address" The drive address is larger than allowed or the drive is not active within the SERCOS ring (deactivated or does not exist).		
0x00F0	"Fatal software error" An internal CLC error has occurred during parameter transmission (see <u>C-0-0041</u>), that has affected the exchange of data.		
0x1001	IDN does not exist.		
0x1009	Incorrect access to Element 1.		
0x13E8	Transmission error.		
0x13E9	Drive does not exist.		
0x13EA	Cancellation of data transmission when requested.		
0x13EB	Request data channel is closed.		
0x13EC	System faults		
0x2001	Name does not exist.		
0x2002	Name transmitted too short.		
0x2003	Name transmitted too long.		
0x2004	Name cannot be changed.		
0x2005	Name currently write-protected.		
0x3002	Attribute transmitted too short.		
0x3003	Attribute transmitted too long.		
0x3004	Attribute cannot be changed.		
0x3005	Attribute currently write-protected.		
0x4001	Unit does not exist.		
0x4002	Unit transmitted too short.		
0x4003	Unit transmitted too long.		
0x4004	Unit cannot be changed.		
0x4005	Unit currently write-protected.		
0x5001	Minimum input value does not exist.		
0x5002	Minimum input value transmitted too short.		
0x5003	Minimum input value transmitted too long.		
0x5004	Minimum input value cannot be changed.		
0x5005	Minimum input value currently write-protected.		



Code	Error Messages in Serial Protocol		
0x6001	Maximum input value does not exist.		
0x6002	Maximum input value transmitted too short.		
0x6003	Maximum input value transmitted too long.		
0x6004	Maximum input value cannot be changed.		
0x6005	Maximum input value currently write-protected.		
0x7002	Date transmitted too short.		
0x7003	Date transmitted too long.		
0x7004	Date can not be changed.		
0x7005	Date currently write-protected.		
0x7006	Date smaller than min. input value.		
0x7007	Date larger than max. input value.		
0x7008	Incorrect date.		
0x7009	Date is write-protected by password.		
0x700A	The operating date is currently write-protected as it has been configured cyclically (IDN is configured with MDT or AT; therefore, writing via the service channel is not allowed)		
0x700B	Invalid list element IDN is not supported, value outside of the input limits).		
0x700C	Operating data write-protected at the moment because of other settings (e.g. parameter, operating mode, drive release, drive ON, etc.).		
0x700D	"Length of date cannot currently be changed" The length of the date cannot be changed in the current mode.		
0x700E	"Length of the date cannot currently be changed" The length of the date is permanently write-protected.		
0x7010	Command already active		
0x7011	Command cannot be interrupted		
0x7012	Command cannot be executed at present (e.g. command cannot be activated in this phase)		
0x7013	Command cannot be executed (invalid or incorrect parameters)		
0x710C	Date outside of figure range		
0x710D	Length of date cannot be changed at present		
0x710E	Length of date can not be changed.		
0x8001	"Service channel is currently assigned (BUSY)" The required access is not currently possible as the service channel is assigned. Data transmission is not executed.		
0x8002	"Fault in service channel" Access to the required drive is not currently possible.		
0x800B	Transmission has been cancelled by the control unit as it must currently communicate with the same drive (higher priority).		
0x800C	Unauthorized access (service channel is still active); last transmission has not yet been completed and a new request has been started.		

8.16 Global SERCANS Error

The global SERCANS errors are not directly related to the message transmitted. These are fatal communication errors that result in the breakdown of communication with one or more drives.

The following global SERCANS er	rror codes have been defined:
---------------------------------	-------------------------------

Code	Error Messages in Serial Protocol		
0x8006	HS timeout		
0x8007	Double AT breakdown.		
0x8008	Optical waveguide ring not closed.		
0x8009	Optical waveguide ring interrupted.		
0x800A	"Test operation: zero bit current or continuous light". Test operation is set on the SERCANS assembly in order to check the optical transmission route on the SERCOS interface.		
0xC001	Invalid command control word.		
0xC002	IDN is not a command.		
0xC003	Command channel cannot currently be activated.		
0xD001	Drive error (status class 1, S-0-0011).		
0xD004	Command cannot be executed in drive.		
0xF001	"Configuration error". An error occurred on configuration of the setpoint or actual value channel: a) too many setpoint or actual values have been configured, or the setpoint or actual values are not supported.		
0xF002	"Error in calculating time slot" a) Telegram configured is too long b) Communication cycle time is too short		
0xF003	Incorrect phase details from the NC		
0xF004	"Error in life counter". The control no longer accesses the DPR of SERCANS cyclically.		
0xF005	SERCANS: internal error		
0xF006	"Copy times too long". The copy times of the command values and actual values taken together are larger than the time between the end of the last ATs and the beginning of the MDTs.		
0xF007	Checksum error (Y parameter).		
0xF008	Breakdown of input signal SYNCIN		
0xF009	Error in storing the system parameter or the system parameter has been changed. A check of the min/max values failed.		
0xF00A	Parameter is write-protected.		

8.17 Structure of Error File after Download

If an error is generated during downloading, this error is recorded in an error file and the *DownloadError* key is set to the value "YES" in the download file. The error file corresponds to the download file with the ending ".ERR".

An error is assigned to the section in which it occurs. The following error keys are set.

XX indicates a serial number

Key: ErrorLine_XX Line in which the error has occurred (optional).

Key: ErrorLink_XX Shows the connection with the error.

Key: ErrorSection_XX Identifies the section in which the error has occurred.

Key: ErrorText_XX Error text generated during the program flow.

Key: ErrorToken_XX Error number

Key: TextFileName_XX Name of the text file used (without language extension).

Examples for the MWCX Device Group

NC Cycle Download: CCA	<u>Example:</u> Download file [Common] ErrorSection_01= ListOfCycPrograms ErrorToken_01 = 1014
	[ListOfCycPrograms] ErrorLink_01= K:\Program Files\Indramat\Mtgui\Project_000\Cyc-Prg-00- 01.dat ErrorToken_01 = 1014
	Beispiel: NC cycle program file [Data] ErrorColumn_01 = 7 ErrorLine_01 = 6 ErrorText_01 = Format error in the NC program ErrorToken_01 = 1014
NC D-Correction Download: DCA	Example: Download file [Common] ;Errors of a general kind. Cannot be assigned to a specific section. E.g., Compiler not found ErrorToken_01 = 1234 ErrorText_01=."Download error" ErrorSection_01= DCorrection_1 [DCorrectionPackage_Info] ErrorToken_01=5678



	ErrorText_01=Missing info value		
	[DCorrection_1] ErrorToken_01=1014 ErrorText_01=Transmission error		
NC Program Download: NCA	Example: Download file [Common] ErrorSection_01= ListOfNCPrograms ErrorToken_01 = 1014		
	[ListOfNCPrograms] ErrorLink_01= K:\Program Files\Indramat\Mtgui\Project_000\\NC-PRG- 00-01.Dat ErrorToken_01 = 1014		
	<u>Example:</u> NC program file [Data]		
	ErrorColumn_01 = 7 ErrorLine_01 = 6		
	ErrorText_01 = Format error in the NC program		
	$ErrorToken_01 = 1014$		
NC Events Download: NEA	Example: Download file		
	[Common]		
	;Errors of a general kind. Cannot be assigned to a specific section. E.g., Compiler not found		
	$ErrorToken_01 = 1234$		
	ErrorText_01=."Download error"		
	ErrorSection_01= NCEvents_1		
	[NCEventsPackage_Info]		
	ErrorToken_01=5678		
	ErrorText_01=Missing info value		
	[NCEvents_1]		
	ErrorToken_01=1014		
	ErrorText_01=Transmission error		
NC Zero Point Download: NUA	Example: Download file		
	[Common]		
	;Errors of a general kind. Cannot be assigned to a specific section. E.g., Compiler not found		
	ErrorToken_01 = 1234		
	ErrorText_01=."Download error"		
	ErrorSection_01= OffsetDataPackage_Info		
	[OffsetDataPackage_Info]		
	ErrorToken_01=5678		
	ErrorText_01=Missing info value		
	[OffsetData_0\0\0] ; process 0, zero point data base 0, axis X		



ErrorToken_01=1014 ErrorText_01=Transmission error

NC Variables Download: NVA

Example: Download file [Common] ;Errors of a general kind. Cannot be assigned to a specific section. E.g., Compiler not found ErrorToken_01 = 1234 ErrorText_01=."Download error" ErrorSection_01= NCVariables_1

[NCVariablesPackage _Info] ErrorToken_01=5678 ErrorText_01=Missing info value

[NCVariables_1] ErrorToken_01=1014 ErrorText_01=Transmission error



9 Answers to Frequently Asked Questions

9.1 Function Interface FAQs

In this chapter, you'll find a collection of **F**requently **A**sked **Q**uestions from our customers' feedback on the Rexroth Indramat Function Interface.

- **Question 1** A message box appears when starting my application. Has the message box been issued by the function interface?
 - **Note:** As message boxes are entered in the Windows NT Task Manager as "applications", it is easy to see what has actually issued the message box.
 - Answer To do this, open Windows NT Task Manager e.g. using key combination: <Strg>+<Shift>+<Esc>

Highlight the message box entry in the "applications" tab page and click with the right-hand mouse button.

Note: The key combination <Ctrl>+<F10> does not work here for the right mouse button!

Select the "Switch to Process" command in the context menu that opens for the highlighted object.

If one of the following processes is displayed

- LOGINTFC.exe
- BOFINTFC.exe
- COMINTFC.exe

then this is a basic process of the function interface.

- Question 2 Can group requests also be issued via the "DataTransfer" routine?
 - Answer No, the "DataTransfer" routine is only for issuing single read or write requests. Group requests are issued via the routines for cyclic reading via pipes.
- **Question 3** Why does the login procedure for my application to the function interface take so long?
 - **Answer** During the function interface initialization phase numerous safety checks are carried out.



9.2 Windows NT FAQs

This chapter contains FAQs regarding Windows NT from customer feedback.

- Question 1 How can I log in with my name and password automatically (AutoLogin)?
 - Answer You must make the following entries in the Windows NT registry using the registry editor "REGEDT32" under key

HKEY_LOCAL_MACHINE\ Software Microsoft\ Windows NT\ Current Version Winlogon:

Value	Туре	Content	Info
AutoAdminLogon	REG_SZ	1	Switch AutoLogin on/off
DefaultUserName	REG_SZ	<user name=""></user>	User login name
DefaultPassword	REG_SZ	<password></password>	User password (a password must exist)
DefaultDomainName	REG_SZ	<domain name=""></domain>	Login must be carried out on another computer

Note: No further message box will appear. If you want to log in using another name then you must keep the <Shift> key pressed during the starting procedure. You will now be prompted to enter your name and password.

If no password is entered in the registry then AutoLogin will only function once and Windows will then reset "AutoAdminLogon" to "0". Entering the password is absolutely essential. Please note that the password is then visible in the registry for anyone to see!



10 Reference to Literature

10.1 Information in Rexroth Indramat Literature

[1] More detailed information regarding acceleration value and value range is contained in the Rexroth Indramat documentation:

NC Programming Instructions, chapter entitled "Interpolation Requirements/ Programmable Acceleration ACC", DOK-MTC200-NC**PRO*Vxx-AW0x-DE.

[2] More detailed information regarding the arguments of the trigonometric functions is contained in the Rexroth Indramat documentation:

NC Programming Instructions, chapter "Angle Dimension for Trigonometrical Functions RAD, DEG", DOK-MTC200-NC**PRO*Vxx-AW0x-DE.

[3] More detailed information regarding the axis speeds is contained in the Rexroth Indramat documentation:

NC Programming Instructions, chapter entitled "Interpolation Functions/ Linear Interpolation, Rapid Traverse Rate G00", DOK-MTC200-NC**PRO*Vxx-AW0x-DE

Parameter Description, chapter entitled "Maximum Track Speed", DOK-MT*CNC-PAR*DES*Vxx-AW0x-DE.

[4] More detailed information regarding the structure of an NC block is contained in the Rexroth Indramat documentation:

NC Programming Instructions, chapter entitled "Elements of an NC Block",

DOK-MTC200-NC**PRO*Vxx-AW0x-DE.

[5] Additional information regarding the reference spindle as well as NC programming of the cutting speed is contained in the Rexroth Indramat documentation:

NC Programming Instructions, chapter entitled "Spindle Speed, Constant Cutting Speed G96 / Selection of Reference Spindle SPF", DOK-MTC200-NC**PRO*Vxx-AW0x-DE.

[6] More detailed information regarding D-corrections is contained in the Rexroth Indramat documentation:

"NC Programming Instructions Vxx", chapter entitled "D-Corrections", DOK-MTC200-NC**PRO*Vxx-AW0x-DE.

[7] More detailed information regarding events and their treatment is contained in the Rexroth Indramat documentation:

"NC Programming Instructions Vxx", chapter entitled "Events", DOK-MTC200-NC**PRO*Vxx-AW0x-DE.

[8] More detailed information regarding tool management is contained in the Rexroth Indramat documentation:

"NC Programming Instructions Vxx", chapter "Commands for Tool Management / Cutter Selection E", DOK-MTC200-NC**PRO*Vxx-AW0x-DE.



[9] More detailed information regarding feedrate override is contained in the Rexroth Indramat documentation:

"CNC/PLC Interface Description xxVRS", chapter entitled "Feedrate and Spindle Override PxxCSOVRD", DOK-MTC200-SPS*GWY*Vxx-AW0x-DE.

[10] More detailed information regarding the feedrate is contained in the Rexroth Indramat documentation:

"CNC NC Programming Instructions Vxx", chapter entitled "Feedrate", DOK-MTC200-NC**PRO*Vxx-AW0x-DE.

[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 Rexroth Indramat documentation:

"NC Programming Instructions Vxx", chapter entitled "Table of G-Code Groups", DOK-MTC200-NC**PRO*Vxx-AW0x-DE.

[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 Rexroth Indramat documentation:

"NC Programming Instrucitons Vxx", chapter entitled "Table of M Function Groups", DOK-MTC200-NC**PRO*Vxx-AW01x-DE.

[13] More detailed information regarding the diagnostics system and the accompanying types of message is contained in the Rexroth Indramat documentation:

"xxVRS GUI", Application Description, Chapter 3 "Diagnostics", DOK-MTC200-GBO*GEN*Vxx-AW0x-DE.

[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 Rexroth Indramat documentation:

"MTC200/MT-CNC MCI Operating Instructions xxVRS", chapter entitled "Machine Parameters", DOK-MTC200-GBO*MCI*Vxx-AW0x-DE

"Parameter Description", DOK-MT*CNC-PAR*DES*Vxx-AW0x-DE.

[15] More detailed information regarding the elements of an NC record and the note is contained in the Rexroth Indramat documentation:

"CNC NC Programming Instructions Vxx", chapter entitled "NC Word", DOK-MTC200-NC**PRO*Vxx-AW0x-DE.

[16] More detailed information regarding the display possibilities within user interfaces, as well as the definition of axis data, is contained in the Rexroth Indramat documentation:

"MTC200/MT-CNC xxVRS GUI", chapter "Survey of Axis Data", DOK-MTC200-GBO*GEN*Vxx-AW0x-DE.

[17] More detailed information regarding the NC data structure is contained in the Rexroth Indramat documentation:

"NC Programming Instructions Vxx", chapter entitled "Program and Data Organization", DOK-MTC200-NC**PRO*Vxx-AW0x-DE.

[18] More detailed information regarding the rapid override is contained in the Rexroth Indramat documentation:

"CNC/PLC Interface Description xxVRS", chapter entitled "Feedrate and Spindle Override"; "Rapid Override PxxCSOVRD", DOK-MTC200-SPS*GWY*Vxx-AW0x-DE.

[19] Additional information regarding the selection of the reference spindle in the NC program is contained in the Rexroth Indramat documentation:

"NC NC Programming Instructions Vxx", Application Description, chapter entitled "Spindle Speed", "Selecting the Reference Spindle SPF"

DOK-MTC200-NC**PRO*Vxx-AW0x-DE.

[20] Additional information regarding the selection of the spindle speed in the NC program is contained in the Rexroth Indramat documentation:

"NC Programming Instructions Vxx", chapter entitled "Additional Functions M" / "Switching Gear", DOK-MTC200-NC**PRO*Vxx-AW0x-DE.

[21] More detailed information regarding the spindle override is contained in the Rexroth Indramat documentation:

"CNC/PLC Interface Description xxVRS", chapter entitled "Feedrate and Spindle Override PxxCSOVRD", DOK-MTC200-SPS*GWY*Vxx-AW0x-DE.

[22] More detailed information regarding the axis speeds is contained in the Rexroth Indramat documentation:

"MT-CNC Numeric Control for Multiple Axis, Multiple Process Applications", chapter entitled "Maximum Track Acceleration", DOK-MT*CNC-PAR*DES*Vxx-AW0x-DE.

[23] More detailed information regarding the structure and elements of the tool data is contained in the Rexroth Indramat documentation:

"NC NC Programming Instructions Vxx, Application Description", chapter entitled "Access to Tool Data by NC Program TLD", DOK-MTC200-NC**PRO*Vxx-AW0x-DE.

[24] More detailed information regarding the use of zero-point offsets and zero offset tables is contained in the Rexroth Indramat documentation:

"NC NC Programming Instructions Vxx", Application Description, chapter "Zero-Point Offsets, Zero Offest Tables O", DOK-MTC200-NC**PRO*Vxx-AW0x-DE.

[25] Additional information regarding the display of the axis position in the GBO is contained in the Rexroth Indramat documentation:

"MTC200/MT-CNC xxVRS GUI", DOK-MTC200-GBO*GEN*Vxx-AW0x-DE.

[26] More detailed information on resetting the device is contained in the Rexroth Indramat documentation:

"SPS Programming Instructions xxVRS", Chapter "Putting into Operation / Start", DOK-CONTRL-SPS*PRO*Vxx-AW0x-DE.

[27] More detailed information regarding the configuration of the device axes is contained in the Rexroth Indramat documentation:

"Parameter Description", chapter "System Parameters" as well as chapter "Axis Parameters", DOK-MT*CNC-PAR*DES*Vxx-AW0x-DE.



[28] Additional information regarding process parameters and their functions as well as value ranges is contained in the Rexroth Indramat documentation:

"CNC/SPS Interface Description xxVRS Application Description, chapter "External Mechanisms", DOK-MTC200-SPS*GWY*Vxx-AW0x-DE.

[29] Additional information regarding the function of the NC parameters and the structure of the NC parameter records is contained in the Rexroth Indramat documentation:

"MTC200/MT-CNC Parameter Description xxVRS", DOK-MTC200-PAR*DES*Vxx-AW0x-DE.

[30] More detailed information concerning the PLC Programming System is contained in the Rexroth Indramat documentation:

"PLC Programming Instructions xxVRS Application Description" DOK-CONTRL-SPS*PRO*Vxx-AW0x-DE.

[31] More detailed information regarding the structure of NC packages is contained in the Rexroth Indramat documentation:

"MTC200/MT-CNC NC Programming Instructions xxVRS", chapter "Sub-Programs", DOK-MTC200-NC**PRO*Vxx-AW0x-DE.

- [32] More detailed information regarding the structure of tool lists is contained
 - in the Rexroth Indramat documentation: "MTC200/MT-CNC xxVRS GUI", chapter "Tool Data Handling BOF" and chapter "Tool Data Handling GBO", DOK-MTC200-GBO*GEN*Vxx-AW0x-DE.
- [33] More detailed information regarding the use of machine data is contained in the Rexroth Indramat documentation:

"CNC Machine Data xxVRS Application Description" DOK-MT*CNC-MAS*DAT*Vxx-AW0x-DE.

[34] Additional information regarding process parameters and their functions as well as value ranges is contained in the Rexroth Indramat documentation:

"Parameter Description" chapter "Process Parameters" DOK-MT*CNC-PAR*DES*Vxx-AW0x-DE.

[35] Additional information regarding process parameters and their functions as well as value ranges is contained in the Rexroth Indramat documentation:

"MT-CNC Numeric Control for Multiple Axes, Multiple Process Applications", Chapter 2 "Process Parameters", DOK-MT*CNC-PAR*DES*V15-ANW1-DE-E.

[36] Additional information regarding module configuration and the structure of the "Moduldef.ini" file is contained in the following Rexroth Indramat documentation:

"Diagnostics and Message System for HMI System ProVi", chapter "Configure Moduldef.ini", DOK-MTC200-DIAG*PROVI*-AW0x-DE.

[37] More detailed information on selecting the NC program and the NC memory is contained in the Rexroth Indramat documentation:

"MTC200/MT-CNC xxVRS GUI", chapter entitled "Operation Survey of the Administration of NC Programs", DOK-MTC200-GBO*GEN*Vxx-AW0x-DE. [38] More detailed information regarding the contents of parameter records is contained in the Rexroth Indramat documentation:

"MTC200/MT-CNC Parameter Description xxVRS", chapter entitled "Processing / Displaying Contents of Parameter Records", DOK-MTC200-PAR*DES*Vxx-AW0x-DE.

[39] More detailed information regarding NC variables is contained in the Rexroth Indramat documentation:

"NC Programming Instructions Vxx", chapter entitled "Assigning Variables and Mathematical Functions", DOK-MTC200-NC**PRO*Vxx-AW0x-DE.

[40] More detailed information regarding servo lag is contained in the Rexroth Indramat documentation:

"NC Programming Instructions Vxx", chapter entitled "Movement Records and Interpolation Requirements", DOK-MTC200-NC**PRO*Vxx-AW0x-DE.

[41] Additional information regarding the function of the standard and productspecific SERCOS parameters (S and P) is contained in the Rexroth Indramat Documentation:

"DIAX04 Drive with Servo Functions", Appendix A Description of Parameters, DOK-DIAX04-SSE-02VRS**-FKB1-DE-P.

[42] More detailed information regarding the communication phases is contained in the Rexroth Indramat documentation:

"DIAX04 Drive with Servo Functions", General Instructions on Putting into Operation, DOK-DIAX04-SSE-xxVRS**-FKBx-DE.

[43] More detailed information regarding tool management is contained in the Rexroth Indramat documentation:

"MT-CNC Numeric Control for Multiple Axes, Multi-Process Applications, Tool Data Handling", DOK-MT*CNC-BOF*WZH*Vxx-AW0x-DE.

[44] More detailed information regarding zero offsets is contained in the Rexroth Indramat documentation:

MTC200/MT-CNC 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-DE.

[45] More detailed information regarding SERCANS errors is contained in the Rexroth Indramat documentation:

"SERCANS /SERCVME SERCOS Interface Assemblies with Universal μ P 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 Rexroth Indramat Documentation:

"SERCANS SERCOS Interface Assemblies", Chapter 10 "Description of Parameters",

DOK-SERCAN-SER-VxxVRS**-AW0x-DE.





11 Glossary

ANSI

American National Standards Institute, American standards institute which developed the ANSI emulation (refer also to: ANSI code).

ANSI code

Standard code standardized by ANSI making it possible to generate pictures, animation and texts plus sounds from the PC speaker from sequences of ANSI control frequencies. Method primarily used in mailboxes to design a GUI. The ASCII Code is generally also referred to as ANSI Code. These characters are generated in a document by pressing the <AltGr> key together with inputting the relevant code.

ASCII

American Standard Code for Information Interchange, widely used code, especially on domestic and personal computers, for displaying numbers, letters and special characters; designed as a 7-bit code with a character set of 128 characters or as an 8-bit code with a character set of 256 characters including upper and lower case letters. The unassigned (free) eighth and ninth bits, formatted in bytes, are used as parity bits (check bits).

Operating date

The operating date is data block element 7 of a parameter. The value of the parameter is stored in it.

BTV20

The BTV20 is a machine user terminal in which one or more NC control units with PLC or one or more stand-alone PLCs can be integrated. The number of units that can be integrated depends on their configuration. In contrast to the BTV30, the BTV20 offers an application-oriented function keyboard with the following features:

- Front panel made of 4mm aluminum with camfered edges.
- Fully flush, chemical-resistant polyester foil with embossed stamping.
- Integrated EMC-compatible glass plate to protect the display.
- Integrated machine keys with intermediate plate to avoid double entries and accidental operation.
- Key switch to lock security functions.

BTV30

The BTV30 has all the functions and operating elements of a complete industrial PC. In addition to the 10.4" flat colour display and a complete ASCII keyboard with cursor block, keyboard mouse, there is a standard disk drive and the connection for an external keyboard located behind a lockable hinged cover. Ten PC function keys are positioned under the display, while eight machine function keys are located to the right of the display. These are either fed to the outside via a socket or, in the case of an integrated PLC unit, connected directly to the PLC. Genuine key elements are embedded in the stable PVC/ABS front panel, allowing fatigue-free programming even over prolonged periods of time. The display is protected by a stable, EMC-compatible glass plate. When the hinged disk flap is closed the BTV30 complies with IP65 type of protection and is resistant to all known coolants and lubricants.



Client

A client is a computer system or a process that requests the services of another computer system or process. The term also indicates a workplace computer that can use the services and resources (e.g., printer, scanner, plotter) of a server or also of other clients. Generally speaking it has significantly fewer privileges than the server.

DDE

Dynamic Data Exchange is a standard defined by Microsoft for data exchange between programs under MS-Windows from Version 3.0. DDE permits files or parts of files to be linked between two applications that support the DDE standard. A distinction is made between a source application (server) and a target application (client), wherein the target application maps a complete or partial copy of the server file. If the data in the source file is changed, then this information is transferred to the target application via the link and dynamically updated there. However, a DDE communication can also simply be used to exchange commands and instructions between two programs.

DLL

Dynamic Link Library, is a library linked to a program during its run-time. DLLs are special files for Windows from which, for example, functions, dialog boxes or symbols are loaded by applications. They simplify programming and save hard disk space if, for example, the same functions are required by several user programs. A dynamic link library offers various advantages: It need only be loaded when required and until then it does not use any RAM.

Dual port RAM

A Dual port RAM is a memory area between two connected users: the actual controllers, and the GUI (PC). This memory area, used by both users, permits only limited data traffic.

If, for example, the control unit wants to send a message to the user interface (GUI/PC), it sends it first to the dual port RAM. A cyclic mechanism running on the PC side detects and fetches the new information, acknowledges it for the control unit and passes it on for processing (display on the GUI).

If the situation is reversed and the GUI wants to send a message to the control unit, it is likewise sent first to the dual port RAM. Just as with the control unit, a cyclic mechanism detects and fetches this new information, acknowledges it for the PC and passes it on for processing.

This ensures that both users only exercise controlled data trafficking and otherwise work in their own, separate memory areas.

Remote device

This term depends on the point of view (refer also to Local Device). Where a user program (client) is concerned, the device is **not** at the PC on which the client is running, but at a PC within the PC network.

Device

A device is the control hardware, drive device or I/O device.

Device Type

The device type indicates which Rexroth Indramat device is meant e.g., MTC200-P-G2, MTVNC, ISP200-R-G2.



FarDevice

Configuring the PC network requires, in addition to the list of PCs, a list of FarDevices. The FarDevice address has been introduced for clear, linear addressing of devices in the PC network. This means that every device available in the PC network receives an additional address (FarDevice address). Any device that is available in the PC network has a FarDevice address and is referred to as a FarDevice. The list of FarDevices is produced on the basis of devices previously connected to each PC.

Local device

This term depends on the point of view (refer also to Remote Device). As far as the user program is concerned the device is at the PC on which the client is running.

MPI

(**MPI** = **M**ulti-**P**rotocol-Interface). The Rexroth Indramat MPI provides a standardized user interface for the communication interfaces Profibus-FMS, MMS-Ethernet (MAP), TCP/IP and FIPWAY on PC units under the Windows NT 4.0 operating system. The MPI provides the interfaces for implementing client and server user programs. The MPI communication driver makes the connection between the MPI and the function interface. This in turn provides that can be configured via the function interface.

MTC200

MTC200 is a PC-based generation of control units. The MTC200 system integrates all the functions of an NC and PLC control unit including the entire drive technology. Components of this system include MTC-P, MTC-R, MTS-P, MTS-R.

Up to seven independent NC processes can be controlled with the MTC200. The seven NC processes can be divided to a max. of 32 axes. This means that the MTC is both multi-axis and multi-process capable.

MTC-P

The MTC-P is a powerful CNC control unit in ISA bus plug-in card format for insertion into an industrial PC; it is part of the MTC200 range. It consists of a basic unit with the processor system of an NC and an integrated axis processor to which a maximum of eight drives can be attached via a SERCOS interface. Expansion by a maximum of up to three axis processor modules allows 32 drives to be controlled at the highest level of expansion. These can be divided between 7 processes. Together with the MTS-P01.1 PLC control unit, this unit forms a compact and flexible solution for tool machine control.

MTC200-P-G2

See MTC-P.

MTC-R

The RECO-based NC unit, MTC-R, contains a complete MTC-Pcompatible NC processor and an axis processor module for controlling up to 8 digital drives via a SERCOS interface. Up to three further axis processor modules can be plugged in via the PC/104 bus to achieve the maximum capacity of 32 drives. While there is room for one additional PC/104 module in a unit of normal width (for controlling up to 16 drives), a double-width housing is available for additional axis processors.

The MTC-R cannot function on its own; it always requires an MTS-R as an offshore adaptive control. Both units are connected via a local bus for communication between the MTC-R and MTS-R. Both units are then



slotted together into an RMB02.2 or RMB02.4 rack. If required, as described for the MTS-R, additional I/O modules can be used to supplement the local I/O level.

MTC200-R-G2

See MTC-R.

MTS-R

The RECO-PLC is a powerful but small PLC that is compatible with the PLC in the MTC200 control system. The housing is designed in accordance with IP20. It can be used as a stand-alone PLC and together with an MTC-R as a slave PLC. To facilitate networking several PLC control units and connecting to a programming device or PC there is an RS 232/RS 485 programming interface available. For connection to a printer, read/write memory, or screen a free serial interface (RS 232/RS 422) is available.

The MTS-R01.1 occupies one module slot in the RMB02 rack, and the MTS-R02.1 occupies two slots. This means that the ISP200-R is capable of driving the bus for up to 15 further I/O modules. An internal local bus communicates with the MTC-R NC control unit that is part of the system by means of an adapter board.

As an option, the MTS-R01 and the MTS-R02 can be equipped with the open fieldbus interfaces INTERBUS or PROFIBUS-DP. Distributed I/O peripheral units, each with up to 4096 inputs and 4096 outputs, can be connected via these optional interfaces. In addition, the MTS-R02 can be equipped with a serial interface ($2 \times RS 232$ and $2 \times RS 422$).

PC network

The PC network is made up of several PCs connected at the level of the function interface. The PC network comprises the PCs that are used to control a machine (primarily graphics, operation and programming).

Process

The process is the combination of functions and axes relevant to the control work carried out by the MTC200-P-G2 control unit; this combination is processed in a processing unit within the control unit. Each MTC-P / MTC-R (MTC200-P-G2, MTC200-R-G2) has a maximum of 7 processes.

RECO

RECO is a modular I/O system for rapid signal exchange with the PLC. The racks for two or four I/O modules can be mounted on a standardized top-hat rail. There is a choice between analog and digital inputs and outputs as well as serial interfaces.

Registry

See Registry Database.

Registry database

In Windows NT the registry database replaces most of the Win3.x INI files (these files still exist in Windows NT but are generally only used by 16-bit programs). Information concerning configuration is entered in the registry both by Windows NT and by all 32-bit programs.

Registry editor

Entries in the registry database "Registry" are changed using the registry editor. The editor is in the Windows system directory and is called REGEDT32.EXE (\rightarrow "Start" \rightarrow "Run" REGEDT32).

RS232

Serial interface with 9-pin or 25-pin connection conforming to the V.24 standard and developed by the EIA for communication with devices; maximum 115,200 bit/s; often used for connection between computers and modems.

Server

A server is a computer that holds ready applications and documents for connected computers (clients) to access. The term also indicates a program that provides certain services that can be accessed using programs specially adapted to the server.

Shared memory

An area in the computer's RAM that can be accessed by several processes (applications).

System200

The Rexroth Indramat System200 is a comprehensive scalable control and drive program for the entire machine and system structure. There is a choice of various software packages (WIN-HMI, MPI, function interface etc.,) for one single PC hardware platform (MTC200), plus various screens (BTV20, BTV30, etc.), application-optimized drives (DIAX04, ECODRIVE, etc.) and periphery connections (Profibus-DP, Profibus-FMS, SERCOS interface, etc.).

Thread

Threads are objects within processes which carry out programming instructions. They permit various simultaneous actions within a process and enable a process to execute different parts of a program simultaneously on different processors.

WinHMI

(**WinHMI = WIN**dows based - Human Machine Interface). The WinHMI software package is a standard GUI for automated production.





12 List of Figures

Fig. 3-1:	Structural overview of the function interface 3-1
Fig. 3-2:	MPI with Profibus FMS connection 3-3
Fig. 3-3:	Software structure: Rexroth Indramat user interface and DDE server 3-4
Fig. 3-4:	Overview of the connection options 3-5
Fig. 3-5:	Communication between a Client and Rexroth Indramat Devices 3-6
Fig. 3-6:	Communication between several Client and Rexroth Indramat Devices 3-7
Fig. 4-1:	Including the Rexroth Indramat Header Files in the Client 4-2
Fig. 4-2:	Project settings "For Win32 Release": Multithreaded DLL 4-3
Fig. 4-3:	Project settings "For Win32 Debug": Debug Multithreaded DLL 4-3
Fig. 4-4:	"For All Configurations" project settings 4-4
Fig. 4-5:	Synchronization via SYS-Messages 4-22
Fig. 4-6:	"Run" dialog box of Windows NT: Complete re-initialization 4- 39
Fig. 4-7:	"Run" dialog box of Windows NT: Selective re-initialization 4-40
Fig. 4-8:	The "VBDemo" user program 4-41
Fig. 4-9:	Starting VBDemo in the diagnostics mode of the function interface 4-42
Fig. 4-10:	"Run" dialog box of Windows NT: BOFINTFC.EXE\$ 4-43
Fig. 4-11:	Diagnose mode of the function interface 4-43
Fig. 5-1:	Windows NT Task Manager 5-2
Fig. 5-2:	Setting the WindowsNT system property "Performance" 5-3
Fig. 5-3:	Date and time settings 5-4
Fig. 5-4:	Rexroth Indramat system configurator 5-7
Fig. 6-1:	Identifier 6-1
Fig. 6-2:	Example 1: Cyclic reading in ASCII code 6-3
Fig. 6-3:	Example 2: Cyclic reading in ASCII code 6-3
Fig. 6-4:	Example 3: Cyclic reading in binary code 6-4
Fig. 6-5:	Overview of the MPCX device group 6-7
Fig. 6-6:	Overview of the MTCX device group 6-12
Fig. 6-7:	Overview of the MWCX device group 6-18
Fig. 6-8:	Overview of the MSCX device group 6-18
Fig. 6-9:	Overview of the MWMX device group 6-20
Fig. 6-10:	Overview of MWSX device group 6-22
Fig. 6-11:	Overview of the MWAX device group 6-24
Fig. 6-12:	Overview of the MSYX device group 6-24
Fig. 6-13:	Overview of the MWYX device group 6-26
Fig. 6-14:	Logical conjunctions of FI commands 6-29
Fig. 6-15:	Computer identification data 6-30
Fig. 6-16:	Representative devices 6-30



- Fig. 6-17: Command execution times of the MPCX device group 6-31
- Fig. 6-18: Command execution times of the MWCX device group 6-35
- Fig. 6-19: Command execution times of MSCX device groups 6-35
- Fig. 6-20: Command execution times of the MWSX device group 6-35
- Fig. 6-21: Command execution times of the MWAX device group 6-36
- Fig. 6-22: Command execution times of the MSYX device group 6-36
- Fig. 7-1: File structure of the download file 7-82
- Fig. 7-2: Status flags 7-84
- Fig. 7-3: File structure of the upload file 7-86
- Fig. 7-4: Status flags 7-88
- Fig. 7-5: File structure of the download file 7-159
- Fig. 7-6: Description of the status flags 7-161
- Fig. 7-7: File structure of the upload file 7-163
- Fig. 7-8: Status flags 7-165
- Fig. 7-9: Structure for handling tool data records 7-291



13 Index

Α

AAC Active ACceleration 7-39 AAD Active Angle Dimension 7-40 AAS Actual Axis Speed 7-40 ABI Active NC-Block Information 7-42 ACS Active Cutting Speed 7-43 ADM MTA200 Messages 7-369 ADN Active D-Correction Number 7-43 AEM Active Event Monitoring 7-44 AEN Active Edge-Number 7-45 AFO Active Feedrate Override 7-45 AFR Actual Feed Rate 7-46 AGF Active G-Function 7-47 AMF Active M-Function 7-48 AMM Active Mechanism Messages 7-48, 7-371 ANM Active NC Memory Size 7-56 ANSI 6-3 API Active Machine-Parameter Index 7-57 APM Active Part-Program Message 7-59 APN Active Part-Program Message-Number 7-60 APO Actual Axis POsition 7-61, 7-373 APP Active Part-Program Number 7-62 ARF Axis Reference Flags 7-63 ARO Actual Rapid Override 7-64 ART Axis Reference Table 7-65 ASCII 6-3 ASD Actual Spindle Data 7-66 ASE Actual System Error 7-295, 7-449 ASF Active Spindle For Process 7-67 ASG Actual Spindle Gear 7-68 ASM Active System Messages 7-69, 7-309, 7-375 ASN Actual Sequence Number 7-74 ASO Actual Spindle Override 7-75 ASS Actual Spindle Speed 7-76 ATN Active Tool-Number 7-77 ATP Actual Tool-Place Information 7-77 ATR Actual Tooldata-Record 7-79 ATU Actual Tooldata-Update 7-81 AutoLogin 9-2 Availability 2-1 AZB Active Zero-Offset Bank 7-81

В

Basic process 3-1 BFJ Break Function-Interface Jobs 7-1 Binary 6-3

С

Calling the Task Manager 5-2 CCA NC-Cycle Access 7-82, 7-85 CCP Cell Configuration Parameter 7-2 CEI Communication Error Info 7-8 CMA CMOS RAM ASCII Parameter 7-380 CMF CMOS RAM Floatingpoint Parameter 7-381 CMI CMOS RAM Integer Parameter 7-381 Communication process 3-1 Complete reinitialization 4-39 Core 5-10 Core mode 5-10 Core-mode driver 5-10 CPO Command POsition 7-88 CPR Create PRocess 7-9





CRT Control ReseT 7-90, 7-314, 7-382 CSE Clear System Error 7-296, 7-450

D

DAC Device Axis Configuration Parameter 7-90 DataTransfer 4-8, 6-2 DCA NC- D-Correction Access 7-93, 7-95 DCD D-Correction Data 7-96 DCP Device Configuration Parameter 7-97, 7-383 DCR D-Correction Record 7-99 DCT Device Communication Timeout 7-101, 7-296, 7-315, 7-385, 7-450 Device address 6-1 DFJ Delete Function-Interface Jobs 7-10 DFS Delete IF Command Stack 7-11 Diagnosis 4-43 Diagnostic messages 4-43 DIS Data Identification String 7-102, 7-315, 7-386 DLL 2-1 DPN Delete Program NC 7-107 DPP Delete Program Package 7-108 DPR Delete PRocess 7-11 DSI Device Status Information 7-108, 7-297, 7-316, 7-388 DTC Device Tool Management Configuration 7-111 DTG Distance To Go 7-115, 7-391 DTY Device TYpe 7-117, 7-300, 7-319, 7-392, 7-451 Dual Port RAM 2-1, 3-2 DWD Diagnosis Window Data 7-117, 7-320, 7-393

Ε

EAD Existing MTA200 Diagnosis 7-395 EDE Existing Diagnosis Error 7-120, 7-322, 7-396 EDP Existing PLC Diagnosis 7-324 EDW Existing Diagnosis Window 7-121, 7-323, 7-397 END Existing NC Diagnosis 7-122 EPD Existing PLC Diagnosis 7-123 EPO End POsition 7-124 EPT Existing ProVi Types 7-126, 7-326, 7-398 ERI ERror Information 7-12 EST Error STate 7-127, 7-326, 7-399 Example 1 (cyclic reading) 6-3 Example 2 (cyclic reading) 6-3 Example 3 (cyclic reading) 6-4 EXD EXecution Display 7-127, 7-327, 7-400

F

FCP Far Device Configuration Parameter 7-12 FDC Far Device Configuration 7-16 FI Commands for the MTCX Device Group 7-37 FI Commands for the MWCX Device Group 7-39 FIT Further Info Text 7-18 FPC Far PC Configuration Parameter 7-19 Function code 6-2

G

GDB Global Data Buffer 7-21 GPC Global Process Configuration 7-128 GPP Global Process Parameter 7-131

I

ICA Initialization Communication Address 7-22 IFJ Information about Function-Interface Jobs 7-22 IFS IF Command Stack Info 7-24 INDIF000.H 4-2 INDIFX00.H 4-2



Interface designator 6-1 IPP Insert NC-Program Package 7-133

Κ

Key combination <Alt>+<Tab> 5-2

L

LDT PC Local Date Time 7-25 LNG Activated LaNGuage 7-26 LOG channels 3-1 Logic channel 3-1 Logic process 3-1 LogInlf 4-4 LogOutlf 4-7

Μ

Management process 3-1, 4-43 MAP Modul Assign of Process 7-134, 7-401 MAR Map Absolute PCL-Reference 7-135, 7-401 MAR Map Absolute PCL-Referenz 7-328 MCD Modul Configuration Device Information 7-136, 7-329, 7-402 MCM Modul Configuration Modul Information 7-137, 7-330, 7-403 MCP Modul Configuration Process Information 7-138, 7-404 MCS Modul Configuration SFC Information 7-138, 7-331, 7-405 MDA Machine Data Access 7-139, 7-142 MDI Manual Data Input 7-144 MDS Machine Data Single 7-146 MFD Message Files Download 7-149, 7-332, 7-406 MFO Maximal Feedrate Override 7-150 MFR Maximal Feed Rate 7-151 MKS Machine Key Status 7-152, 7-332, 7-406 MKT Machine Key Table 7-152, 7-333, 7-407 MPI (Multi-Protocol-Interface) 3-3 MRO Maximal Rapid Override 7-154 MSG MeSsaGe 7-26, 7-154, 7-301, 7-334, 7-408, 7-452 MSO Maximal Spindle Override 7-155 MSS Maximal Spindle Speed 7-156 MTC MT-CNC Slot Software Version 7-156, 7-335, 7-409 MTCNC00I.SYS 5-10 MTCX Device Group 7-37 MTD Machine Table Data 7-157 MWCXDevice Group 7-39

Ν

NCA NC-Program Access 7-158, 7-162 NCM NC Messages 7-166 NEA NC-Event Access 7-168, 7-170 NEV NC-EVent 7-172 NMM NC-MeMory 7-173 Non-pre-emptive Multitasking 5-1 NPA NC-PArameter 7-173 NPC NC-Package Compiling 7-177 NPD NC-Package Download 7-177 NPI NC-Package Directory 7-180 NPS NC-Program Selection 7-181 NST NT-ShuT-Down 7-27 NTN Next Tool-Number 7-182 NUA NC- Offset Data Access 7-182, 7-185 NVA NC-Variable Access 7-187, 7-189 NVS NC-Variable Single 7-191





0

Objective 2-1 OPD Optimal Position Distance 7-192

Ρ

PAA PArameter Access 7-194, 7-196, 7-410 PAC Process Axis Configuration Parameter 7-198 PAD PArameter Deactivate 7-199 PAF PArameter File Converted 7-28 Parameterized device 3-2 PAS PArameter Set Active 7-200 PDD Provi Diagnosis Data 7-200, 7-336, 7-411 PDT Parameter Definition Table 7-203 PFR Programmed Feed Rate 7-204 PHD PHysical Directory 7-29, 7-205, 7-301, 7-339, 7-414, 7-452 Physical communication address 2-1 POB POrt Byte Access 7-30 POI POsition Information 7-415 POW POrt Word Access 7-31 PPA Part Program Active 7-206 PPD Part Program Directory 7-206 PPN Part Program NC 7-207, 7-209 PPP Part Program Package 7-209 PPS Part Program Sequence 7-210 Pre-emptive Multitasking 5-1 Problem 4-38 Program languages 4-1 PSM PCL Sys Message 7-210, 7-340, 7-416 PSS Programmed Spindle Speed 7-212 PTC Process Tool Management Configuration 7-212 PVA PROVI-Messages-Access 7-214, 7-341, 7-418 PVF PLC-Variable Formatted 7-217, 7-344, 7-421 PVM ProVi Messages 7-224, 7-351, 7-428 PVR PLC Variable Retain Backup 7-228, 7-229, 7-355, 7-356, 7-432, 7-433 PVT PLC-Variable Type 7-230, 7-356, 7-433

R

ReadGroupItem 6-2 Remedy 4-38 REP REPositioning Data 7-233 Requirements 2-1 Rexroth Indramat GUI WIN200 3-4 Rexroth Indramat system configurator 5-6 Right mouse click 4-44 RPR Ready PRocess 7-32

S

SDD SFC Diagnosis Data 7-236, 7-436 SDP Start Device Polling 7-32 SDS Set Device Status 7-240, 7-302, 7-359, 7-440, 7-453 Selector 6-3 Separator 6-1 Serial interface 2-1 SFD SFC Data 7-241, 7-360, 7-441 SFE SFC Error 7-243, 7-362, 7-443 SFM SFC Mode 7-245, 7-364, 7-444 SFW Set Focus to Window 7-33 Shared memory 3-2 Shared Memory area 2-1 SID Software Installation Data 7-245, 7-303, 7-364, 7-445, 7-454 Simulation pair 5-6 SLA Servo LAg 7-246 SLI SPS Long Identification 7-248, 7-365, 7-446 SPA SERCOS PArameter 7-249, 7-304, 7-455 SPH SERCOS PHase 7-251, 7-307, 7-458



SPP Selected Part-Program Number 7-252 SSM Set Sys-Message 7-34 Start bar 4-44 SYS-MSGs 4-1 System configuration 5-6 System configurator 5-6 System Messages 4-1

Т

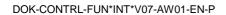
Tab pages 4-44 Task-Manager 5-2 TDA Tool DAta 7-253 TDD Tool Data Download 7-258 TDR Tool Data Record 7-259, 7-261 TIF Tool Insert Finish 7-263 TII Tool Insert Initiate 7-263 TLB TooL Basic Data List 7-264 TLD TooL Data 7-267, 7-269, 7-270, 7-271 TLE TooL Edge Data List 7-273 TMV Tool MoVe 7-275 TPI Tool Position Information 7-276 TQE TorQuE 7-278 TRM ReMove 7-279 TRS Tool ReSet 7-280

W

Windows NT core-mode driver 5-10 Windows NT Task Manager 9-1 WinHMI 3-4 WinMTC 3-4 WLA Watch List Allocation 7-281, 7-366, 7-447 WLF Watch List Free 7-281, 7-367, 7-447

Ζ

ZOD Zero Offset Data 7-282







14 Service & Support

14.1 Helpdesk

Unser Kundendienst-Helpdesk im Hauptwerk Lohr am Main steht Ihnen mit Rat und Tat zur Seite. Sie erreichen uns

- telefonisch: +49 (0) 9352 40 50 60 über Service Call Entry Center Mo-Fr 07:00-18:00
- per Fax: +49 (0) 9352 40 49 41
- per e-Mail: service@indramat.de

Our service helpdesk at our headquarters in Lohr am Main, Germany can assist you in all kinds of inquiries. Contact us

- by phone: +49 (0) 9352 40 50 60 via Service Call Entry Center Mo-Fr 7:00 am - 6:00 pm
- by fax: +49 (0) 9352 40 49 41
- by e-mail: service@indramat.de

14.2 Service-Hotline

Außerhalb der Helpdesk-Zeiten ist der Service direkt ansprechbar unter		After depar	helpdesk tment directl	,	contact	our	service
oder	+49 (0) 171 333 88 26 +49 (0) 172 660 04 06	or			(0) 171 3 (0) 172 6		

14.3 Internet

Unter **www.boschrexroth.de** finden Sie ergänzende Hinweise zu Service, Reparatur und Training sowie die **aktuellen** Adressen *) unserer auf den folgenden Seiten aufgeführten Vertriebsund 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.de** 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.



offices providing service

Please contact our sales / service office in your area first.

*) Data in the present documentation may have become obsolete since printing.

14.4 Vor der Kontaktaufnahme... - Before contacting us...

Wir können Ihnen schnell und effizient helfen wenn Sie folgende Informationen bereithalten:

- detaillierte Beschreibung der Störung und der Umstände.
- Angaben auf dem Typenschild der betreffenden Produkte, insbesondere Typenschlüssel und Seriennummern.
- 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.

14.5 Kundenbetreuungsstellen - Sales & Service Facilities

Deutschland – Germany

vom Ausland: from abroad: (0) nach Landeskennziffer weglassen! don't dial (0) after country code!

Vertriebsgebiet Mitte Germany Centre	SERVICE	SERVICE	SERVICE
Rexroth Indramat GmbH BgmDrNebel-Str. 2 / Postf. 1357 97816 Lohr am Main / 97803 Lohr Kompetenz-Zentrum Europa	CALL ENTRY CENTER MO – FR von 07:00 - 18:00 Uhr	HOTLINE MO – FR von 17:00 - 07:00 Uhr from 5 pm - 7 am	ERSATZTEILE / SPARES verlängerte Ansprechzeit - extended office time -
Tel.: +49 (0)9352 40-0 Fax: +49 (0)9352 40-4885	from 7 am – 6 pm Tel. +49 (0) 9352 40 50 60 <u>service@indramat.de</u>	+ SA / SO Tel.: +49 (0)172 660 04 06 oder / or Tel.: +49 (0)171 333 88 26	- 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	Vertriebsgebiet West Germany West	Gebiet Südwest Germany South-West	Gebiet Südwest Germany South-West
Rexroth Indramat GmbH Landshuter Allee 8-10 80637 München	Bosch Rexroth AG Regionalzentrum West Borsigstrasse 15 40880 Ratingen	Bosch Rexroth AG Service-Regionalzentrum Süd-West Siemensstr.1 70736 Fellbach	Bosch Rexroth AG Regionalzentrum Südwest Ringstrasse 70 / Postfach 1144 70736 Fellbach / 70701 Fellbach
Tel.: +49 (0)89 127 14-0 Fax: +49 (0)89 127 14-490	Tel.: +49 (0)2102 409-0 Fax: +49 (0)2102 409-406	Tel.: +49 (0)711 51046–0 Fax: +49 (0)711 51046–248	Tel.: +49 (0)711 57 61–100 Fax: +49 (0)711 57 61–125
Vertriebsgebiet Nord Germany North	Vertriebsgebiet Mitte Germany Centre	Vertriebsgebiet Ost Germany East	Vertriebsgebiet Ost Germany East
Bosch Rexroth AG Walsroder Str. 93 30853 Langenhagen Tel.: +49 (0) 511 72 66 57-0 Service: +49 (0) 511 72 66 57-256	Bosch Rexroth AG Regionalzentrum Mitte Waldecker Straße 13 64546 Mörfelden-Walldorf	Bosch Rexroth AG Beckerstraße 31 09120 Chemnitz	Bosch Rexroth AG Regionalzentrum Ost Walter-Köhn-Str. 4d 04356 Leipzig
Fax: +49 (0) 511 72 66 57-93 Service: +49 (0) 511 72 66 57-95	Tel.: +49 (0) 61 05 702-3 Fax: +49 (0) 61 05 702-444	Tel.: +49 (0)371 35 55-0 Fax: +49 (0)371 35 55-333	Tel.: +49 (0)341 25 61-0 Fax: +49 (0)341 25 61-111



Europa (West) - Europe (West)

Austria - Österreich	Austria – Österreich	Belgium - Belgien	Denmark - Dänemark	
Bosch Rexroth GmbH Bereich Indramat Stachegasse 13 1120 Wien Tel.: +43 (0)1 985 25 40 Fax: +43 (0)1 985 25 40-93	Bosch Rexroth GmbH Gesch.ber. Rexroth Indramat Industriepark 18 4061 Pasching Tel.: +43 (0)7221 605-0 Fax: +43 (0)7221 605-21	Bosch Rexroth AG Electric Drives & Controls Industrielaan 8 1740 Ternat Tel.: +32 (0)2 5830719 - service: +32 (0)2 5830717 Fax: +32 (0)2 5830731 indramat@boschrexroth.be	BEC A/S Zinkvej 6 8900 Randers Tel.: +45 (0)87 11 90 60 Fax: +45 (0)87 11 90 61	
Great Britain – Großbritannien	Finland - Finnland	France - Frankreich	France - Frankreich	
Bosch Rexroth Ltd. Rexroth Indramat Division 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	Bosch Rexroth Oy Rexroth Indramat division Ansatie 6 017 40 Vantaa Tel.: +358 (0)9 84 91-11 Fax: +358 (0)9 84 91-13 60	Bosch Rexroth S.A. Division Rexroth Indramat 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	Bosch Rexroth S.A. Division Rexroth Indramat 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	Italy - Italien	Italy - Italien	Italy - Italien	
Bosch Rexroth S.A. Division Rexroth Indramat 91, Bd. Irène Joliot-Curie 69634 Vénissieux – Cedex	Bosch Rexroth S.p.A. Via G. Di Vittoria, 1 20063 Cernusco S/N.MI	Bosch Rexroth S.p.A. Via Paolo Veronesi, 250 10148 Torino	Bosch Rexroth S.p.A. Via del Progresso, 16 (Zona Ind.) 35020 Padova	
Tel.: +33 (0)4 78 78 53 65 Fax: +33 (0)4 78 78 53 62	Tel.: +39 02 92 365 1 +39 02 92 365 326 Fax: +39 02 92 365 500 +39 02 92 365 516378	Tel.: +39 011 224 88 11 Fax: +39 011 224 88 30	Tel.: +39 049 8 70 13 70 Fax: +39 049 8 70 13 77	
Italy - Italien	Italy - Italien	Netherlands - Niederlande/Holland	Netherlands - Niederlande/Holland	
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	Bosch Rexroth S.p.A. Viale Oriani, 38/A 40137 Bologna Tel.: +39 051 34 14 14 Fax: +39 051 34 14 22	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	Bosch Rexroth Services B.V. Technical Services Kruisbroeksestraat 1 (P.O. Box 32) 5281 RV Boxtel Tel.: +31 (0)411 65 19 51 Fax: +31 (0)411 67 78 14 services@boschrexroth.nl	
Norway - Norwegen	Spain - Spanien	Spain – Spanien	Sweden - Schweden	
Bosch Rexroth AS Rexroth Indramat Division Berghagan 1 or: Box 3007 1405 Ski-Langhus 1402 Ski Tel.: +47 (0)64 86 41 00 Fax: +47 (0)64 86 90 62	Bosch Rexroth S.A. Divisiòn Rexroth Indramat 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	Goimendi S.A. División Rexroth Indramat 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	Rexroth Mecman Svenska AB Rexroth Indramat Division - Varuvägen 7 (Service: Konsumentvägen 4, Älfsjö) 125 81 Stockholm Tel.: +46 (0)8 727 92 00 Fax: +46 (0)8 647 32 77	
jul.ruud@rexroth.no		Fax: +34 9 43 31 84 27 - service: +34 9 43 31 84 60 <u>sat.indramat@goimendi.es</u>	Fax: +46 (0)8 647 32 77	
Sweden - Schweden	Switzerland West - Schweiz West	Switzerland East - Schweiz Ost		
Rexroth Mecman Svenska AB Indramat Support Ekvändan 7 254 67 Helsingborg Tel.: +46 (0) 42 38 88 -50 Fax: +46 (0) 42 38 88 -74	Bosch Rexroth Suisse SA Département Rexroth Indramat Rue du village 1 1020 Renens Tel.: +41 (0)21 632 84 20 Fax: +41 (0)21 632 84 21	Bosch Rexroth Schweiz AG Geschäftsbereich Indramat Hemrietstrasse 2 8863 Buttikon Tel. +41 (0) 55 46 46 111 Fax +41 (0) 55 46 46 222		

<u>vom Ausland</u>: (0) nach Landeskennziffer weglassen, <u>from abroad</u>: don't dial (0) after country code, <u>Italien</u>: 0 nach Landeskennziffer mitwählen <u>Italy</u>: dial 0 after country code



Europa (Ost) - Europe (East)

vom Ausland: (0) nach Landeskennziffer weglassen from abroad: don't dial (0) after country code

Czech Republic - Tschechien	Czech Republic - Tschechien	Hungary - Ungarn	Poland – Polen
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	DEL a.s. Strojírenská 38 591 01 Zdar nad Sázavou Tel.: +420 566 64 3144 Fax: +420 566 62 1657	Bosch Rexroth Kft. Angol utca 34 1149 Budapest Tel.: +36 (1) 42 23 200 Fax: +36 (1) 42 23 201	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	Rumania - Rumänien	Russia - Russland	Russia - Russland
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	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	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 <u>laura.kanina@boschrexroth.ru</u>	ELMIS 10, Internationalnaya 246640 Gomel, Belarus Tel.: +375/ 232 53 42 70 +375/ 232 53 21 69 Fax: +375/ 232 53 37 69 <u>elmis_ltd@yahoo.com</u>
Turkey - Türkei	Slowenia - Slowenien		
Bosch Rexroth Otomasyon San & Tic. A.S. Fevzi Cakmak Cad No. 3 34630 Sefaköy Istanbul Tel.: +90 212 541 60 70 Fax: +90 212 599 34 07	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	Australia - Australien	China	China
AIMS - Australian Industrial Machinery Services Pty. Ltd. 28 Westside Drive Laverton North Vic 3026 Melbourne Tel.: +61 3 93 243 321	Bosch Rexroth Pty. Ltd. No. 7, Endeavour Way Braeside Victoria, 31 95 Melbourne Tel.: +61 3 95 80 39 33	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	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: +61 3 93 243 329 Hotline: +61 4 19 369 195 terryobrien@aimservices.com.au	Fax: +61 3 95 80 17 33 mel@rexroth.com.au	Fax: +86 21 58 66 55 23 gf.zhu sh@boschrexroth.com.cn	Fax: +86 10 65 05 03 79
China	China	China	Hongkong
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	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	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	Bosch Rexroth (China) Ltd. 6 th Floor, Yeung Yiu Chung No.6 Ind Bldg. 19 Cheung Shun Street Cheung Sha Wan, Kowloon, Hongkong Tel.: +852 22 62 51 00
+86 20 8755-0011 Fax: +86 20 8755-2387	Fax: +86 411 46 78 932	Fax: +86 21 6351 3138	Fax: +852 27 41 33 44 alexis.siu@boschrexroth.com.hk
India - Indien	India - Indien	India - Indien	Indonesia - Indonesien
Bosch Rexroth (India) Ltd. Rexroth Indramat Division Plot. A-58, TTC Industrial Area Thane Turbhe Midc Road Mahape Village Navi Mumbai - 400 701	Bosch Rexroth (India) Ltd. Rexroth Indramat Division Plot. 96, Phase III Peenya Industrial Area Bangalore - 560058	Bosch Rexroth (India) Ltd. 1st Floor, S-10 Green Park ext. Market New Delhi – 110016	PT. Rexroth Wijayakusuma Building # 202, Cilandak Commercial Estate JI. Cilandak KKO, Jakarta 12560
Tel.: +91 22 7 61 46 22 Fax: +91 22 7 68 15 31	Tel.: +91 80 41 70 211 Fax: +91 80 83 94 345 mohanvelu.t@boschrexroth.co.in	Tel.: +91 1 16 56 68 88 Fax: +91 1 16 56 68 87	Tel.: +62 21 7891169 (5 lines) Fax: +62 21 7891170 - 71
Japan	Japan	Korea	Korea
Bosch Rexroth Automation Corp. Service Center Japan Yutakagaoka 1810, Meito-ku, NAGOYA 465-0035, Japan	Bosch Rexroth Automation Corp. Rexroth Indramat Division 1F, I.R. Building Nakamachidai 4-26-44, Tsuzuki-ku YOKOHAMA 224-0041, Japan	Bosch Rexroth-Korea Ltd. Electric Drives and Controls Bongwoo Bldg. 7FL, 31-7, 1Ga Jangchoong-dong, Jung-gu Seoul, 100-391	Bosch Rexroth-Korea Ltd. 1515-14 Dadae-Dong, Saha-Ku Rexroth Indramat Division Pusan Metropolitan City, 604-050
Tel.: +81 52 777 88 41 +81 52 777 88 53 +81 52 777 88 79 Fax: +81 52 777 89 01	Tel.: +81 45 942 72 10 Fax: +81 45 942 03 41	Tel.: +82 234 061 813 Fax: +82 222 641 295	Tel.: +82 51 26 00 741 Fax: +82 51 26 00 747 gyhan@rexrothkorea.co.kr
Malaysia	Singapore - Singapur	South Africa - Südafrika	Taiwan
Bosch Rexroth Sdn.Bhd. 11, Jalan U8/82, Seksyen U8 40150 Shah Alam Selangor, Malaysia	Bosch Rexroth Pte Ltd 15D Tuas Road Singapore 638520	TECTRA Automation (Pty) Ltd. 71 Watt Street, Meadowdale Edenvale 1609	Rexroth Uchida Co., Ltd. No.17, Alley 24, Lane 737 Cheng Bei 1 Rd., Yungkang Tainan Hsien
Tel.: +60 3 78 44 80 00 Fax: +60 3 78 45 48 00 <u>hockhwa@hotmail.com</u> rexroth1@tm.net.my	Tel.: +65 68 61 87 33 Fax: +65 68 61 18 25 <u>sanjay.nemade</u> @boschrexroth.com.sg	Tel.: +27 11 971 94 00 Fax: +27 11 971 94 40 Hotline: +27 82 903 29 23 georgv@tectra.co.za	Tel.: +886 6 25 36 565 Fax: +886 6 25 34 754 indra.charlie@msa.hinet.net
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 sonkawin@hotmail.com			



Nordamerika – North America

USA	USA Central Region - Mitte	USA Southeast Region - Südwest	USA SERVICE-HOTLINE
Headquarters - Hauptniederlassung Bosch Rexroth Corporation Rexroth Indramat Division 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	Bosch Rexroth Corporation Rexroth Indramat Division 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	Bosch Rexroth Corporation Rexroth Indramat Division Southeastern Technical Center 3625 Swiftwater Park Drive Suwanee, Georgia 30124 Tel.: +1 770 9 32 32 00 Fax: +1 770 9 32 19 03	- 7 days x 24hrs - +1-800-860-1055
USA East Region – Ost	USA Northeast Region – Nordost	USA West Region – West	
Bosch Rexroth Corporation Rexroth Indramat Division 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	Bosch Rexroth Corporation Rexroth Indramat Division Northeastern Technical Center 99 Rainbow Road East Granby, Connecticut 06026 Tel.: +1 860 8 44 83 77 Fax: +1 860 8 44 85 95	Bosch Rexroth Corporation7901 Stoneridge Drive, Suite 220Pleasant Hill, California 94588Tel.:+1 925 227 10 84Fax:+1 925 227 10 81	
Canada East - Kanada Ost	Canada West - Kanada West	Mexico	Mexico
Bosch Rexroth Canada Corporation Burlington Division 3426 Mainway Drive Burlington, Ontario Canada L7M 1A8	Bosch Rexroth Canada Corporation 5345 Goring St. Burnaby, British Columbia Canada V7J 1R1	Bosch Rexroth Mexico S.A. de C.V. Calle Neptuno 72 Unidad Ind. Vallejo 07700 Mexico, D.F.	Bosch Rexroth S.A. de C.V. Calle Argentina No 3913 Fracc. las Torres 64930 Monterrey, N.L.
Tel.: +1 905 335 55 11 Fax: +1 905 335-41 84 <u>michael.moro@boschrexroth.ca</u>	Tel. +1 604 205-5777 Fax +1 604 205-6944 <u>david.gunby@boschrexroth.ca</u>	Tel.: +52 5 754 17 11 +52 5 754 36 84 +52 5 754 12 60 Fax: +52 5 754 50 73 +52 5 752 59 43 mariofelipe.hernandez@boschrexroth.com.m ≚	Tel.: +52 8 333 88 3436 +52 8 349 80 9193 Fax: +52 8 346 78 71 mario.quiroga@boschrexroth.com.mx

Südamerika – South America

Argentina - Argentinien	Argentina - Argentinien	Brazil - Brasilien	Brazil - Brasilien
Bosch Rexroth S.A.I.C. "The Drive & Control Company" Acassusso 48 41/47 1605 Munro Provincia de Buenos Aires Tel.: +54 11 4756 01 40 Fax: +54 11 4756 01 36 victor.jabif@boschrexroth.com.ar	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 nakase@usa.net nakase@nakase.com gerencia@nakase.com (Service)	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	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





