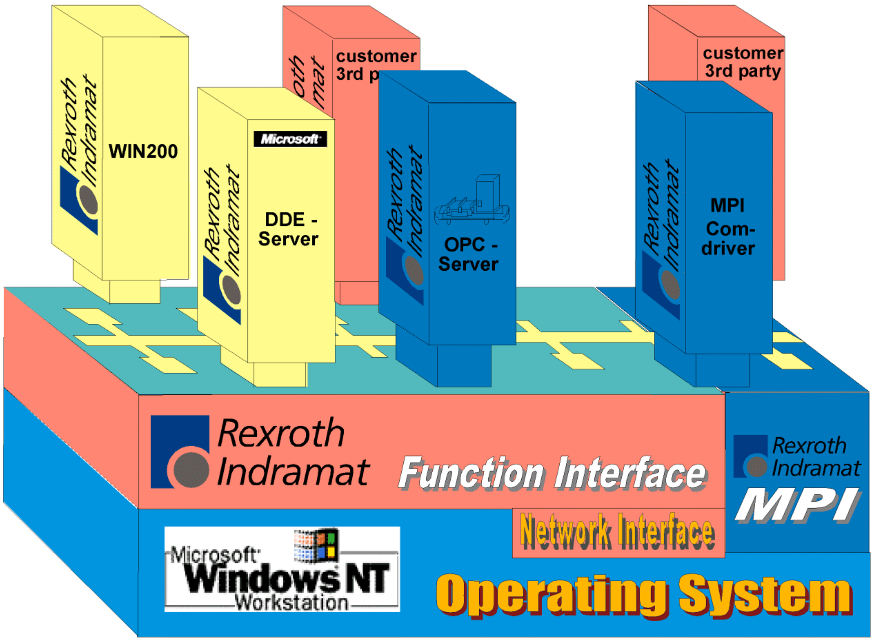


MTC200/MTA200/ISP200
Function Interface
05VRS

Application Description

SYSTEM200

DOK-CONTRL-FUN*INT*V05-AW01-EN-P

	MTC200/MTA200/ISP200
Title	Function Interface
	05VRS
Nature of Documentation	Application Description
Documentation Type	DOK-CONTRL-FUN*INT*V05-AW01-EN-P
Internal Comment	• Document Number: 120-0400-B313-01
Purpose of Documentation?	• a survey of the functions of the function interface • a definition of the application possibilities as well as • planning and developing user-friendly GUIs in C/C++ and Visual Basic.

Changes

Document IDs of previous releases	State	Comment
109-0768-4187-01/EN	01.97	First issue V16
109-0768-4187-01/EN	03.98	Revision 03VRS
120-0400-B313-01/EN	08.99	Revision 04VRS
120-0400-B339-01/EN	03.01	Revision 05VRS

Validity We reserve the right to make changes to the documentation as well as to the delivery of the products

Publisher Rexroth Indramat GmbH
Bgm.-Dr.-Nebel-Str. 2 • D-97816 Lohr a. Main, Germany

Telephone +(44) 9352/40-0 • Tx 689421 • Fax +(44) 9352/40-4885

http://www.rexroth.com/indramat

Dept. ECS2 (FW), ESM (EW/JA)

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

Contents

1 New in Versions 05VRS

1.1 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 SPS 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 SPS variables, arrays and structures (MTCX device group).
- The FI command "PVS" is for the reading and writing of SPS variables, arrays and structures (MTCX device group).
- The FI command "PVT" reads the declaration of SPS variables, including structures and arrays (MTCX device group).
- The FI command "TDR" returns the complete basic data and cutter 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 remaining tool life of a tool to 100% (MTCX device group)
- The FI command "TII" initiates a tool replacement (MTCX device group).
- In chapter 6.4, Logical Connections of the FI Command, the table directly to the left of the individual commands contains a help file.
- In chapter 6.3, Survey of FI Commands, the table has been split into separate sections for each device group. It now contains direct links to the individual commands in the help file.
- New device group MSYX (SYNAX200-P, SYNAX200-R) incorporated into the documentation.

1.2 Version 04V03

General Information

- Documentation of previously undocumented and new commands for the software standard 05-18V06 WIN-HMI.
- Inclusion of a table with logical links of the FI commands in chapter 5.
- Inclusion of a table above the command times in chapter 05.
- Inclusion of the component types for the CNC and SPS hardware in the file "IND_DEV.INI". Expansion of the FI command "DTY" by the output of the component types "Componenttype1=" and "Componenttype2=".

Version ID, Rexroth Indramat Software Components

Entries in the "C:\IND_BASE\INDRAMAT.INI" file:

- IfDllMode = 04.20
- Version 04V03

Software components contained normally within the function interface :

- All Rexroth Indramat System200 GUIs of Version 18V06.

FI Commands Changes / Additions

- FI command "AMM7" for outputting of 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 CNC processes. The FI command "PTC2” returns the tool management data of a defined CNC 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 name of an NC program package (MTCX device group).
- The FI command "PVF" handles the formatted reading and writing of SPS variables, arrays and structures (MTCX, MISX and MTAX device groups).
- The FI command "PVT" reads the type of SPS variables, including structures and arrays (MTCX, MISX and MTAX device groups).
- The FI command "SID1" returns information regarding the installation. This information includes the installation path, the software version being used and 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, name, 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 of data from the zero offset table has been expanded by the FI commands "ZOD1" and "ZOD2" (MTCX device group). The FI command "ZOD" is no longer used!
- New, speed-optimised FI command "GPP" for reading-out the global process parameters (MTCX device group).
- New, speed-optimised 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 "Initialisation" (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 the output of 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 of 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 case of a form error of 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).

1.3 Version 04V02

General Information

- New chapter in the documentation "Answers to Frequently Asked Questions (FAQ)".
- The wasting of resources has been mostly removed in Service Pack 2 of GUI 18V05.

Version ID, Rexroth Indramat Software Components

Entries in the "C:\IND_BASE\INDRAMAT.INI" file:

- IfDllMode = 04.10
- IfVersion = 04V02

Software components contained normally within the function interface :

- All Rexroth Indramat System200 GUIs of Version 18V05 with Service Pack 3.

FI Commands Changes / Additions

- Module commands MCD1, MCM1 and MCS1 released for the MISX device group.
- Module commands MCD1, MCM1, MCP1, MCS1, MAP1 released 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 Changes / Additions

- Waste of resources in logic process resolved.
- Expansion from 15 to a max. 255 group requests during cyclic requests (see "Routines for Cyclic Reading via Pipes").

1.4 Version 04V01

General Information

- Inclusion of the PRO-Version as a software option in the installation program.

Version ID, Rexroth Indramat Software Components

Entries in the "C:\IND_BASE\INDRAMAT.INI" file:

- IfDllMode = 04.10
- IfVersion = 04V01

Software components contained normally within the function interface :

- All Rexroth Indramat System200 GUIs of 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 optimiser (correction of timeout recognition).
- New SYS-Message "MSG_PC__ALIVE" in PC network.

1.5 Version 04V00

In contrast to the previous 03VRS versions, fundamental changes have been made in this version:

General Information

- Delivery of a Visual Basic example connection to the function interface (application including source codes).
- Delivery of a printed documentation as online help in NT/95 help format.
- Provision of an installation program for the function interface.
- New! FI commands for an NC download. (MTCX Device Group)

Version ID, Rexroth Indramat Software Components

Entries in the "C:\IND_BASE\INDRAMAT.INI" file:

- IfDllMode = 04.00
- IfVersion = 04V00

Software components contained normally within the function interface :

- All Rexroth Indramat System200 GUIs of Version 18V04 with Service Pack 2.

FI Commands Changes / Additions

- FI commands "XYZ" and "XYZ1" implemented with new formatted output: AAC1, AAS1, ADN1, AFO1, APO1, ARO1, ASO1, AZB1, MFO1, MRO1, MSO1. The FI command "XYZ" should no longer be used!
- The FI command "ABN" has been replaced by the FI commands "ASM", "AMM".
- New functions for the BOF/GBO for calling WIN32 applications.
- New functions for a WIN32 application at the function interface for calling the BOF/GBO.
- Expansion of the data structure for the BOF/GBO.

- Expanded function calls for the device configuration.
- Message for activating or deactivating a PC in the PC network.
- Expansions to the device-independent access functions.

Basic Processes Changes / Additions

- Error correction of the telegram optimiser (correction of time-out recognition).
- Expansions in the SPS data optimiser.
- Enlarged input buffer for the telegram optimiser.
- Reworking of the internal interface.
- Error correction in data provision by mean of the "ReadGroupItem" routine.
- Error correction of the communication channel.
- Error corrections in the internal DLL interfaces.
- Correction of the INDIF200.DLL (correction of the binary result of spindle 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".

1.6 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

The Rexroth Indramat Function Interface is a unified data interface from Rexroth Indramat for application programs (often referred to in the following as clients) based on the Windows NT platform.

Requirements Regarding the 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 simple to handle as possible. The main requirements are to be able to access CNC/PLC data of large sizes at fast access and reaction speeds. Several clients can access the data.

Targets The Rexroth Indramat Function Interface has exactly this goal, 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 program languages Visual C++ or Visual Basic. The user is therefore provided with a powerful interface with which he can communicate with Rexroth Indramat devices and user interfaces using mnemonic function calls. The Function Interface is therefore a universal solution for data communication

Validity This description is valid for the following versions:

- BOF/GBO: 19Vxx
- Function Interface 05Vxx
- Windows NT Workstation/Server: 4.0
- Visual C++: 5.0
- Visual Basic: 5.0

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 for communication services. The services, i.e. the functions of the DLL , fulfil the communication tasks that are required for reading in individual data, the cyclic reading of data, the cyclic reading of data groups, the writing of data and the processor communication with the Rexroth Indramat user interfaces and devices. The Function Interface can communicate with a maximum of seven independent user programs (clients). A user program can thereby be, e.g. a customized user interface, a Rexroth Indramat DDE-Server or a communication driver to another data interface. In the direction of the device, up to eight parallel communication channels are supported, whereby a communication channel establishes a connection to one of a maximum of 16 data terminals. The physical communication address can here be a serial interface (RS232/RS485), a Dual-Port-RAM or a Shared Memory area .

3 Structure and Configuration Examples

3.1 The Structure of the Function Interface

The Function Interface when seen as an entire component consists of the three basic processes :

- Logic process
- Communication process
- BOF process

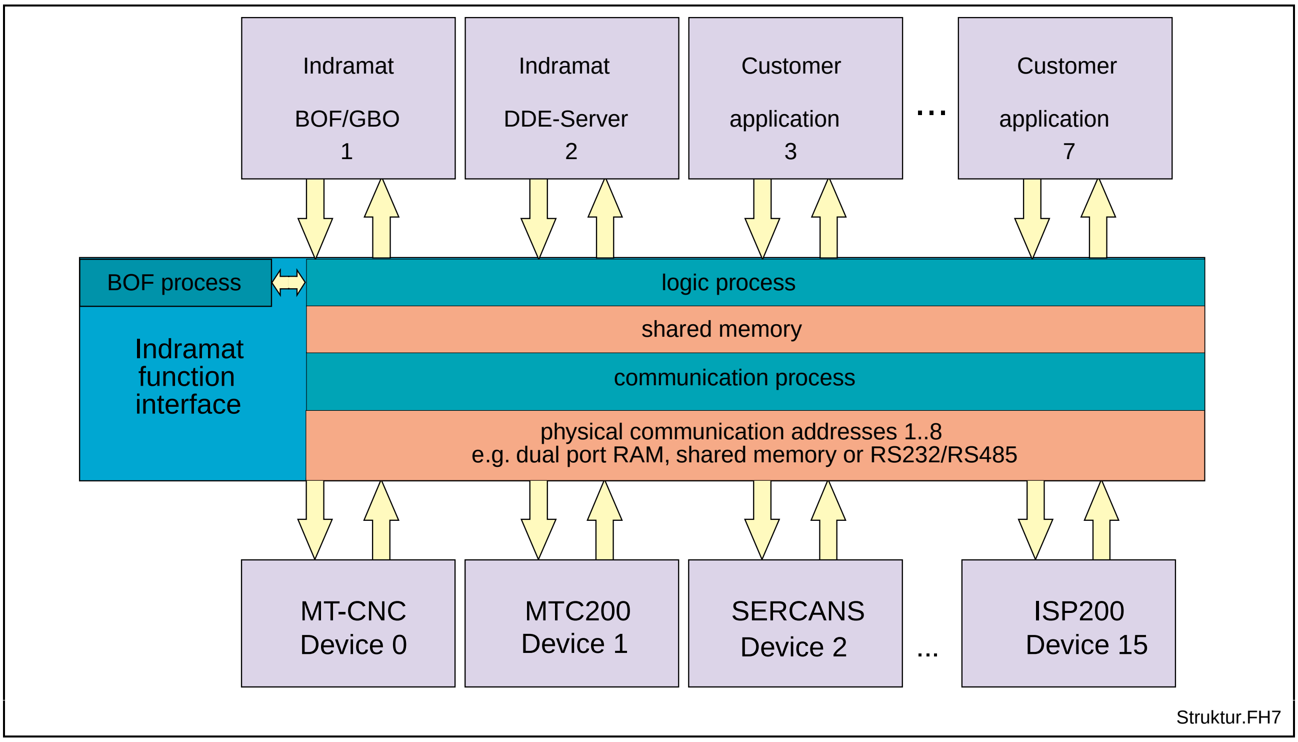


Fig. 3-1: Structural Survey of the Function Interface

Logic Process

The logic process provides the user program (client) of the actual interface with the services provided in the previous chapter. To do this, it opens a logic channel (LOG Channel) for every connected client. The number of active LOG channels thereby directly depends on the number of the connected clients. Furthermore, the logic process is a data interface to all defined devices and to the administration and status terminals that are administered by the BOF process. From the point of view of the client, 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. Chapter 4, 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 (MT-CNC, MTC200-P, ISP200-P, 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 memory with the configured Rexroth Indramat devices via the dual port RAM . The communication process opens a communication channel (thread) for each of these configured devices. It thereby allows communication via various communication methods and via several parallel interfaces.

Windows NT Core-Mode Driver (MTCNC00I.SYS)

Under operating system Windows NT, the core-mode driver "MTCNC00I.SYS" is required for access to the hardware components (serial interface, dual port RAM etc.); see chapter 5.3, Installation of the Windows NT Core-Mode Driver.

This core-mode driver makes the connection between the function interface and a serial interface or dual port RAM. The core-mode driver for the function interface supports the physical RS232-C interface, dual port RAM for the MTC200-P, IPS200-P and RS485.

ATTENTION

The function interface only supports RS485 interfaces that have been authorised and tested by Rexroth Indramat via the core-mode driver.

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

BOF Process

The BOF process is designed as an internal user program (client) 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 BOF process thereby gathers, e.g. MTC200-P control data with data from the PC hard drive which a client can then access. The BOF process thereby fulfils administrative tasks.

3.2 Configuration Examples and Connection Possibilities

MPI Connection with Profibus FMS

The following figure shows the connection of the Rexroth Indramat MPI (Multi-Protocol-Interface) with Profibus 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 GUI (**GUI** = **G**raphical **U**ser **I**nterface) runs within the DOS environment under Windows NT. The connection to the second LOG channel (Logic Channel 2) is realised via a virtual device driver . The function interface is informed of certain events (PLC program download, parameter download, etc.) by the Rexroth Indramat GUI via this connection. These events are made available to the client by the function interface in the form of system messages (SYS Messages); see chapter „Access Functions for Working with SYS Messages“. The MPI connection to the function interface is made via the third LOG channel (Logic Channel 3).

The figure also makes clear that three parallel Windows NT processes access the data from a Rexroth Indramat device (MT-CNC control). As a rule, every process has the same access rights.

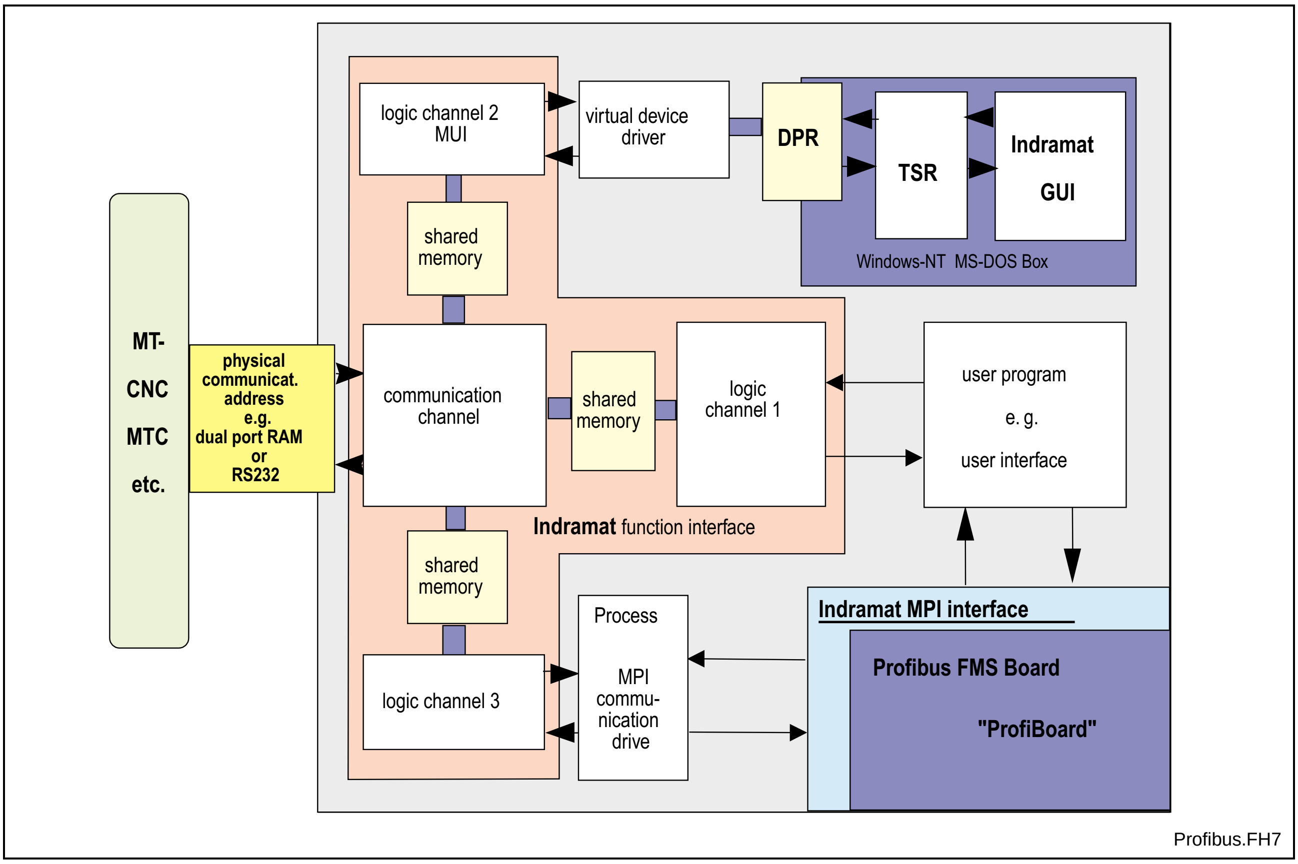


Fig. 3-2: MPI Connection with Profibus FMS

Rexroth Indramat GUI and DDE Server

The following figure shows the software structure with the 19VRS Rexroth Indramat GUI (HMI+BOF) as well as when using the Rexroth Indramat DDE server.

The "WIN-MTC" and "WIN-HMI" components are a part of the Rexroth Indramat GUI called WIN200. Components are converted fluently from BOF to WIN200. The basis is WIN200. The DOS environment is integrated into WIN200 “based on menus”, i.e. certain menus that are a part of the BOF are called by the WIN200. The program jumps back to the WIN200 user interface. This allows you to swap menus from the user interface into menus of the respective WIN type.

The DDE server allows the connection via standard communication mechanisms to external program packages such as, e.g. WONDERWARE "InTouch". Furthermore, using the NetDDE option, the DDE server allows a connection to be made via a network.

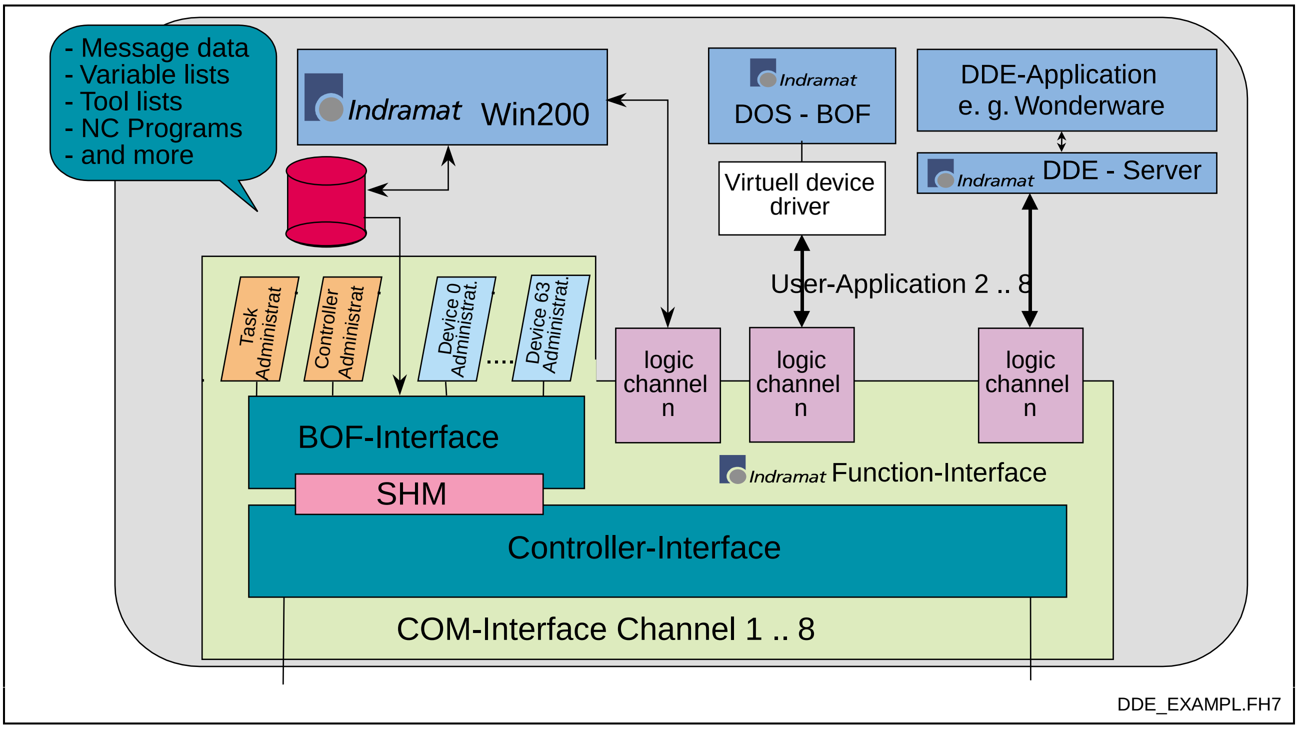


Fig. 3-3: Software Structure: Rexroth Indramat GUI and DDE Server

Rexroth Indramat GUI and OPC Server

OPC™ stands for **O**LE for **P**rocess **C**ontrol. OLE (**O**bject **L**inking and **E**mbedding) was originally introduced by Microsoft for the communication between software components. Today, we refer to the terms COM (**C**omponent **O**bject **M**odel) or DCOM.

The goal of OPF is to create a unified communication interface for process data from any source such as, e.g. PLC and NC controls.

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

- Only a minimum of know-how regarding the controls is required in order to communicate with the control software.
- If an application has to communicate with different makes of control then no adjustment has to be made.

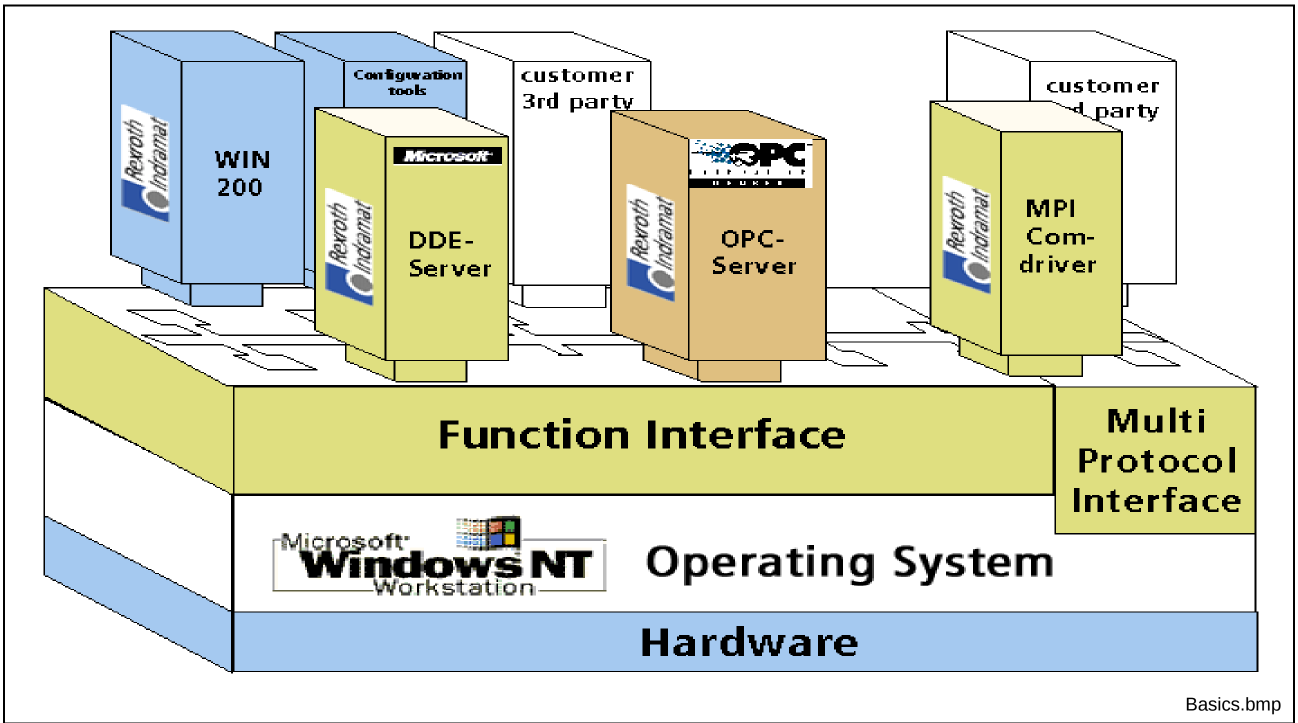


Fig. 3-4: Survey of Connection Possibilities

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 and MTC200-P). On the one hand, the device 00 (MTC200-R) hereby communicates via the serial interface (COM1), and device 01 (MTC200-P) via a dual port RAM with the communication process. 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 corresponding device address is specified in the function interface command (F1 command) (see Chapter 6, Design and Availability of the F1 Command).

Note: Several cyclic requests (F1 commands) can easily be combined at both devices. See chapter 4.3, 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 condition. F1 commands that have been requested are thereby checked as to their validity for the configured device group. Any errors in command mnemonics can thereby already be intercepted in the top level.

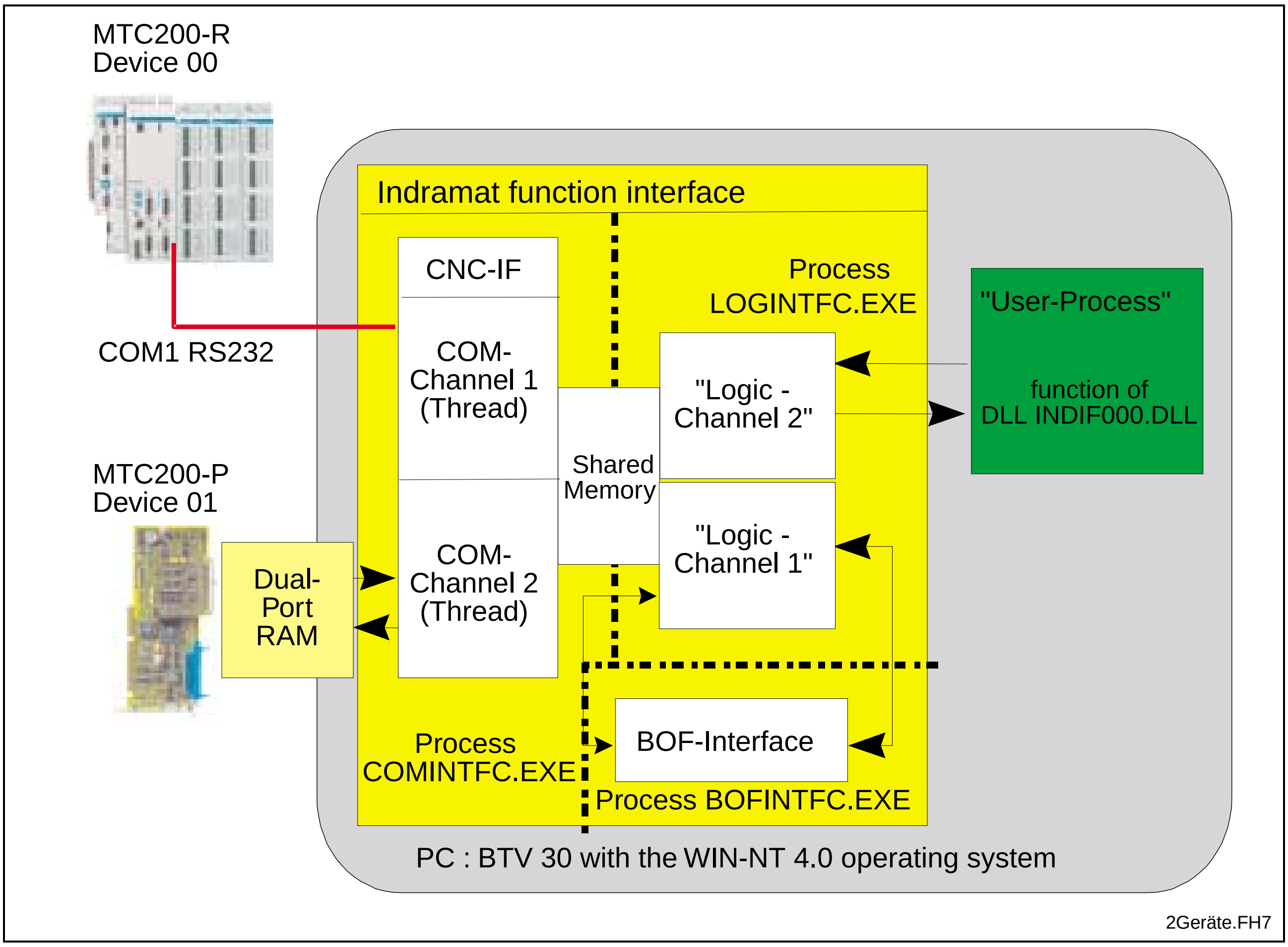


Fig. 3-5: Communication between a Client and Rexroth Indramat Devices

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 operation of several clients.

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

Here, the function interface allows parallel communication via various interfaces. In direction of the client, four programs are connected to the function interface in the following example. Every client can communicate with every device, independent of the other clients. When operating with 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.

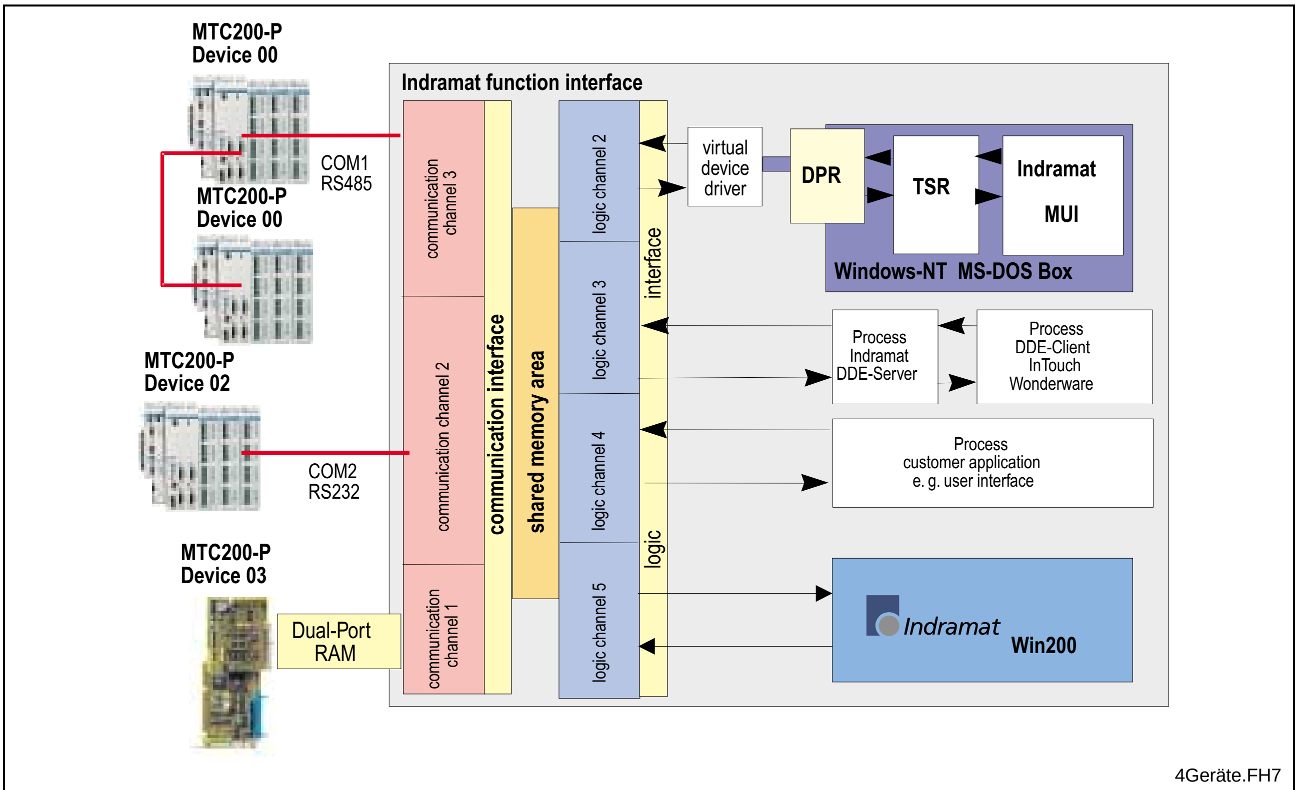


Fig. 3-6: Communication between Several Clients and Rexroth Indramat Devices

Structure of Function Interface with Configuration Data

The following figure shows the functional connection between the configuration data and the data structures stored in the function interface.

The required configuration interfaces and the existing device types are configured via the Rexroth Indramat System Configurator. The configuration that is created is stored in the "IND_DEV.INI" file (see Chapter 5.5, Directory and File Structure of the Function Interface).

The device parameters are set via the user interface. In this way, e.g. the existing CNC processes and the number and characteristics of the NC axes are set.

During the initialization phase of the function interface, the following device types are checked:

- Version of function interface and GBO version,
- Configuration settings in the "IND_DEV.INI" file,
- Validity of the PLC program,
- Validity of the NC parameter set, and the
- Initialization of the machine keys.

For the data that is most frequently required, the BOF process creates various data memory blocks. The data memory blocks that are created have the following meaning and contain the following information:

Common MAP Connection names of the client and lifesign of the application.

System MAP Entire device configuration. Current language setting of the Rexroth Indramat GUI. Number of configured devices as well as the interface parameters of the devices.

Data MAP x Device name. Index identification of the PLC program and the parameter set (MTC). Process and axis configuration (MTC). Additional frequently used configuration information. The function interface creates a Data MAP for every configured device. The design of the respective Data MAP depends on the device type that is recognized.

The Data MAPs are automatically updated by the function interface e.g., after a parameter or PLC program download. The function interface provides system messages (SYS messages) to allow the client to react to such changes. An application connected to the function interface is, e.g. informed of a parameter download via these SYS messages.

Note: Dealing with SYS messages is described in chapter 4.5, Access Functions for Working with SYS Messages.

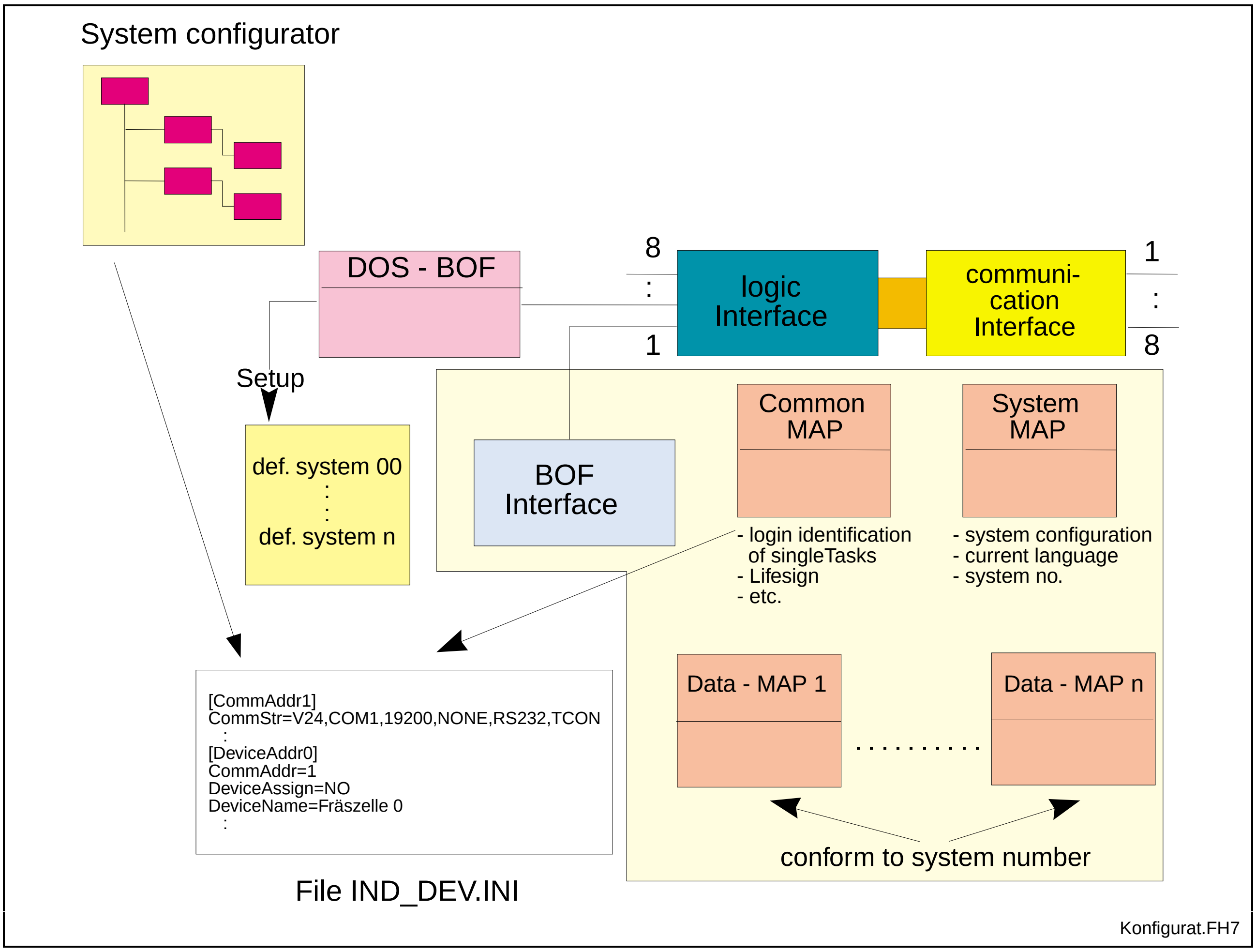


Fig. 3-7: Structure of the Function Interface with the Configuration Data

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 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 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 drive can also be freely selected.

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

- During an SPS 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 condition. The system messages for the SPS program and/or parameter download are to be considered in the logic of the client.
- Reading and writing of SPS data is limited. Using the FI command PVS (see 7, Function Interface Commands), SPS variables with a maximum length of 240 bytes can be read and written. SPS structures and arrays can have a dynamic length. Extremely precise planning for communication with the SPS is required.
- In principle, an SPS variable can be written using the function interface. However, only those SPS variables should be written in the application that are correspondingly found in the SPS program.

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

- Signals from the process/axis interface should never be directly changed by the program. Use a read/write buffer in the SPS.

ATTENTION

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 nullified.

Software for Developing of Clients (PRO Version)

The runtime version of the function interface is on the diskettes labelled:

- SWD-FUN*PC-RUN-05VRS-MS-C1,44 (Disk Version)
- SWD-FUN*PC-RUN-05VRS-MS-CD650 (CD-ROM Version)

The configuration software required for the development of clients in the programming languages Visual C/C++ and Visual Basic is available on installing the diskette:

- SWD-FUN*PC-PRO-05VRS-MS-C1,44 (PRO-Version)

Note: The Visual Basic examples for the individual function interface routines are also delivered with the PRO version.

Visual C/C++

The following C sources (*.C) and header files (*.H) are to be included in accordance with the settings given in the chapter on the C++ development environment (page 4-4) in the client:

File	Usage
INDIF000.C	Load procedure for the DLL functions.
INDRAMAT.C	Load procedure for the global DLL functions.
INDIF000.H	Function definition of the individual routines.
INDIFX00.H	Function definition of the individual routines.
INDRAMAT.H	Global function interface routines.

Note: The source files are contained in the directory "C:\MT-CNC\Sample\FI\VCDemo\" after the standard installation (see chapter 5.4, Installing the Function Interface).

Visual Basic: No settings should be made in the development environment of Visual Basic 5.0. Only the following source file should be included in the client:

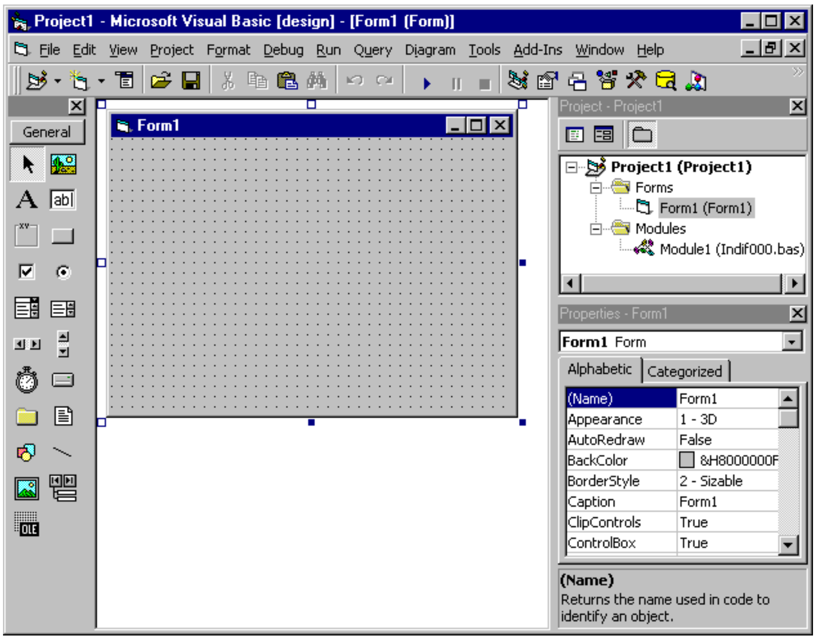
File	Usage
INDIF000.BAS	Load procedure for the DLL functions; Function definitions of the individual routines;

Note: The source files are contained in the directory "C:\MT-CNC\Sample\FI\VBDemo\" after the standard installation (see chapter 5.4, Installing the Function Interface).

Via the menu item Project:

- Add Module

the source file "INDIF000.BAS" is included as a new module in the client.



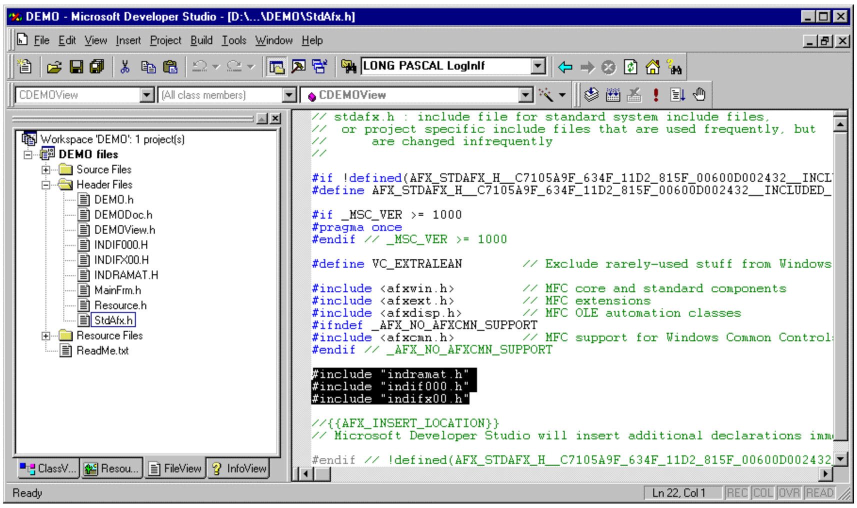
VBprojekt.bmp

Fig. 4-1: Including the "INDIF000.BAS" file in the Visual Basic user project.

Settings of the C++ Development Environment

To allow the functions of the "INDIF000.DLL" library (see chapter, Content Of The Directory "[LW]:\...\IND_DLL\", p. 5-26), to be globally available to the function interface, the following header files must be incorporated into the client using the syntax "#include":

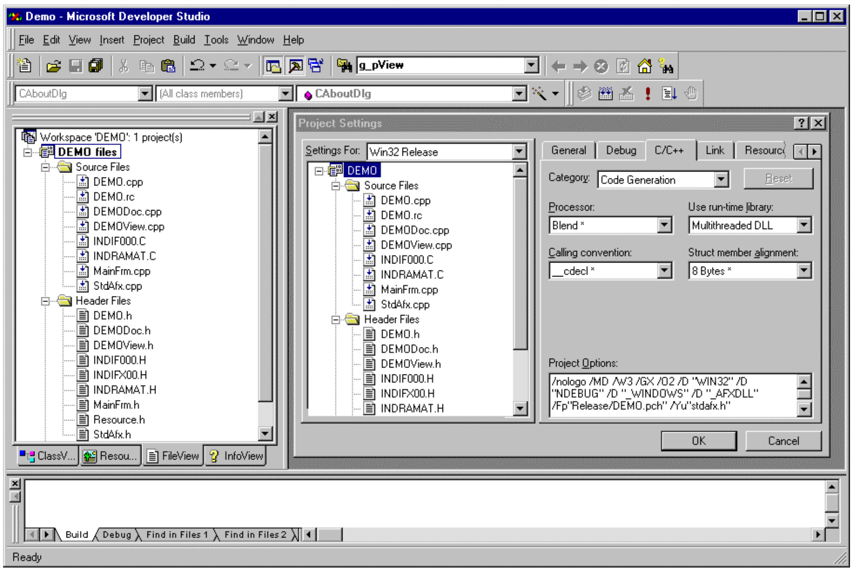
- INDIF000.H,
- INDIFX00.H and
- INDRAMAT.H



StdAfx.bmp

Fig. 4-2: Including the Rexroth Indramat Header Files in the Client

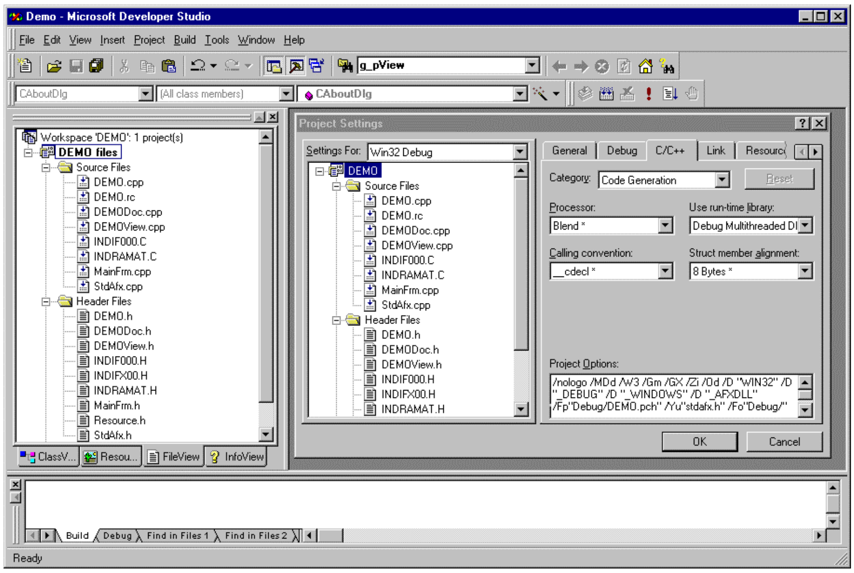
For Visual C++ 5.0 in den Project-Settings: "For Win32 Release" must be selected in the "C/C++" card under the category "Code-Generation" in the combo box "Use run-time library" of the "Multithreaded DLL" entry.



ProjectSettings1.bmp

Fig. 4-3: Project Settings "For Win32 Release": Multithreaded DLL

In the 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++" card.

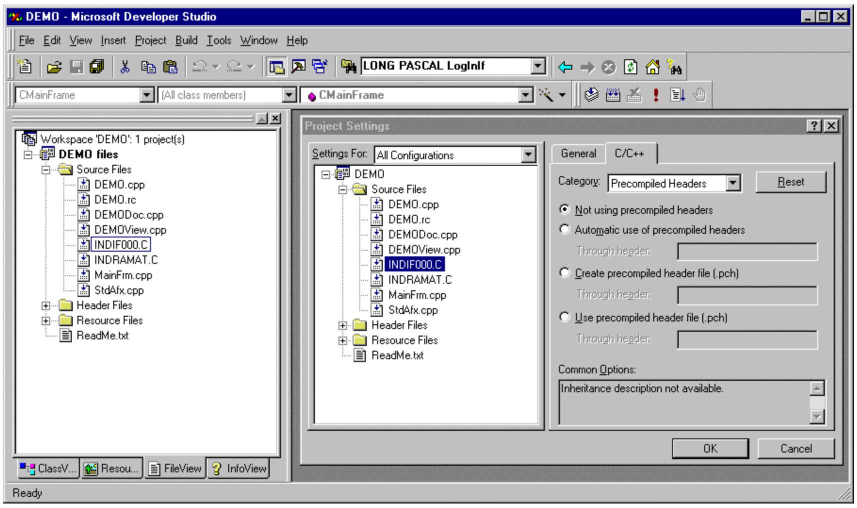


ProjectSettings2.bmp

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

Furthermore, from the project settings, select the "Not using precompiled Headers" setting from the "Precompiled Headers" category in card "C/C++" and "For All Configurations" for the C sources:

- INDIF000.C and
- INDRAMAT.C



ProjectSettings3.bmp

Fig. 4-5: Project Settings "For All Configurations"

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 first always be called. If work with the function interface has already been completed, then the logout routine ”LogOutIf” should be called.

Login Routine "LogInIf"

Explanation A client connects to the administration structure of the function interface via the ”LogInIf” routine.

Syntax

```
LONG PASCAL LogInIf (    CHAR *lpcTaskName,
                         CHAR *lpcCommandLine,
                         CHAR *lpcParentWinName,
                         HANDLE *lhTerminateEvent,
                         UCHAR lucIfChannel,
                         UCHAR lucIfChannelGrp,
                         HANDLE *lhSysMsgEvent,
                         UCHAR *lucTaskId,
                         DWORD *ldwIFChannelId   );
```

Pass Parameters

Parameter	Explanation
[IN] lpcTaskName	Pointer to the name of the client
[IN] lpcCommandLine	Pointer to the command line for the BOF and logic process. As a rule, the pass parameters are passed on to the client here. The function interface can hereby be switched to diagnostics mode via the command line 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)
[OUT] lhTerminateEvent	HANDLE to the termination event of a process.
[IN] lucIfChannel	Decides whether or not the process requests a LOG channel 0= no LOG channel request >0 = LOG channel is requested (normal case)
[IN] lucIfChannelGrp	Maximal number of function calls within a group request [1...MAXGRP]. Default: 10 (refer to entry in the "INDIFX00.H" file)
[OUT] lhSysMsgEvent	HANDLE on the Sys-Msg-Event.
[OUT] lucTaskId	TaskID, that is assigned to a client on logging in for administration reasons [1..MAX_TASK_ANZAHL] (see entry in the "INDIFX00.H" file).
[OUT] ldwIFChannelId	Assigned ID of the Communication Channel [2...8]

Return Values

0: Request successful
1...n: Request unsuccessful (see chapter 8, Error Codes)

Note: Additionally, an error can be requested with the "ReadGroupItem" Routine, (page 4-13) in the form of a general error result line. Additional information is contained in chapter 8.1, General Error Result Lines

LogInIf Example (Visual Basic: VBDEMO.FRM)

```
Private Sub Form_Load()

’INPUT-Values of the LogInIf-routine
’*********************************
Dim TaskName As String          'Application’s name
Dim CommandLine As String       'Command for starting conditions, e.g. "/C=t /B=w"
Dim ParentWinName As String     'Titlebar’s (Window’s )name
Dim IfChannel As Byte           'Function Interface Channel
Dim IfChannelGrp As Byte        'Value for group request
Dim ResBuf As String * 32768    'Result buffer

'Return-Values of the LogInIf-routine
'*********************************
Dim TaskId As Byte
Dim IfChannelId As Long

'General declarations
'******************
Dim lRet As Long                    'Routine's return value
Dim ErrMsg As String                'Error message string
Dim nHookList(0 To 4) As Integer    'Number of FI-System Messages (FI-SYS-MSGs)
Dim lpThreadId As Long

'Timer interval initialization
'***************************
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
ParentWinName = "VBDemo"    'Titlebar’s (Window’s )name
IfChannel = 1               'Function Interface Channel 1 requested
IfChannelGrp = 10           'Max. value for group request
lRet = 1                    'Default Return value = 1 for error handling

'Call LogInIf-Routine (Start Interface)
'************************************
lRet = LogInIf(TaskName, CommandLine, ParentWinName, SysThread.hTerminateEvent,
IfChannel, IfChannelGrp, SysThread.hSysMsgEvent, 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
Else    'Function-interface channel identification output
    VBDemoStatus.BackColor = QBColor(10) 'set BackgroundColor to bright green
    VBDemoStatus.Caption = "Login succeeded on FI-Channel " & IfChannelId
End If

'Creating Function Interface System Message List (FI-SysMsg)
'*******************************************************
nHookList(0) = 4                'Number of FI-SYS-MSGs
nHookList(1) = MSG_PCLUPDBEG    'PLC Download Begin
nHookList(2) = MSG_PCLUPDEND    'PLC Download End
nHookList(3) = MSG_PARUPDBEG    'Parameter Download Begin
nHookList(4) = MSG_PARUPDEND    'Parameter Download End

lRet = HookIfMsgList(nHookList(0))      'Ptr-Handed over in Basic is equalto C

If lRet Then   'error handling
    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, _
```

```
                        0, _
                        AddressOf SysThread.SysMsgThreadProc, _
                        0, _
                        0, _
                        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
```

```
// General Declaration
//********************************
LONG lRet;
CHAR acErrMsg[80];
// Starting parameters of the LogInIf-Routine
//***********************************************
HANDLE ghTerminateEv = NULL;
HANDLE ghSysMsgEv = NULL;
UCHAR gucTaskld = 0;
DWORD gdwlFChannelld  = 0;
// LogInIf-Routine (Start Interface)
//**************************************
IRet = LogInIf("VCDemo.exe",  // Name of user program,
m_lpCmdLine,                  // Command, e.g. "/C=t",
"Demo",                       // Window´s Name,
&ghTerminateEv,               // HANDLE to terminate event,
1,                            // Interface channel requested,
10,                           // Max. number of function calls in groupe,
&ghSysMsgEv,                  // HANDLE to Sys-Msg-Event,
&gucTaskld,                   // Task-ID,
&gdwlFChannelld);             // Communication channel - ID

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

Log out Routine "LogOutIf"

Explanation A client logs out itself with the administration 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 8, Error Codes)

Note: Additionally, an error can be requested with the "ReadGroupItem" Routine, (page 4-13) in the form of a general error result line. Additional information is contained in chapter 8.1, General Error Result Lines.

LogOutIf Example (Visual Basic:

```
Public Sub Form_Terminate()
'IN-/Output Values
'*****************
Dim lRet As Long            'Routine's return value
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
```

LogOutIf Example (Visual C++)

```
// General Declarations
//*******************************
LONG lRet;
CHAR acErrMsg[80];
//LogOutIf-Routine (Stop Interface)
//*************************************
IRet = LogOutIf();
// Fehler Handling
//*****************
if (lRet)
{
     sprintf(acErrMsg,"Function-Interface LogOutIf ErrorCode: %ld ",lRet);
     MessageBox (GetFocus(),acErrMsg,"Function Interface Error", MB_OK);
}
```

4.3 Data Transfer and Result Evaluation Routines

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

Single Request

Group Element 1	Line 1	Column 1	...	Column j
	:	:	:	:
	Line m	Column 1	...	Column j

Note: In case of an error, (return value <> 0), the result buffer contains a general error result line that may have to be evaluated in a separate routine (see chapter 8, Error Codes).

Note: As only one command line and **no** group of command lines (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 ”GetNumberOfGroups” returns the number of groups in the result. The ”GetNumberOfRows” routine determines the number of lines (rows) of a group, and the ”GetNumberOfItems” routine determines the number of columns in the lines.

Group Request (ONLY in cyclical requests)

During a group request, several command lines (single requests) are requested simultaneously. The command lines of a group request are separated by spaces. Exactly one group element is delivered in the result for each of these command lines in a group request. The data delivered in the result buffer is thereby constructed as follows:

Group Element 1	Line 1	Column 1	...	Column j
	:	:	:	:
	Line i	Column 1	...	Column j
:	:	:	:	:
Group Element n	Line 1	Column 1	...	Column j
	:	:	:	:
	Line i	Column 1	...	Column j

Example of a Group Request

During a group request (BR_NPS... BR_ABN... BR_AGF...), the single group elements can be accessed with *[bGroup]* The following mean:

- *1st* Group Element *(bGroup* = 1): BR_NPS...
- *2nd* Group Element *(bGroup* = 2): BR_ABN...
- *3rd* Group Element *(bGroup* = 3): BR_AGF...

Note: A maximum of 256 command lines (FI commands) can be gathered together as a group request.

"DataTransfer" Routine

Explanation

Data is read or written in accordance to the configured functions using the "DataTransfer" routine (see chapter 7, Function Interface Commands").

Syntax

```
LONG PASCAL DataTransfer (     CHAR *pszFunction,
                               CHAR acValue[ ],
                               LONG ValLen,
                               LONG ValType,
                               CHAR acResBuf[ ],
                               LONG lMaxResLen,
                               LONG lResBufType   );
```

Pass Parameters

Parameter	Explanation
[IN] pszFunction	Command line
[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 6, Design and Availability of the F1 Command, Data Code)
[OUT] acResBuf	Result buffer
[IN] lMaxResLen	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 default value.
[IN] lResBufType	Data code of result data (see chapter 6, Design and Availability of the F1 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 (S. 4-13).

Return Values

0: Request successful
1...n: Request unsuccessful (see chapter 8, Error Codes)

Note:	Additionally, an error can be requested in detail with the "ReadGroupItem" Routine, (page 4-13) in the form of a general error result line. Additional information is contained in chapter 8.1, General Error Result Lines

DataTransfer Example (Visual Basic: FRM)

```
Private Sub DataTransferFunc()

'Read/Write Data from/to the various devices via the function-interface
'***********************************************************************
Dim ResBuf As String * 32768      'Result buffer
Dim lRet As Long                  'Routine's return value
Dim lLen As Long                  'Value's length
Dim pszFunction As String         'FI-command
Dim ErrMsg As String              'Error message string
Dim szBuf As String * 32768       'Buffer for controller data
Dim DataValidation As Boolean     'Flag for data validation
Dim szVal As String               'Write value

pszFunction = SingleRequest.Text  'Hand over FI-Command from Edit box
szVal = WriteValue.Text           'Hand over Write Value from Edit box

'DataTransfer to function-interface
'**********************************
lRet = DataTransfer(pszFunction, szVal, Len(szVal), 1, ResBuf, 32768, 1)

If lRet Then   'error handling
    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 successfully 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 lItemLen;
CHAR acItembuf[50];
BOOL boItemValid;

// Access to Function Interface
//*********************************

lRet = DataTransfer (szFunction,     // Command line,
szValue,                             // Value,
strlen(szValue),                     // Length of value,
1,                                   // Data code of value,
acResultbuf,                         // Results buffer,
RESULT_BUF_SIZE,                     // Length of results buffer,
```


Rexroth Indramat

```
1);                                   // Data code of results buffer
// Error Handling
  if (lRet)
{
      sprintf(acErrMsg,"Function-Interface DataTransfer ErrorCode: %ld ",lRet);
      MessageBox (GetFocus(),acErrMsg,"Function Interface Error", MB_OK);
 }
// Determine Number of Lines
//****************************
lRet = GetNumberOfRows(acResultbuf,  // Results data,
1,                                   // Group element,
& lNumOfRows);                       // Number of lines
 // 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++)
{
     // Determine Number of Elements
     //*******************************
     lRet = GetNumberOfItems(acResultbuf,   // Results data,
     1,                                     // Group element,
     i,                                     // Line,
     &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++)
     {
        // Evaluate all Results of a Line
       lRet=ReadGroupItem(acResultbuf,      // Results data,
       1,                                   // Group element,
       i,                                   // Line,
       j,                                   // Column,
       acItembuf,                           // Single result,
       50,                                  // Length of single result buffer,
       &lItemLen,                           // Length of result,
       &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

Explanation

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

Syntax

```
LONG PASCAL ReadGroupItem (     CHAR acResBuf[ ]
                                BYTE bGroup,
                                LONG lRow,
                                LONG lItem,
                                CHAR acItemBuf[ ],
                                LONG lItemBufLen,
                                LONG *plItemLen,
                                BOOL *pboItemValid );
```

Pass Parameters

Parameter	Explanation and Value Areas
[IN] CHAR acResBuf[ ]	Buffer for the entire result
[IN] BYTE bGroup	Details on group element [1...n]
[IN] LONG lRow	-1: Output of an entire table as well as all lines of a request. 1...n: of the respective result line
[IN] LONG lItem	-1: Output of a line 0: Output of the requested command with administration information [1...n]: Individual result (element of a line)
[OUT] CHAR acItemBuf[ ]	Buffer for requested partial result
[IN] LONG lItemBufLen	Length of buffer for partial result
[OUT] LONG *plItemLen	Length of partial result
[OUT] BOOL *pboItemValid	TRUE: if the value of the partial result is valid.

Return Values

0: Request successful
1...n: Request unsuccessful (see chapter 8, Error Codes)

Example of Routine "ReadGroupItem"

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, lLenBuf, &lItemLen, &boItemValid)	E1
ReadGroupItem(acResBuf, **1**, **1**, **2**, acItemBuf, lLenBuf, &lItemLen, &boItemValid)	E2
ReadGroupItem(acResBuf, **1**, **2**, **2**, acItemBuf, lLenBuf, &lItemLen, &boItemValid)	E5
ReadGroupItem(acResBuf, **1**, **3**, **4**, acItemBuf, lLenBuf, &lItemLen, &boItemValid)	E9
ReadGroupItem(acResBuf, **1**, **2**, **3**, acItemBuf, lLenBuf, &lItemLen, &boItemValid)	Error code
ReadGroupItem(acResBuf, **1**, **1**, **-1**, acItemBuf, lLenBuf, &lItemLen, &boItemValid)	E1 E2 E3
ReadGroupItem(acResBuf, **1**, **2**, **-1**, acItemBuf, lLenBuf, &lItemLen, &boItemValid)	E4 E5
ReadGroupItem(acResBuf, **1**, **3**, **-1**, acItemBuf, lLenBuf, &lItemLen, &boItemValid)	E6 E7 E8 E9
ReadGroupItem(acResBuf, **1**, **-1**, **-1**, acItemBuf, lLenBuf, &lItemLen, &boItemValid)	E1 E2 E3 E4 E5 E6 E7 E8 E9
ReadGroupItem(acResBuf, **1**, **1**, **0**, acItemBuf, lLenBuf, &lItemLen, &boItemValid)	Returns e.g.: 001234567800_CC_ ....

Example of Visual Basic/ C++

(see "DataTransfer" Routine, page 4-10)

GetNumberOfGroups" Routine

Explanation

The ”GetNumberOfGroups” routine returns the number of group elements.

Syntax

LONG PASCAL ReadGroupItem (CHAR *pszValBuf,
LONG *plGroupSize);

Pass Parameters

Parameter	Explanation
[IN] CHAR *pszValBuf	Buffer for the entire result
[OUT] LONG *plGroupSize	Number of group element

Return Values

0: Request successful
1...n: Request unsuccessful (see chapter 8, Error Codes)

Note: Additionally, the error can be requested in detail with the "ReadGroupItem" Routine, (page 4-13) in the form of a general error result line. Additional information is contained in chapter 8.1, General Error Result Lines.

Example of Visual C++

(see "DataTransfer" Routine, page 4-10)

"GetNumberOfRows" Routine

Explanation

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

Syntax

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

Pass Parameters

Parameter	Explanation
[IN] CHAR *pszValBuf	Buffer for the entire result
[IN] BYTE bGroupIndex	Number of group elements
[OUT] LONG *plNumberOfRow	Number of lines of a group element

Return Values

0: Request successful
1...n: Request unsuccessful (see chapter 8, Error Codes)

Note: Additionally, the error can be requested in detail with the "ReadGroupItem" Routine, (page 4-13) in the form of a general error result line. Additional information is contained in chapter 8.1, General Error Result Lines

Example of Visual Basic/ C++

(see "DataTransfer" Routine, S. 4-10)

GetNumberOfItems" Routine

Explanation

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

Syntax

LONG PASCAL GetNumberOfItems (CHAR *pszValBuf,
BYTE bGroupIndex,
BYTE bRowIndex,
LONG *plNumberOfItems);

Pass Parameters

Parameter	Explanation
[IN] CHAR *pszValBuf	Buffer for the entire result
[IN] BYTE bGroupIndex	Number of group elements
[IN] BYTE bRowIndex	Line index: 0: Number of partial results
[OUT] LONG plNumberOfItems	Number of partial results for a particular line

Return Values

0: Request successful
1...n: Request unsuccessful (see chapter 8, Error Codes)

Note: Additionally, the error can be requested in detail with the "ReadGroupItem" Routine, (page 4-13) in the form of a general error result line. Additional information is contained in chapter 8.1, General Error Result Lines

Example of Visual Basic/ C++

(see "DataTransfer" Routine, S. 4-10)

4.4 Routine for Cyclical Reading via Pipes

The pipe access functions are used for cyclical reading of device data via the function interface. Thereby, several command lines can be passed simultaneously via a group request. The command lines of a group request are separated by a space (refer here also to the "ReadGroupItem" Routine on page 4-13).

Note: A maximum of 256 command lines (FI commands) can be gathered together 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 the cyclical reading of the data.

Syntax

```
LONG PASCAL StartCyclicPipe (   WORD wPipe,
                                CHAR *pszFunktion,
                                LONG lBufSize,
                                LONG lGroupSize,
                                DWORD dwSleep );
```

Pass Parameters

Parameter	Explanation
[IN] wPipe	Pipe number [1...1000]
[IN] *pszFunktion	String group in accordance to the defined function calls.
[IN] lBufSize	Size of result buffer [Byte]
[IN] lGroupSize	Number of group elements [1...n]
[IN] dwSleep	Read delay time [ms]

Return Values

0: Request successful
1...n: Request unsuccessful (see chapter 8, Error Codes)

Note: Additionally, the error can be requested in detail with the "ReadGroupItem" Routine, (page 4-13) in the form of a general error result line. Additional information is contained in chapter 8.1, General Error Result Lines

StartCyclicPipe (Visual Basic:

```
Public Sub StartCyclicFunc()
'Start of a cyclic request
'*************************
Dim lRet As Long                'Routine's return value
Dim ErrMsg As String            'Error message string
Dim pszFunction As String       'FI-command
pszFunction = CyclicRequest.Text    'Hand over FI-Command from Edit box
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
    CyclicRun = True                    'Flag for a cyclic request is started
    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,
szGroupFunction,                     // Group function call,
RESULT_BUF_SIZE,                     // Size of results buffer,
2,                                   // Number of group elements,
500);                                // Read delay time [ms]
// Error Handling
//*******************
if (lRet)
{
    sprintf(acErrMsg,"Function-Interface StartCyclicPipe ErrorCode: %ld ",lRet);
    MessageBox (GetFocus(),acErrMsg,"Function Interface Error", MB_OK);
}
```

"ReadCyclicPipe" Routine

Explanation

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

Syntax

LONG PASCAL ReadCyclicPipe (
WORD wPipe,
CHAR acResult[ ],
LONG lBufSize,
BYTE *pbGroupFault,
LONG *plAttr);

Pass Parameters

Parameter	Explanation
[IN] WORD wPipe	Number of the pipe
[OUT] CHAR acResult[]	Buffer for the entire result
[IN] LONG lBufSize	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

0: Request successful
1...n: Request unsuccessful (see chapter 8, 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, p. 4-13. Additional information is contained in chapter 8.1, General Error Result Lines

ReadCyclicPipe Example (Visual Basic: VBDEMO.FRM)

```
Private Sub CyclicOutputTimer_Timer()

'IN-/Output Values
'*****************
Dim lRet As Long                    'Routine's return value
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 length
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
            If lRet Then   'error handling
                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 successfully completed"
End If
End Sub
```

ReadCyclicPipe Example (Visual C++)

```
// General Declarations
//********************************
LONG lRet;
CHAR acErrMsg[80];
int i;
// Starting Parameters of the ReadCyclicPipe-Routine
//**********************************************************
CHAR acResultbuf[RESULT_BUF_SIZE];
UCHAR bIndexItemFault;
LONG lAttr;
// Starting Parameters of the GetNumberOfGroups-Routine
//*************************************************************
LONG lNumOfGroups;
// Read Pipe
//************
lRet = ReadCyclicPipe(wPipeNo,              // Pipe Number,
acResultbuf,                                // Results buffer,
RESULT_BUF_SIZE,                            // Length of results buffer,
&bIndexItemFault,                           // Index of group-
                                            // element with errors,
&lAttr);                                    // Results attribute
// Fehler Handling
if (lRet)
{
     sprintf(acErrMsg,"Function-Interface ReadCyclicPipe ErrorCode: %ld ",lRet);
    MessageBox (GetFocus(),acErrMsg,"Function Interface Error", MB_OK);
}
// Determine Number of Groups
//****************************
lRet = GetNumberOfGroups(acResultbuf,              // Results buffer,
       &lNumOfGroups);       // Number of groups,
// Error handling
if (lRet)
{
sprintf(acErrMsg,"Function-Interface GetNumberOfGroups ErrorCode: %ld",lRet);
   MessageBox (GetFocus(),acErrMsg,"Function Interface Error", MB_OK);
}
// Results Evaluation
//************************
for (i=1; i<=lNumOfGroups; i++)
{
     // For every group result evaluation
     // e.g.  LONG lItemLen;
     //       CHAR acItembuf[50];
     //       int iItemValid;
     //
     lRet=ReadGroupItem(acResultbuf, // Results buffer,
     i,                              // Group element,
     1,                              // Line,
     1,                              // Element,
     acItembuf,                      // Single result buffer,
     50,                             // Length of single result buffer,
     &lItemLen,                      // Length of single result,
     &iItemValid);                   // IS single 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”.

Syntax **LONG PASCAL StopCyclicPipe (WORD wPipe);**

Pass Parameters

Parameter	Explanation
[IN] WORD wPipe	Pipe number

Return Values

0: Request successful
1...n: Request unsuccessful (see chapter 8, Error Codes)

Note: Additionally, an error can be requested with the "ReadGroupItem" Routine, (page 4-13) in the form of a general error result line. Additional information is contained in chapter 8.1, General Error Result Lines

StopCyclicPipe Example (Visual Basic: VBDEMO.FRM)

```
Public Sub StopCyclicFunc()
'Stop of a cyclic request
'************************

'IN-/Output Values
'*****************
Dim lRet As Long            'Routine's return value
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 stoped

End If
End Sub
```

StopCyclicPipe Example (Visual C++)

```
// General Declarations
//*******************************
LONG lRet;
CHAR acErrMsg[80];

// Close Pipe
//******************
lRet = StopCyclicPipe(wPipeNo);             // Pipe number
// Error Handling
//*******************
if (lRet)
{
     sprintf(acErrMsg,"Function-Interface StopCyclicPipe ErrorCode: %ld ",lRet);
     MessageBox (GetFocus(),acErrMsg,"Function Interface Error", MB_OK);
}
```

"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 administration structure of the function interface established by "StartCyclicPipe" routine (page 4-16, also refer to "ResumeCyclicPipe" page 4-21).

Syntax

LONG PASCAL SuspendCyclicPipe (WORD wPipe);

Pass Parameters

Parameter	Explanation
[IN] WORD wPipe	Number of the pipe

Return Values

0: Request successful
1...n: Request unsuccessful (see chapter 8, Error Codes)

Note: Additionally, an error can be requested with the "ReadGroupItem" Routine, (page 4-13) in the form of a general error result line. Additional information is contained in chapter 8.1, General Error Result Lines.

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 return value
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)

     If lRet Then   'error handling
         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++)

```
// General Declarations
//*******************************
LONG lRet;
CHAR acErrMsg[80];

// Suspend Pipe
//************************
lRet = SuspendCyclicPipe(wPipeNo);   // Pipe number

// Error Handling
//*******************

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

"ResumeCyclicPipe" Routine

Explanation

The ”ResumeCyclicPipe” routine reactivates the data request of a pipe that has been set to standby mode by the "SuspendCyclicPipe" Routine (p. 4-20).

Syntax

LONG PASCAL SuspendCyclicPipe (WORD wPipe);

Pass Parameters

Parameter	Explanation
[IN] WORD wPipe	Number of the pipe

Return Values

0: Request successful
1...n: Request unsuccessful (see chapter 8, Error Codes)

Note: Additionally, an error can be requested with the "ReadGroupItem" Routine, (page 4-13) in the form of a general error result line. Additional information is contained in chapter 8.1, General Error Result Lines.

ResumeCyclicPipe Example (Visual Basic: VBDEMO.FRM)

```
Public Sub ResumeCyclicFunc()
'Activates a suspended cyclic Pipe
'*********************************

'IN-/Output Values
'*****************
Dim lRet As Long            'Routine's return value
Dim ErrMsg As String        'Error message 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++)

```
// General Declarations
//********************************
LONG lRet;
CHAR acErrMsg[80];

// Resume Pipe
//*******************

lRet = ResumeCyclicPipe(wPipeNo);    // Pipe Number
// Error Handling
//******************

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

4.5 Access Functions for Working with SYS Messages

Although every client works in its own process area independent of other clients, it can only access the function interface in co-operation with the others.

For example, the client can not access control data for reading or writing during an NC-program download from the Rexroth Indramat GUI.

To avoid this type of resource conflict, the function interface informs the other side regarding desired resource requests by means of a SYS-message event.

Depending on the SYS message that has arrived, the client must now block or can once more open access to the corresponding resource.

To allow the client to only be informed on the status changes that it requires per SYS message, it connects itself for particular SYS messages with the ”HookIfMsgList” Routine (see page 4-23).

The program waits for the SYS-message event in a thread, usually parallel to the actual main program. After the event has arrived, the message number must first be fetched using the "GetIfMsg" Routine

(see page 4-24). After internal processing of the status change, (block, open, etc.), the message must be acknowledged by the "SetIfMsgConf" Routine

(see page 4-25), as only then will the resource status change be effective.

”HookIfMsgList” Routine

Explanation

The ”HookIfMsgList” routine connects a client for particular SYS messages. The application can then be informed at any time by the SYS-message event when one of these SYS messages occurs.

Syntax

LONG PASCAL HookIfMsgList (WORD *lpwSysMsgList);

Pass Parameters

Parameter	**Explanation**
[IN] WORD *lpwSysMsgList	List of messages

Note: These messages must be acknowledged by the process.

Return Values

0: Request successful
1...n: Request unsuccessful (see chapter 8, Error Codes)

Note: Additionally, the error can be requested in detail with the "ReadGroupItem" Routine, (page 4-13) in the form of a general error result line. Additional information is contained in chapter 8.1, General Error Result Lines.

Example for Visual Basic

(See Login Routine "LogInIf", S. 4-6)

HookIfMsgList Example (Visual C++)

```
//General Declarations
//*******************************
LONG lRet;
CHAR acErrMsg[80]

// Starting Parameters of the HookIfMsgList-Routine
//*******************************************************
WORD wSysMsgList[3] = {2,     // Number of messages,
MSG_PCLUPDBEG,                // Message constants for
MSG_PCLUPDEND};               // PLC download start and // end
// Login for SYS-MSG
//*****************************
lRet = HookIfMsgList((WORD*) &wSysMsgList);         // SYS-MSG list
// Error Handling
//*******************
if (lRet)
{
     sprintf(acErrMsg,"Function-Interface HookIfMsgList  ErrorCode: %ld ",lRet);
    MessageBox (GetFocus(),acErrMsg,"Function Interface Error", MB_OK);
}
```

"GetIfMsg" Routine

Explanation

After the client has been informed by a SYS-message event, the ”GetIfMsg” routine is called to fetch a waiting SYS-MSG message.

Syntax

LONG PASCAL GetIfMsg (WORD *pwMsgNr,
CHAR *pcSysMsgBuffer,
WORD *pwSysMsgBufferLen,
CHAR *pcTaskName);

Pass Parameters

Parameter	Explanation
[OUT] WORD *pwMsgNr	Contains the SYS-MSG message currently waiting
[OUT] CHAR *pcSysMsgBuffer	Pointer to the data contained in the SYS-MSG BUFFER; contains the device addresses as ASCII characters
[OUT] WORD *pwSysMsgBufferLen	Length of the data contained in the SYS-MSG-BUFFER
[OUT] CHAR *pcTaskName	Pointer to the task name that triggered the SYS-MSG message

Return Values

0: Request successful
1...n: Request unsuccessful (see chapter 8, Error Codes)

Note: Additionally, an error can be requested with the "ReadGroupItem" Routine, (page 4-13) in the form of a general error result line. Additional information is contained in chapter 8.1, General Error Result Lines

Example for Visual Basic

(see SYS-MSG Example (Visual Basic: SYSTHREAD.BAS), page 4-27)

GetIfMsg Example (Visual C++)

```
//General Declarations
//********************************
LONG lRet;
CHAR acErrMsg[80];
// Starting Parameters of the GetIfMsg-Routine
//**************************************************
CHAR acSysMsgBuffer[MSG_BUFFER_LEN] ="";
CHAR acTaskName[MAX_TASK_NAME_LENGTH] ="";
WORD wMsgNr;
WORD wSysMsgBufferLen;
 // Fetch SYS-Message
//*****************************
lRet=GetIfMsg(&wMsgNr,              // SYS-MSG number,
acSysMsgBuffer,                     // Buffer for SYS-MSG info,
&wSysMsgBufferLen,                  // Data length of the SYS-MSG buffer,
acTaskName);                        // Buffer for names of triggerong // Task
// Error Handling
//********************
if (lRet)
{
     sprintf(acErrMsg,"Function-Interface GetIfMsg ErrorCode: %ld ",lRet);
     MessageBox (GetFocus(),acErrMsg,"Function Interface Error", MB_OK);
}
```

"SetIfMsgConf" Routine

Explanation

The "SetIfMsgConf" access function is used for acknowledging the SYS-MSG message that has been received with "GetIfMsg". The SYS-MSG acknowledgement contains the value SYS-MSG message "_Q".

Syntax

LONG PASCAL SetIfMsgConf (WORD wMsgNr_Quit);

Pass Parameters

Parameter	Explanation
[IN] WORD wMsgNr_Quit	SYS-MSG acknowledgement

Return Values

0: Request successful
1...n: Request unsuccessful (see chapter 8, Error Codes)

Note: Additionally, an error can be requested with the "ReadGroupItem" Routine, (page 4-13) in the form of a general error result line. Additional information is contained in chapter 8.1, General Error Result Lines.

Example for Visual Basic

(see SYS-MSG Example (Visual Basic: SYSTHREAD.BAS), page 4-27)

SetIfMsgConf Example (Visual C++)

Assumption: wQMsgNr contains the SYS-MSG acknowledgement

```
// General Declarations
//********************************
LONG lRet;
CHAR acErrMsg[80];
// Acknowledge SYS-MSG
//*************************
lRet = SetIfMsgConf(wQMsgNr);              // SYS-MSG acknowledgement message
// Error Handling
//********************
if (lRet)
{
     sprintf(acErrMsg,"Function-Interface SetIfMsgConf ErrorCode: %ld ",lRet);
     MessageBox (GetFocus(),acErrMsg,"Function Interface Error", MB_OK);
}
```

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 (see "SetIfMsgConf" Routine, p. 4-25).

Note: The device address that has issued the system message is returned as an ASCII character in the buffer "CHAR *pcSysMsgBuffer" of the "GetIfMsg" Routine (p. 4-24). The buffer may possibly receive additional SYS-MSG information such as, e.g. 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 BOF/GBO.	Communication between all Rexroth Indramat devices is interrupted.
MSG_FWAUPDEND	End of the firmware download.	Communication is re-commenced. 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 BOF/GBO.	Communication between all Rexroth Indramat devices is interrupted.
MSG_PARUPDEND	End of the parameter download.	Communication is re-commenced Note: Process and axis configuration data etc., may have changed.
MSG_PCLUPDBEG	The SPS program download commences, i.e. the SPS program is loaded into the System200 components by the BOF/GBO.	Communication between all Rexroth Indramat devices is interrupted.
MSG_PCLUPDEND	End of the SPS program download.	Communication is re-commenced.
MSG_MEMUPDBEG	BOF/GBO begins to delete the data memory in the System200 components.	Communication between all Rexroth Indramat devices is interrupted.
MSG_MEMUPDEND	BOF/GBO has deleted the data memory in the System200 components.	Communication is re-commenced. Note: Configuration data has been deleted.
MSG_SYSERRGEN	If there is a system error, this SYS-MSG is issued, i.e. the SPS can no longer be accessed at the moment.	**No** interruption of communication to the Rexroth Indramat devices is necessary. (Is used for visualization of a system error from a particular Rexroth Indramat device).
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.)

SYS Message	What happens?	Reaction from the Client
MSG_MECERRGEN	If there is a fault in the mechanism, this SYS-MSG is issued.	**No** interruption of communication to the Rexroth Indramat devices is necessary. (Is used for visualization of a system error from a particular Rexroth Indramat device).
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_PC__ALIVE	A PC/device logs in/out of the PC system. 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" Routine . 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
        Exit Do     'End of the threadloop
    ElseIf lWait = 1 Then
    'SysMsgs 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 nMsgNr
                Case MSG_PCLUPDBEG
                    Demo.SuspendCyclicFunc  'Termination of a cyclic request
                    lRet = SetIfMsgConf(MSG_PCLUPDBEG_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 = "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
                Demo.ResumeCyclicFunc   'Start of a cyclic request
                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
        Exit Do     'End of Threadloop
End If
Loop
 End Sub
```

4.6 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 9, Answers to Commonly Posed Questions: FAQ).

Furthermore, we have discovered that using Windows NT without a mouse always tends to present difficulties and we have therefore listed the most important (see page 4-36) in the form of a table.

Problem	Remedy
In your application, you issue an FI command and receive: • no answer - **or** - • an unexpected answer **- or -** • an error code (see chapter 8, Error Codes)	Frequent causes: Device address has not been given **or** been incorrectly given! ⇒ Check the correct details of the FI command (see chapter 6, Design and Availability of the F1 Command and chapter 7, Function Interface Commands). ⇒ Issue the FI command that is causing problems using the VBDemo program (see Issuing FI Commands using the “VBDemo” Application, p. 4-32)
Your client no longer reacts	⇒ See Clearing Memory using the "KILLTASK.EXE" Tool, page 4-30)
Your client terminates "DR. WATSON" with a memory protection error.	⇒ See Clearing Memory using the "KILLTASK.EXE" Tool, page 4-30) ⇒ Correct the programming error and re-start your application.
The entire system (Windows NT, client and Rexroth	⇒ Check the Windows NT settings for improved performance, idling activity, swapping of the core-

Indramat GUI) react slowly.	mode driver and idling activity according to chapter 5.2, Setting the Windows NT System Properties.
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 removed from memory.	⇒ Start the Task manager, i.e. using the key combination <Ctrl>+<Shift>+<Esc> (see chapter Windows NT Task Manager, p. 5-2). ⇒ Click on the “Processes” card. ⇒ Terminate the three basic processes of the function interface and, when necessary, your client: • LOGINTFC.EXE (logic process) • COMINTFC.EXE (communication process) • BOFINTFC.EXE (BOF process) using the <End Process> button.
Your application terminates because: • required files are missing - **or** - • path entries do not exist or are incorrect.	⇒ Check to make sure the required files are located in their respective directories. ⇒ Check the path entries. **Note!** • Avoid absolute paths in your application, as any later change to the drive from, e.g. C:\ to D:\ or in the directory structure are not supported. • The system directory as well as the Windows NT drive can also be freely selected.

Clearing Memory using the "KILLTASK.EXE" Tool

This tool can be used when creating software for clearing the memory. After a standard installation (see chapter 5, Installing Windows NT and the Function Interface) it is located in the default directory "C:\MT-CNC\IND_DRV\".

The tool provides you with the following two possibilities 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, and
- (BOF process) BOFINTFC.EXE

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

ATTENTION

⇒ 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 nullified.

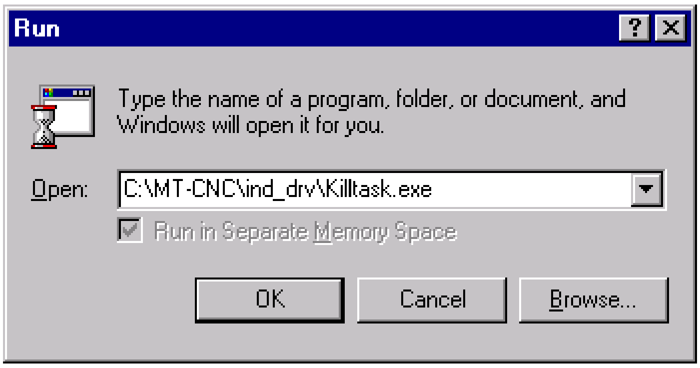
To completely reinitialize, proceed as follows:

⇒ Click on Start and then on the “Run” option.

⇒ Click on the <Find> button to search for the "KILLTASK.EXE" tool.

Note: After a standard installation (see chapter 5, Installing Windows NT and the Function Interface) the “KILLTASK.EXE” application is located in the default directory "C:\MT-CNC\IND_DRV\".

⇒ Click on the <OK> button.
All applications connected to the function interface – and the basic processes of the function interface itself – are removed from memory.



Killtask00.bmp

Fig. 4-6: Windows NT Dialog Box “Run": Complete Reinitialization

Selective Reinitialization

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

To selectively reinitialize, proceed as follows:

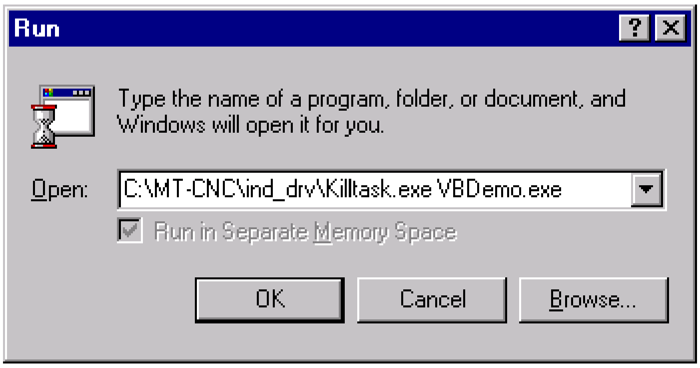
⇒ Click on Start and then on the “Run” option.

Note: You can search for the "KILLTASK.EXE" application by clicking on the „Find...“ button. After a standard installation (see chapter 5, Installing Windows NT and the Function Interface) this file is located in the default directory "C:\MT-CNC\IND_DRV\".

⇒ In the text box, enter the name of the application that is to be removed from the memory and from the administration structure of the function interface (here VBDemo.exe).

⇒ Then click on the <OK> button.

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



Killtask01.bmp

Fig. 4-7: Windows NT Dialog Box “Run": Selective Reinitialization

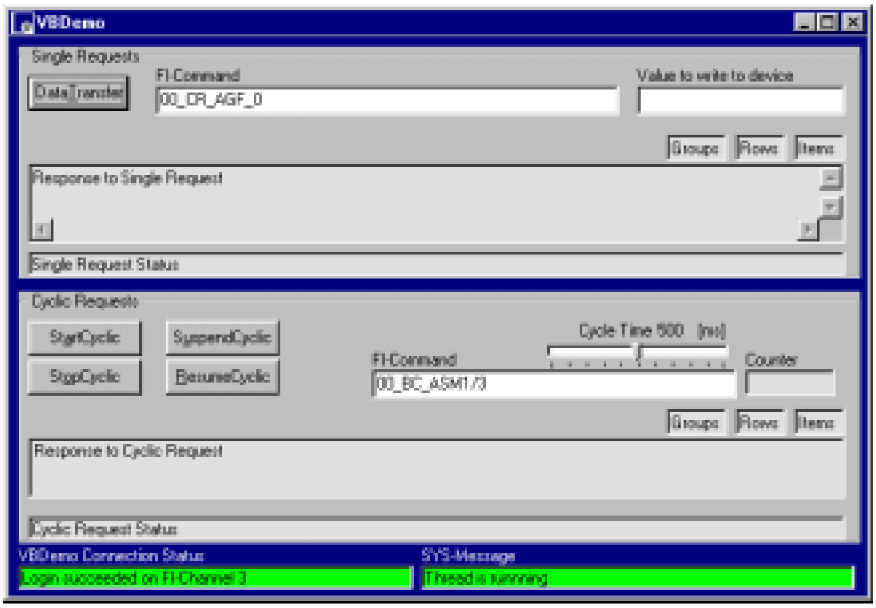
Issuing FI Commands using the “VBDemo” Application

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

To start the application, proceed as follows:

⇒ Click on start, point to programs, then to Rexroth Indramat and finally to FI.

⇒ Click on VBDemo.



VBDemo.bmp

Fig. 4-8: The “VBDemo” Client

“Single Requests” Dialog Box

This dialog box allows single requests to be issued that both read and write using the "DataTransfer" Routine (p. 4-10).

To do this, enter the FI command in the entry field “FI-Command”. 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 7, Function Interface Commands).

Then issue the FI command you have entered to the function interface by clicking on the <DataTransfer> button.

The answer from the function interface is displayed in the text box "Response to Single Request".

“Cyclic Requests” Dialog Box

This dialog box allows cyclic requests to be issued using the "StartCyclicPipe" Routine (S. 4-16).

To do this, enter the FI command in the entry field “FI-Command” (see chapter 7, Function Interface Commands).

Then issue the FI command entered cyclically to the function interface by clicking in the <StartCyclic> button.

The answer from the function interface is displayed in the text box "Response to Cyclic Request".

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. The "StartCyclicPipe" Routine (S. 4-19) is hereby processed.

Note:	Clicking on the <SuspendCyclic> button processes the "SuspendCyclicPipe" Routine (p. 4-20) and sets the cyclic request to standby mode. To reactivate the cyclic request, click on the <ResumeCyclic> button, whereby the "ResumeCyclicPipe" Routine (p. 4-21) is processed.

"VBDemo Connection Status" Dialog Box

Displays the connection status of the user at the function interface. There are two conditions:

- The dialog box is shaded green and shows the function interface channel (LOG channel) that has been assigned to the application.
- The dialog box is shaded red and shows the error code which has been caused on connection by the Login Routine "LogInIf" (p. 4-6).

"SYS-Messages" Dialog Box

This dialog box displays the function interface system messages to which the client “VBDemo” is to respond (see chapter 4.5, Access Functions for Working with SYS Messages).

Starting “Vdemo” in Diagnostics Mode

To start the “VBDemo” program in diagnostics mode, proceed as follows:

⇒ Open the Windows NT Explorer: To do this, click on Start, point to Programs and then click on the Windows NT Explorer.

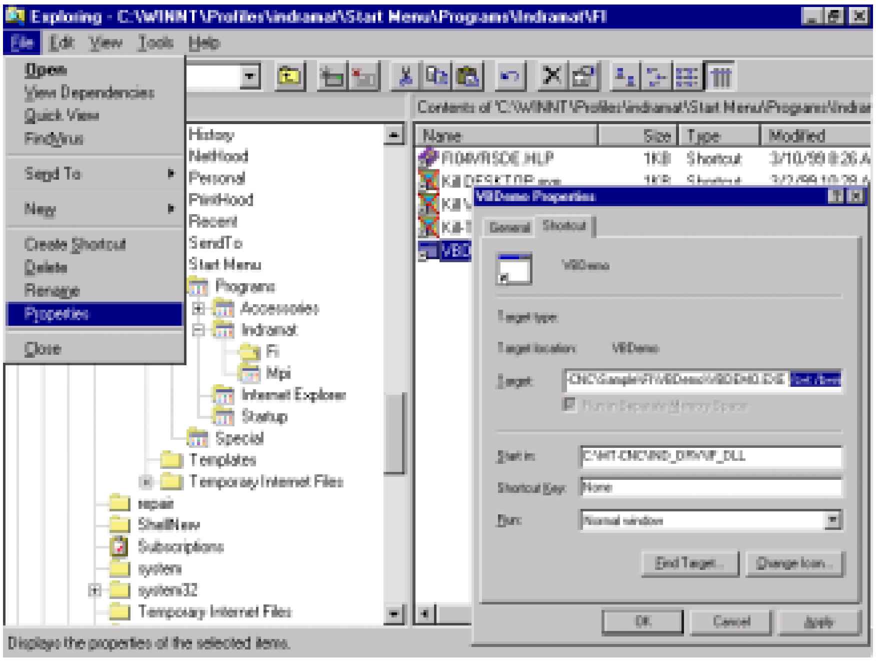
⇒ Via Winnt, go to Profiles into the User Profile by which the function interface was installed.

⇒ Click on the Start Menu, point to Programs, then to Rexroth Indramat and finally to FI.

⇒ Click on VBDemo and open the Properties dialog box via the Explorer menu file.

⇒ Click on the Link card and enter the start parameter "/c=t /b=w" in the „Target“ text field.

⇒ Click on the <Close> button and VBDemo will be started in diagnostics mode the next time it is called.



VBDemo_Explorer.bmp

Fig. 4-9: Start VBDemo in Diagnostics Mode of the Function Interface

Outputting Diagnostic Messages

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

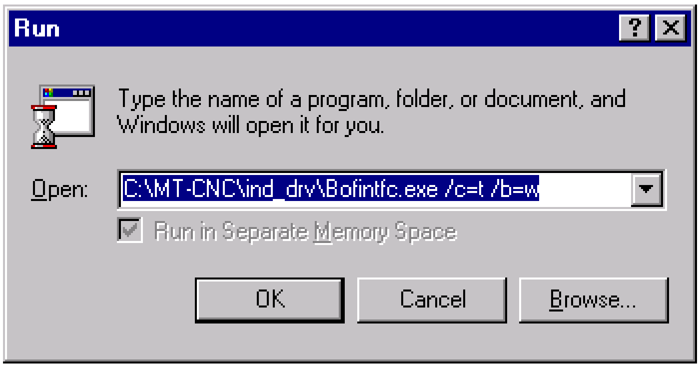
To start the function interface in diagnostics mode, proceed as follows:

⇒ Click on Start and then on the “Run” option.

Note: You can search for the BOF-process "BOFINTFC.EXE" by clicking on the „Find“ button. After a standard installation (see chapter 5, Installing Windows NT and the Function Interface) this file is located in the default directory "C:\MT-CNC\IND_DRV\".

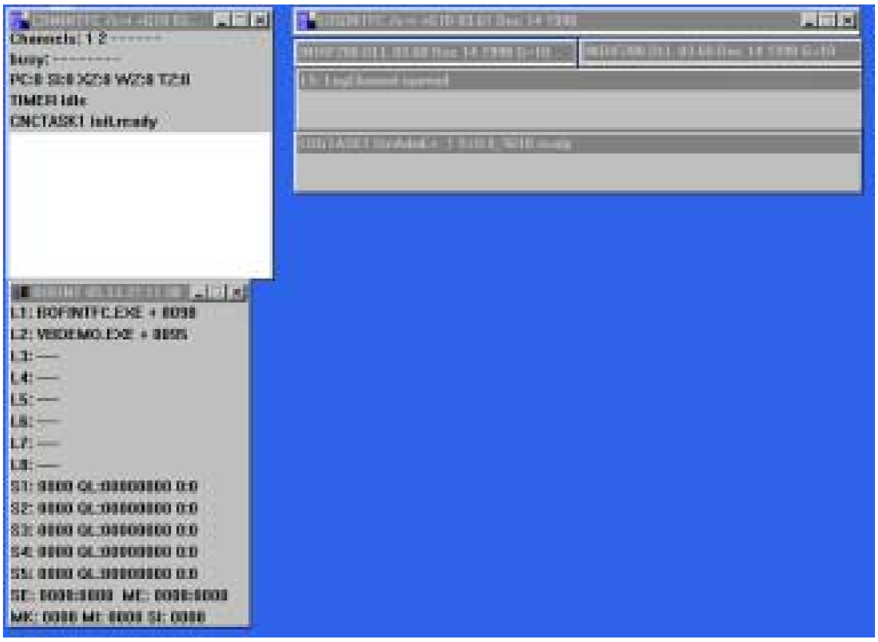
⇒ Enter the start parameter "/c=t /b=w" in lower case letters in the text box (observe space between entries).

⇒ Then click on the <OK> button.
The function interface is started in diagnostics mode.



AusführenDiagnose.bmp

Fig. 4-10: Windows NT Dialog Box “Run": BOFINTFC.EXE



Diagnose.bmp

Fig. 4-11: Diagnostics Mode of the Function Interface

Meaning of the Counters

5 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 → PC.

SI Number of communication errors that have occurred in the direction of transmission from PC → 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, no valid telegram could be transmitted 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, no valid telegram could be transmitted to the device.

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

Data accesses of 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 BOF process (BOFINT), those applications are shown that are known in the administration mechanism of the BOF process.

Windows NT Key Combinations

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

Action	Key Combination
Open start bar	<Ctrl> + <Esc>
Navigate within the opened start bar or within the submenus opened in the start bar	<Arrow key left, right>, or <Arrow key up, down>
Select (start) the applications in the opened submenus of the start bar	<Enter>
Start Windows NT Task Manager	<Ctrl> + <Shift> + <Esc>
Moving within the Windows NT menu	<Tab>
"Right mouse click" on button moved to	<Ctrl> + <F10>
Switching within a menu to other cards	<Ctrl>+<Tab>
Switching between opened applications	<Alt> + <Tab>

5 Installing Windows NT and the Function Interface

5.1 The Windows NT Operating System

The use of 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. Network cards and their drivers require considerable computing power which may then be lacking in the rest of the system. Hardware must therefore be selected with great care and utmost precision.

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

Multitasking and Windows NT

While under Windows 3.1x the co-called “cooperative” or "non-preemptive" multitasking method still controls several applications running concurrently, genuine “preemptive” multitasking is integrated into Windows NT.

Non-preemptive 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 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.

Preemptive Multitasking

The operating system decides itself how much computing time is to be allocated to the individual applications. Switching between applications is now much more fluid a process as the operating system is able to distribute computing time faster and in shorter intervals, creating the impression that several instructions really can work “simultaneously and completely” in parallel.

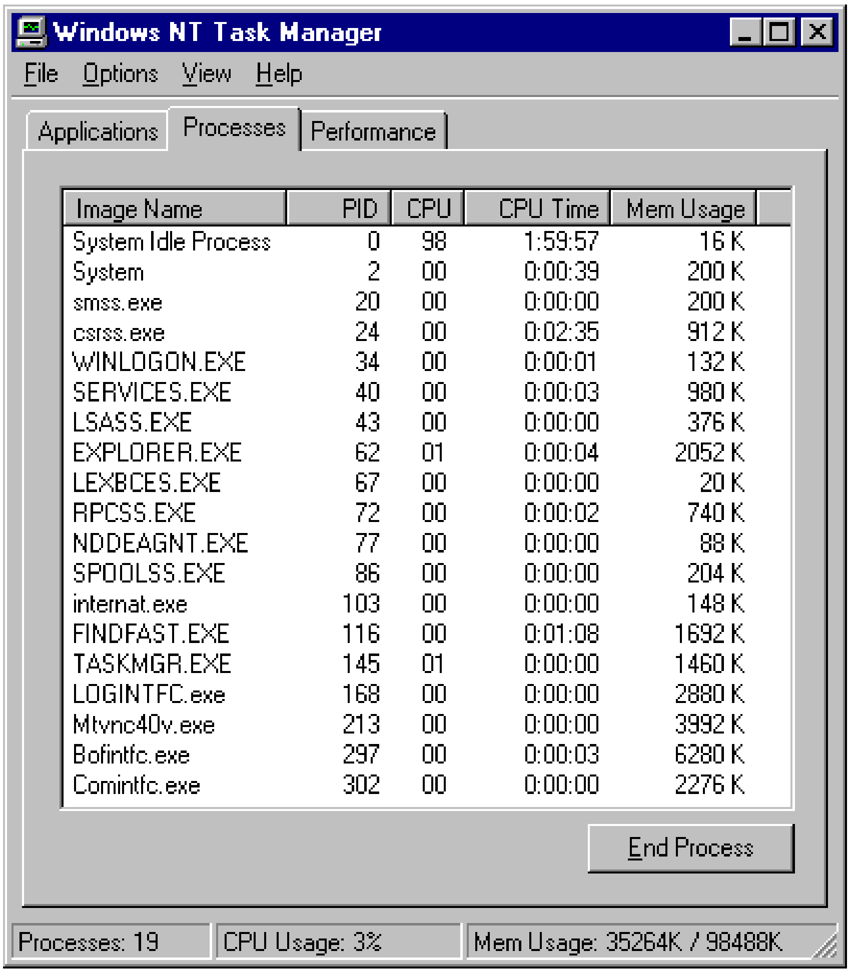
Windows NT Task Manager

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, other applications can also be started or the operating system can switch to other applications that are already open.

Calling the Task-Manager

⇒ using the key combination <Ctrl>+<Shift>+<Esc>

⇒ clicking with the right mouse button on the taskbar



Taskmanager.bmp

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 the Task Manager.

5.2 Setting the Windows NT System Properties

Performance

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

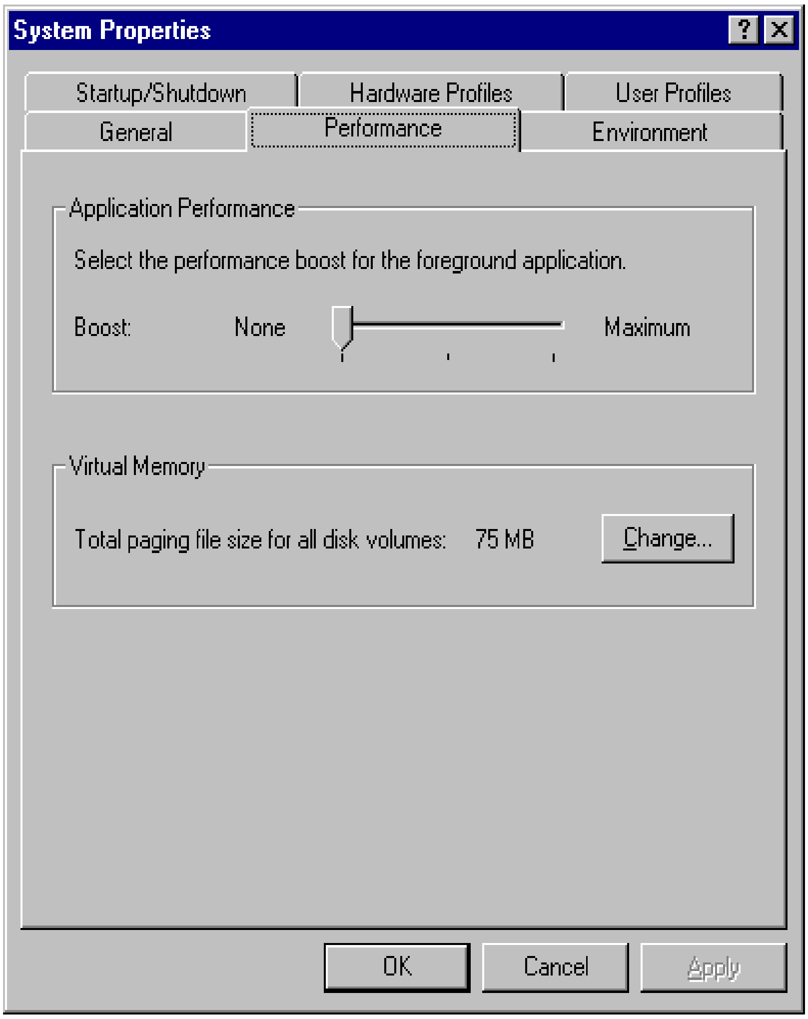
Note: Safe and error-free operation of the function interface is only ensured when the performance boost for the application in the foreground is set to "none".

To make this setting, proceed as follows:

⇒ Click on start, point to Settings, then to System Control and finally to System.

⇒ Click on the "System Properties" card and set the thumb to “none”.

⇒ Then click on the <OK> button.



Leistungsmerkmale.bmp

Fig. 5-2: Setting the Windows NT System Properties "Performance".

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

Date/Time Properties

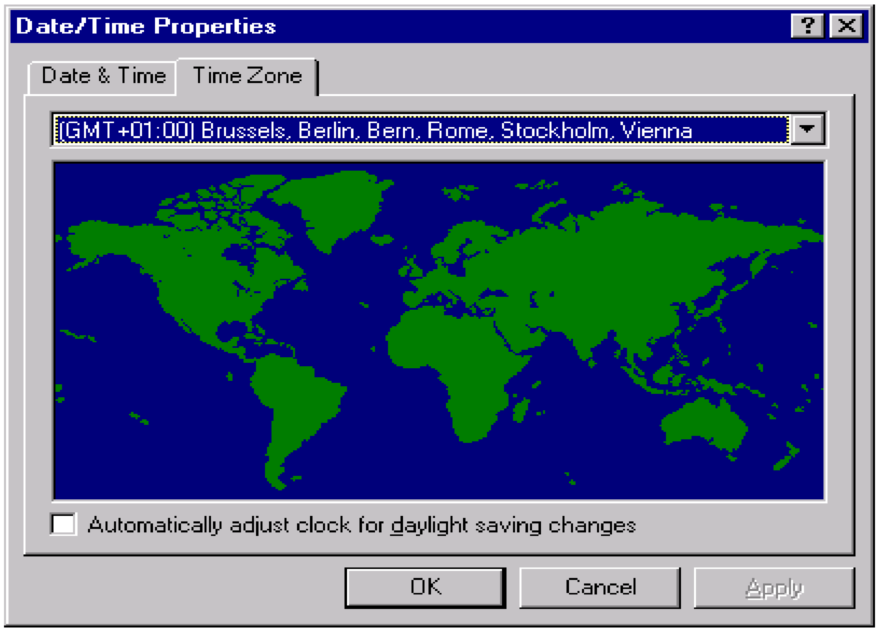
To exchange BOF/GBO files between two PCs, the time zone settings must be identical on both PCs. Furthermore, the automatic clock adjustment for daylight saving (switching between summer and winter times) must be deactivated.

To make this setting, proceed as follows:

⇒ Click on start, point to Settings, then to System Control and finally to Date/Time Properties.

⇒ Click on the Time Zone card and deactivate the toggle button “Automatically adjust clock for daylight saving changes”.

⇒ Then click on the <OK> button.



Zeitzone.bmp

Fig. 5-3: Date/Time Properties

Environment

To use the specific functions of the Rexroth Indramat Function Interface, the individual applications must be informed regarding the directory structure of the DLLs. In the "Environment" card, the installation program adds the path entries [LW]:\...\IND_DLL and C:\IND_BASE in the "User variables for ..." text fields, whereby [LW] stands for the installation drive.

Note: Entries for the user variables for the path are carried out during the standard installation (see chapter 0 Fig. 5-8: System Control: Device "MTCNC00I"

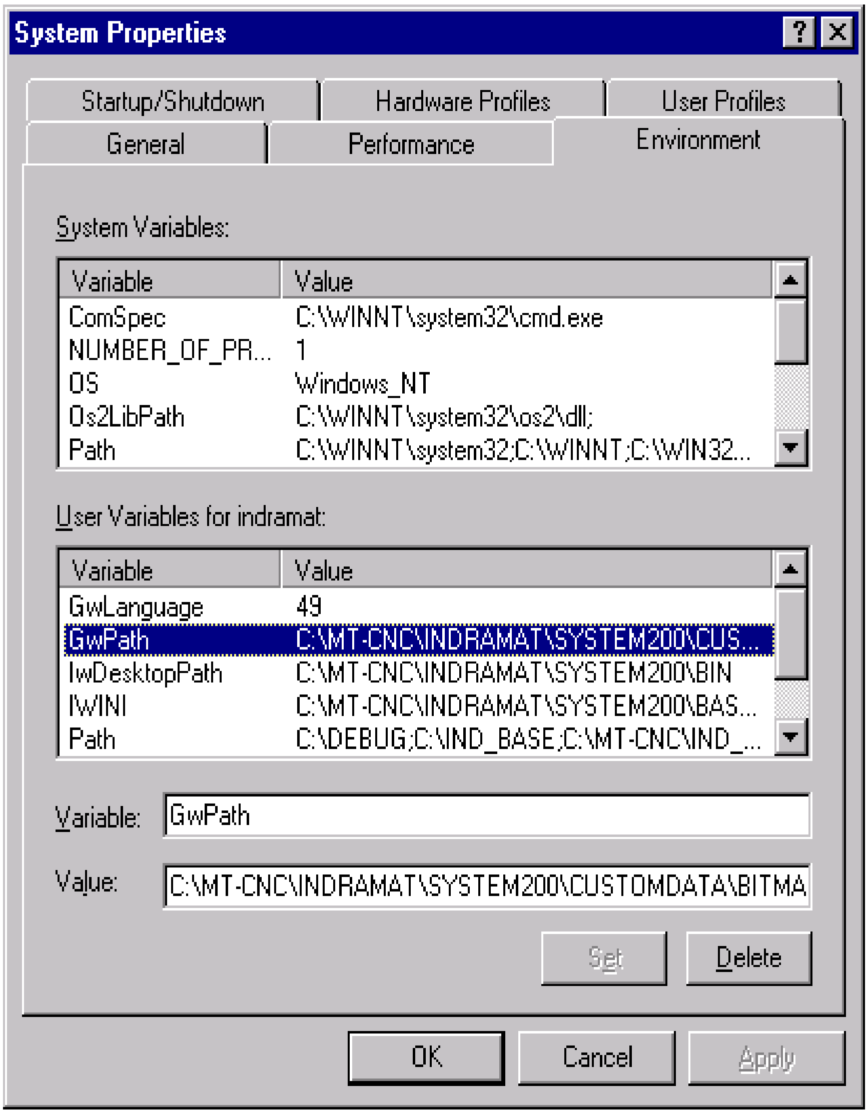
) , i.e. "C:\IND_BASE" and "C:\MT-CNC\IND_DLL" are added.

To check these entries, proceed as follows:

⇒ Click on start, point to Settings, then to System Control and finally to System.

⇒ Click on the "Environment " card and then on Path.

⇒ Then click on the <Cancel> button.



Umgebung.bmp

Fig. 5-4: Setting the Windows NT System Properties "Environment".

Rexroth Indramat

Note: The other entries can vary from the entries in your system or user profile.

Swapping the Core-Mode Driver

During intensive use of memory, the Windows NT Operating System swaps parts of the core-mode driver to the page file. This can have a dramatically negative effect on the performance of the entire system. Windows NT allows you to prevent the swapping of core-mode drivers.

Note: This setting is carried out by the installation (setup) program of the function interface.

You can make manual changes by calling up the registry editor "REGEDT32" .

⇒ To do this, click on Start and then on the “Run” option.

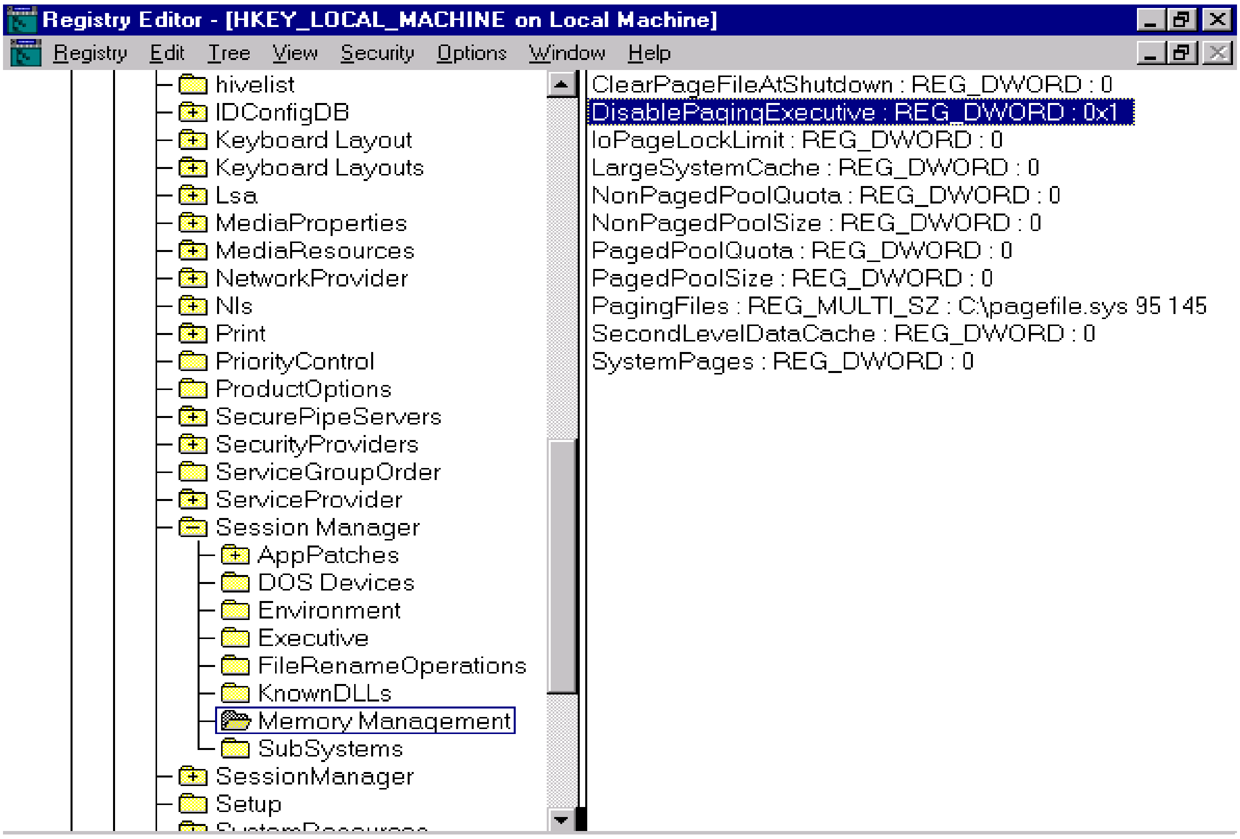
⇒ Enter "REGEDT32.EXE" in the text field.

⇒ Then click on the <OK> button and the registry editor starts.

The settings are changed by the following key:

HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session-Manager\MemoryManagement

Here, the value of the "DisablePagingExecutive" variable is set by the installation program to "1". This change keeps the system code in the RAM and prevents it from being swapped During installation of the Windows NT Operating System, the value of this variable was set to "0", whereby temporary swapping is allowed.



Regedt32.bmp

Fig. 5-5: Registry Editor: "DisablePagingExecutive" Variable

Note: Error-free operation of the function interface is only assured when the "DisablePagingExecutive" parameter is set to the value "1".

Idling Activity

To ensure a more efficient use of CPU resources when a program in the foreground is idle (DOS box), the idling activity of the "_default.pif" file should be set to "High". This setting means that a background program receives less CPU resources, i.e. when the DOS box is idle, as much as possible of the CPU power is passed on to other active applications.

To make this setting, proceed as follows:

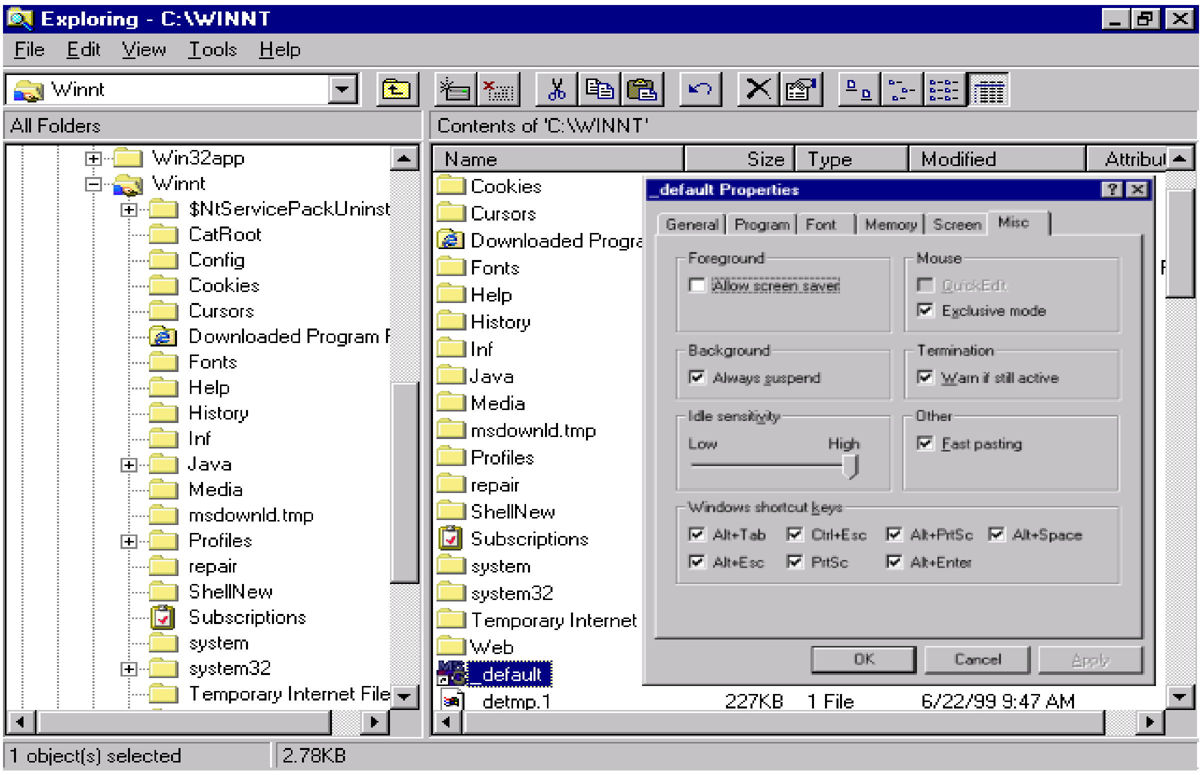
⇒ Open the Windows NT Explorer: To do this, click on Start, point to Programs and then click on the Windows NT Explorer.

⇒ Enter the Winnt file

⇒ Click on the "_default.pif" file and open the Properties dialog box via the Explorer menu file.

⇒ Click on the "Other" card and set the thumb for idling time to “High”.

⇒ Then click on the <OK> button.



Leerlaufaktivität_Explorer.bmp

Fig. 5-6: Properties of the "_default.pif" File: Idling Activity

Note: The other settings can vary from the entries in your system or user profile.

5.3 Installing Windows NT Core-Mode Driver

The Windows NT Core-Mode Driver "MTCNC00I.SYS" establishes the connection between the function interface and a serial interface or dual port RAM (see chapter 3.1, The Structure of the Function Interface).

Note: Installation of this driver is carried out by the installation program of the function interface.

You can also install this manually by calling up the registry editor "REGINI" via the Windows NT prompt.

⇒ To do this, click on Start and point to Programs.

⇒ Click on the prompt.

⇒ Enter "cd winnt\system32\drivers" to switch to the driver directory (see chapter Contents of the Driver Directory "[LW]:\Winnt\System32\Drivers\", page 5-23).

⇒ Confirm your entry by pressing the <Return> key.

⇒ Enter "regini mtcnc00i.ini" to start the registry editor and to install the Windows NT Core-Mode Driver.

⇒ Confirm your entry by pressing the <Return> key.

⇒ Enter "Exit" and then confirm your entry with the <Return> key to close the prompt.

Eingabeaufforderung.bmp

Fig. 5-7: Prompt: Installing the Core-Mode Driver

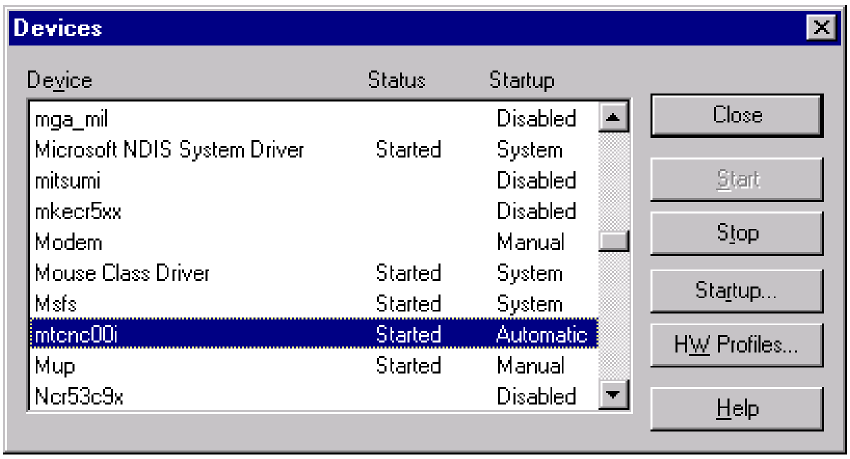
Note: After installing the Windows NT Core-Mode Driver, you must restart Windows NT or your computer.

To check whether the installation has been carried out successfully, proceed as follows:

⇒ Click on Start, point to Settings and then to System Control.

⇒ Then click on Devices.

After successful installation, the Windows NT Core-Mode Driver "MTCNC00I.SYS" is listed as device "mtcnc00i" with the status "Started" and start mode "Automatic".



System_Geräte.bmp

Fig. 5-8: System Control: Device "MTCNC00I"

5.4 Installing the Function Interface

ATTENTION

It is the responsibility of the user to store all user data before proceeding with the installation. Furthermore, make sure that only one version of the function interface is ever installed on the target system at any given time.

Installing from CD-ROM

⇒ To install the function interface, place the CD-ROM labeled:

- SWD-FUN*PC-RUN-05VRS-MS-CD650

in the appropriate drive (e.g. drive D) of your PC and the installation is then automatically started.

Installing from Diskettes

⇒ When installing from diskettes, place the 1st diskette labeled:

- SWD-FUN*PC-PRO-05VRS-MS-C1,44

in the appropriate drive (e.g. Drive A) of your PC.

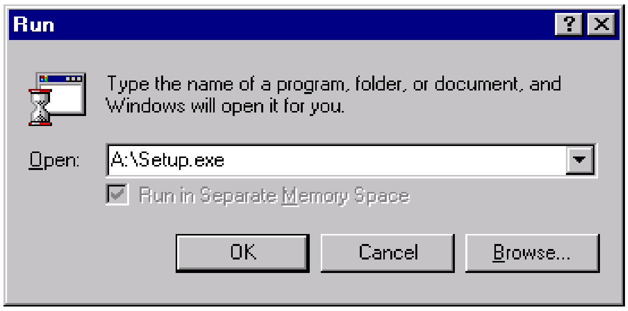
Note: In the following description, we assume that Drive A is being used for an installation from diskette.

To start the function interface, proceed as follows:

⇒ Click on Start and then on the “Run” option.

⇒ Enter in the text field of the dialog box "A:\Setup.exe".

⇒ Then click on the <OK>button and the installation of the function interface is commenced.



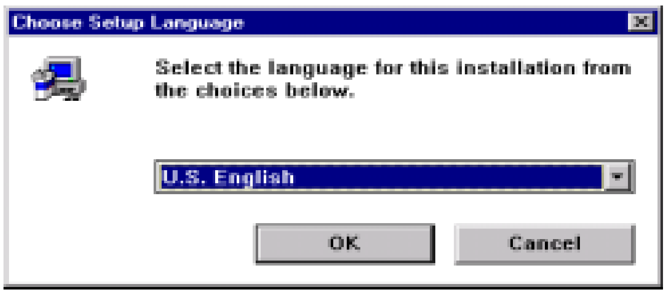
AusführenSetup.bmp

Fig. 5-9: Windows NT Dialog Box “Run": Diskette Installation

You can select the language for the InstallShield® assistant in the first dialog box.

⇒ To do this, click on the arrow to the right next to the standard setting "U.S. English" and select the desired language for the installation program by clicking on it.

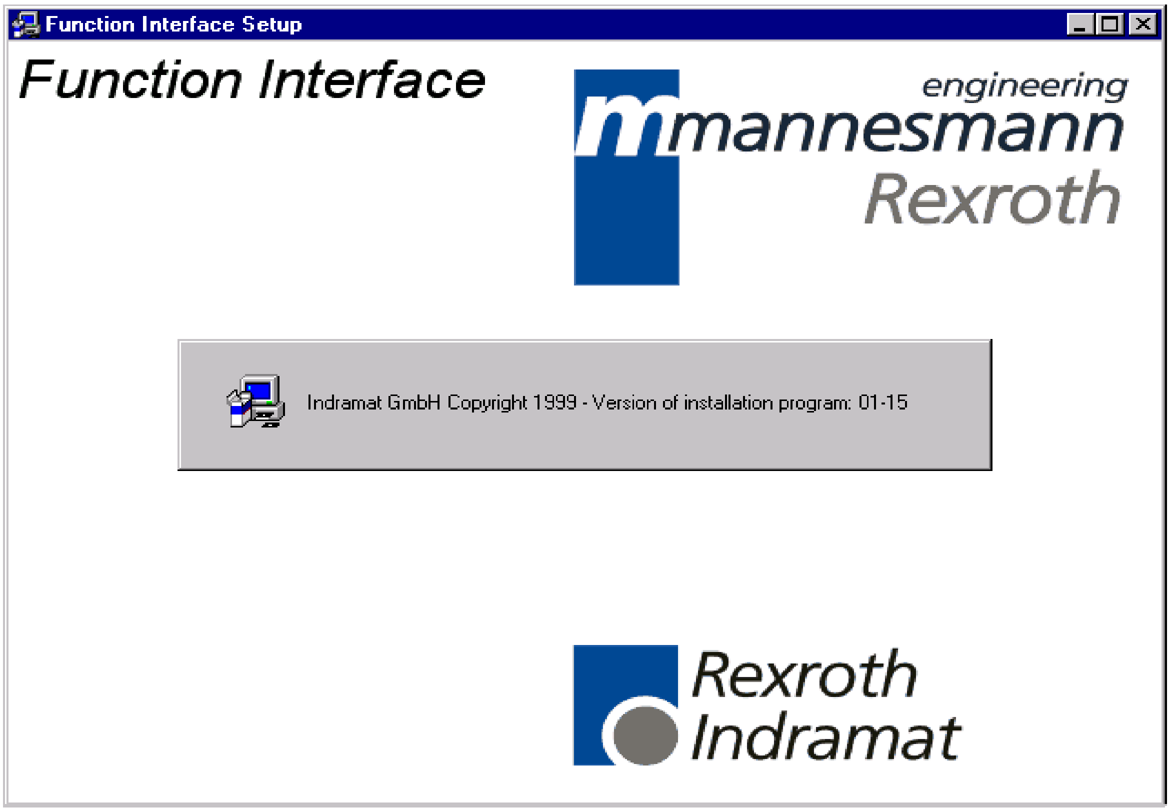
⇒ Then click on the <OK> button.



Sprachauswahl.bmp

Fig. 5-10: Selecting the Language of the InstallShield® Assistant

After selecting the language, the main screen of the installation program appears on the screen. This screen contains copyright information and the version of the installation program for approx. three seconds.



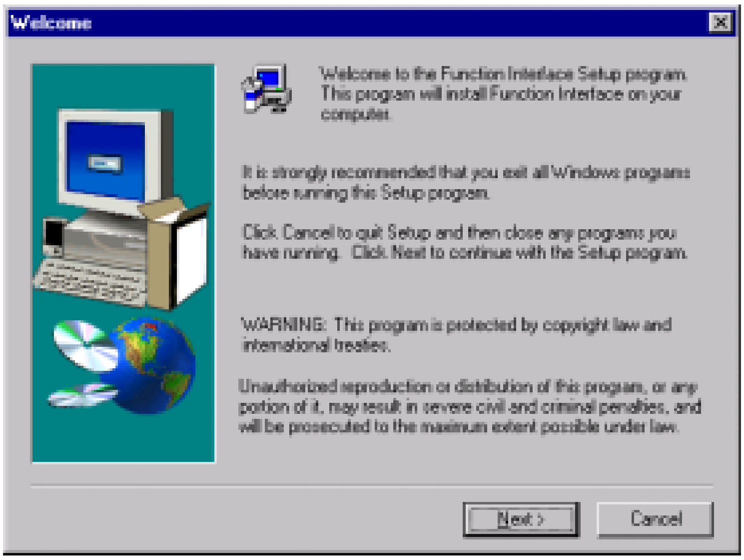
Copyright.bmp

Fig. 5-11: Copyright Information and Version of the Installation Program

A window then appears welcoming you to the function interface setup program.

Note: Please observe the instructions that you are given here. The installation program is cancelled by clicking on the <Cancel>-button.

⇒ Click on the <Next> button to proceed with the installation program.



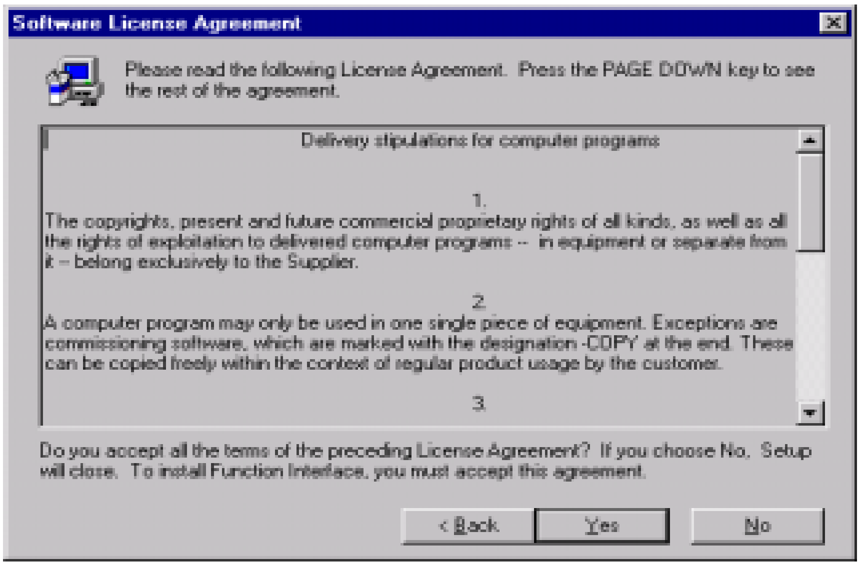
Willkommen.bmp

Fig. 5-12: Welcome Screen of the Installation Program

The software license agreement is shown in the next window. You can display the entire license agreement by pressing the Page Down key (cursor key).

Note: You must accept this license agreement to be able to install the function interface. Clicking on the <No> button cancels the installation. You can return to the previous window by clicking on the <Back> key.

⇒ If you agree to all terms contained within the software license agreement then click on the <Yes> button.



Lizenzvertrag.bmp

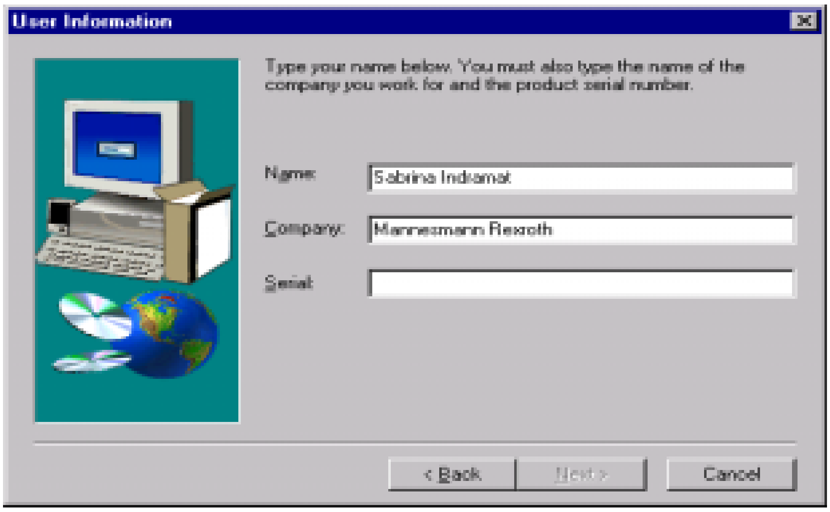
Fig. 5-13: Terms of the Software License Agreement

In the following window, you will be prompted to enter certain user information.

⇒ Enter your name, the name of your company and the serial number of the user in the corresponding text fields.

Note: The serial number is printed on the label of the installation diskette or on the CD-ROM. Clicking on the <Cancel> button cancels the installation. You can return to the previous window by clicking on the <Back> key.

⇒ Then click on the <Next> button to proceed with the installation program.



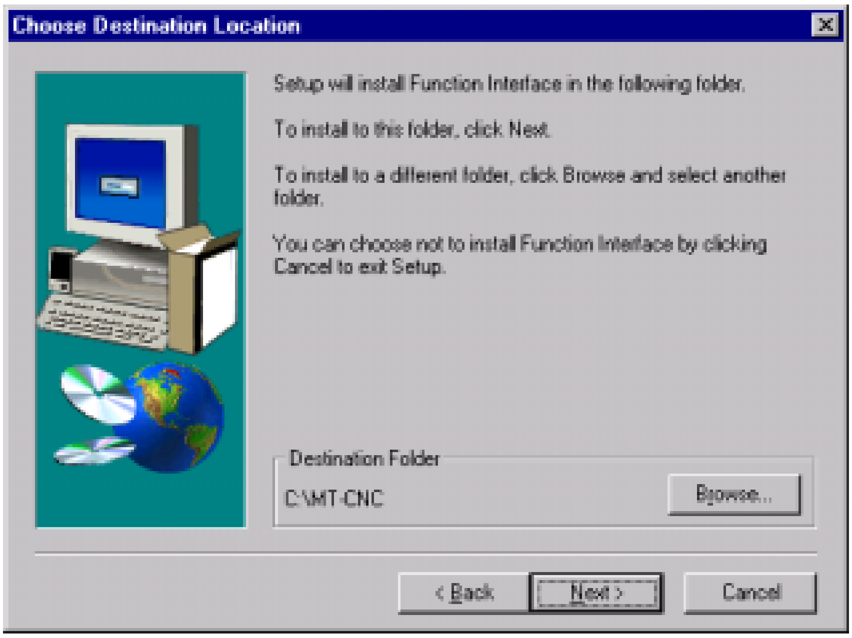
Benutzerinfo.bmp

Fig. 5-14: Entering User Information

You can select the target path for the installation in the window that now pops up. During a standard installation, the target path "C:\MT-CNC" is suggested.

Note:	We recommend that you accept this name, either on the default drive or on another. **Please make sure that only one version of the function interface is ever installed on the target system at any given time.** To change the standard path, click on the <Browse> button. Then select the corresponding path and the directory.

⇒ Clicking on the <Next> button continues with the installation.



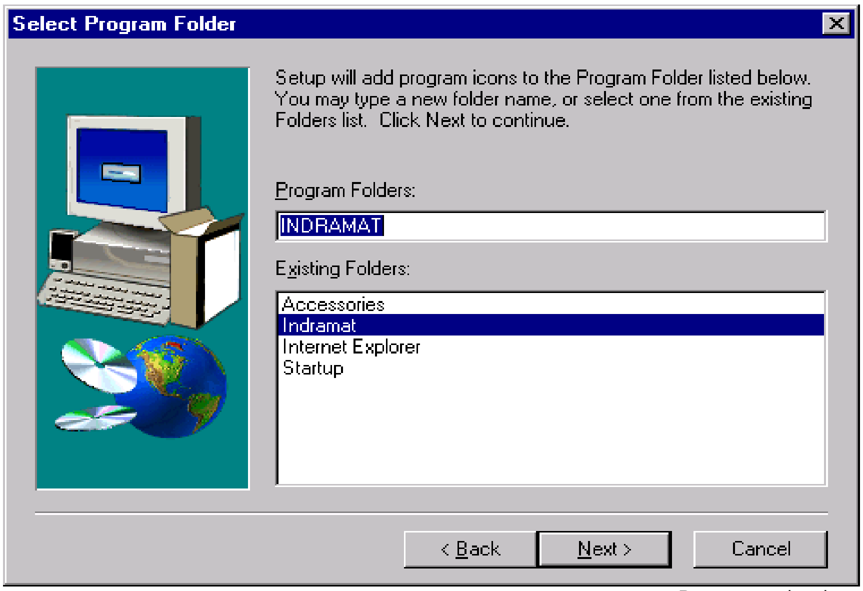
Zielpfad.bmp

Fig. 5-15: Selecting the Target Path for the Installation

In the next installation step, new symbols are added to the existing program folders. You also have the possibility of creating new program folders at this stage.

⇒ To add symbols to a program folder that already exists, click on the folder; otherwise, enter the name for the new program folder in the "Program Folders:" text field.

⇒ Clicking on the <Next> button continues with the installation.



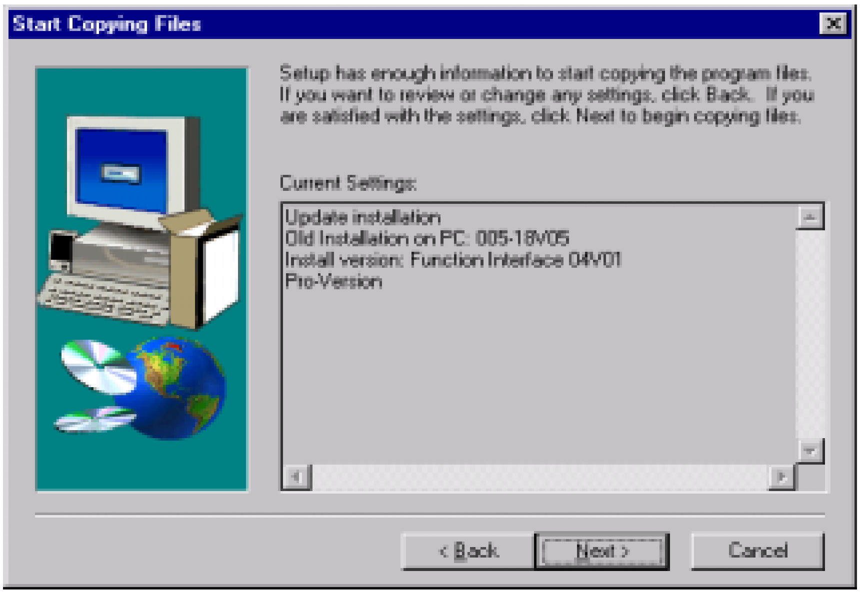
Programmordner.bmp

Fig. 5-16: Selecting the Program Folder

The next installation step shows you the settings that you have made for the installation. If you want to change the current settings, click on the <Back>-button to return to the previous installation step.

⇒ Clicking on the <Cancel> button cancels the installation.

⇒ If you want to keep your settings and start the copying procedure then click on the <Next> button.

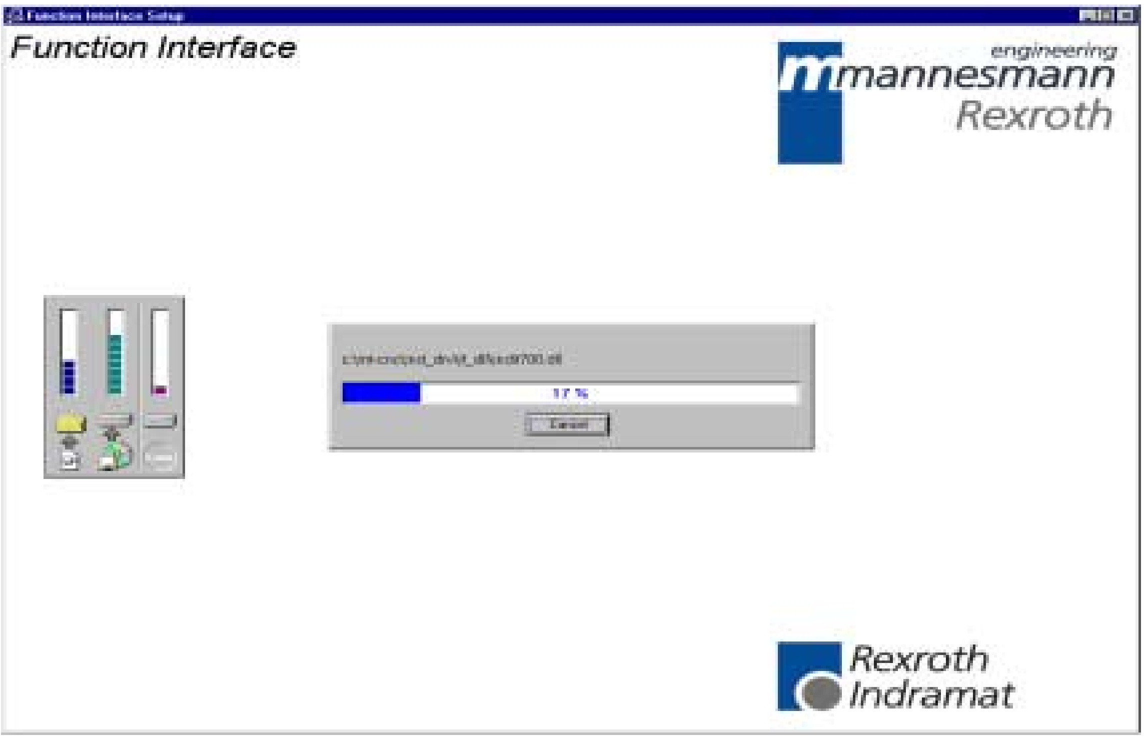


Kopiervorgang.bmp

Fig. 5-17: Confirmation Window to Start Copying Files

The following figure shows the progress of the copying procedure, which may last several minutes. Three bars showing the installation status of the currently copied function interface components as well as the copying status of the installation medium (Diskette, CD-ROM) and the amount of remaining free space on the hard drive are displayed to the left of the screen. If you do not have sufficient free capacity on your hard drive for the installation then the installation is cancelled and a corresponding message is displayed.

The status window in the middle of the screen represents the progress of the entire installation in the form of a bar graph.



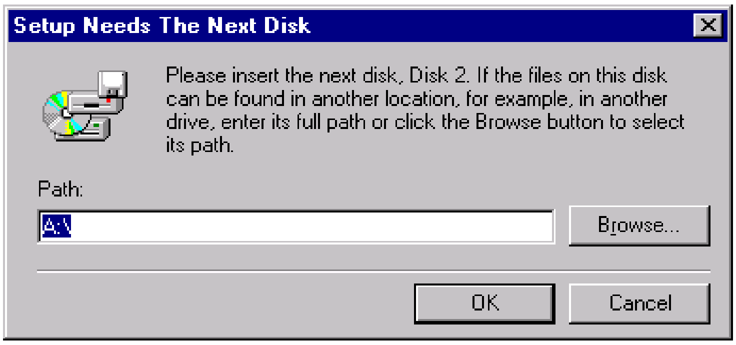
Statusinfos.bmp

Fig. 5-18: Status Information on the Copying Procedure

Note: Clicking on the <Cancel> button cancels the installation.

If you are installing from diskette, you will be prompted to insert additional disks.

⇒ Insert the corresponding diskette in the drive and then click on the <OK> button to proceed with the installation.



Diskette2.bmp

Fig. 5-19: Inserting the next Diskette

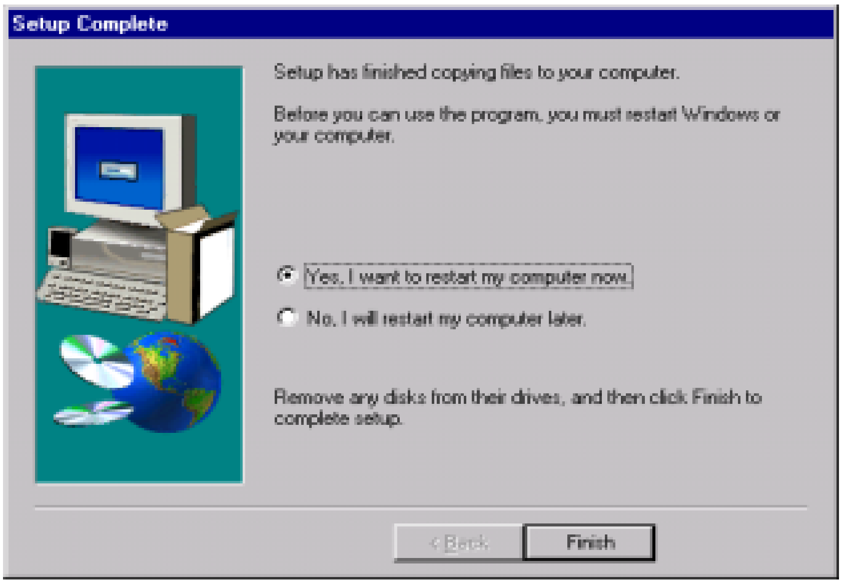
After the copying procedure has been completed, you can then place icons for installed applications on the desktop.

⇒ Click on the <Yes> button if icons are to be installed on the desktop.

Frage.bmp

Fig. 5-20: Installing Application Icons on the Desktop

You are now informed that the installation has been successfully completed and that you should re-start your computer.



Neustart.bmp

Fig. 5-21: Setup Complete: Restart Computer

Note: Before you are able to start working with the function interface, you must re-start your computer or Windows NT.

5.5 Directory and File Structure of the Function Interface

Contents of the "INDRAMAT.INI" File

The global settings for the function interface are stored in this file. The function interface searches for the "INDRAMAT.INI" file in the "C:\IND_BASE" directory. The default setting points to the path [LW]:\MT-CNC\IND_DRV, whereby the name of the subdirectory \IND_DRV is fixed.

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
[IfConfig]	This contains the configuration settings for the function interface	
IfInstDir=	e.g.: C:\MT-CNC\IND_DRV	Directory in which the three basic processes of the function interface are installed. This entry is set by the installation program.
AndInstDir=	e.g.: C:\Andron	!Optional! Directory for Andron control software. Details refer to the "ANDRON.EXE" application.
IfDllMode=	e.g.: 04.10 03.xx [00..70], 04.xx [00,10]	Here the mode is specified that is to be supported by the function interface. The IfDllMode of a more recent version of the function interface can, e.g. be operated in the same mode as the previous version for troubleshooting.
IfVersion=	e.g.: 04V01	Current version of the function interface.
GBOVERSION=	e.g.: 005-18V05	Current version of the Rexroth Indramat GUI (BOF/GBO).
INDRAMAT_x=	x=1....9 Name of file	Reference to directory C:\IND_BASE. The existence of the files named here is checked on starting the function interface. The following applies: A file name without an extension is a DLL. E.g. Indramat = INDRAMT.DLL Several file identifiers are separated by a "comma".
Identifier	**Values**	**Explanation**
IND_DLL_x=	x=1...9 Name of file	Reference to directory [LW]:\....\IND_DLL. The existence of the files named here is checked on starting the function interface. The following applies: A file name without an extension is a DLL. E.g. NDFS100 = INDFS100.DLL. Several file identifiers are separated by a "comma".
IF_DLL_x=	x=1...9 Name of file	Reference to directory LW]:\....\IND_DRV\IF_DLL. The existence of the files named here is checked on starting the function interface. The following applies: A file name 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=	e.g.: 01V05	Version ID of the System200 software component WIN-HMI
TYP=	e.g.: HMI	System200 software component WIN-HMI
ServicePack=	e.g.: 2 [1,2,...]	Service Pack ID of the installed System200 software components
SP_Release=	[1,...,F]	State of release ID of the Service Pack (F = Final Version)

Example Entries in the "INDRAMT.INI" File

```
[IfConfig]
IfInstDir=C:\MT-CNC\IND_DRV
AndInstDir=C:\Andron
IfDllMode=04.10
IFVERSION=04V02
GBOVERSION=005-18V05
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=01V05
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

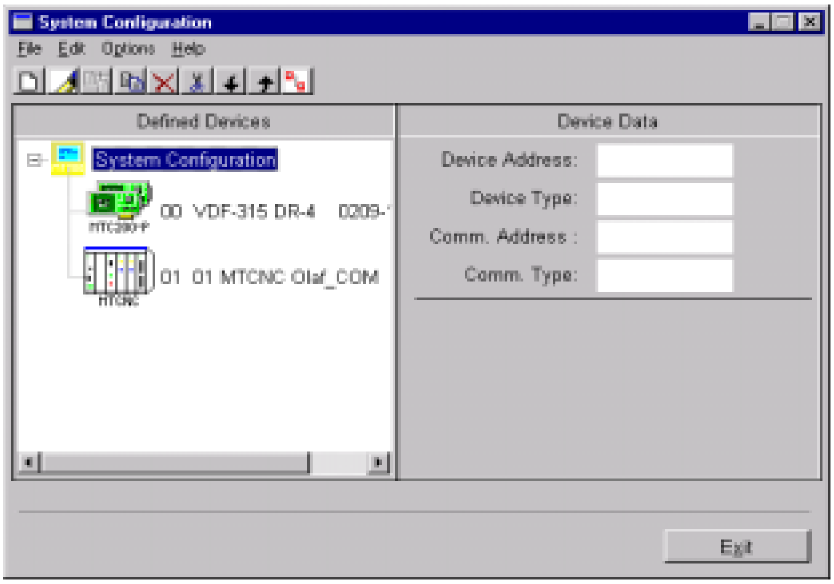
The configuration of the individual communication addresses and the settings of the various Rexroth Indramat devices are set in this file. The "IND_DEV.INI" file is edited by the system configurator and is located in the "[LW]\...\CONFIG\" directory.

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 are used in doing 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. The parameter records of the real device can hereby 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 function key F1 while the program is running.



Systemkonfigurator.bmp

Fig. 5-22: Rexroth Indramat System Configurator

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

Identifier	Values	Explanation
[CommAddrX]	X = 1...8	Assignment of the communication channel (thread) of the function interface.
CommStr=	V24, Port [COM1...4], baud rate, parity, type of interface, packet counter	Communication via RS232 serial interface, e.g. V24,COM1,19200,NONE,RS232,TCON Communication via RS232 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,$2000
	- or - SHM, Channel No. [1...15]	Communication channel to the MTVNC via a shared memory, e.g. SHM, 1
Timeout=	>= 1000 [msec] Preset = 3500 [msec]	! OPTIONAL ! Time in which an answer must be received from the device.
- or – only for dual port RAM (e.g. for MTC200-P)		
CommStr=	DPR, address [$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 an answer must be received from the device.
[DeviceAddrX]	X = 0..0,15	Device address
Componenttype1=	e.g. MTS-P01.2 NONE, MPU, MTS-P, MTS-P01.2, MTS-P02.2, MTS-R-M1, PPC-R	Name of the SPS component type
Componenttype1=	e.g. MTC-P NONE, MPU, MTC-P, MTC-R, PPC-R	Name of the CNC component type
DeviceName=	max. 32 ASCII characters	Device name; e.g.. Processing Center 12T34
DeviceType=	e.g. MTC200-P MTVNC MTCNC MTC200-P MTC200-R ISP200-P ISP200-R TRA200-R ECODRIVE03 MTA200-P SERCANS-A SERCANS-P SYNAX-P SYNAX-R	Device type: MTC with SPS PC variant Virtual MTC MT-CNC 02 / 03 MTC with SPS PC variant MTC with SPS RECO variant Standalone SPS PC variant Standalone SPS RECO variant TRANS200 RECO variant Ecodrive03 ANDRON control SERCANS-A card (via serial interface) SERCANS-P card (via serial interface) SYNAX PC variant SYNAX RECO variant
DeviceAssign=	0...15, NO	Assignment of a simulation pair. The MTVNC is e.g. hereby assigned to a real MTCNC.
DeviceStatus=	ON, OFF	Assignment of whether or not the device is incorporated into the administration structure of the function interface.
MtvncMode=	OFF, RUN, STANDBY	! Only in case of virtual MTC (MTVNC) ! Condition of the MTVNC during inactive use.
MtvncMemory=	256, 257 .. 16383 [KB] Preset = 512 [KB]	! Only in case of virtual MTC (MTVNC) ! Size of the PC memory used by the MTVNC.
CommAddr=	1...8	Assignment of the communication address. Corresponds to the parameter [CommAddr1...8].

Identifier	Values	Explanation
PLC=	YES, NO	SPS support for the device. E.g. one MTVNC, TRANS200-R has no SPS, 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] CommStr=DPR,$D000,$0000,$2000,RAM0,TCON PortAddr=$31C PortVal=$28	Communication address 1 Settings for the communication via dual port RAM Port address of the MTC/MTS card Physical memory address of the MTC/MTS card.
[CommAddr2] CommStr=DMA,$D000,$0000,$2000	Communication address 2 Assignment of the DMA channel.
[CommAddr3] CommStr=V24,COM1,19200,NONE,RS232,TCON	Communication address 3 Settings for communication via RS232.
[CommAddr4] CommStr=DPR,$D200,$0000,$2000,RAM0,TCON PortAddr=$318 PortVal=$29	Communication address 4 Settings for the communication via dual port RAM Port address of the MTC/MTS card Physical memory address of the MTC/MTS card.
[CommAddr5] CommStr=SHM,1	Communication address 5 Settings for communication via shared memory.
[DeviceAddr0] CommAddr=1 Componenttype1= MTS-P Componenttype2= MTC-P DeviceAssign=NO DeviceName=VDF-315 DR-4 0209-15 DeviceStatus=ON DeviceType=MTC200-P PLC=YES	Device address 00 Assigned communication channel SPS components MTS-P CNC components MTC-P No MTVNC assigned Device name Device is available and ready for operation Device type SPS support
[DeviceAddr1] CommAddr=3 Componenttype1= MPU Componenttype2= MPU DeviceAssign=3 DeviceName= processing center 12T34 DeviceStatus=OFF DeviceType=MTCNC PLC=YES	Device address 01 Assigned communication channel SPS components MPU CNC components MPU Device address 03 (MTVNC) assigned (simulation pair) Device name Device is not ready for operation Device type SPS support
[DeviceAddr2] CommAddr=4 Componenttype1= MTS-P02.02 Componenttype2= MTC-P DeviceAssign=NO DeviceName= processing time 12T35 DeviceStatus=ON DeviceType=MTC200-P PLC=YES	Device address 02 Assigned communication channel SPS components MTS-P CNC components MTC-P No MTVNC assigned Device name Device is available and ready for operation Device type SPS support

Entry	Explanation
[DeviceAddr3] CommAddr=5 Componenttype1= NONE Componenttype2= NONE DeviceAssign=1 DeviceName= V processing time 12T34 DeviceStatus=ON DeviceTyp=MTVNC MtvncMemory=512 MtvncMode=RUN PLC=NO	Device address 03 Assigned communication channel SPS component not available CNC component not available Assigned to device address 01 (simulation pair) Device name Device is available and ready for operation Device type Size of the PC memory Condition during inactive use No SPS support
[DeviceAddr4] CommAddr=3 Componenttype1= NONE Componenttype2= PPC-R DeviceAssign=NO DeviceName= TRANS200 DeviceStatus=ON DeviceType=TRANS200-R PLC=NO	Device address 04 Assigned communication channel SPS component not available CNC component PPC-R No MTVNC assigned Device name Device is ready for operation Device type No SPS support
[DeviceOrder] Order=3,0,1,2,4	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 System Directory "[LW]:\Winnt\System32\"

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 Driver Directory "[LW]:\Winnt\System32\Drivers\"

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 "C:\IND_BASE\" Directory

Rexroth Indramat files and function libraries that are required for the start and initialization phase are stored in the "C:\IND_BASE\" directory.

File	Explanation
INDRAMAT.INI	File with global function interface settings
INDRAMAT.DLL	Access to global settings (GetInstPath, etc.)

Contents of the "[LW]:\...\DOCUMENTATION\" Directory

The drive as well as the path "[LW]:\...\" are preset during the standard installation routine to "C:\MT-CNC\". The following Windows 95/NT help files of the printed English and German manuals are stored in the "C:\MT-CNC\DOCUMENTATION\" directory:

File	Explanation
FI04VRSDE.HLP	Windows 95/NT help file in German
FI04VRSDE.CNT	Definition file of the Windows 95/NT help file
FI04VRSEN.HLP	Windows 95/NT help file in English (in preparation)
FI04VRSEN.CNT	Definition file of the Windows 95/NT help file

Contents of the "[LW]:\...\SAMPLE\FI\VBDEMO\" Directory

The drive as well as the path "[LW]:\...\" are preset during the standard installation routine to "C:\MT-CNC\". The files from the example program for connecting a Visual Basic application to the function interface are located in the "C:\MT-CNC\SAMPLE\FI\VBDEMO\" directory.

File	Explanation
VBDEMO.EXE	Example program
The following Visual Basic source files of the "VBDemo.exe" application are delivered with the planning software: SWD-FUN*PC-PRO-05VRS-MS-C1,44 (PRO-Version)	
VBDEMO.VBP	Visual Basic project file
VBDEMO.VBW	Visual Basic Project work area file
VBDEMO.FRX	Binary form file
VBDEMO.FRM)	Form file
INDIF000.BAS	4
SYSTHREAD.BAS)	Basic module for the reaction to SYS messages
WIN32_API.BAS	Basic module for Windows API functions/subroutines

Contents of the "[LW]:\...\SAMPLE\FI\VCDEMO\" Directory

The drive as well as the path "[LW]:\...\" are preset during the standard installation routine to "C:\MT-CNC\". The files for connecting a Visual C/C++ application to the function interface are located in the "C:\MT-CNC\SAMPLE\FI\VCDEMO\" directory.

File	Explanation
The following source files are delivered with the planning software. **SWD-FUN*PC-PRO-05VRS-MS-C1,44 (PRO-Version)**	
INDIF000.C	Load procedure for the DLL functions.
INDRAMAT.C	Load procedure for the global DLL functions.
INDIF000.H	Function definition of the individual routines.
INDIFX00.H	Function definition of the individual routines.
INDRAMAT.H	Global function interface routines.

Contents of the "[LW]:\...\IND_DLL\" Directory

The drive as well as the path "[LW]:\...\" are preset during the standard installation routine to "C:\MT-CNC\". The following Rexroth Indramat function libraries are contained in the "C:\MT-CNC\IND_DLL\" directory:

File	Explanation
INDFS100.DLL	Processing the file ID
INDMA900.DLL	Processing the MAP file
INDMA110.DLL	Connecting the MAP file
INDUT140.DLL	Using various system utilities
INDOF160.DLL	Using various system utilities

Contents of the "[LW]:\...\IND_DRV\" Directory

The drive as well as the path "[LW]:\...\" are preset during the standard installation routine to "C:\MT-CNC\". The following files are contained in the "C:\MT-CNC\IND_DRV\" directory:

File	Explanation
MISX.DAT	Definition file for the MISX device group
MTCX.DAT	Definition file for the MTCX device group
MSCX.DAT	Definition file for the MSCX device group
MPCX.DAT	Definition file for the MPCX device group
MTAX.DAT	Definition file for the MTAX device group
MECX.DAT	Definition file for the MECX device group
MTRX.DAT	Definition file for the MTRX device group
MSYX.DAT	Definition file for the MSYX device group
VERSION.DAT	Definition file for the version ID
COMINTFC.EXE	Communication process
BOFINTFC.EXE	BOF process
BOFINTFC.DAT	BOF process definition file
LOGINTFC.EXE	Logic process
LOGINTFC.DAT	Definition file of the logic process
NETINTFC.EXE	Application for connection of client/server
KILLTASK.EXE	Application for terminating function interface clients (see chapter 4.6 Tips and Tricks for working with the Interface)

Example Entries in the "VERSION.DAT" File

The version ID of the individual parts of the program as well as the version of he 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	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]:\...\IND_DRV\IF_DLL\" Directory

The drive as well as the path "[LW]:\...\" are preset during the standard installation routine to "C:\MT-CNC\". The following function libraries of the function interface are contained in the "C:\MT-CNC\IND_DRV\IF_DLL\" directory:

File	Explanation
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 connections and SPS and CNC 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 MTCX 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 MTAX device groups.
INDIF850.DLL	Functions for access to the MTRX device groups.
INDIF860.DLL	Functions for access to the MSYX device groups.
INDIFA00.DLL	Functions for HMI support of the MTCX device groups.
INDIFZ00.DLL	Functions for access to the MTAX device groups.

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.

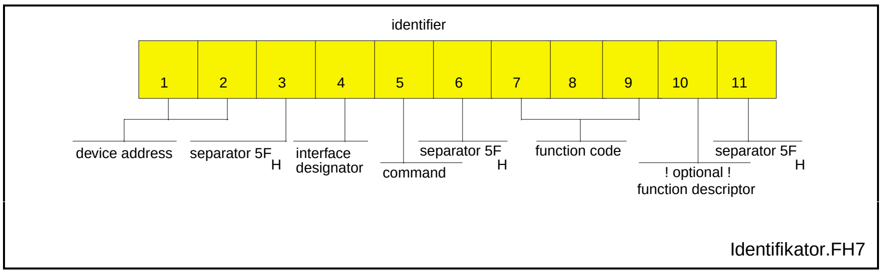


Fig. 6-1: Identifier

Device address

The device address corresponds to the system address within the Rexroth Indramat BOF/GBO. This means, e.g. that device 00 corresponds to system 0. Please observe, however, that the Rexroth Indramat BOF/GBO always requires a device 00. The addresses are listed specific to the device group in the following table as well as in chapter 7, Function Interface Commands.

Address	Group	Affiliated device types
[xx]	MPCX	PCs
[00...15]	MTCX	MTCNC, MTC200-P, MTC200-R, MTVNC
[00]	MSCX	SERCANS-A, SERCANS-P
[00...63]	MISX	ISP200-P, ISP200-R
[00...63]	MTAX	MTA200-P (ANDRON controller)
[00...63]	MSYX	SYNAX200-P, SYNAX200-R

Separator

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

Interface designator

BOF Process
Controller (logic process and communication process)

All data access via the interface identifier „B“ are managed by the BOF process so as to ensure, e.g. 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 BOF process ensures that a different user program cannot open the file once 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 (MTCX and MTAX, etc.).

Read and Write Command

W = **W**rite	(write)
R = Single **R**ead	(read)
C = **C**yclic Read	(cyclic read)
B = **B**reak Cyclic Read	(interruption in cyclic read)

(Read) command „**R**“

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

Write command „**W**“

A new value for the specified SPS 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 in the event of the 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

With its three letters, the function code provides information about the data to be accessed. The identifier is encoded in the form of the data type designation. After the three letters, a function descriptor for the respective function code may be optionally 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 CNC- process 0, from partial program1 of the NC set 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 request. This is why this access is specified in greater detail with a function descriptor.

Selector

Example

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

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

Example 1 (cyclic reading)

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

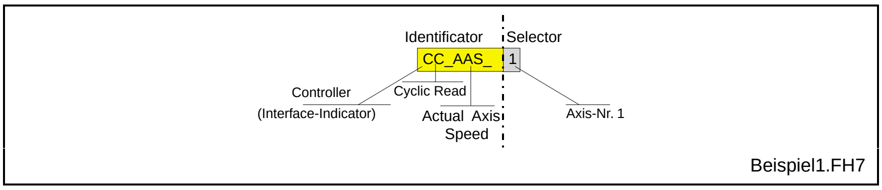


Fig. 6-2: Example 1: Cyclic reading in ASCII code

Example 2 (cyclic reading)

Cyclic reading of the current feed rate in the CNC process 3 of device address 02.

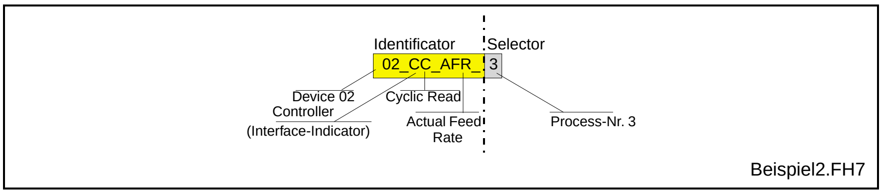


Fig. 6-3: Example 2: Cyclic reading in ASCII code

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 reply in the result buffer.

The following coding types are supported:

1 = ASCII default !

2 = Binary

3 = ANSI

4 = Unicode

Example 3 (cyclic reading)

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

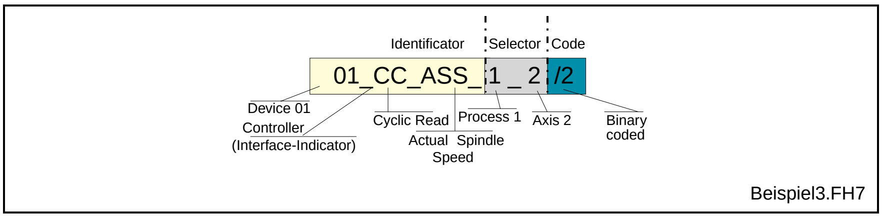


Fig. 6-4: Example 3: Cyclic reading in binary code

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 of the MTCX Device Groups

Parameter	Value range
Axis number	1...32
CNC memory	1=A, 2=B
NC block No.	1...10000
NC program number	0...99
NC packet	1...99
Spindle number	S1, S2, S3
CNC process number	0...6
Mechanism number	0...31
Drive address	0...254
Tool number	0...9999999
Duplo No.	1...9999
Data block	0 = basic tool data 1...9 = cutter data
Data element	1...28 for basic tool data 1..0,40 = cutter data
Memory	M = Magazine/turret S = Spindle G = Gripper X = Index data
Position	1...999 in the case of M 1...4 in the case of S,G 0...16770215 in the case of X

Meanings of the Axes for the MTCX 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 MTCX Device Group

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

Base Units

Base unit	Measurement System			
	Polar in mm	**Polar in inch**	**Rotatory in units**	**Specific to main spindle**
Speed	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
Displacement	mm	inch	units	deg
Speed	1/min	1/min	1/min	1/min
Cutting speed	m/min	inch/min	units/min	--

6.3 Survey of FI Commands

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

7Survey of the MPCX Device Group

Com.	**Description**	**Process**	R**ead**	W**rite**	C**yclic**
CCP1	**C**ell **C**onfiguration **P**arameter	B	R		
CCP2	**C**ell **C**onfiguration **P**arameter	B	R		
CCP3	**C**ell **C**onfiguration **P**arameter	B	R		
CCP4	**C**ell **C**onfiguration **P**arameter	B	R		
CCP5	**C**ell **C**onfiguration **P**arameter	B	R		
DFJ1	**D**elete **F**unction Interface **J**ob	B	R		
DFJ2	**D**elete **F**unction Interface **J**ob	B	R		
ERI1	**ER**ror **I**nformation	B	R		
FCP1	**F**ar Device **C**onfiguration **P**arameter	B	R		C
FCP2	**F**ar Device **C**onfiguration **P**arameter	B	R		C
FCP3	**F**ar Device **C**onfiguration **P**arameter	B	R		C
FDC1	**F**ar **D**evice **C**onfiguration	B	R		C
FIT1	**F**urther **I**nfo **T**ext	B	R		
FPC1	**F**ar **P**C **C**onfiguration	B	R		C
IFJ1	**I**nformation about **F**unction Interface **J**obs	B	R		C
IFJ2	**I**nformation about **F**unction Interface **J**obs	B	R		C
LNG	Active **L**a**NG**uage	B	R		C
SFW1	**S**et **F**ocus to **W**indow	B		W	C
SFW2	**S**et **F**ocus to **W**indow	B		W	C

Survey of the MTCX Device Group

Com.	**Description**	**Process**	R**ead**	W**rite**	C**yclic**
AAC1	**A**ctual **AC**celeration	C	R		C
AAD	**A**ctive **A**ngle **D**imension	C	R		C
AAS1	**A**ctual **A**xis **S**peed	C	R		C
AAS2	**A**ctual **A**xis **S**peed	C	R		C
ABI	**A**ctual NC-**B**lock **I**nformation	B	R		C
ACS	**A**ctual **C**utting **S**peed	C	R		C
ADN1	**A**ctive **D**-Correction **N**umber	C	R		C
AEM	**A**ctive **E**vent **M**onitoring	C	R		C
AEN	**A**ctive **E**dge-**N**umber	C	R		C
AFO1	**A**ctive **F**eedrate **O**verride	C	R		C
AFR	**A**ctive **F**eed**R**ate	C	R		C
Com.	**Description**	**Process**	R**ead**	W**rite**	C**yclic**
AGF	**A**ctive **G**-**F**unction	C	R		C

AMF	**A**ctive **M**-**F**unction	C	R		C
AMM1	**A**ctive **M**echanism **M**essage	B	R		C
AMM2	**A**ctive **M**echanism **M**essage	B	R		C
AMM3	**A**ctive **M**echanism **M**essage	B	R		C
AMM4	**A**ctive **M**echanism **M**essage	B	R		C
AMM5	**A**ctive **M**echanism **M**essage	B	R		C
API1	**A**ctual **P**arameter **I**ndex	B	R		C
API2	**A**ctual **P**arameter **I**ndex	B	R		C
APM	**A**ctive **P**art-Program **M**essage	C	R		C
APN	**A**ctive **P**art-Program Message Number	C	R		C
APO1	**A**ctual Machine **PO**sition	C	R		C
APO2	**A**ctual Machine **PO**sition	C	R		C
APP	**A**ctive **P**art-**P**rogram number	C	R		C
ARO1	**A**ctual **R**apid **O**verride	C	R		C
ASF	**A**ctual **S**pindle **F**or Process	C	R		C
ASG	**A**ctual **S**pindle **G**ear	C	R		C
ASM1	**A**ctive **S**ystem-Fault **M**essage	B	R		C
ASM2	**A**ctive **S**ystem-Fault **M**essage	B	R		C
ASM3	**A**ctive **S**ystem-Fault **M**essage	B	R		C
ASM4	**A**ctive **S**ystem-Fault **M**essage	B	R		C
ASM5	**A**ctive **S**ystem-Fault **M**essage	B	R		C
ASN	**A**ctual **S**equence **N**umber	C	R		C
ASO1	**A**ctual **S**pindle **O**verride	C	R		C
ASS	**A**ctual **S**pindle **S**peed	C	R		C
ATN	**A**ctive **T**ool-**N**umber	C	R		C
ATP1	**A**ctual **T**ool **P**lace Information	C	R		C
ATP2	**A**ctual **T**ool **P**lace Information	C	R		C
ATP3	**A**ctual **T**ool **P**lace Information	C	R		C
AZB1	**A**ctive **Z**ero Offset **B**ank	C	R		C
CPO1	**C**ommand **PO**sition (NOMINAL)	C	R		C
CPO2	**C**ommand **PO**sition by log AxisNo	C	R		C
CRT	**C**ontrol **R**ese**T**	C		W	
DAC1	**D**evice **A**xis **C**onfiguration Parameter	B	R		C
DAC2	**D**evice **A**xis **C**onfiguration Parameter	B	R		C
DCD1	**D**-**C**orrection **D**ata	C	R		C
DCP1	**D**evice **C**onfiguration **P**arameter	B	R		C
DCP2	**D**evice **C**onfiguration **P**arameter	B	R		C
DCR1	**D**-**C**orrection **R**ecord	C	R	W	C
DIS1	**D**ata **I**dentification **S**tring Parameter	C	R		
DIS2	**D**ata **I**dentification **S**tring PLC Program	C	R		
DIS3	**D**ata **I**dentification **S**tring NC Packet	C	R		
Com.	**Description**	**Process**	R**ead**	W**rite**	C**yclic**
DIS4	**D**ata **I**dentification **S**tring Tool List	C	R		

DIS5	**D**ata **I**dentification **S**tring Machine	C	R		
DIS6	**D**ata **I**dentification **S**tring NC Program	C	R		
DPN	**D**elete Part **P**rogram **N**C	B		W	
DPP	**D**elete Part **P**rogram **P**ackage	B		W	
DTC1	**D**evice **T**ool Management **C**onfiguration	B	R		C
DTG1	**D**istance **T**o **G**o	C	R		C
DTG2	**D**istance **T**o **G**o by log. AxisNo	C	R		C
DTY1	**D**evice **TY**pe	C	R		
EPO1	ProgrammEd **PO**sition (END)	C	R		C
EPO2	ProgrammEd **PO**sition (END)	C	R		C
GPC1	**G**lobal **P**rocess **C**onfiguration	B	R		C
GPC2	**G**lobal **P**rocess **C**onfiguration	B	R		C
GPP1	**G**lobal **P**rocess **P**arameter	B	R		C
GPP2	**G**lobal **P**rocess **P**arameter	B	R		C
IPP	**I**nsert **P**rogram **P**ackage	B		W	
MAP1	**M**odule **A**ssign of **P**rocess	B	R		C
MAR	**M**ap **A**bsolute PCL-**R**eference	B	R		
MCD1	**M**odule **C**onfiguration: **D**evice Information	B	R		C
MCM1	**M**odule **C**onfiguration: **M**odule Information	B	R		C
MCP1	**M**odule **C**onfiguration: **P**rocess Information	B	R		C
MCS1	**M**odule **C**onfiguration: **S**FC Information	B	R		C
MFO1	**M**aximal **F**eedrate **O**verride	C	R		C
MFR	**M**aximal **F**eed**R**ate	C	R		C
MRO1	**M**aximal **R**apid **O**verride	C	R		C
MSO1	**M**aximal **S**pindle **O**verride	C	R		C
MSS	**M**aximal **S**pindle **S**peed	C	R		C
MTD	**M**achine **T**able **D**ata	C	R	W	C
NEV	**N**C **Ev**ent	C	R	W	C
NMM	**N**C **M**e**M**ory selection	C		W	
NPA1	**N**C **Pa**rameter	B	R		C
NPA2	**N**C **Pa**rameter	B	R		C
NPA3	**N**C **Pa**rameter	B	R		C
NPA4	**N**C **Pa**rameter	B	R		C
NPC1	**N**C **P**ackage **C**ompiling	B	R		C
NPD1	**N**C **P**ackage **D**ownload	B		W	
NPD2	**N**C **P**ackage **D**ownload	B		W	
NPD3	**N**C **P**ackage **D**ownload Quickly	B		W	
NPD4	**N**C **P**ackage **D**ownload Quickly	B		W	
NPI	**N**C **P**ackage D**I**rectory	B	R		C
NPS	**N**C **P**rogram **S**election	C		W	
NTN	**N**ext Tool **N**umber	C	R		C
Com.	**Description**	**Process**	R**ead**	W**rite**	C**yclic**
NVS	**N**C **V**ariable **S**ingle	C	R	W	C

OPD1	**O**ptimal **P**osition **D**istance by Axis sign.	C	R		C
OPD2	**O**ptimal **P**osition **D**istance by phys. AxisNo	C	R		C
PAC1	**P**rocess **A**xis **C**onfiguration Parameter	B	R		C
PAC2	**P**rocess **A**xis **C**onfiguration Parameter	B	R		C
PFR	**P**rogrammed **F**eed**R**ate	C	R		C
PPD	**P**art **P**rogram **D**irectory	B	R		C
PPN	**P**art **P**rogram **N**C	B	R	W	C
PPP	**P**art **P**rogram **P**ackage	B			C
PPS	**P**art-**P**rogram-**S**equence	C	R		
PSS	**P**rogrammed **S**pindle **S**peed	C	R		C
PTC1	**P**rocess **T**ool Management **C**onfiguration	B	R		C
PTC2	**P**rocess **T**ool Management **C**onfiguration	B	R		C
PVF	**P**LC **V**ariable **F**ormatted	C		W	C
PVS	**P**LC-**V**ariable **S**ingle	C	R	W	C
PVT	**P**LC **V**ariable **T**ype	B	R		
SID1	**S**oftware **I**nstallation **D**ata	B	R		C
SLA1	Actual **S**ervo **LA**g	C	R		C
SLA2	Actual **S**ervo **LA**g	C	R		C
SLI	**SPS L**ong **I**dentification	B	R		C
SPA1	**S**ercos **PA**rameter	B	R	W	C
SPH	**S**ercos **PH**ase	C	R	W	
SPP	**S**elected **P**art **P**rogram Number	C	R		C
TDA1	**T**ool **DA**ta	B	R	W	
TDA2	**T**ool **DA**ta	B	R		
TDR1	**T**ool **D**ata **R**ecord of Place	C	R		C
TDR2	**T**ool **D**ata **R**ecord	C	R		C
TFD1	**T**ext **F**iles **D**ownload	B	R		
TIF	**T**ool **I**nsert **F**inish	C	R		
TII	**T**ool **I**nsert **I**nitiated	C	R		
TLB1	**T**oo**L B**asicdata List	B	R		C
TLB2	**T**oo**L B**asicdata List	B	R		C
TLD1	**T**oo**L D**ata of Place	C	R	W	C
TLD2	**T**oo**L D**ata of Tool	C	R	W	C
TLD3	**T**oo**L D**ata of Place	C	R	W	C
TLD4	**T**oo**L D**ata of Tool	C	R	W	C
TLE1	**T**oo**L E**dgedata List	B	R		C
TLE2	**T**oo**L E**dgedata List	B	R		C
TMV	**T**ool **M**o**V**e	C	R		
TQE1	Actual **T**or**Q**u**E**	C	R		C
TQE2	Actual **T**or**Q**u**E**	C	R		C
TRM	**T**ool **R**e**M**ove	C	R		
Com.	**Description**	**Process**	R**ead**	W**rite**	C**yclic**
TRS	**T**ool **R**e**S**et	C	R		

ZOD	**Z**ero **O**ffset **D**ata	C		W	C
ZOD1	**Z**ero **O**ffset **D**ata	C	R		C
ZOD2	**Z**ero **O**ffset Data	C	R		C

Survey of the MSCX Device Group

Com.	**Description**	**Process**	R**ead**	W**rite**	C**yclic**
ASE	**A**ctual **S**ystem **E**rror	C	R		C
CSE	**C**lear **S**ystem **E**rror	C		W	
DTY1	**D**evice **Ty**pe	C	R		
SID1	**S**oftware **I**nstallation **D**ata	B	R		C
SPA1	**S**ercos **Pa**rameter	B	R	W	C
SPH	**S**ercos **Ph**ase	C	R	W	C

Survey of the MISX Device Group

Com.	**Description**	**Process**	R**ead**	W**rite**	C**yclic**
ASM1	**A**ctive **S**ystem Fault **M**essage	B	R		C
ASM2	**A**ctive **S**ystem Fault **M**essage	B	R		C
ASM3	**A**ctive **S**ystem Fault **M**essage	B	R		C
ASM4	**A**ctive **S**ystem Fault **M**essage	B	R		C
ASM5	**A**ctive **S**ystem Fault **M**essage	B	R		C
CRT	**C**ontrol **R**ese**T**	C		W	
DIS2	**D**ata **I**dentification **S**tring PLC Program	C	R		
DTY1	**D**evice **Ty**pe	C	R		
MAR	**M**ap **A**bsolute PCL **R**eference	B	R		
MCD1	**M**odule **C**onfiguration: **D**evice Information	B	R		C
MCM1	**M**odule **C**onfiguration: **M**odule Information	B	R		C
MCS1	**M**odule **C**onfiguration: **S**FC- Information	B	R		C
PVF	**P**LC **V**ariable **F**ormatted	C		W	C
PVS	**P**LC **V**ariable **S**ingle	C	R	W	C
PVT	**P**LC **V**ariable **T**ype	B	R		
SID1	**S**oftware **I**nstallation **D**ata	B	R		C
SLI	**SPS** **L**ong **I**dentification	B	R		C
TFD1	**T**ext **F**iles **D**ownload	B	R		

Survey of the MTAX Device Group

Com.	**Description**	**Process**	R**ead**	W**rite**	C**yclic**
AMM7	**A**ctive **M**echanism **M**essage	B	R		C
APO2	**A**ctual Machine **PO**sition	C	R		C
ASM1	**A**ctive **S**ystem Fault **M**essage	B	R		C
ASM2	**A**ctive **S**ystem Fault **M**essage	B	R		C
ASM3	**A**ctive **S**ystem Fault **M**essage	B	R		C
ASM4	**A**ctive **S**ystem Fault **M**essage	B	R		C
ASM5	**A**ctive **S**ystem Fault **M**essage	B	R		C
CMA	**CM**OS RAM **A**SCII Parameter	C	R	W	
CMF	**CM**OS RAM **F**loating Point Parameter	C	R	W	
CMI	**CM**OS RAM **I**nteger Parameter	C	R	W	
CRT	**C**ontrol **R**ese**T**	C		W	
DCP1	**D**evice **C**onfiguration **P**arameter	B	R		C
DCP2	**D**evice **C**onfiguration **P**arameter	B	R		C
DIS2	**D**ata **I**dentification **S**tring PLC Program	C	R		
DTG2	**D**istance **T**o **G**o by log. AxisNo	C	R		C
DTY1	**D**evice **Ty**pe	C	R		
MAR	**M**ap **A**bsolute PCL **R**eference	B	R		
MAP1	**M**odule **A**ssign of **P**rocess	B	R		C
MCD1	**M**odule **C**onfiguration: **D**evice Information	B	R		C
MCM1	**M**odule **C**onfiguration: **M**odule Information	B	R		C
MCP1	**M**odule **C**onfiguration: **P**rocess Information	B	R		C
MCS1	**M**odule **C**onfiguration: **S**FC- Information	B	R		C
PVF	**P**LC **V**ariable **F**ormatted	C		W	C
PVS	**P**LC **V**ariable **S**ingle	C	R	W	C
PVT	**P**LC **V**ariable **T**ype	B	R		
SID1	**S**oftware **I**nstallation **D**ata	B	R		C
SLI	**SPS L**ong **I**dentification	B	R		C

Survey of the MSYX Device Group

Com.	**Description**	**Process**	R**ead**	W**rite**	C**yclic**
ASE	**A**ctual **S**ystem **E**rror	C	R		C
CSE	**Cl**ear **S**ystem **E**rror	C		W	
DTY	**D**evice **TY**pe	C	R		
SID1	**S**oftware **I**nstallation **D**ata	B	R		C
SPA1	**S**ercos **PA**rameter	B	R	W	C
SPH	**S**ercos **PH**ase	C	R	W	C

6.4 Logical Connection between FI Commands

All FI commands are gathered together in the following table, ordered from a logical point of view.

GROUP:	DEVICE GRP.:	FI COMMAND:
Axes	MTCX	AAD, AAS1, AAS2, CPO1, CPO2, DTG1, DTG2, EPO1, EPO2, OPD1, OPD2, PAC1, PAC2, SLA1, SLA2, TQE1, TQE2
	MTAX	DTG2
D Correction	MTCX	ADN1, DCD1, DCR1
Download	MTCX	NPD1, NPD2, NPD3, NPD4, TFD1
	MISX	TFD1
Event	MTCX	AEM, NEV
Device	MPCX	CCP1, CCP2, CCP3, CCP4, CCP5, FCP1, FCP2, FCP3, FDC1
	MTCX	DCP1, DCP2, DTC1, DTY1, MCD1
	MSCX	DTY1
	MISX	DTY1, MCD1
	MTAX	DCP1, DCP2, DTC1, DTY1, MCD1
Configuration	MPCX	CCP1, CCP2, CCP3, CCP4, CCP5, FCP1, FCP2, FCP3, FDC1
	MTCX	DAC1, DAC2, DCP1, DCP2, DTY1, GPC1, GPC2, PAC1, PAC2, PTC1, PTC2
	MSCX	DTY1
	MISX	DTY1
	MTAX	DCP1, DCP2, DTY1
Machine data	MTCX	DIS5, MTD
Messages	MPCX	FIT1, ERI1
	MTCX	AMM1, AMM2, AMM3, AMM4, AMM5, ASM1, ASM2, ASM3, ASM4, ASM5, SLI, TFD1
	MSCX	ASE, CSE
	MISX	ASM1, ASM2, ASM3, ASM4, ASM5, SLI, TFD1
	MTAX	AMM7, ASM1, ASM2, ASM3, ASM4, ASM5, SLI
Module	MTCX	MAP1, MCD1, MCM1, MCP1, MCS1
	MISX	MCD1, MCM1, MCS1
	MTAX	MAP1, MCD1, MCM1, MCP1, MCS1
NC processing	MTCX	ABI, AGF, AMF, APM, APN, APP, ASN, DIS1, DIS2,DIS3, DIS4, DIS5, DIS6, DPN, DPP, IPP, NEV, NMM, NPA1, NPA2, NPA3, NPA4, NPC1, NPD1, NPD2, NPD3, NPD4, NPI, NPS, NVS, PPD, PPN, PPP, PPS, SPP
Override	MTCX	AFO1, ARO1, ASO1, MFO1, MRO1, MSO1
Position value	MTCX	APO1, APO2, CPO1, CPO2, DTG1, DTG2, EPO1, EPO2, OPD1, OPD2, SLA1, SLA2
	MTAX	APO2, DTG2
Process	MTCX	GPC1, GPC2, GPP1, GPP2, MAP1, MCP1, PAC1, PAC2, PTC1, PTC2
	MTAX	MAP1, MCP1
Cut	MTCX	AEN, TLE1, TLE2

GROUP:	DEVICE GRP.:	FI COMMAND:
Sercos	MTCX	SPA1, SPH
	MSCX	SPA1, SPH
	MSYX	SPA1, SPH
Spindle	MTCX	AAD, AAS1, AAS2, ACS, ASF, ASG, ASO1, ASS, MSO1, MSS, PSS
PLC	MTCX	DIS2, MAR, PVF, PVS, PVT, SLI
	MISX	DIS2, MAR, PVF, PVS, PVT, SLI
	MTAX	DIS2, MAR, PVF, PVS, PVT, SLI
Feed	MTCX	AAC1, AAD, AAS1, AAS2, ACS, ADN1, AFO1, AFR, ARO1, AZB1, CPO1, CPO2, DCD1, DCR1, DTG1, DTG2, MFO1, MFR, MRO1, OPD1, OPD2, PFR, PSS, SLA1, SLA2, TQE1, TQE2, ZOD, ZOD1, ZOD2
	MTAX	DTG2
Tool	MTCX	AEN, ATN, ATP1, ATP2, ATP3, DIS4, DTC1, NTN, PTC1, PTC2, TDA1, TDA2, TDR1, TDR2, TIF, TII, TLB1, TLB2, TLD1, TLD2, TLD3, TLD4, TLE1, TLE2, TMV, TRM, TRS

6.5 Command Execution Times

Legend for the Command Execution Times

The command execution times determined are typical test values. Their reproducibility 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 test values determined are subject to a rasterization of 10ms. In principle, therefore, a tolerance of +/- 10ms should be assumed. In addition, sporadic test values will be determined that lie outside of this tolerance range. Deterministic behavior can therefore not be the basis.

The execution times determined do however help you to get a feeling for the processing times of the commands. Thereby, various variants of the device access can be tried “at your desk” and the best type of access can be found.

For a better comparison, the specifications of the PC and device configuration with which the command execution times have been determined are listed in the following:

Computer Type

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

Processor	Memory	Operating System
Pentium 166 MHz	32 Mbytes	Windows NT 4.0

Fig. 6-5: Computer Specifications

Device Configuration

To determine the command execution times, a representative device was selected from each device class and the complete range of commands of the device was tested. The communication port used between the PC and the device is thereby 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.

The following table lists the respective representative devices of the various device classes:

	MPCX	**MTCX**	**MSCX**	**MISX**	**MTAX**	**MSYX**
Device	PC	MTA200-P	SERCANS-A	ISP200-P	MTA200-P	SYNAX200-P
SPS Components	None	MTS-P	None	MTS-P02.2	MTS-P	None
NC Components	None	MTC-P	None	None	None	None
Communication Configuration	None	DPR TCON	V24 19200 baud TCON	DPR TCON	SHM	DPR TCON

Fig. 6-6: 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 ANDRON.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 connected PC

Note: ***1)** The command marked is a job command. The time given refers to the start of the job. To get the total command execution time, you must add to this the time that the job runs in the background.

***2)** In evaluating the command execution time, the comment is of decisive importance.

Command Execution Times of 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_MTCX	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_MTCX	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

Command Execution Times of the MTCX 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

Com.	Description	Example	[ms]
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_ARO1_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_MTCX	50
ASM5	Active System-Fault Message	00_BR_ASM5_74_0	30
ASN	Actual Sequence Number	00_CR_ASN_0	20
ASO1	Actual Spindle Override	00_CR_ASO1_0_1	20
ASS	Actual Spindle Speed	00_CR_ASS_0_1	20
ATN	Active Tool-Number	00_CR_ATN_0	20
ATP1	Actual Tool Place Information	00_CR_ATP1_0	20
ATP2	Actual Tool Place Information	00_CR_ATP2_0	20
ATP3	Actual Tool Place Information	00_CR_ATP3_0	20
AZB1	Active Zero-Offset Bank	00_CR_AZB1_0	20
CPO1	Command POsition (SOLL)	00_CR_CPO1_0_2_1	20
CPO2	Command POsition by log.AxisNo	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. AxisNo	00_CR_DTG2_3_1	20
DTY1	Device TYpe	00_CR_DTY1	20

Com.	Description	Example	[ms]
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
IPP	Insert Program Package	00_BW_IPP_2_1 Value: TEST	110
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	Maximal Feedrate Override	00_CR_MFO1_0	20
MFR	Maximal FeedRate	00_CR_MFR_0	20
MRO1	Maximal Rapid Override	00_CR_MRO1_0	20
MSO1	Maximal Spindle Override	00_CR_MSO1_0_1	20
MSS	Maximal Spindle Speed	00_CR_MSS_0_1	20
MTD	Machine Table Data	00_CR_MTD_90_0_0_1_7	20
NEV	NC-EVent	00_CR_NEV_0_1	20
NMM	NC-MeMory selection	00_CW_NMM Value: 2	10
NPA1	NC-PArameter	00_BR_NPA1_01_A00.000	90
NPA2	NC-PArameter	00_BR_NPA2_01_A00.000_A00.004	90
NPA3	NC-PArameter	00_BR_NPA3_01_A00.000_3	100
NPA4	NC-PArameter	00_BR_NPA4_01_A00.000	120
NPC1	NC Package Compiling	00_BR_NPC1_1	30 *1)
NPD1	NC-Package Download	00_BW_NPD1_1_1	2290 *1)
NPD2	NC-Package Download	00_BW_NPD2_1_1	2380 *1)
NPD3	NC-Package Download Quickly	00_BW_NPD3_1_1 Value: 1	610
NPD4	NC-Package Download Quickly	00_BW_NPD4_1_1 Value: 1	770
NPI	NC-Package DIrectory	00_BR_NPI	20
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	Optimal Position Distance	00_CR_OPD1_0_2	20
OPD2	Optimal Position Distance by log. AxisNo	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
Com.	**Description**	**Example**	**[ms]**
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 Long Identification	00_BR_SLI	30
SPA1	Sercos PArameter	00_BR_SPA1_1_S-0-0001_40	120
SPH	Sercos PHase	00_CW_SPH_1 Value: 2	20
SPP	Selected Part-Program Number	00_CR_SPP_0	20
TDA1	Tool DAta	00_BR_TDA1_0_M_21	60
TDA2	Tool DAta	00_BR_TDA2_0_1_1	70
TDR1	Tool Data Record of Place	00_CR_TDR1_0_M_21_0	30
TDR2	Tool Data Record	00_CR_TDR2_0_1_1_0	20
TIF	Tool Insert Finish	00_CR_TIF_0_M_25	20
TII	Tool Insert Initiated	00_CR_TII_0_M_25	20
TLB1	TooL Basicdata List	00_BR_TLB1_0_M_1_10_2_5_6_7	380 *2)
TLB2	TooL Basicdata List	00_BR_TLB2_0_2_5_6_7	700 *2)
TLD1	TooL Data of Place	00_CR_TLD1_0_M_1_1_1	20
TLD2	TooL Data of Tool	00_CR_TLD2_0_1_1_0_5	20
TLD3	TooL Data of Place	00_CR_TLD3_0_M_2_1	30
TLD4	TooL Data of Tool	00_CR_TLD4_0_1_1_1	30
TLE1	TooL Edgedata List	00_BR_TLE1_0_1_M_1_3_2_3	260 *2)
TLE2	TooL Edgedata 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

Command Execution Times of 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

Command Execution Times of the MISX 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_MTCX	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 Long Identification	00_BR_SLI	10

Command Execution Times of the MTAX 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_MTCX	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 Floatingpoint 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. AxisNo	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 of 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	

7 Function Interface Commands

The following FI Commands are valid for the MPCX device group. Please note that the device class "XX" must always be set before the FI command, e.g. XX_BR_CCP1 (refer here to chapter 6.1 Elements of the FI Command).

Outputting the Device Configuration: CCP

MPCX Device Group

Name

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

Explanation

The configuration settings are read in from the "IND_DEV.INI" file. The configuration of the individual communication addresses and the settings of the various Rexroth Indramat devices are set in this file (refer to chapter, “Contents of the "IND_DEV.INI" File”, p. 5-19).

FI Command

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

XX_BR_CCP1 **(Single Read)**

Construction of Answer

The following table shows the general construction of the answer of the FI command CCP1. The answer consists of a maximum of n=16 lines (n=16 configurable devices), each with 13 lines.

Line 1...n:	Column 1	...	Column 13

Value Range/Meaning of Columns

1 =	Device address	IND_DEV.INI entry: [DeviceAddrX]
2 =	Device name	IND_DEV.INI entry: DeviceName=
3 =	Device type:	IND_DEV.INI entry: DeviceType=
4 =	SPS support	IND_DEV.INI entry: PLC=
5 =	Device status	IND_DEV.INI entry: DeviceStatus=
6 =	Assignment of a simulation pair	IND_DEV.INI entry: DeviceAssign=
7 =	Device mode	IND_DEV.INI entry: MtvncMode=
8 =	Communication channel	IND_DEV.INI entry: [CommAddrX]
9 =	Description of the communication channel	IND_DEV.INI entry: CommStr=
10 =	Timeout value	IND_DEV.INI entry: Timeout=
11 =	Device group	(see chapter, 6.1 Elements of the FI Commanf, Identifier)
12 =	Type of SPS component	IND_DEV.INI entry: Componenttype1=
13 =	Type of CNC component	IND_DEV.INI entry: Componenttype1=

Example CCP1

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

FI Command

Output of the configuration settings of the selected device type.

BR_CCP2_(1) **(Single Read)**

(1)= device type [MTCNC, MTC200-P, MTC200-R, MTVNC, SERCANS-A, SERCANS-P, ISP200-P, ISP200-R, TRA200-P, TRA200-R, MTA200-P]

Construction of Answer

The following table shows the general construction of the answer of the FI command CCP2. The answer consists of a maximum of n=16 lines (n=16 configurable devices), each with 13 lines.

Line 1...n:	Column 1	...	Column 13

Value Range/Meaning of Columns

1 =	Device address	IND_DEV.INI entry: [DeviceAddrX]
2 =	Device name	IND_DEV.INI entry: DeviceName=
3 =	Device type:	IND_DEV.INI entry: DeviceType=
4 =	SPS support	IND_DEV.INI entry: PLC=
5 =	Device status	IND_DEV.INI entry: DeviceStatus=
6 =	Assignment of a simulation pair	IND_DEV.INI entry: DeviceAssign=
7 =	Device mode	IND_DEV.INI entry: MtvncMode=

8 =	Communication channel	IND_DEV.INI entry: [CommAddrX]
9 =	Description of the communication channel	IND_DEV.INI entry: CommStr=
10 =	Timeout value	IND_DEV.INI entry: Timeout=
11 =	Device group	(see chapter 6.1 Elements of the FI Command, Identifier)
12 =	SPS component type	IND_DEV.INI entry: Componenttype1=
13 =	CNC component type	IND_DEV.INI entry: Componenttype1=

Example CCP2

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

FI Command

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

BR_CCP3_(1) **(Single Read)**

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

Construction of Answer

The following table shows the general construction of the answer of the FI command CCP3. The answer consists of a maximum of n=16 lines (n=16 configurable devices), each with 13 lines.

Line 1...n:	Column 1	...	Column 13

Value Range/Meaning of Columns

1 =	Device address	IND_DEV.INI entry: [DeviceAddrX]
2 =	Device name	IND_DEV.INI entry: DeviceName=
3 =	Device type:	IND_DEV.INI entry: DeviceType=
4 =	SPS support	IND_DEV.INI entry: PLC=
5 =	Device status	IND_DEV.INI entry: DeviceStatus=
6 =	Assignment of a simulation pair	IND_DEV.INI entry: DeviceAssign=
7 =	Device mode	IND_DEV.INI entry: MtvncMode=
8 =	Communication channel	IND_DEV.INI entry: [CommAddrX]

9 =	Description of the communication channel	IND_DEV.INI entry: CommStr=
10 =	Timeout value	IND_DEV.INI entry: Timeout=
11 =	Device group	(see chapter 6.1 Elements of the FI Command, Identifier)
12 =	Type of component	IND_DEV.INI entry: Componenttype1=
13 =	Type of component	IND_DEV.INI entry: Componenttype1=

Example CCP3

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

Assumption: The following device types have been defined:

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

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

FI Command

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

BR_CCP4_(1)	**(Single Read)**
(1) = device group	[MTCX, MSCX, MISX, MTRX, MTAX] (see chapter 6.1 Elements of the FI Commando, Identifier)

Construction of Answer

The following table shows the general construction of the answer of the FI command CCP4. The answer consists of a maximum of n=16 lines (n=16 configurable devices), each with 13 lines.

Line 1...n:	**Column 1**	**...**	**Column 13**

Value Range/Meaning of Columns

1 =	Device address	IND_DEV.INI entry: [DeviceAddrX]
2 =	Device name	IND_DEV.INI entry: DeviceName=
3 =	Device type:	IND_DEV.INI entry: DeviceType=
4 =	SPS support	IND_DEV.INI entry: PLC=
5 =	Device status	IND_DEV.INI entry: DeviceStatus=
6 =	Assignment of a simulation pair	IND_DEV.INI entry: DeviceAssign=
7 =	Device mode	IND_DEV.INI entry: MtvncMode=
8 =	Communication channel	IND_DEV.INI entry: [CommAddrX]

9 =	Description of the communication channel	IND_DEV.INI entry: CommStr=
10 =	Timeout value	IND_DEV.INI entry: Timeout=
11 =	Device group	(see chapter 6.1 Elements of the FI Command, Identifier)
12 =	Type of component	IND_DEV.INI entry: Componenttype1=
13 =	Type of component	IND_DEV.INI entry: Componenttype2=

Example CCP4

Reads the configuration settings of the defined MSCX devices. Assumption: The following device types have been defined:

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

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

FI Command

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

BR_CCP5_(1) **(Single Read)**

(1) = device address [00...63]

Construction of Answer

The following table shows the general construction of the answer of the FI command CCP5. The answer consists of a line with 13 columns.

Line 1...n:	Column 1	...	Column 13

Value Range/Meaning of Columns

1 =	Device address	IND_DEV.INI entry: [DeviceAddrX]
2 =	Device name	IND_DEV.INI entry: DeviceName=
3 =	Device type:	IND_DEV.INI entry: DeviceType=
4 =	SPS support	IND_DEV.INI entry: PLC=
5 =	Device status	IND_DEV.INI entry: DeviceStatus=
6 =	Assignment of a simulation pair	IND_DEV.INI entry: DeviceAssign=
7 =	Device mode	IND_DEV.INI entry: MtvncMode=
8 =	Communication channel	IND_DEV.INI entry: [CommAddrX]
9 =	Description of the communication channel	IND_DEV.INI entry: CommStr=
10 =	Timeout value	IND_DEV.INI entry: Timeout=
11 =	Device group	(see chapter 6.1 Elements of the FI Command, Identifier)

12 =	Type of component	IND_DEV.INI entry: Componenttype1=
13 =	Type of component	IND_DEV.INI entry: Componenttype2=

Example CCP5 Reads the configuration settings of device address 00.

Assumption: The following device types have been defined:

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

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

Removing Function Interface Jobs: DFJ

MPCX Device Group

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

Explanation Jobs, also referred to as FI-Jobs, are removed from the administration structure of the functions interface. These are jobs that have either the status "READY" or "ERROR" . All interface jobs are removed using the FI command DFJ1; DFJ2 removes the selected job.

Note: Refer here also to Activate NC Download: NPD, p. 7-23 in chapter 7.2 “FI Commands for the MTCX Device Group”.

FI Command Removes all FI-Jobs from the administration structure of the function interface.

XX_BR_DFJ1 **(Single Read)**

Construction of Answer The following table shows the general construction of the answer of the FI command DFJ1. The answer consists of a maximum of n=19 lines (n=19 maximum number of FI-Jobs), each of two columns.

Line 1...n:	Column 1	Column 2

Value Range/Meaning of Columns

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

2 = FI Command

Example DFJ1 Delete all FI-Jobs.

Assumption:

An NC program has been successfully compiled using the FI command “NPC” and has then been transmitted to the device (control) using the “NPD” command. (refer to “FI Commands of the MTCX Device Group”).

- Job ID of the NC compiler program: 01
- Job ID of the NC download program: 02

FI Command		XX_BR_DFJ1
Line	Column	Answer
1	1	01
	2	02_BR_NPC1_1 /3
2	1	02
	2	02_BR_NPD1_1_1 /3

FI Command Removes the selected FI-job from the administration structure of the function interface.

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

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

Construction of Answer The following table shows the general construction of the answer of the FI command DFJ2. The answer consists of a line with 13 columns.

Line 1	Column 1	Column 2

Value Range/Meaning of Columns

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

2 = FI Command [String, in accordance to chapter 6.1, Elements of the FI Command]

Example DFJ2 Deletes the FI-Job 01.

Assumption:

An NC program has been successfully compiled using the FI command “NPC” and has then been transmitted to the device (control) using the “NPD” command. (refer to “FI Commands of the MTCX Device Group”)

- Job ID of the NC compiler program: 01
- Job ID of the NC download program: 02

FI Command		XX_BR_DFJ2_01
Line	Column	Answer
1	1	01
	2	02_BR_NPC1_1 /3

Error Information: ERI

MPCX Device Group

Name **ERI** **ER**ror **I**nformation

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

FI Command Read error text and additional text.

BR_ERI1_(1)_(2) **(Single Read)**

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

(2) = error number [LONG]

Construction of Answer The following table shows the general construction of the answer of the FI command ERI. 2 lines, each with one column, are outputted. Line 1 contains the error text and line 2 contains the additional text.

Lines 1..2	Column 1

Meaning of the Column

1 = error text [language-dependent]

2 = additional text [language-dependent]

Example ERI Read the error text including the additional error text with error number 26.

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

Far Configuration Parameters: FCP

MPCX Device Group

Name **FCP** **F**ar Device **C**onfiguration **P**arameter

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

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

Case A PC is in PC Network The list of the FarDevices defined in the network configuration data, is outputted on the PC (see file "FAR_DEV.INI"). Furthermore, the local devices are outputted that are not defined as FarDevices.

Case B PC stand-alone The list of local devices is outputted if one or more of the following points apply:

- There is no network configuration data on the PC (see file "FAR_DEV.INI").
- 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 Reading out the addressable devices on the PC.

XX_BR_FCP1{_(1)) **(Single Read)**

(1) = device selection [L= only local, F= only FAR] ! Optional !

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

XX_BR_FCP2_(1){_(2)} **(Single Read)**

(1) = device group	[MPCX, MTCX, MISC, MTAX, MTRX]
(2) = device selection	[L= only local, F= only FAR] ! Optional !

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

XX_BR_FCP3_(1){_(2)} **(Single Read)**

(1)= device type	[MTCNC, MTC200-P, MTC200-R, MTVNC, SERCANS-A, SERCANS-P, ISP200-P, ISP200-R, TRA200-P, TRA200-R, MTA200-P]
(2) = device selection	[L= only local, F= only FAR] ! Optional !

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

Result when network configuration data is available:

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

Value Range/Meaning of the Columns

1 = FarDevice address	[00...15]
2 = Device name	max. 28 ASCII characters
3 = Device type	[MTCNC, MTC200-P, MTC200-R, MTVNC, SERCANS-A, SERCANS-P, ISP200-P, ISP200-R, TRA200-P, TRA200-R, MTA200-P]
4 = Local device address	[00...15]
5 = PC No.	[00...15, XX]
6 = Local device	[YES, NO, --]
7 = Device status	ON, OFF
8 =Assignment of a	[00...15, NO]

simulation pair.

9 = Device group [MPCX, MTCX, MISC, ...]

10 = Online [YES, NO, --]

Explanation of column 1
FarDevice Address

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

Explanation of Column 7
Device Status

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

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

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

Note: If a PC is disabled then its corresponding devices are also in the “Disable” condition.

Explanation of Column 10
Online?

This column indicates whether there is currently a connection to the PC via which the device can be addressed. A 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: In case of B, YES is always outputted.

Example FCP1
Case A

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

<u>Assumption:</u> The following device types have been defined:

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

FI Command		**XX_BR_FCP1**
Line	**Column**	**Answer**
1	1	15
	2	Drill left
	3	MTCNC
	4	05
	5	02
	6	YES
	7	ON
	8	11
	9	MTCX
	10	YES

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

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

Example FCP1 Case B

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

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

- Device address 05: MTCNC
- Device address 01: MTVNC

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

FI Command		XX_BR_FCP1
Line	Column	Answer
1	1	05
	2	Drill left
	3	MTCNC
	4	05
	5	XX
	6	YES
	7	ON
	8	NO
	9	MTCX
	10	YES
2	1	01
	2	Drill left
	3	MTVNC
	4	01
	5	XX
	6	YES
	7	ON
	8	NO
	9	MTCX
	10	YES

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

Far Device Configuration Parameters: FDC

MPCX Device Group

Name **FDC** **F**ar **D**evice **C**onfiguration

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

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

Case A PC is in 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 file "FAR_DEV.INI").
- 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)**

Construction of Answer The following table shows the general construction of the answer of the FI command FDC1.

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

Value Range/Meaning of Columns

Line 1

1 = PC network exists?	[YES, NO]
2 = Name of the PC network	max. 28 ASCII characters
3 = max. Number of PCs	(Integer)
4 = max. Number of devices	(integer)

Line 2

1 = PC No.	[00...15, XX]

Line 3

1 = Host name/ Ethernet host name possibly expanded by name of domain	(string)

Line 4

1 = Computer name/ NETBIOS name of computer	(string)

Line 5

1 = IP address of network card 1	(string)
...	...
4 = IP address of network card 4	(string)

Line 6

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

Example FDC1 Case A

Read the general data of the PC network.

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

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

FI Command		**XX_BR_FDC1**
Line	**Column**	**Answer**
1	1	YES
	2	Operation10
	3	20
	4	64
2	1	XX
3	1	Machine1.line1
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 within the network or has been defined within it.

FI Command		**XX_BR_FDC1**
Line	**Colum n**	**Answer**
1	1	NO
	2	--
	3	1
	4	16
2	1	XX
3	1	Machine1.line1
4	1	MACHINE1
5	1	172.16.0.1
6	1	--

Further Info Text: FIT

MPCX Device Group

Name **FIT** Further Info Text

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

FI Command Read additional (further) text.

BR_FIT1_(1)_(2) **(Single Read)**

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

(2) = error number [LONG]

Construction of Answer One line with one column is outputted for the additional text.

Line	Column

Meaning of the Column

Additional Text [language-dependent]

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

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

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

Far PC Configuration Parameters: FPC

MPCX Device Group

Name **FPC** **F**ar **P**C **C**onfiguration Parameter

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

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

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

Construction of Answer The following table shows the general construction of the answer of the FI command “FPC1”. The number of lines depends on the actual configuration. Result when network configuration data is available:

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

Value Range/Meaning of the Columns

1 = PC No.	[00...15, XX]
2 = Port	[IP address, hostname]
3 = Name of PC	max. 28 ASCII characters
4 = Local device	[YES = PC is the local PC, NO = PC is a remote PC]
5 = Device status	[OFF = PC is disabled, ON = PC is enabled] corresponds to the "Disable" entry of section "PC<pcnr>"
6 = Master?	[YES = PC is MasterPC (Head-PC), NO] corresponds to the "MasterPC" entry of section "PC<pcnr>"
7 = Online	[YES, NO, --]

Explanation of Column 7 Online? This column indicates whether there is currently a connection to the PC via which the device can be addressed. A 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: In case of B, YES is always outputted.

Example FPC1 Case A Read the list of PCs that are defined in the function interface. Assumption: Two PCs are defined:

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

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

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

Example FPC1 Case B

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

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

Information regarding Function Interface Jobs: IFJ

MPCX Device Group

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

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

Note: Refer here to “NC Download” in chapter 7.2 FI Commands for the MTCX Device Group.

FI Command Returns status information on all active FI-Jobs.

XX_BR_IFJ1 **(Single Read)**

Construction of Answer The following table shows the general construction of the answer of the FI command IFJ1 . The answer consists of a maximum of n=19 lines (n=19 maximum number of FI-Jobs), each with 16 columns.

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

Value Range/Meaning of Columns

1 =	Job ID	[01...20]
2 =	FI Command	[String, in accordance to chapter 6.1, Elements of the FI Command]
3 =	Job classification	[1 = NC-Download, 2 = compile NC program package]
4 =	Job-Status	[RUN, READY, ERROR]
5 =	Number of error lines in the error info buffer	
6 =	Max. Processing time [ms] until TIMEOUT	
7 =	Start time of the job	[hh:mm:ss:ms]
8 =	Processing time up to now in ms	
9 =	Function interface connection (login) name of the application	
10 =	Progress type	[1 = details of progress in %, 2 = details of absolute progress]
11 =	Details of progress as percentage value	[Value, --], depends on column 10 "Progress type"
12 =	Details of 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 8 Error Codes)
16 =	Error info buffer	

Note: The results of the column depend on the FI-Job that has been started. Refer here to “NC Download” in chapter 7.2 FI Commands for the MTCX Device Group.

Example IFJ1 Read the status information of all active FI-Jobs.

<u>Assumption:</u>

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

FI Command		XX_BR_IFJ1
Line	**Column**	**Answer**
1	1	01
	2	02_BR_NPC1_1 /3
	3	2
	4	READY
	5	0
	6	600000
	7	16:15:22:123
	8	120
	9	VBDemo
	10	1
	11	100 %
	12	--
	13	--
	14	--
	15	0
	16	--

FI Command

Returns information regarding the selected and active FI-Job.

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

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

Note: Information regarding the construction of the answer can be gathered from the FI-Command "XX_BR_IFJ1" described in the above.

Activated Language of the Rexroth Indramat GUI: LNG

MPCX Device Group

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

Explanation The country code of the activated language of the Rexroth Indramat GUI is outputted.

FI Command

XX_BR_LNG **(Single Read)**

Construction of Answer The answer of the FI-Command LNG consists of one line with one column for the country code of the activated language.

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

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

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

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

7.1 FI Commands for the MTCX Device Group

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

Group	Accompanying Types	Address
MTCX	MTCNC, MTC200-P, MTC200-R, MTVNC	[00...15]

Active Acceleration Value: AAC

MTCX Device Group

Name **AAC** **A**ctive **AC**celeration

Explanation The current acceleration value of a CNC 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, e.g. the axes of the workpiece carrier is to be moved depending on the weight of the workpiece.

FI Command Output of the active acceleration value of a CNC process of the selected device from the MTCX device group.

CR_AAC1_(1) **(Single Read)**

CC_AAC1_(1) **(Cyclic Read)**

CB_AAC1_(1) **(Break Cyclic Read)**

(1) = CNC process number [0...6]

Construction of Answer The following table shows the general construction of the answer of the FI command AAC. One line with three columns is outputted for the NC command, the acceleration value and the unit.

Line 1	Column 1	....	Column 3

Value Range of the Columns

1 = NC command [ACC]

2 = Acceleration value [0...100]

3 = Unit [%]

Example AAC1 Read the active acceleration value in CNC 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 You can find more details regarding the acceleration value and the value range in the Rexroth Indramat documentation:

NC Programming Instructions, chapter „Interpolation requirements/ Programmable Acceleration ACC",
DOK-MTC200-NC**PRO*Vxx-AW0x-EN

Active Angle Dimension (RAD/DEG): AAD

MTCX Device Group

Name **AAD** **A**ctive **A**ngle **D**imension

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

FI Command Output of the active acceleration value of a CNC process of the selected device from the MTCX device group.

CR_AAD_(1) **(Single Read)**

CC_AAD_(1) **(Cyclic Read)**

CB_AAD_(1) **(Break Cyclic Read)**

(1) = CNC process number [0...6]

Construction of Answer The answer of the FI-Command AAD consists of one line with one column for the unit [RAD/DEG].

Example AAD Read the active angle dimension in CNC process 0 of device address 00.

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

Reference to Literature You can find more details regarding the arguments of the trigonometric functions in the Rexroth Indramat documentation:

NC Programming Instructions, chapter „Angle Dimension for Trigonometric Functions RAD, DEG“, DOK-MTC200-NC**PRO*Vxx-AW0x-EN

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

MTCX Device Group

Name

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

Explanation

The current axis speeds and spindle speeds of a CNC process of the selected device are read out. The FI command "AAS1" refers to the CNC process number and to the source of the axis meaning, whereby the FI command “AAS2” allows the output of the current speed related to the physical axis number.

FI Command

Output of the current axis speed related to the CNC process number and to the code of the axis meaning.

CR_AAS1_(1)_(2) **(Single Read)**

CC_AAS1_(1)_(2) **(Cyclic Read)**

CB_AAS1_(1)_(2) **(Break Cyclic Read)**

(1) = CNC process number [0...6]

(2) = Axis meaning [0...11; 20] (see chapter 6.2, Data Tables)

Construction of Answer

The following table shows the general construction of the answer of the FI command AAS. One line with three columns is outputted for the name of the axis, the axis speed and the unit.

Line 1	Column 1	...	Column 3

Value Range/Meaning of Columns

1 = Axis name [acc. to settings of axis parameter]

2 = Speed [acc. to settings of axis parameter]

3 = unit [acc. to settings of axis parameter]

Note: If the specified axis is not defined in the selected CNC process then the answer in all columns is [--].

Example AAS1

Read the current axis speed of the Z axis in CNC process of device address 00.

FI Command	00_CR_AAS1_0_2		
Answer			
Line	**Column 1**	**Column 2**	**Column 3**
1	Z	-158.2	[mm/min]

Reference to Literature

You can find more details regarding the axis speeds in the Rexroth Indramat documentation:

NC Programming Instructions, chapter „Interpolation Functions/ Straight Line Interpolation, Quick Motion G00“, DOK-MTC200-NC**PRO*Vxx-AW0x-EN

Parameter Description, chapter „Maximum Track Acceleration“, DOK-MT*CNC-PAR*DES*Vxx-AW0x-EN

FI Command

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

CR_AAS2_(1) **(Single Read)**

CC_AAS2_(1) **(Cyclic Read)**

CB_AAS2_(1) **(Break Cyclic Read)**

(1) = Physical axis number [1...32, acc. to settings of the system parameters]

Construction of Answer

The following table shows the general construction of the answer of the FI command AAS2 . One line with three columns is outputted for the name of the axis, the axis speed and the unit.

Line 1	Column 1	...	Column 3

Value Range/Meaning of Columns

1 = Axis name [acc. to settings of axis parameter]

2 = Speed [acc. to settings of axis parameter]

3 = Unit [acc. to settings of the process parameter]

Note: If the specified axis is not defined in the CNC process then the answer in all columns is [--].

Example AAS2

Read the current speed of spindle S (physical axis number 4) of device address 00.

FI Command	**00_CR_AAS2_4**		
Answer			
Line	**Column 1**	**Column 2**	**Column 3**
1	S	4000.0	[1/min]

Reference to Literature

You can find more details regarding the axis speeds in the Rexroth Indramat documentation:

Parameter Description, chapter „Maximal Track Speed“, DOK-MT*CNC-PAR*DES*Vxx-AW0x-EN

Active NC Block: ABI

MTCX Device Group

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

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

FI Command Output of the active NC record as well as the previous and following NC records of a CNC process of the selected device from the MTCX 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) = CNC process number [0...6]

(2) = Number of previous NC Records [1..4] ! Optional !

(3) = Number of following NC Records [1..4] ! Optional !

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

Construction of Answer The number of lines (1...n = 9) of the answer depends on the number of NC records requested. Each line consists of a column containing the respective NC record.

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

Example ABI Read the active NC record and the previous and two following NC records of the CNC process 0 of device class 00.

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

Reference to Literature You can find more details regarding the construction of an NC record in the Rexroth Indramat documentation:

NC Programming Instructions, chapter „Elements of an NC Record“, DOK-MTC200-NC**PRO*Vxx-AW0x-EN

Active Cutting Speed of the Reference Spindle: ACS

MTCX Device Group

Name **ACS** **A**ctive **C**utting **S**peed

Explanation Output of the active cutting speed of the reference spindle of a CNC process of the selected device from the MTCX device group.

FI Command

CR_ACS_(1) **(Single Read)**

CC_ACS_(1) **(Cyclic Read)**

CB_ACS_(1) **(Break Cyclic Read)**

(1) = CNC process number [0...6]

Construction of Answer The following table shows the general construction of the answer of the FI command ACS. One line with three columns is outputted for the S number of the reference spindle, the cutting speed and the according to the settings of the system parameters.

Line 1	Column 1	...	Column 3

Value Range/Meaning of Columns

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

Note: If no reference spindle is defined in the selected CNC process then the value of column 1 is [*S]; columns 2 and 3 receive the value [--].

Example ACS Read the active cutting speed in CNC process 0 of device address 00.

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

Reference to Literature Additional information regarding the reference spindle as well as the NC programming of the cutting speed is contained in the Rexroth Indramat documentation:

NC Programming Instructions, chapter „Spindle Speed, Constant Cutting Speed G96/Selection of Reference Spindle SPF“, DOK-MTC200-NC**PRO*Vxx-AW0x-EN

Active D-Correction Number: ADN

MTCX Device Group

Name **ADN** **A**ctive **D-**Correction **N**umber

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

FI Command Output of the active D-correction number of a CNC process of the selected device from the MTCX device group.

FI Command

CR_ADN1_(1) **(Single Read)**

CC_ADN1_(1) **(Cyclic Read)**

CB_ADN1_(1) **(Break Cyclic Read)**

(1) = CNC process number [0...6]

Construction of Answer One line with two columns is outputted for the active D-correction number of the indicated CNC process. The following hereby mean:

1 = Identifier D

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

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

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

Reference to Literature You can find more details regarding D-correction in the Rexroth Indramat documentation:

“NC Programming Instructions Vxx", chapter „D-Correction“, DOK-MTC200-NC**PRO*Vxx-AW0x-EN

Active Event Monitoring: AEM

MTCX Device Group

Name **AEM** **A**ctive **E**vent **M**onitoring

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

FI Command Output of the status of the event monitoring of a CNC process of the selected device from the MTCX device group.

CR_AEM_(1) **(Single Read)**

CC_AEM_(1) **(Cyclic Read)**

CB_AEM_(1) **(Break Cyclic Read)**

(1) = CNC process number [0...6]

Construction of Answer One line and one column are outputted for the status of the event monitoring. The following hereby mean:

- EEV = activation of event monitoring
- DEV = suppression of event monitoring

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

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

Reference to Literature You can find more details regarding events and their treatment in the Rexroth Indramat documentation:

“NC Programming Instructions Vxx", chapter „Events“,
DOK-MTC200-NC**PRO*Vxx-AW0x-EN

Active Edge Number: AEN

MTCX Device Group

Name **AEN** **A**ctive **E**dge-**N**umber

Explanation The active edge number of a CNC process is outputted. Changing the active cutter in the NC program results in the provision of the corresponding correction and tool life data which the tool management then accesses during subsequent processing.

FI Command Output of the active edge number of a CNC process of the selected device from the MTCX device group.

CR_AEN_(1) **(Single Read)**

CC_AEN_(1) **(Cyclic Read)**

CB_AEN_(1) **(Break Cyclic Read)**

(1) = CNC process number [0...6]

Construction of Answer One line with two columns is outputted for the identifier "E = Edge" and for the active edge number. The active cutter corresponds thereby to the single-digit decimal number [1...9], that is assigned the address letter "E".

Example AEN Read the active edge number of CNC process 0 of device address 00.

FI Command	00_CR_AEN_0	
Answer		
Line	**Column 1**	**Column 2**
1	E	1

Reference to Literature You can find more details regarding tool administration in the Rexroth Indramat documentation:

“NC Programming Instructions Vxx", chapter „Commands for Tool Administration/Cutter Selection E“,
DOK-MTC200-NC**PRO*Vxx-AW0x-EN

Active Feedrate Override: AFO

MTCX Device Group

Name

AFO **A**ctive **F**eedrate **O**verride

Explanation

The current value of the feedrate override of a CNC process is outputted. The override is evaluated in the NC independent of he operating mode and is effective for all axis movements (except for a reference run of the digital axes).

FI Command

Output of the current value of the feedrate override of a CNC process of the selected device from the MTCX device group.

CR_AFO1_(1) **(Single Read)**

CC_AFO1_(1) **(Cyclic Read)**

CB_AFO1_(1) **(Break Cyclic Read)**

(1) = CNC process number [0...6]

Construction of Answer

The following table shows the general construction of the answer of the FI command AFO . One line with three columns is outputted for the identifier, the current value of the feedrate override and the unit [%].

Line 1	Column 1	....	Column 3

Value Range/Meaning of Columns

1 = Identifier [OVR=Override]

2 = Current value of the feedrate override [0...255]

3 = Unit [%]

Note: The permissible range of the override evaluation by the SPS program is between 0 and 255 %. The NC limits the axis and/or processor speed to the maximal values set in the parameters if an override value is set that is too large.

Example AFO1

Read the current value of the feedrate override in CNC 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

You can find more details regarding feedrate override in the Rexroth Indramat documentation:

"CNC/SPS Interface Description xxVRS", chapter "Feedrate and Spindle Override PxxCSOVRD"
DOK-MTC200-SPS*GWY*Vxx-AW0x-EN

Actual (Current) Feedrate: AFR

MTCX Device Group

Name **AFR** **A**ctual **F**eed**R**ate

Explanation The current value of the feedrate of a CNC process is outputted. The details of the feedrate in an NC program is expressed by a feedrate word with address letter "F" and a feedrate that is specified directly as constant or via an expression.

FI Command Output of the current value of the feedrate of a CNC process of the selected device from the MTCX device group.

CR_AFR_(1) **(Single Read)**

CC_AFR_(1) **(Cyclic Read)**

CB_AFR_(1) **(Break Cyclic Read)**

(1) = CNC process number [0...6]

Construction of Answer The following table shows the general construction of the answer of the FI command AFR. One line with three columns is outputted for the identifier, the current value of the feedrate and the unit [%].

Line 1	Column 1	...	Column 3

Value Range/Meaning of Columns

1 = Identifier [F = Feedrate]

2 = Value [format acc. to settings of the parameters]

3 = Unit [acc. to settings of the process parameter]

Example AFR Read the current feedrate in CNC process 0 of device address 00.

FI Command	00_CR_AFR_0		
Answer			
Line	**Column 1**	**Column 2**	**Column 3**
1	F	5000.0	[mm/min]

Reference to Literature You can find more details regarding the feedrate in the Rexroth Indramat documentation:

“CNC NC Programming Instructions Vxx", chapter „Feedrate“, DOK-MTC200-NC**PRO*Vxx-AW0x-EN

Active G Functions: AGF

MTCX Device Group

Name **AGF** **A**ctive **G** **F**unction

Explanation Read-out of the active G functions of a CNC process of the selected device from the MTCX device group.

FI Command

CR_AGF_(1){_(2)} **(Single Read)**

CC_AGF_(1){_(2)} **(Cyclic Read)**

CB_AGF_(1){_(2)} **(Break Cyclic Read)**

(1) = CNC process number [0...6]

(2) = G-code group [1...21] ! Optional !

Note: If the optional parameter is not specified, then all active G codes of all G-code groups are outputted.

Construction of Answer One line is outputted, whereby the number of columns depends on the number of G-code groups that are requested. When the optional parameter has not been specified, the answer consists of one line with 21 columns. If the optional parameter has been specified then the answer 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 answer consists of the characters [--].

Example AGF Read the active G function of G-code group 17 in CNC process 0 of device address 00.

FI Command	**00_CR_AGF_0_17**
Answer	
Line	**Column 1**
1	G30

Reference to Literature Additional 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 „Table of G-Code Groups“, DOK-MTC200-NC**PRO*Vxx-AW0x-EN

Active M Functions: AMF

MTCX Device Group

Name **AMF** **A**ctive **M** **F**unction

Explanation Read-out of the active M functions of a CNC process of the selected device from the MTCX device group.

FI Command

CR_AMF_(1){_(2)} **(Single Read)**

CC_AMF_(1){_(2)} **(Cyclic Read)**

CB_AMF_(1){_(2)} **(Break Cyclic Read)**

(1) = CNC process number [0...6]

(2) = M function group [1...16] ! Optional !

Note: If the optional parameter is not specified then all active M functions of all M function groups are outputted.

Construction of Answer One line is outputted, whereby the number of columns depends on the number of M function groups that are requested. When the optional parameter has not been specified, the answer consists of one line with 16 columns. If the optional parameter has been specified then the answer 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 CNC process 0 of device address 00.

FI Command	**00_CR_AMF_0_2**
Answer	
Line	**Column 1**
1	M005

Reference to Literature Additional 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 Instructions Vxx", chapter „Table of M Function Groups“, DOK-MTC200-NC**PRO*Vxx-AW0x-EN

Active Mechanism Messages: AMM

MTCX Device Group

Name **AMM** **A**ctive **M**echanism **M**essages

Explanation Messages regarding active mechanism errors and mechanism diagnostics are outputted. 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 additional text are all outputted.

FI Command Output of the currently pending mechanism messages of all active devices.

BR_AMM1 **(Single Read)**

BC_AMM1 **(Cyclic Read)**

BB_AMM1 **(Break Cyclic Read)**

Note: The AMM1 FI command refers to all devices within this device group. Therefore, any valid device address can be indicated in the command line (see Example AMM1).

Construction of Answer The following table shows the general construction of the answer of the FI command AMM1 . The answer consists of a maximum of n=512 lines (n=16 devices x 32 mechanisms = 512), each with 11 lines.

Line 1...n:	Column 1	...	Column 11

Value Range/Meaning of Columns

1 =	Device address	[00...15]
2 =	Device name	[max. 32 ASCII characters]
3 =	Mechanism number	[0...31]
4 =	Mechanism name	[max. 28 ASCII characters]
5 =	Type of message	[F = Fault/Error, D = Diagnosis]
6 =	Message source	[CNC, SPS]
7 =	Type of message (2)	[S = status, O = operator, E = external, I = internal]
8 =	Message number	[0...600]
9 =	Short text	[max. 54 ASCII characters]
10 =	Additional Text	[x= exists, -- = does not exist]
11 =	2 bytes of additional info for the message number	is required to resolve the information „@“ (see AMM5)

Reference to Literature Additional 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-EN

Example AMM1 Read the current mechanism messages of all active devices.
Assumption: the following device addresses and mechanisms are defined:

- Device address 01 with 2 mechanisms 0 and 1, and
- Device address 03 with one mechanism 0.

FI Command		03_BR_AMM1
Line	**Column**	**Answer**
1	1	01
	2	Drill center
	3	0
	4	Station 1
	5	D
	6	CNC
	7	S
	8	79
	9	Station waits until tool-change command has completed.
	10	x
	11	0
2	1	01
	2	Drill center
	3	1
	4	Station 2
	5	F
	6	CNC
	7	O
	8	1
	9	No external 24V supply.
	10	x
	11	0
3	1	03
	2	Milling center
	3	0
	4	Camshaft 30.40.25.0S
	5	D
	6	CNC
	7	S
	8	71
	9	Circular interpolation
	10	x
	11	0

FI Command

Output of the currently pending mechanism messages of the selected device.

BR_AMM2	**(Single Read)**
BC_AMM2	**(Cyclic Read)**
BB_AMM2	**(Break Cyclic Read)**

Construction of Answer

The following table shows the general construction of the answer of the FI command AMM2. The answer consists of up to a maximum of n=31 lines, each with 11 columns.

Line 1...n:	Column 1	...	Column 11

Value Range/Meaning of Columns

1 =	Device address	[00...15]
2 =	Device name	[32 ASCII characters]
3 =	Mechanism number	[0...31]
4 =	Mechanism name	[max. 28 ASCII characters]
5 =	Type of message	[F = Fault/Error, D = Diagnosis]
6 =	Message source	[CNC, SPS]
7 =	Type of message (2)	[S = status, O = operator, E = external, I = internal]
8 =	Message number	[0...600]
9 =	Short text	[max. 54 ASCII characters]
10 =	Additional Text	[x= exists, -- = does not exist]
11 =	2 bytes of additional info for the message number	is required to resolve the information „@“ (see AMM5)

Example AMM2

Read the current mechanism messages of device address 01.
<u>Assumption:</u> Device address 01 with 2 defined mechanisms 0 and 1.

FI Command		01_BR_AMM2
Line	**Column**	**Answer**
1	1	01
	2	Drill center
	3	0
	4	Station 1
	5	D
	6	CNC
	7	S
	8	79
	9	Station waits until tool-change command has completed.
	10	x
	11	0
2	1	01
	2	Drill center
	3	1
	4	Station 2
	5	F
	6	CNC
	7	O
	8	1
	9	No external 24V supply.
	10	x
	11	0

Reference to Literature

Additional 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-EN

FI Command

Output of the currently pending messages of the mechanisms listed for the selected device of the MTCX device group.

BR_AMM3_(1) **(Single Read)**

BC_AMM3_(1) **(Cyclic Read)**

BB_AMM3_(1) **(Break Cyclic Read)**

(1) = Selection list for a max. of 10 mechanisms [0_1_2_ ... _31]

Construction of Answer

The following table shows the general construction of the answer of the FI command AMM3. The number of lines (1 .. n=32) depends on the number of requested mechanism messages. Each line consists of again of 11 columns.

Line 1...n:	Column 1	...	Column 11

Value Range/Meaning of Columns

1 =	Device address	[00...15]
2 =	Device name	[max. 32 ASCII characters]
3 =	Mechanism number	[0...31]
4 =	Mechanism name	[max. 28 ASCII characters]
5 =	Type of message	[F = Fault/Error, D = Diagnosis]
6 =	Message source	[CNC, SPS]
7 =	Type of message (2)	[S = status, O = operator, E = external, I = internal]
8 =	Message number	[0...600]
9 =	Short text	[max. 54 ASCII characters]
10 =	Additional Text	[x= exists, -- = does not exist]
11 =	2 bytes of additional info for the message number	is required to resolve the information „@“ (see AMM5)

Reference to Literature

Additional 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-EN

Example AMM3

Read the current messages of mechanisms 0 and 1 of device address 01.

Assumption: Device address 01 with 2 defined mechanisms 0 and 1.

FI Command		01_BR_AMM3_0_1
Line	**Column**	**Answer**
1	1	01
	2	Drill center
	3	0
	4	Station 1
	5	D
	6	CNC
	7	S
	8	79
	9	Station waits until tool-change command has completed.
	10	x
	11	0
2	1	01
	2	Drill center
	3	1
	4	Station 2
	5	F
	6	CNC
	7	O
	8	1
	9	No external 24V supply.
	10	x
	11	0

FI Command

Output of the currently pending messages of the mechanisms listed for the devices of the MTCX device group.

BR_AMM4_(1) **(Single Read)**

BC_AMM4_(1) **(Cyclic Read)**

BB_AMM4_(1) **(Break Cyclic Read)**

(1) = Selection list for a max. of 10 mechanisms [Format: x.y]

Format x.y	Value range
X	Device address [00...15]
Y	Mechanism number [0...31]

Construction of Answer

The following table shows the general construction of the answer of the FI command AMM4. The number of lines (n=10 mechanisms maximum) depends on the number of requested mechanism messages. Each line consists of again of 11 columns.

Line 1...n:	Column 1	...	Column 11

Value Range/Meaning of Columns

1 =	Device address	[00...15]
2 =	Device name	[max. 32 ASCII characters]
3 =	Mechanism number	[0...31]
4 =	Mechanism name	[max. 28 ASCII characters]
5 =	Type of message	[F = Fault/Error, D = Diagnosis]
6 =	Message source	[CNC, SPS]
7 =	Type of message (2)	[S = status, O = operator, E = external, I = internal]
8 =	Message number	[0...600]
9 =	Short text	[max. 54 ASCII characters]
10 =	Additional Text	[x= exists, -- = does not exist]
11 =	2 bytes of additional info for the message number	is required to resolve the information „@“ (see AMM5)

Reference to Literature

Additional 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-EN

Example AMM4

Read the current messages of mechanisms 0 and 1 of device address 01 as well as the messages of mechanism 0 of device address 03.

<u>Assumption:</u> The following device addresses and mechanisms are defined:

- Device address 01 with 2 mechanisms 0 and 1, and
- Device address 03 with one mechanism 0.

FI Command		01_BR_AMM4_01.0_01.1_03.0
Line	**Column**	**Answer**
1	1	01
	2	Drill center
	3	0
	4	Station 1
	5	D
	6	CNC
	7	S
	8	79
	9	Station waits until tool-change command has completed.
	10	x
	11	0
2	1	01
	2	Drill center
	3	1
	4	Station 2
	5	F
	6	CNC
	7	O

	8	1
	9	No external 24V supply.
	10	x
	11	0
3	1	03
	2	Milling center
	3	0
	4	Camshaft 30.40.25.0S
	5	D
	6	CNC
	7	S
	8	71
	9	Circular interpolation
	10	x
	11	0

FI Command

Device and mechanism related output of the additional text for the selected message number for the devices of the MTC device group.

BR_AMM5_(1)_(2)_(3)	**(Single Read)**
(1) = Mechanism number	[0...31]
(2) = Message number	[0...600]

(3) = 2 bytes of additional info for the message number

Construction of Answer

The following table shows the general construction of the answer of the FI command AMM5. The number of lines n=512 lines (n=16 devices x 32 mechanisms = 512) depends on the number of requested mechanism messages. Each line consists here of 9 columns.

Line 1...n:	Column 1	...	Column 9

Value Range/Meaning of Columns

1 =	Device address	[00...15]
2 =	Device name	[max. 32 ASCII characters]
3 =	Mechanism number	[0...31]
4 =	Mechanism name	[max. 28 ASCII characters]
5 =	Type of message	[F = Fault/Error, D = Diagnosis]
6 =	Message source	[CNC, SPS]
7 =	Type of message (2)	[S = status, O = operator, E = external, I = internal]
8 =	Message number	[0...600]
9 =	Additional Text	[max. 14 lines with a max. 78 characters/line]

Example AMM5

Read the additional 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 until the active tool-change command has been completely processed.

Reference to Literature

Additional 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-EN

Active Machine-Parameter Index: API

MTCX Device Group

Name

API **A**ctive Machine-**P**arameter **I**ndex

Explanation

Information regarding the active machine-parameter records of all defined devices of the MTCX device group are outputted. The following are outputted: the device addresses, index, display of BOF/GBO, name, size, date and time of creation or the of the last change and details of the defined processes of the active machine-parameter record.

FI Command

BR_API1 (Single Read)

BC_API1 (Cyclic Read)

BB_API1 (Break Cyclic Read)

Note: The API1 FI command refers to all devices within this device group. Therefore, any valid device address can be indicated in the command line (see Example API1).

Construction of Answer

The following table shows the general construction of the answer of the FI command API1. The answer consists of up to a maximum of n=16 lines, each with 8 columns.

Line 1...n:	Column 1	...	Column 8

Value Range/Meaning of Columns

1 =	Device address	[00...15]
2 =	Index of active parameter record	[1...99]
3 =	Display of BOF/GBO	[max. 62 ASCII characters]
4 =	Name of parameter record	[max. 32 ASCII characters]
5 =	Size of parameter record [Byte]	[max. 7 ASCII characters]
6 =	Date of creation or of the last change in the parameter record	[8 ASCII characters in format:] [DD.MM.YY]
7 =	Time of creation or of the last change in the parameter record	[8 ASCII characters in format:] [HH:MM:SS]
8 =	Details of defined process	[max. 7 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 receives the device address and columns 2 to 8 the value [--].

Example API1

Read the information on the active machine-parameter records of all defined devices.
Assumption: The following device addresses of the MTCX device group have been defined:

- Device address 00: MTC200-P
- Device address 01: MTCNC and
- Device address 02: MTVNC

FI Command		01_BR_API1
Line	**Column**	**Answer**
1	1	00
	2	1
	3	00 MSD 0209-15 15625 28.01.99 13:29:10 M123456
	4	MSD 0209-15
	5	15625
	6	28.01.99
	7	13:29:10
	8	M123456
2	1	01
	2	88
	3	88 PCI 12.45.12.34 10584 11.11.98 11:11:11 M12
	4	PCI 12.45.12.34
	5	10584
	6	11.11.98
	7	11:11:11
	8	M12
3	1	02
	2	11

	3	11 Labor 5 DRV 24464 01.03.99 14:25:10 M13456
	4	Lab 5 DRV
	5	24464
	6	01.03.99
	7	14:25:10
	8	M13456

Reference to Literature

Additional information regarding the machine parameters and their classification within the system, process, axis and APR-SERCOS parameters can be located in the Rexroth Indramat documentation:

"MTC200/MT-CNC MCI Operating Instructions xxVRS", chapter "Machine Parameters",
DOK-MTC200-GBO*MCI*Vxx-AW0x-EN

"Parameter Description",
DOK-MT*CNC-PAR*DES*Vxx-AW0x-EN

FI Command

BR_API2 **(Single Read)**

BC_API2 **(Cyclic Read)**

BB_API2 **(Break Cyclic Read)**

Construction of Answer

The following table shows the general construction of the answer of the FI command API2. The answer consists of a line with eight columns.

Line 1	Column 1	...	Column 8

Value Range/Meaning of Columns

1 =	Device address	[00...15]
2 =	Index of active parameter record	[1...99]
3 =	Display of BOF/GBO	[max. 62 ASCII characters]
4 =	Name of parameter record	[max. 32 ASCII characters]
5 =	Size of parameter record [Byte]	[max. 7 ASCII characters]
6 =	Date of creation or of the last change in the parameter record	[8 ASCII characters in format:] [DD.MM.YY]
7 =	Time of creation or of the last change in the parameter record	[8 ASCII characters in format:] [HH:MM:SS]
8 =	Details of defined process	[7 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 receives the device address and columns 2 to 8 the value [--].

Example API2

Read the information on the active machine-parameter record of device address 02. Assumption: The following device addresses of the MTCX device group have been defined:

- Device address 00: MTC200-P
- Device address 01: MTCNC and
- Device address 02: MTVNC

FI Command		00_BR_API2
Line	**Column**	**Answer**
1	1	00
	2	1
	3	00 MSD 0209-15 15625 28.01.99 13:29:10 M123456
	4	MSD 0209-15
	5	15625
	6	28.01.99
	7	13:29:10
	8	M123456

Reference to Literature

Additional information regarding the machine parameters and their classification within the system, process, axis and APR-SERCOS parameters can be located in the Rexroth Indramat documentation:

"Parameter Description",
DOK-MT*CNC-PAR*DES*Vxx-AW0x-EN

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

MTCX Device Group

Name

APM **A**ctive Part-**P**rogram **M**essage

Explanation

The active note of the NC record as well as the NC record number of a CNC process of the MTCX device group is outputted. Every NC record can contain a note that is displayed in the diagnostics menu of the Rexroth Indramat GUI after processing of the NC record. The note in the diagnostics line remains active until it is overwritten by a new note. (refer also to: Active Note in NC Program (only NC Record Number): APN

FI Command

CR_APM_(1) **(Single Read)**

CC_APM_(1) **(Cyclic Read)**

CB_APM_(1) **(Break Cyclic Read)**

(1) = CNC process number [0...6]

Construction of Answer

The following table shows the general construction of the answer of the FI command APM. One line with two columns is outputted for the NC record number and the NC note.

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 CNC process 0 of device address 00.

FI Command	00_CR_APM_0	
Answer		
Line	**Column 1**	**Column 2**
1	0002	Technological instructions

Reference to Literature

You can find more details regarding the elements of an NC record and the note in the Rexroth Indramat documentation:

“CNC NC Programming Instructions Vxx", chapter „NC Word“,
DOK-MTC200-NC**PRO*Vxx-AW0x-EN

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

MTCX Device Group

Name **APN** **A**ctive Part-**P**rogram Message-**N**umber

Explanation The NC record number of the active note of a CNC process of the MTCX device group is outputted. Every NC record can contain a note that is displayed in the diagnostics menu of the Rexroth Indramat GUI after processing of the NC record. The note in the diagnostics line remains active until it is overwritten by a new note. (refer also to: Active Note in NC Program (Note and NC Record Number): APM

FI Command

CR_APN_(1) **(Single Read)**

CC_APN_(1) **(Cyclic Read)**

CB_APN_(1) **(Break Cyclic Read)**

(1) = CNC process number [0...6]

Construction of Answer One line with one column is outputted for the NC record number of the active note.

Value Range/Meaning of Columns

1 = NC record number of the note [0000...9999]

Note: If the current NC program does not contain a note, then the result of column 1 is [0000].

Example APN Read the NC record number of the active note in CNC process 0 of device address 00.

FI Command	**00_CR_APN_0**
Answer	
Line	**Column 1**
1	0002

Reference to Literature You can find more details regarding the elements of an NC record and the note in the Rexroth Indramat documentation:

“CNC NC Programming Instructions Vxx", chapter „NC Word“,
DOK-MTC200-NC**PRO*Vxx-AW0x-EN

Current (Actual) Position of an Axis: APO

MTCX Device Group

Name

APO **A**ctual Axis **PO**sition

Explanation

The current (actual) position of a selected axis of the MTCX device group 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 of the position of the selected axis of the device specified, related to the code of the axis meaning.

CR_APO1_(1)_(2)_(3) **(Single Read)**

CC_APO1_(1)_(2)_(3) **(Cyclic Read)**

CB_APO1_(1)_(2)_(3) **(Break Cyclic Read)**

(1) = CNC process number [0...6]

(2) = Axis meaning [0...11; 20];

(3) = Coordinate system [1 = Machine coordinates
2 = Program coordinates]

Construction of Answer

The following table shows the general construction of the answer of the FI command APO1. One line with three columns for the name of the axis, the position and the unit is outputted in accordance to the settings of the process parameters.

Line 1	Column 1	...	Column 3

Value Range/Meaning of Columns

1 = Axis name [acc. to settings of axis parameter]

2 = Position [acc. to settings of the process parameter]

3 = unit [acc. to settings of axis process parameter] mm, inch]

Example APO1

Read the current position of the Z axis in machine coordinates in CNC process 0 of device address 00.

FI Command	**00_CR_APO1_0_2_1**		
Answer			
Line	**Column 1**	**Column 2**	**Column 3**
1	Z	-5.9897	[mm]

FI Command

Output of the position of the selected axis of the device specified, related to the physical axis number.

CR_APO2_(1)_(2) **(Single Read)**

CC_APO2_(1)_(2) **(Cyclic Read)**

CB_APO2_(1)_(2) **(Break Cyclic Read)**

(1) = Physical axis number [1...32, acc. to settings of the system parameters]

(2) = Coordinate system [1 = Machine coordinates
2 = Program coordinates]

Construction of Answer

The following table shows the general construction of the answer of the FI command APO2 . One line with three columns for the name of the axis, the position and the unit is outputted in accordance to the settings of the process parameters.

Line 1	Column 1	...	Column 3

Value Range/Meaning of Columns

1 = Axis name [acc. to settings of axis parameter]

2 = Position [acc. to settings of the process parameter]

3 = unit [acc. to settings of axis process parameter] mm, inch]

Note: If the specified axis is not defined in the selected CNC process then the answer in all columns is [--].

Example APO2

Read the current position of the Z axis (physical axis number = 3) in machine coordinates at the device address 00.

FI Command	00_CR_APO2_3_1		
Answer			
Line	Column 1	Column 2	Column 3
1	Z	-5.9897	[mm]

Reference to Literature

Additional 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-EN

Active NC Program Number APP

MTCX Device Group

Name **APP** **A**ctive **P**art-**P**rogram number

Explanation The active NC program number of a CNC process is read out.

FI Command

CR_APP_(1) **(Single Read)**

CC_APP_(1) **(Cyclic Read)**

CB_APP_(1) **(Break Cyclic Read)**

Construction of Answer

(1) = CNC process number [0...6]

The following table shows the general construction of the answer of the FI command APP. One line with 2 columns is outputted for the NC memory and the NC program number.

Line 1	Column 1	Column 2

Value Range/Meaning of Columns

1 = NC memory [A = memory A; B = memory B]

2 = program number [01...99]

Example APP Read the active NC program number in CNC process 0.

FI Command	00_CR_APP_0	
Answer		
Line	**Column 1**	**Column 2**
1	A	01

Reference to Literature You can find more details regarding the construction of the NC data structure in the Rexroth Indramat documentation:

“NC Programming Instructions Vxx", chapter „Program and Data Organization“,
DOK-MTC200-NC**PRO*Vxx-AW0x-EN

Current (Actual) Rapid Override: ARO

MTCX Device Group

Name **ARO** **A**ctual **R**apid **O**verride

Explanation The current value of the rapid override of a CNC process of the MTCX device group is outputted. This value is evaluated by the NC for all axis movements that are executed with “G00”. The permissible range of the override evaluation by the SPS program is between 0 and 255 %.

FI Command Output of the current value of the rapid override of a CNC process of the selected device from the MTCX device group.

CR_ARO1_(1) **(Single Read)**

CC_ARO1_(1) **(Cyclic Read)**

CB_ARO1_(1) **(Break Cyclic Read)**

(1) = CNC process number [0...6]

Construction of Answer The following table shows the general construction of the answer of the FI command ARO. One line with three columns is outputted for the identifier, the current value of the rapid override and the unit [%].

Line 1	Column 1	...	Column 3

Value Range/Meaning of Columns

1 = Identifier [ROV= rapid override]

2 = Current value of the rapid override [0...255]

3 = Unit [%]

Note: The permissible range of the override evaluation by the SPS program is between 0 and 255 %. The NC limits the axis and/or processor speed to the maximal values set in the parameters if an override value is set that is too large.

Example ARO1 Read the current value of the rapid override in CNC process 0 of device address 00.

FI Command	00_CR_AFO1_0		
Answer			
Line	**Column 1**	**Column 2**	**Column 3**
1	ROV	100	[%]

Reference to Literature You can find more details regarding rapid override in the Rexroth Indramat documentation:

"CNC/SPS Interface Description xxVRS”, chapter "Feedrate and Spindle Override”, “Rapid Override PxxCSOVRD"
DOK-MTC200-SPS*GWY*Vxx-AW0x-EN

Active Spindle for Process: ASF

MTCX Device Group

Name **ASF** **A**ctive **S**pindle **F**or Process

Explanation The active (selected) spindle of the selected CNC process is outputted. As there can be several spindles in a CNC process, it is necessary for certain NC functions such as, e.g. G96 (constant cutting speed) that these are active on another spindle as well as on the first spindle. The following NC functions are dependent on the selected main spindle:

- G33 thread cutting
- G63/G64 tapping
- G65 tapping; spindle serves as leading axis
- G95 feed per turn and
- G96 constant cutting speed.

FI Command

CR_ASF_(1) **(Single Read)**

CC_ASF_(1) **(Cyclic Read)**

CB_ASF_(1) **(Break Cyclic Read)**

(1) = CNC process number [0...6]

Construction of Answer The answer of the FI-Command ASF consists of one line with one column for the selected active spindle.

Active Spindle for Process: [S1, S2, S3, *S]

Note: If no active spindle is selected in the CNC process, then the answer of column 1 is [*S].

Example ASF Read the selected active spindle in CNC process 0 of device address 00.
Assumption:

- a main circular-axis spindle (S1) has been defined in CNC 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**
Answer	
Line	**Column 1**
1	S1

Reference to Literature Additional information regarding the selection of the active spindle in the NC program is contained in the Rexroth Indramat documentation:

"CNC NC Programming Instructions Vxx", Application Description, chapter „Spindle Speed“, “Selecting the Active Spindle SPF”
DOK-MTC200-NC**PRO*Vxx-AW0x-EN

Current (Actual) Spindle Gear: ASG

MTCX Device Group

Name **ASG** **A**ctual **S**pindle **G**ear

Explanation Output of the current spindle gear of a CNC process of the selected device from the MTCX device group. The control signals of the gear selection are only evaluated by the CNC when one gear with at least two gear levels has been entered within the axis parameter.

FI Command

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

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

CB_ASG_(1)_(2) **(Break Cyclic Read)**

(1) = CNC process number [0...6]

(2) = spindle number [1...3]

Construction of Answer The answer of the ASG FI command consists of one lien with two columns for the identifier and for the current spindle gear level of the selected CNC process.

Line 1	Column 1	Column 2

Value Range/Meaning of Columns

1 = Identifier [g = gear]

2 = Current (Actual) Spindle Gear: [1...3, -]

Note: If no current spindle gear level is selected in the CNC process or in the NC program then column 1 receives the value [g] and column 2 the value [-].

Example ASG Read the current spindle gear of spindle number 1 in CNC process 0 of device address 00.

FI Command	00_CR_ASG_0_1	
Answer		
Line	**Column 1**	**Column 2**
1	g	1

Reference to Literature Additional information regarding the selection of the spindle gear in the NC program is contained in the Rexroth Indramat documentation:

"NC Programming Instructions Vxx", chapter „Additional Functions M" / "Switching Gear",
DOK-MTC200-NC**PRO*Vxx-AW0x-EN

Active System Error Messages: ASM

MTCX Device Group

Name

ASM **A**ctive **S**ystem **M**essages

Explanation

The active system error messages that effect the functioning of the entire electrical device are outputted depending on the FI command, the device address, device name, message number, type of message, short text and additional text are all outputted.

FI Command

Output of all existing current system error messages pending of all active devices from the MTCX device group.

BR_ASM1 **(Single Read)**

BC_ASM1 **(Cyclic Read)**

BB_ASM1 **(Break Cyclic Read)**

Note: The ASM1 FI command refers to all devices within this device group. Therefore, any valid device address can be indicated in the command line (see Example ASM1).

Construction of Answer

The following table shows the general construction of the answer of the FI command ASM1 . The number of lines (1 .. n=15) depends on the number of defined devices. Every 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 1...n:	Column 1	...	Column 7

Value Range/Meaning of Columns

1 =	Device address	[00...15]
2 =	Device name	[max. 32 ASCII characters]
3 =	Message number	[0...150]
4 =	Type of message	[F = Fault/Error, D = Diagnosis]
5 =	Short text	[max. 54 ASCII characters]
6 =	Additional Text	[x= exists, -- = does not exist]
7 =	2 bytes of additional info for the message number	is required to resolve the information „@“ (see ASM5)

Example ASM1

Read the current system error messages of all defined devices of the MTCX device group.

Assumption: the following three devices are defined:

- Device address 01:
- Device address 07 and
- Device address 10:

FI Command		07_BR_ASM1
Line	**Column**	**Answer**
1	1	01
	2	Drill center
	3	71
	4	F
	5	SPS battery voltage too low.
	6	X
	7	0
2	1	07
	2	Milling center 1
	3	74
	4	F
	5	SLM time monitoring
	6	X
	7	0
3	1	10
	2	Milling center 2
	3	1
	4	D
	5	Error has been corrected.
	6	X
	7	0

FI Command

Output of existing current system error message of the selected device from the MTCX device group.

BR_ASM2 **(Single Read)**

BC_ASM2 **(Cyclic Read)**

BB_ASM2 **(Break Cyclic Read)**

Construction of Answer

The following table shows the general construction of the answer of the FI command ASM2. The answer consists of a line of 7 columns for the device address, device name, message number, message status, short text and indication of whether there is an additional text for this error message.

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

Value Range/Meaning of Columns

1 =	Device address	[00...15]
2 =	Device name	[max. 32 ASCII characters]
3 =	Message number	[0...150]
4 =	Type of message	[F = Fault/Error, D = Diagnosis]
5 =	Short text	[max. 54 ASCII characters]
6 =	Additional Text	[x= exists, -- = does not exist]
7 =	2 bytes of additional info for the message number	is required to resolve the information „@“ (see ASM5)

Example ASM2

Read the current system error messages of device address 01.

Assumption: the following three devices are defined:

- Device address 01:
- Device address 07 and
- Device address 10:

FI Command		01_BR_ASM2
Line	**Column**	**Answer**
1	1	01
	2	Drill center
	3	71
	4	F
	5	SPS battery voltage too low.
	6	X
	7	0

FI Command

Output of all current system error messages of the device listed from the MTCX 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 MTCX devices [00_01_ ... _15]

Construction of Answer

The following table shows the general construction of the answer of the FI command ASM3 . The number of lines (1 .. n=15) depends on the number of defined listed MTCX devices. Every 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 1...n:	Column 1	...	Column 7

Value Range/Meaning of Columns

1 =	Device address	[00...15]
2 =	Device name	[max. 32 ASCII characters]
3 =	Message number	[0...150]
4 =	Type of message	[F = Fault/Error, D = Diagnosis]
5 =	Short text	[max. 54 ASCII characters]
6 =	Additional Text	[x= exists, -- = does not exist]
7 =	2 bytes of additional info for the message number	is required to resolve the information „@“ (see ASM5)

Example ASM3

Read the current system error messages of the selected MTCX devices. Assumption: The following device types have been 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	SPS battery voltage too low.
	6	X
	7	0
2	1	10
	2	Milling center 2
	3	1
	4	D
	5	Error has been corrected.
	6	X
	7	0

FI Command

Output of all current system error messages of all defined devices (in accordance to the system configuration) from the MTCX device group.

BR_ASM4_(1) **(Single Read)**

BC_ASM4_(1) **(Cyclic Read)**

BB_ASM4_(1) **(Break Cyclic Read)**

(1) = device group [MTCX, MISX]

Construction of Answer

The following table shows the general construction of the answer of the FI command ASM4. The number of lines (1 .. n=15) depends on the number of defined MTCX devices. Every 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 1...n:	Column 1	...	Column 7

Value Range/Meaning of Columns

1 =	Device address	[00...15]
2 =	Device name	[max. 32 ASCII characters]
3 =	Message number	[0...150]
4 =	Type of message	[F = Fault/Error, D = Diagnosis]
5 =	Short text	[max. 54 ASCII characters]
6 =	Additional Text	[x= exists, -- = does not exist]
7 =	2 bytes of additional info for the message number	is required to resolve the information „@“ (see ASM5)

Example ASM4

Read the current system error messages of all defined devices of the MTCX device group.
Assumption: The following device types have been defined:

- Device address 01 and
- Device address 10:

FI Command		01_BR_ASM4_MTCX
Line	**Column**	**Answer**
1	1	01
	2	Drill center

	3	71
	4	F
	5	SPS battery voltage too low.
	6	X
	7	0
2	1	10
	2	Milling center 2
	3	1
	4	D
	5	Error has been corrected.
	6	X
	7	0

FI Command

Output of the additional text for the currently existing error message, related to the device and the message number.

BR_ASM5_(1)_(2) **(Single Read)**

(1) = message number [0...150]

(2) = 2 bytes of additional info for the message number

Construction of Answer

The following table shows the general construction of the answer of the FI command ASM5. The answer consists of a line with 5 columns for device addresses, device names, message number and additional text.

Line 1...n:	Column 1	...	Column 5

Value Range/Meaning of Columns

1 =	Device address	[00...15]
2 =	Device name	[max. 32 ASCII characters]
3 =	Message number	[0...150]
4 =	Type of message	[F = Fault/Error, D = Diagnosis]
6 =	Additional Text	[max. 14 lines with a max. 78 characters/line]

Example ASM5

Read the additional text of the system error with message number 74 of device address 01.

FI Command		03_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 Additional 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-EN

Current (Actual) NC Sequence Number: ASN

MTCX Device Group

Name **ASN** **A**ctual **S**equence **N**umber

Explanation Output of the current NC sequence number of a CNC process of the selected device from the MTCX device group.

FI Command

CR_ASN_(1) **(Single Read)**

CC_ASN_(1) **(Cyclic Read)**

CB_ASN_(1) **(Break Cyclic Read)**

(1) = CNC process number [0...6]

Construction of Answer The answer of the FI-Command ASN consists of one line with one column for the active NC sequence number [N0000...N9999].

Note: If no valid NC program exists then column 1 receives the value [N0000].

Example ASN Read the active NC sequence number of CNC process 0 of device address 00.

FI Command	**00_CR_ASN_0**
Answer	
Line	**Column 1**
1	N0002

Reference to Literature You can find more details regarding the construction of an NC sequence in the Rexroth Indramat documentation:

“NC Programming Instructions Vxx", chapter „Elements of an NC Sequence“,
DOK-MTC200-NC**PRO*Vxx-AW0x-EN

Current (Actual) Spindle Override: ASO

MTCX Device Group

Name **ASO** **A**ctual **S**pindle **O**verride

Explanation The current value of the spindle override of a CNC process of the MTCX device group is outputted. The override is valid for all non-interpolated axes, i.e. for spindle axes and magazine axes. The override is evaluated in the NC independent of the operating mode and is effective for all axis movements (except for a reference run of the digital axes).

FI Command Output of the current value of the override of the selected device of the MTCX device group, related to the CNC process and the spindle number.

CR_ASO1_(1)_(2)	**(Single Read)**
CC_ASO1_(1)_(2)	**(Cyclic Read)**
CB_ASO1_(1)_(2)	**(Break Cyclic Read)**
(1) = CNC process number	[0...6]
(2) = spindle number	[1...3]

Construction of Answer The following table shows the general construction of the answer of the FI command ASO1. One line with three columns is outputted for the identifier, the current value of the override and the unit [%].

Line 1	Column 1	...	Column 3

Value Range/Meaning of Columns

1 = identifier	[S= Spindle]
2 = current value of the override with unit	[0...255]
3 = unit	[%]

Note: The permissible range of the override evaluation by the SPS program is between 0 and 255 %. The NC limits the axis and/or processor speed to the maximal values set in the parameters if an override value is set that is too large.

If the spindle number is not defined within the selected process then the result in column 1 is [--].

Example ASO1 Read the current value of the override in CNC process 0 of device address 00.

FI Command	**00_CR_ASO1_0_1**		
Answer			
Line	**Column 1**	**Column 2**	**Column 3**
1	S	60	[%]

Reference to Literature You can find more details regarding the spindle override in the Rexroth Indramat documentation:

"CNC/SPS Interface Description xxVRS”, chapter "Feedrate and Spindle Override PxxCSOVRD"
DOK-MTC200-SPS*GWY*Vxx-AW0x-EN

Current (Actual) Spindle Speed: ASS

MTCX Device Group

Name **ASS** **A**ctual **S**pindle **S**peed

Explanation Output of the current spindle speed (axis speed) of a CNC process of the selected device from the MTCX device group.

FI Command Output of the current axis speed of a CNC process, related to the spindle number.

CR_ASS_(1)_(2) **(Single Read)**
CC_ASS_(1)_(2) **(Cyclic Read)**
CB_ASS_(1)_(2) **(Break Cyclic Read)**
(1) = CNC process number [0...6]
(2) = Spindle number [1...3]

Construction of Answer The following table shows the general construction of the answer of the FI command ASS. One line with three columns for the name of the axis, the axis speed and the unit is outputted in accordance to the settings of the process parameters.

Line 1	Column 1	...	Column 3

Value Range/Meaning of Columns

1 = Axis name [S, S1, S2, S3]
2 = Spindle speed [acc. to settings of axis parameter]
3 = Unit [1/min; acc. to parameter settings]

Note: If the spindle number is not defined in the selected CNC process then the result in column 1 is [--], the result in column 2 is [0.0] and that of column 3 is [1/min].

Example ASS Read the current axis speed of the 1st spindle in CNC process 0 of device address 00.

FI Command	00_CR_ASS_0_1		
Answer			
Line	**Column 1**	**Column 2**	**Column 3**
1	S1	4000.0	[1/min]

Reference to Literature You can find more details regarding the axis speeds in the Rexroth Indramat documentation:

"MT-CNC Numeric Control for Multiple Axis, Multiple Process Applications", chapter "Maximum Track Acceleration", DOK-MT*CNC-PAR*DES*Vxx-AW0x-EN

Active Tool Number: ATN

MTCX Device Group

Name

ATN **A**ctive **T**ool **N**umber

Explanation

Read-out of the active tool number of a CNC process of the selected device from the MTCX device group.

FI Command

CR_ATN_(1) **(Single Read)**

CC_ATN_(1) **(Cyclic Read)**

CB_ATN_(1) **(Break Cyclic Read)**

(1) = CNC process number [0...6]

Construction of Answer

The answer of the FI command ATN consists of one line with two columns for the identifier and the number of the active tool.

Line 1	Column 1	Column 2

Value Range/Meaning of Columns

1 = Identifier [T = tool]

2 = Number of active tool [1...9999999]

Note: If no tool is active in the selected CNC process then column 1 receives the value [T] and column 2 the value [0].

Example ATN

Read the number of the active tool in CNC process 0 of device address 00.

FI Command	**00_CR_ATN_0**	
Answer		
Line	**Column 1**	**Column 2**
1	T	4

Reference to Literature

You can find more details regarding the construction and the elements of the tool data in the Rexroth Indramat documentation:

"CNC NC Programming Instructions Vxx, Application Description", chapter „Access to Tool Data by NC Program TLD“, DOK-MTC200-NC**PRO*Vxx-AW0x-EN

Read Current Tool-Place Information: ATP

MTCX Device Group

Name **ATP** **A**ctual **T**ool-**P**lace Information

Explanation Information regarding the tool-place and the current edge of the pre-selected tool is returned by the “ATP” command. The answer telegram of the controls also returns additional information on the current position of the tool magazine. For this reason, the “ATP” access has 3 filter possibilities. The following information is returned by the control with the FI command “ATP”:

- **ATP1** Command / Actual position of the tool magazine and tool-place information of the active tool.
- **ATP2** Edge and place information of the active tool.
- **ATP3** Command / Actual position of the tool magazine.

The FI command refers to the indicated NC process. If the control is not able to return any information then the corresponding part-result "--" is transmitted.

FI Command Command / Actual position of the tool magazine and tool-place information of the active tool.

CR_ATP1_(1) **(Single Read)**

CC_ATP1_(1) **(Cyclic Read)**

CB_ATP1_(1) **(Break Cyclic Read)**

(1) = CNC process number [0...6]

Construction of Answer The following table shows the general construction of the answer of the FI command ATP1. One line with 4 columns is outputted for the returned values.

Line 1	Column 1	...	Column 4

Value Range/Meaning of the Columns

1 = Command pos. of magazine [1...999]

2 = Actual pos. of magazine [1...999]

3 = Active cutter number [1...9]

4 = Tool-place (type and place number) [Mx= magazine/turret [x=1...999]
Sx = spindle [x=1...4]
Gx = grip [x=1...4]]

Note: Details of the current command and actual position of the tool magazine refer to the reference point of the magazine control.

Example ATP1 Read the position of the tool magazine and tool-place information of the active tool from CNC process 0 of device 00.

FI Command		00_CR_ATP1_0
Line	**Column**	**Answer**
1	1	3
	2	3
	3	1
	4	S1

FI Command

Edge and place information of the active tool.

CR_ATP2_(1)	**(Single Read)**
CC_ATP2_(1)	**(Cyclic Read)**
CB_ATP2_(1)	**(Break Cyclic Read)**

(1) = CNC process number [0...6]

Construction of Answer

The following table shows the general construction of the answer of the FI command ATP2. One line with 2 columns is outputted for the returned values.

Line 1	Column 1	Column 2

Value Range/Meaning of the Columns

1 = active cutter number [1...9]

2 = tool-place (type and place number) [Mx= magazine/turret [x=1...999]
Sx = spindle [x=1...4]
Gx = grip [x=1...4]]

Example ATP2

Read the tool-place information of the active tool from CNC process 0 of device 00.

FI Command		00_CR_ATP2_0
Line	**Column**	**Answer**
1	1	1
	2	S1

FI Command

Output of the position of the tool magazine.

CR_ATP3_(1)	**(Single Read)**
CC_ATP3_(1)	**(Cyclic Read)**
CB_ATP3_(1)	**(Break Cyclic Read)**

(1) = CNC process number [0...6]

Construction of Answer

The following table shows the general construction of the answer of the FI command ATP3. One line with 2 columns is outputted for the returned values.

Line 1	Column 1	Column 2

Value Range/Meaning of the Columns

1 = Command pos. of magazine [1...999]

2 = Actual pos. of magazine [1...999]

Note: Details of the current command and actual position of the tool magazine refer to the reference point of the magazine control.

Example ATP3

Read the command and actual position of the tool magazine from CNC process 0 of device 00.

FI Command		00_CR_ATP3_0
Line	**Column**	**Answer**
1	1	3
	2	3

Active Zero-Offset Bank: AZB

MTCX Device Group

Name **AZB** **A**ctive **Z**ero-Offset **B**ank

Explanation Read-out of the number of the active zero-offset bank of a CNC process of the selected device from the MTCX device group.

The zero-point 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 are held in the shift banks.

FI Command

CR_AZB1_(1)	**(Single Read)**
CC_AZB1_(1)	**(Cyclic Read)**
CB_AZB1_(1)	**(Break Cyclic Read)**
(1) = CNC process number	[0...6]

Construction of Answer The answer of the FI command AZB1 consists of one line with two columns for the identifier (O = offset) and the number of the active drift bank [0...2].

Line 1	Column 1	Column 2

Example AZB Read the number of the active drift bank in CNC process 0 of device address 00.

FI Command	**00_CR_AZB1_0**	
Answer		
Line	**Column 1**	**Column 2**
1	O	2

Reference to Literature You can find more details regarding the use of zero-point offsets and drift banks in the Rexroth Indramat documentation:

“CNC NC Programming Instructions Vxx", Application Description, chapter „Zero-Point Offsets, Drift Banks O“, DOK-MTC200-NC**PRO*Vxx-AW0x-EN

Command Position of an Axis: CPO

MTCX Device Group

Name **CPO** **C**ommand **PO**sition

Explanation Output of the current command position of an axis selected on the code of the axis meaning of the MTCX device group.

FI Command

CR_CPO1_(1)_(2)_(3) **(Single Read)**

CC_CPO1_(1)_(2)_(3) **(Cyclic Read)**

CB_CPO1_(1)_(2)_(3) **(Break Cyclic Read)**

(1) = CNC process number [0...6]

(2) = Axis meaning [0...11; 20] (see chapter 6.2, Data Tables)

(3) = Coordinate system [1 = machine coordinates
2 = program coordinates]

Construction of Answer The following table shows the general construction of the answer of the FI command CPO1. One line with three columns for the name of the axis, the position and the unit is outputted in accordance to the settings of the process parameters.

Line 1	Column 1	...	Column 3

Value Range/Meaning of Columns

1 = Axis name [acc. to settings of axis parameter]

2 = Position [acc. to settings of the process parameter]

3 = Unit [acc. to settings of axis process parameter] mm, inch]

Note: If the specified axis is not defined in the selected CNC process then the answer in all columns is [--].

Example CPO1 Read the current command position of the Z axis in machine coordinates in CNC process 0 of device address 00.

FI Command	00_CR_CPO1_0_2_1		
Answer			
Line	**Column 1**	**Column 2**	**Column 3**
1	Z	-5.9897	[mm]

Reference to Literature Additional information regarding the display of the axis position in the GBO is located in the Rexroth Indramat documentation:

"MTC200/MT-CNC xxVRS GUI", DOK-MTC200-GBO*GEN*Vxx-AW0x-EN

FI Command Output of the command position of an axis of the device specified, related to the physical axis number.

CR_CPO2_(1)_(2) **(Single Read)**

CC_CPO2_(1)_(2) **(Cyclic Read)**

CB_CPO2_(1)_(2) **(Break Cyclic Read)**

(1) = Physical axis number	[1...32, acc. to settings of the system parameters]
(2) = Coordinate system	[1 = machine coordinates 2 = program coordinates]

Construction of Answer

The following table shows the general construction of the answer of the FI command CPO2. One line with three columns for the name of the axis, the position and the unit is outputted in accordance to the settings of the process parameters.

Line 1	Column 1	...	Column 3

Value Range/Meaning of Columns

1 = Axis name	[acc. to settings of axis parameter]
2 = Position	[acc. to settings of the process parameter]
3 = Unit	[acc. to settings of axis process parameter] mm, inch]

Note: If the specified axis is not defined in the selected CNC process then the answer in all columns is [--].

Example CPO2

Read the current command position of the Z axis (physical axis number = 3) in machine coordinates in CNC process 0 of device address 00.

FI Command	00_CR_CPO2_3_1		
Answer			
Line	Column 1	Column 2	Column 3
1	Z	-5.9897	[mm]

Reference to Literature

Additional information regarding the display of the axis position in the GBO is located in the Rexroth Indramat documentation:

"MTC200/MT-CNC xxVRS GUI", DOK-MTC200-GBO*GEN*Vxx-AW0x-EN

Trigger Control Reset: CRT

MTCX Device Group

Name

CRT **C**ontrol **R**ese**T**

Explanation

The control reset allows the selected device to be reset during an active system error. If there is no system error at the selected device then the job is ignored.

ATTENTION

Carrying out a reset completely re-initializes the device.
During initialization, communication is thereby temporarily interrupted (inherent to design).

FI Command

CW_CRT **(Single Write)**

Value to be written

Trigger reset 0

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

Construction of Answer

The return value of the "DataTransfer" routine is [0] when the write procedure has been successfully completed. In case of an error, more information can requested by the routine "ReadGroupItem" in the form of a general error result line (refer here to chapter 8, Error Codes).

Example CRT

Trigger a control reset on the selected device.

FI Command	00_CW_CRT
Value to be written	0

Reference to Literature

You can find more details on resetting the device in the Rexroth Indramat documentation:

"SPS Programming Instructions xxVRS", Chapter "Putting into Operation / Start", DOK-CONTRL-SPS*PRO*Vxx-AW0x-EN

Device Axis Configuration Parameter: DAC

MTCX Device Group

Name **DAC** **D**evice **A**xis **C**onfiguration Parameter

Explanation The configuration of the device axes that are configured in the active machine parameter record is read out. The following belong to the configuration data of the device axes: axis number, corresponding process, assigned processes, type of axis, APR number, APR axis number, main axis meaning, secondary axis meaning, main axis name, secondary axis name and corresponding axis number.

FI Command Output of the current parameters of all configured device axes.

BR_DAC1 **(Single Read)**

Construction of Answer The following table shows the general construction of the answer of the FI command DAC1. The number of answer lines [1...32 per CNC process] depends on the number of configured device axes. Each line consists of 11 columns.

Line 1...n:	Column 1	...	Column 11

Note: If there is no active machine parameter record in the device then the columns [1...11] are not applicable.

Value Range/Meaning of Columns

1 =	Physical axis number	[1...32]
2 =	CNC process number	[0...6]
3 =	Assigned processes	[0...6,--]
4 =	Type of axle	[see chapter 6.2, Data Tables]
5 =	APR number	[1...5]
6 =	APR axis number	[1...8]
7 =	Main axis meaning	[see chapter 6.2, Data Tables]
8 =	Secondary axis meaning	[see chapter 6.2, Data Tables]
9 =	Main axis name	[Xi, Yi, Zi, Ui, Vi, Wi, Ai, Bi ,Ci, Si, --] (i=[ ], [1...3])
10 =	Secondary axis name	[Xi, Yi, Zi, Ui, Vi, Wi, Ai, Bi ,Ci, Si, --] (i=[ ], [1...3])
11 =	Assigned axis number	[1...32, --]

Reference to Literature You can find more details regarding the configuration of the device axes in the Rexroth Indramat documentation:

"Parameter Description", chapter "System Parameters" as well as chapter "Axis Parameters", DOK-MT*CNC-PAR*DES*Vxx-AW0x-EN

Example DAC1 Read the current parameters of all configured device axes of the active machine parameter record of device address 00.

Assumption: The following three device axes have been defined:

- digital linear axis (axis number 1)
- digital linear axis (axis number 2)
- main circular axis spindle (axis number 3).

FI Command		**00_BR_DAC1**
Line	**Column**	**Answer**
1	1	1
	2	0
	3	--
	4	81
	5	1
	6	1
	7	0
	8	--
	9	X
	10	--
	11	--
2	1	2
	2	0
	3	--
	4	81
	5	1
	6	2
	7	1
	8	--
	9	Y
	10	--
	11	--
3	1	3
	2	0
	3	--
	4	85
	5	1
	6	4
	7	8
	8	--
	9	S
	10	--
	11	--

FI Command

Output of the current parameter of the selected device axis type.

BR_DAC2_(1) **(Single Read)**

(1) = axis type [1 = only digital axes, 2 = only analog axes]

Construction of Answer

The following table shows the general construction of the answer of the FI command DAC2. The number of answer lines [1...32] depends on the number of configured device axes. Each line consists of 11 columns.

Line 1...n:	Column 1	...	Column 11

Note: If there is no active machine parameter record in the device then the columns [1...11] are not applicable.

Value Range/Meaning of Columns

1 =	Axis number	[1...32]
2 =	CNC process number	[0...6]
3 =	Assigned processes	[0...6,--]
4 =	Type of axle	[see chapter 6.2, Data Tables, Axis Types]
5 =	APR number	[1...5]
6 =	APR axis number	[1...8]
7 =	Main axis meaning	[see chapter 6.2, Data Tables, Axis Meanings]
8 =	Secondary axis meaning	[see chapter 6.2, Data Tables, Axis Meanings]
9 =	Main axis name	[Xi, Yi, Zi, Ui, Vi, Wi, Ai, Bi ,Ci, Si, --] (i=[ ], [1...3])
10 =	Secondary axis name	[Xi, Yi, Zi, Ui, Vi, Wi, Ai, Bi ,Ci, Si, --] (i=[ ], [1...3])
11 =	Assigned axis number	[1...32, --]

Example DAC2

Read the current parameters of all configured digital device axes of the active machine parameter record of device address 00.

Assumption: A digital, linear axis with axis number 1 has been defined.

FI Command		00_BR_DAC2
Line	**Column**	**Answer**
1	1	1
	2	0
	3	--
	4	81
	5	1
	6	1
	7	0
	8	--
	9	X
	10	--
	11	--

Read D-Correction Data: DCD

MTCX Device Group

Name **DCD** **D**-**C**orrection **D**ata

Explanation The values of a D-correction register of the selected CNC process are read out.

The D-corrections are additive to the tool geometry data of the register that effects the tool management, i.e. they are additive to the existing geometry registers L1, L2, L3 and R.

99 D-correction numbers are available for each of the seven CNC processes. Each D-correction number thereby contains the registers L1, L2, L3 and R. Value assignment of the D-correction register is via the Rexroth Indramat GUI (BOF/GBO) or via the function interface.

FI Command Output of a D-correction register value of a CNC process of the selected device.

CR_DCD1_(1)_(2)_(3) **(Single Read)**

CC_DCD1_(1)_(2)_(3) **(Cyclic Read)**

CB_DCD1_(1)_(2)_(3) **(Break Cyclic Read)**

(1) = CNC process number [0...6]

(2) = D-correction number: [1...99]

(3) = number of the D-correction register: [1=L1, 2=L2, 3=L3, 4=R]

Construction of Answer The answer consists of one line with three columns for the identifier (length correction L1 to L3 and radius correction R), the value of the requested D-correction register and the unit in accordance to the settings of the process parameters.

Line	Column 1	Column 2	Column 3

Value Range/Meaning of Columns

1 =	Identifier	[L1, L2, L3, R]
2 =	Value of D-correction	[formatting acc. to settings of the process parameter]
3 =	Unit	[mm, inch; acc. to settings of the process parameters]

Note: If the requested D-correction number or the D-correction register is not assigned a value then the value 0 is outputted as answer – formatted according to the settings in the process parameters.

Example DCD1 Read the value of the D-correction register 4 at device address 00 in CNC process 0 of the D-correction number 1 (radius correction R).

FI Command	**00_CR_DCD1_0_1_4**		
Line	**Column 1**	**Column 2**	**Column 3**
1	R	0.0860	[mm]

Device Configuration Parameters: DCP

MTCX Device Group

Name **DCP** **D**evice **C**onfiguration **P**arameter

Explanation The device configuration parameters that are entered in the active machine parameter record as well as in the "IND_DEV.INI" file are outputted. The configuration parameters of the device include the device address, the device name, device type, mechanism number, mechanism name, and the process type.

FI Command Output of the configuration parameters of all defined devices.

BR_DCP1 **(Single Read)**

Note: The DCP1 FI command refers to all defined devices. Therefore, any valid device address can be indicated in the command line (see Example DCP1).

Construction of Answer The following table shows the general construction of the answer of the FI command DCP1 . The answer consists of a maximum of n=512 lines (n=16 devices x 32 mechanisms = 512), each with 7 lines.

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

Note: If no active machine parameter record exists in the device, then columns [1...7] of the respective device are not applicable.

Value Range/Meaning of Columns

1 =	Device address	[00...15]
2 =	Device name.	[max. 32 ASCII characters]
3 =	Device type:	[MTCNC, MTC200-P, MTC200-R, MTVNC, MTRA-P, MTRA-R]
4 =	Mechanism number	[0...31]
5 =	Mechanism name	[max. 28 ASCII characters]
6 =	Process type	[1= internal, 2 = external process]
7 =	Process type (2)	[1 = CNC process, 2 = SPS process]

Example DCP1

Read the device configuration parameters of all defined devices.

Assumption: Three devices have been defined

- Device address 00 (MTCNC)
- Device address 01 (MTC200-P) and
- Device address 02 (MTC200-P)

FI Command		00_BR_DCP1
Line	**Column**	**Answer**
1	1	00
	2	rotary transfer machine
	3	MTCNC
	4	1
	5	Master
	6	1
	7	2
2	1	01
	2	0
	3	MTC200-P
	4	0
	5	Milling machine 01
	6	1
	7	1
3	1	02
	2	0
	3	MTC200-P
	4	1
	5	Milling machine 02
	6	1
	7	1

Reference to Literature

Additional information regarding process parameters and their functions as well as value ranges are located in the Rexroth Indramat documentation:

"CNC/SPS Interface Description xxVRS Application Description, chapter "External Mechanisms"
DOK-MTC200-SPS*GWY*Vxx-AW0x-EN

FI Command

Output of the configuration parameters of the selected device.

BR_DCP2 **(Single Read)**

Construction of Answer

The following table shows the general construction of the answer of the FI command DCP2. The answer consists of a line with 7 columns.

Line 1	Column 1	...	Column 7

Note: If no active machine parameter record exists in the device, then columns [1...7] of the respective device are not applicable.

Value Range/Meaning of Columns

1 =	Device address	[00...15]
2 =	Device name.	[max. 32 ASCII characters]
3 =	Device type:	[MTCNC, MTC200-P, MTC200-R, MTVNC, MTRA-P, MTRA-R]
4 =	Mechanism number	[0...31]
5 =	Mechanism name	[max. 28 ASCII characters]
6 =	Process type	[1= internal, 2 = external process]
7 =	Process type (2)	[1 = CNC process, 2 = SPS process]

Example DCP2

Read the device configuration parameter 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

Additional information regarding process parameters and their functions as well as value ranges are located in the Rexroth Indramat documentation:

"CNC/SPS Interface Description xxVRS Application Description, chapter "External Mechanisms"
DOK-MTC200-SPS*GWY*Vxx-AW0x-EN

D-Correction Register DCR

MTCX Device Group

Name **DCR** **D**-**C**orrection **R**ecord

Explanation The values of a D-correction record of the selected CNC process are read out.

The D-corrections are additive to the tool geometry data of the register that effects the tool management, i.e. they are additive to the existing geometry registers L1, L2, L3 and R.

99 D-correction numbers are available for each of the seven CNC processes. Each D-correction number thereby 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 Output of a D-correction record of a CNC process of the selected device.

CR_DCR1_(1)_(2)	**(Single Read)**
CC_DCR1_(1)_(2)	**(Cyclic Read)**
CB_DCR1_(1)_(2)	**(Break Cyclic Read)**

(1) = CNC process number [0...6]

(2) = D-correction number: [1...99]

Note: If the value of a single D-correction register is to be read then the command **CR_DCD1** should be used.

Construction of Answer The answer consists of four lines, each with three columns for the identifier (length correction L1 to L3 and radius correction R), the value of the requested D-correction register and the unit in accordance to the settings of the process parameters.

Line 1...4	Column 1	...	Column 3

Value Range/Meaning of Columns

1 =	Identifier	[L1, L2, L3, R]
2 =	Value of D-correction	[formatting of values acc. to settings of the process parameter]
3 =	Unit	[mm, inch; acc. to settings of the parameters]

Note: If the requested D-correction number or the D-correction register is not assigned a value then the value 0 is outputted as answer – formatted according to the settings in the process parameters.

Example DCR1 without optional Parameters

Read the value of all D-correction registers at device address 00 of CNC 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

Output of all D-correction register values of a CNC process of the selected device.

CW_DCR_(1)_(2)	**(Single Write)**
(1) = CNC process number	[0...6]
(2) = D-correction number	[1...99]

Value to be written

D-Correction register	[L1<value> L2<value> L3<value> R<value>] [opt. unit]

If there is no optional detail of the unit {mm, inch}, then the values refer to the base programming unit of the process If the unit entered different from the base programming unit then the values entered are converted into the values of the base programming unit.

ATTENTION

⇒ In the conversion from mm -> inch, rounding errors are unavoidable, as precision is lost!

Note: The single values are separated by a space, whereby the formatting should be carried out according to the settings of the process parameters. (refer to example DCR1: write D-correction register).

Construction of Answer

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

(P_ACK) = **P**ositive **ACK**nowledge Value has been written

Example DCR
Write D-Correction Register

Write all D-correction registers at device address 00 of CNC process 0 of D-correction number 1 with the following 5 values:

- 1st value L1: 1.2586
- 2nd value L2: 3.5892
- 3rd value L3: 0.0000 and
- 4th value R: 0.0860
- 5th value unit: mm (optional)

Note: The values to be written are passed in the “Data Transfer” routine to the "acValue" parameter and must be separated from each other by a space " ".

FI Command		**00_CW_DCR_0_1** **Values to be written:** **L1 1.2586 L2 3.5892 L3 0 R 0.086 mm**
Line	**Column**	**Answer**
1	1	(P_ACK)

Reference to Literature

You can find more details regarding D-correction in the Rexroth Indramat documentation:

“MTC200/MT-CNC NC Programming Instructions xxVRS", chapter „D-Corrections“,
DOK-MTC200-NC**PRO*Vxx-AW0x-EN

Long Identification of NC/SPS Data Records: DIS

MTCX Device Group

Name **DIS** **D**ata **I**dentification **S**tring

Explanation Reads the long identification (directory entries) of NC/SPS 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/SPS data records are outputted:

- NC parameter record (FI command: DIS1)
- SPS 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 of the directory entries of the valid NC parameter record in the selected device.

CR_DIS1 **(Single Read)**

Construction of Answer The following table shows the general construction of the answer of the FI command DIS1. The answer consists of a line with five columns.

Line 1	Column 1	...	Column 5

Value Range/Meaning of Columns

1 =	Number in NC parameter directory	[01...99]
2 =	Name of the NC parameter record	[max. 32 ASCII characters]
3 =	Length of the NC parameter record	[bytes]
4 =	Date of creation/last change to NC package	[DD.MM.YY]
5 =	Time of creation/last change to NC parameter record	[HH:MM:SS]

Note: If there is no valid NC parameter record in the selected device then column 1 contains an empty string and columns 2 to 5 are not applicable (redundant).

Example DIS1 Read the directory entries of the NC parameter record at device address 00. Assumption:

- There is a valid NC parameter record in the selected device.

FI Command		00_CR_DIS1
Line	**Column**	**Answer**
1	1	01
	2	KEY1
	3	3579
	4	16.05.99
	5	10:41:08

Reference to Literature

Additional information regarding the function of the NC parameters and the construction of the NC parameter records is contained in the Rexroth Indramat documentation:

"MTC200/MT-CNC Parameter Description xxVRS", DOK-MTC200-PAR*DES*Vxx-AW0x-EN

FI Command

Output of the directory entries of the valid SPS program in the selected device.

CR_DIS2 (Single Read)

Construction of Answer

The following table shows the general construction of the answer of the FI command DIS2. The answer consists of a line with six columns.

Line 1	Column 1	...	Column 6

Value Range/Meaning of Columns

1 =	Number in SPS directory	[01...99]
2 =	Name of the SPS program	[max. 8 ASCII characters]
3 =	Length of the SPS program	[bytes]
4 =	Date of creation/last change to SPS program	[DD.MM.YY]
5 =	Time of creation/last change to SPS program	[HH:MM:SS]
6 =	Date of creation/last change to SPS program	[DD.MM.YYYY]

Note: If there is no valid SPS program in the selected device then column 1 contains an empty string and columns 2 to 6 are not applicable (redundant).

Example DIS2

Read the directory entries of the SPS program at address 00. Assumption:

- There is a valid SPS program in the selected device.

FI Command		00_CR_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

You can find more details regarding the SPS Programming System in the Rexroth Indramat documentation:

"SPS Programming Instructions xxVRS" Application Description" DOK-CONTRL-SPS*PRO*Vxx-AW0x-EN

FI Command

Output of the directory entries of the valid NC package of the selected NC memory.

CR_DIS3 **(Single Read)**

(1) = NC memory [1 = NC memory A; 2 = NC memory B]

Construction of Answer

The following table shows the general construction of the answer of the FI command DIS3. The answer consists of a line with five columns.

Line 1	Column 1	...	Column 5

Value Range/Meaning of Columns

1 =	Number in NC package directory	[01...99]
2 =	Name of the NC package	[max. 32 ASCII characters]
3 =	Length of the NC package	[bytes]
4 =	Date of creation/last change to NC package	[DD.MM.YY]
5 =	Time of creation/last change to NC package	[HH:MM:SS]

Note: If there is no valid NC package in the selected NC memory of the indicated device then column 1 contains an empty string and columns 2 to 5 are not applicable (redundant).

Example DIS3

Read the directory entries of the NC package in NC memory A at device address 00.

FI Command		00_CR_DIS3_1
Line	**Column**	**Answer**
1	1	01
	2	KEY1
	3	3579
	4	16.05.99
	5	10:41:08

Reference to Literature

You can find more details regarding the construction of NC packages in the Rexroth Indramat documentation:

“MTC200/MT-CNC NC Programming Instructions xxVRS", chapter „Sub-Programs“,
DOK-MTC200-NC**PRO*Vxx-AW0x-EN

FI Command

Output of the directory entries of the valid tool list of the selected CNC process.

CR_DIS4_(1) (Single Read)

(1) = CNC process number [0...6]

Construction of Answer

The following table shows the general construction of the answer of the FI command DIS4. The answer consists of a line with five columns.

Line 1	Column 1	...	Column 5

Value Range/Meaning of Columns

1 =	Number in the tool lists directory	[01...99]
2 =	Name of the tool list	[max. 32 ASCII characters]
3 =	Length of the tool list	[bytes]
4 =	Date of creation/last change to the tool list	[DD.MM.YY]
5 =	Time of creation/last change to the tool list	[HH:MM:SS]

Note: If there is no valid tool list in the selected CNC process of the indicated device then column 1 contains an empty string and columns 2 to 5 are not applicable (redundant).

Example DIS4

Read the directory entries of the tool list of CNC process 0 at device address 00.

FI Command		00_CR_DIS4_0
Line	Column	Answer
1	1	01
	2	KEY1
	3	2048
	4	17.09.99
	5	10:45:08

Reference to Literature

You can find more details regarding the construction of tool lists 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-EN

FI Command

Output of the directory entries of the valid machine data record in the selected device.

CR_DIS5 (Single Read)

Construction of Answer

The following table shows the general construction of the answer of the FI command DIS5. The answer consists of a line with five columns.

Line 1	Column 1	...	Column 5

Value Range/Meaning of Columns

1 =	Number in machine data directory	[01...99]
2 =	Name of the data record	[max. 32 ASCII characters]
3 =	Length of data record	[bytes]
4 =	Date of creation/last change of data record	[DD.MM.YY]
5 =	Time of creation/last change of data record	[HH:MM:SS]

Note: If there is no valid machine data in the selected device then column 1 contains an empty string and columns 2 to 5 are not applicable (redundant).

Example DIS5

Read the directory entries of the machine data record in device address 00. Assumption:

- There is valid machine data in the selected device

FI Command		00_CR_DIS5
Line	**Column**	**Answer**
1	1	01
	2	KEY1
	3	3180
	4	18.12.98
	5	21:20:02

Reference to Literature

You can find more details regarding the use of the machine data in the Rexroth Indramat documentation:

"SPS Machine Data xxVRS" Application Description”
DOK-MT*CNC-MAS*DAT*Vxx-AW0x-EN

FI Command

Output of the directory entries of the valid NC program.

CR_DIS6_(1)_(2)_(3) **(Single Read)**

(1) = NC memory	[1 = NC memory A; 2 = NC memory B]
(2) = CNC process number	[0...6]
(3) = NC program number	[1...99]

Construction of Answer

The following table shows the general construction of the answer of the FI command DIS6. The answer consists of a line with six columns.

Line 1	Column 1	...	Column 6

Value Range/Meaning of Columns

1 =	Package number	[01...99]
2 =	Number of the NC program	[01...99]
3 =	Name of the NC program	[max. 32 ASCII characters]
4 =	Length of the NC program	[bytes]
5 =	Date of creation/last change to NC program	[DD.MM.YY]
6 =	Time of creation/last change to NC program	[HH:MM:SS]

Note: If there is no valid NC package in the selected CNC process then column 1 contains an empty string and columns 2 to 6 are not applicable (redundant).

Example DIS6

Read the directory entries of the third NC program (NC package number 2, NC memory A, CNC process 0) at device address 00.

FI Command		00_CR_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

You can find more details regarding the construction of the NC data structure in the Rexroth Indramat documentation:

“NC Programming Instructions Vxx", chapter „Program and Data Organization“,
DOK-MTC200-NC**PRO*Vxx-AW0x-EN

Delete NC Program: DPN

MTCX Device Group

Name **DPN** **D**elete **P**rogram **N**C

Explanation An NC program located in an NC package directory is deleted.

FI Command

BW_DPN_(1)_(2)_(3)_(4) **(Single Write)**

(1) = NC package directory number [1...99]

(2) = CNC process number [0 6]

(3) = NC program number [1...99]

(4) = with check / without check [1 / 0]

Construction of Answer One line with one column is outputted for the acknowledgement of the FI command issued. The following meanings then apply, depending on parameter 4 (check):

with check (1)	
(BOF_C_NCPROG_CREATED)	NC program not deleted

without check (0)	
(BOF_FCT_OK) = **BOF_F**un**CT**ion_**OK**	NC program has been deleted

Example DPN

- The NC program numbered 1 in NC package directory 3 of process 2 is to be deleted.

FI Command	**00_BW_DPN_1_2_3_0**
Answer	
Line	**Column 1**
1	(BOF_FCT_OK)

Delete NC Program Package: DPP

MTCX Device Group

Name **DPP** **D**elete **P**rogram **P**ackage

Explanation An NC program package is deleted in the NC package directory of the selected MTCX device group.

FI Command **BW_DPP_(1)** **(Single Write)**

(1) = NC program package [1...99]

Construction of Answer One line with one column is outputted for the acknowledgement of the FI command issued. The following hereby mean:

(BOF_FCT_OK) = **BOF_F**un**CT**ion_**OK**	program package has been deleted.

Example DPP

- The NC program package numbered 1 in the NC package directory is to be deleted.

FI Command	**00_BW_DPP_1**
Answer	
Line	**Column 1**
1	(BOF_FCT_OK)

Device Tool Management Configuration: DTC

MTCX Device Group

Name **DTC** **D**evice **T**ool Management **C**onfiguration

Explanation Returns the most important system parameter data of the tool management.

FI Command Read tool management data.

BR_DTC1 **(Single Read)**

BC_DTC1 **(Cyclic Read)**

Construction of Answer One line with 10 columns is outputted for the returned values.

Line 1	**Column 1**	**...**	**Column 10**

Value Range/Meaning of the Columns

1 = Tool management	[YES, NO]
2 = Setup list	[[STATION], [PROGRAM]]
3 = Max. number of cutters	[1...9]
4 = Wear register	[YES, NO]
5 = Offset register	[YES, NO]
6 = Comment	[YES, NO]
7 = Wear factors	[YES, NO]
8 = Tool life	[YES, NO]
9 = Geometry limit values	[YES, NO]
10 = Tool technology	[[TURN./MILL.], [GRINDING]]

Note: If there is no tool management (column 1: NO), then all part-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.]

Distance to Go of Axis Movement: DTG

MTCX Device Group

Name

DTG **D**istance **T**o **G**o

Explanation

The distance to go of the movement of a selected axis is output. The FI command "DTG1" returns the distance to go of an axis, related to the code of the axis meaning. The FI command "DTG2", on the other hand, returns the distance to go of an axis, related to the physical axis number.

FI Command

Output of the distance to go of the selected axis of the device specified, related to the code of the axis meaning.

CR_DTG1_(1)_(2)_(3) **(Single Read)**

CC_DTG1_(1)_(2)_(3) **(Cyclic Read)**

CB_DTG1_(1)_(2)_(3) **(Break Cyclic Read)**

(1) = CNC process number [0...6]

(2) = Axis meaning [0...11; 20];

(3) = Coordinate system [1 = machine coordinates
2 = program coordinates]

Construction of Answer

The following table shows the general construction of the answer of the FI command DTG1. One line with three columns for the name of the axis, the distance to go and the unit is outputted in accordance to the settings of the process parameters.

Line 1	Column 1	...	Column 3

Value Range/Meaning of Columns

1 = Axis name [acc. to settings of axis parameter]

2 = Distance to go [acc. to settings of the process parameter]

3 = Unit [acc. to settings of axis process parameter] mm, inch]

.

Note: If the specified axis is not defined in the selected CNC process then the answer in all columns is [--].

Example DTG1

Read the distance to go of the Z axis in machine coordinates in CNC process 0 of device address 00.

FI Command	00_CR_DTG1_0_2_1		
Answer			
Line	Column 1	Column 2	Column 3
1	Z	-5.9897	[mm]

FI Command

Output of the distance to go of the movement of the selected axis of the device specified, related to the physical axis number.

CR_DTG2_(1)_(2) **(Single Read)**

CC_DTG2_(1)_(2) **(Cyclic Read)**

CB_DTG2_(1)_(2) **(Break Cyclic Read)**

(1) = Physical axis number [1...32, acc. to settings of the system parameters]

(2) = Coordinate system [1 = machine coordinates
2 = program coordinates]

Construction of Answer

The following table shows the general construction of the answer of the FI command DTG2 . One line with three columns for the name of the axis, the distance to go and the unit is outputted in accordance to the settings of the process parameters.

Line 1	Column 1	...	Column 3

Value Range/Meaning of Columns

1 = Axis name [acc. to settings of axis parameter]

2 = Distance to go [acc. to settings of the process parameter]

3 = Unit [acc. to settings of axis process parameter] mm, inch]

Note: If the specified axis is not defined in the selected CNC process then the answer in all columns is [--].

Example DTG2

Read the distance to go of the movement of the Z axis (physical axis number = 3) in machine coordinates at the device address 00.

FI Command	00_CR_DTG2_3_1		
Answer			
Line	Column 1	Column 2	Column 3
1	Z	-5.9897	[mm]

Reference to Literature

Additional 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-EN

Device Type and Accompanying Components: DTY

MTCX Device Group

Name **DTY** **D**evice **TY**pe

Explanation The device type as well as the accompanying components of the selected device addresses are outputted.

FI Command **CR_DTY1** **(Single Read)**

Construction of Answer The following table shows the general construction of the answer of the FI command DTY1. A line with three columns for the device type is outputted as well as the names of the first device component and the name of the second device component.

Line 1	Column 1	...	Column 3

Value Range/Meaning of Columns

1 =	Device type:	(see chapter 6.1 Elements of the FI Command, Identifier)
2 =	Component type1	IND_DEV.INI entry: Componenttype1=
3 =	Component type 2	IND_DEV.INI entry: Component-type2=

Example DTY1 Output the device type and the accompanying components of device address 00.

FI Command	00_CR_DTY1		
Answer			
Line	Column 1	Column 2	Column 3
1	MTC200-P	MTS-P	MTC-P

End Point of an Axis Movement: EPO

MTCX Device Group

Name

EPO **E**nd**PO**int

Explanation

The end point of the movement of a selected axis is outputted. 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 of the end point of the selected device related to the code of the axis meaning.

CR_EPO1_(1)_(2)_(3) **(Single Read)**

CC_EPO1_(1)_(2)_(3) **(Cyclic Read)**

CB_EPO1_(1)_(2)_(3) **(Break Cyclic Read)**

(1) = CNC process number [0...6]

(2) = Axis meaning [0...11; 20];

(3) = Coordinate system [1 = machine coordinates
2 = program coordinates]

Construction of Answer

The following table shows the general construction of the answer of the FI command EPO1. One line with three columns for the name of the axis, the end point of the movement and the unit is outputted in accordance to the settings of the process parameters.

Line 1	Column 1	...	Column 3

Value Range/Meaning of Columns

1 = Axis name [acc. to settings of axis parameter]

2 = End point [acc. to settings of the process parameter]

3 = Unit [acc. to settings of the process parameter:] mm, inch]

Note: If the specified axis is not defined in the selected CNC process then the answer in all columns is [--].

Example EPO1

Read the distance to go of the Z axis in machine coordinates in CNC process 0 of device address 00.

FI Command	00_CR_EPO1_0_2_1		
Answer			
Line	**Column 1**	**Column 2**	**Column 3**
1	Z	-5.9897	[mm]

FI Command

Output of the end point of the selected axis of the device specified, related to the physical axis number.

CR_EPO2_(1)_(2)	**(Single Read)**
CC_EPO2_(1)_(2)	**(Cyclic Read)**
CB_EPO2_(1)_(2)	**(Break Cyclic Read)**
(1) = Physical axis number	[1...32, acc. to settings of the system parameters]
(2) = Coordinate system	[1 = machine coordinates 2 = program coordinates]

Construction of Answer

The following table shows the general construction of the answer of the FI command EPO2. One line with three columns for the name of the axis, the end point of the movement and the unit is outputted in accordance to the settings of the process parameters.

Line 1	Column 1	...	Column 3

Value Range/Meaning of Columns

1 = Axis name	[acc. to settings of axis parameter]
2 = Position	[acc. to settings of the process parameter]
3 = Unit	[acc. to settings of the process parameter:] mm, inch]

Note: If the specified axis is not defined in the selected CNC process then the answer in all columns is [--].

Example EPO2

Read the end point of the movement of the Z axis (physical axis number = 3) in machine coordinates at the device address 00.

FI Command	**00_CR_EPO2_3_1**		
Answer			
Line	**Column 1**	**Column 2**	**Column 3**
1	Z	-5.9897	[mm]

Reference to Literature

Additional 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-EN

Global Process Parameter Configuration: GPC

MTCX Device Group

Name **GPC** **G**lobal **P**rocess **C**onfiguration

Explanation The configuration of the global process parameter of the active machine parameter record of the selected device from the MTCX 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 CNC 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, tipping of axis results in a reset and the re-positioning of the tool memory axis.

Note: The FI commands "GPPx" (refer to Global Process Parameter : GPP

FI Command Output of the configuration of the global process parameters of all defined CNC processes of the active machine parameter record.

BR_GPC1 **(Single Read)**

Construction of Answer The following table shows the general construction of the answer of the FI command GPC1. The answer consists of one of a maximum of n=7 lines (n= max. number of defined CNC processes [0...6] = 7), each with 12 columns.

Line 1	Column 1	...	Column 12

Note: If there is no active machine parameter record in the device then the columns [1...12] are not applicable.

Value Range/Meaning of Columns

1 =	CNC process number	[0...6]
2 =	Name of the CNC process	[max. 20 ASCII characters]
3 =	Base programming unit	mm, inch]
4 =	Programmed number of positions after decimal point	[4, 5]
5 =	Displayed positions after the decimal point	[0...4]
6 =	Max. zero-point-data bank number	[0...9]
7 =	D-correction	[YES, NO]
8 =	Basic setting required	[YES, NO]
9 =	Reference required	[YES, NO]
10 =	Transformation from Cartesian to polar possible	[YES, NO]
11 =	Tipping of axis causes reset	[YES, NO]
12 =	Re-position tool memory axis	[YES, NO]

Example GPC1

Read the configuration of the global process parameters of all defined CNC processes of the active machine parameter record of device address 00.
Assumption: The following three CNC processes have been defined:

- Sled 1 (CNC process number 0),
- Turret 1 (CNC process number 1) and
- Turret 2 (CNC process number 3).

FI Command		00_BR_GPC1
Line	**Column**	**Answer**
1	1	0
	2	Sled 1
	3	[mm]
	4	4
	5	3
	6	0
	7	YES
	8	NO
	9	NO
	10	NO
	11	YES
	12	NO
2	1	1
	2	Turret 1
	3	[mm]
	4	4
	5	3
	6	0
	7	NO
	8	YES
	9	YES
	10	NO
	11	YES
	12	NO
3	1	3
	2	Turret 2
	3	[mm]
	4	4
	5	3
	6	0
	7	NO
	8	YES
	9	NO
	10	NO
	11	YES
	12	NO

Reference to Literature

Additional information regarding process parameters and their functions as well as value ranges are located in the Rexroth Indramat documentation:

"Parameter Description" chapter "Process Parameters"
DOK-MT*CNC-PAR*DES*Vxx-AW0x-EN

FI Command

Output of the global process parameters of the active machine parameter record of the selected device related to the CNC process.

BR_GPC2_(1) **(Single Read)**

(1) = CNC process number [0...6]

Construction of Answer

The following table shows the general construction of the answer of the FI command GPC2. The answer consists of a line with 12 columns.

Line 1	Column 1	...	Column 12

Note: If there is no active machine parameter record in the device or the selected CNC process is not defined then the columns [1...12] are not applicable.

Value Range/Meaning of Columns

1 =	CNC process number	[0...6]
2 =	Name of the CNC process	[max. 20 ASCII characters]
3 =	Base programming unit	mm, inch]
4 =	Programmed number of positions after decimal point	[4, 5]
5 =	Displayed positions after the decimal point	[0...4]
6 =	Max. zero-point-data bank number	[0...9]
7 =	D-correction	[YES, NO]
8 =	Basic setting required	[YES, NO]
9 =	Reference required	[YES, NO]
10 =	Transformation from Cartesian to polar possible	[YES, NO]
11 =	Tipping of axis causes reset	[YES, NO]
12 =	Re-position tool memory axis	[YES, NO]

Example GPC2

Read the global process parameter in CNC process 0 of the active machine parameter record of device address 00.
Assumption: The following three CNC processes have been defined:

- Sled 1 (CNC process number 0),
- Turret 1 (CNC process number 1) and
- Turret 2 (CNC process number 3).

FI Command		00_BR_GPC2_0
Line	**Column**	**Answer**
1	1	0
	2	Sled 1
	3	[mm]
	4	4
	5	3
	6	0
	7	YES
	8	NO
	9	NO
	10	NO
	11	YES
	12	NO

Reference to Literature Additional information regarding process parameters and their functions as well as value ranges are located in the Rexroth Indramat documentation:

"Parameter Description" chapter “Process Parameters” DOK-MT*CNC-PAR*DES*Vxx-AW0x-EN.

Global Process Parameter : GPP

MTCX Device Group

Name **GPP** **G**lobal **P**rocess **P**arameter

Explanation The global process parameter of the active machine parameter record of the selected device from the MTCX device group is read out. This includes the programmable and actually displayed digits after the decimal point for the displacement, the name of the CNC process, the base programming unit and the max. zero-point-data bank number.

Note: The FI commands "GPPx" (refer to Global Process Parameter Configuration: GPC

) should be preferred to the FI commands "GPCx" as the access speed has been optimized by these.

FI Command Output of the configuration of the global process parameters of all defined CNC processes of the active machine parameter record.

BR_GPP1 **(Single Read)**

Construction of Answer The following table shows the general construction of the answer of the FI command GPC1. The answer consists of one of a maximum of n=7 lines (n= max. number of defined CNC processes [0...6] = 7), each with six columns.

Line 1	Column 1	...	Column 6

Note: If there is no active machine parameter record in the device then the columns [1...6] are not applicable.

Value Range/Meaning of Columns

1 =	CNC process number	[0...6]
2 =	Name of the CNC process	[max. 20 ASCII characters]
3 =	Base programming unit	mm, inch]
4 =	Programmed number of positions after decimal point	[4, 5]
5 =	Displayed positions after the decimal point	[0...4]
6 =	Max. zero-point-data bank number	[0...9]

Example GPP1

Read the global process parameters of all defined CNC processes of the active machine parameter record of device address 00.
Assumption: The following three CNC processes have been defined:

- Sled 1 (CNC process number 0),
- Turret 1 (CNC process number 1) and
- Turret 2 (CNC process number 3).

FI Command		00_BR_GPP1
Line	**Column**	**Answer**
1	1	0
	2	Sled 1
	3	[mm]
	4	4
	5	3
	6	0
2	1	1
	2	Turret 1
	3	[mm]
	4	4
	5	3
	6	0
3	1	3
	2	Turret 2
	3	[mm]
	4	4
	5	3
	6	0

Reference to Literature

Additional information regarding process parameters and their functions as well as value ranges are located in the Rexroth Indramat documentation:

"Parameter Description" chapter “Process Parameters”
DOK-MT*CNC-PAR*DES*Vxx-AW0x-EN

FI Command

Output of the global process parameters of the active machine parameter record of the selected device related to the CNC process.

BR_GPP2_(1) **(Single Read)**

(1) = CNC process number [0...6]

Construction of Answer

The following table shows the general construction of the answer of the FI command GPP2. The answer consists of a line with six columns.

Line 1	Column 1	...	Column 6

Note: If there is no active machine parameter record in the device or the selected CNC process is not defined then the columns [1...6] are not applicable.

Value Range/Meaning of Columns

1 =	CNC process number	[0...6]
2 =	Name of the CNC process	[max. 20 ASCII characters]
3 =	Base programming unit	mm, inch]
4 =	Programmed number of positions after decimal point	[4, 5]
5 =	Displayed positions after the decimal point	[0...4]
6 =	Max. zero-point-data bank number	[0...9]

Example GPP2

Read the global process parameter in CNC process 0 of the active machine parameter record of device address 00.

Assumption: The following three CNC processes have been defined:

- Sled 1 (CNC process number 0),
- Turret 1 (CNC process number 1) and
- Turret 2 (CNC process number 3).

FI Command		00_BR_GPP2_0
Line	**Column**	**Answer**
1	1	0
	2	Sled 1
	3	[mm]
	4	4
	5	3
	6	0

Reference to Literature

Additional information regarding process parameters and their functions as well as value ranges are located in the Rexroth Indramat documentation:

"Parameter Description" chapter “Process Parameters”
DOK-MT*CNC-PAR*DES*Vxx-AW0x-EN.

Insert NC Program Package: IPP

MTCX Device Group

Name **IPP** **I**nsert NC-**P**rogram **P**ackage

Explanation Enters (inserts) an NC program package into the NC package directory of the BOF structure.

FI Command **BW_IPP_(1){_(2)}** **(Single Write)**

(1) = number in NC package-directory [1...99]

(2) = is the NC package directory entry empty? [0 = without check (preset); 1 = with check] ! Optional !

Note: If there is already an NC program package at the selected number of the NC package directory of the BOF structure then an error is returned if a check is to be made.

Value to be written Name of the NC package [max. 32 ASCII characters]

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

Construction of Answer One line with one column is outputted for the acknowledgement of the FI command issued. The following hereby mean:

(P_ACK) = **P**ositive **ACK**nowledge NC package has been entered.

Example IPP Enter the NC program package named "KEY1" into number 1 in the NC package directory of the BOF structure. <u>Assumption:</u>

- A check is to be made of whether or not the selected entry is empty in the NC package directory.

FI Command	**00_BW_IPP_1_1** **Value to be written: KEY1**
Answer	
Line	**Column 1**
1	(P_ACK)

Module Assignment of a Process: MAP

MTCX Device Group

Name **MAP** **M**odule **A**ssign of **P**rocess

Explanation The module to which a particular process is assigned is read out from the "Moduldef.ini'" file. This data is located in the [LW]:\MT-CNC\CONFIG directory and contains all module configuration data. The process data is located in three sections:

[DeviceAddrX\ModulY\Process]

whereby "X" stands for the device addressed and "Y" for the configuration of the module numbers.

FI Command Determine the module to which the process belongs. Information is read out from the module configuration of the MTCX device group.

BR_MAP1_(1) **(Single Read)**

BC_MAP1_(1) **(Cyclic Read)**

BB_MAP1_(1) **(Break Cyclic Read)**

(1) = mechanism number [0...31]

Construction of Answer The following table shows the general construction of the answer of the FI command MAP1. One line with one column is outputted for module number that has been determined.

Value Range of the Column 1 = module number [0...99]

Example MAP1 Read the module number that is assigned to the CNC process number 4 from the module configuration.

Assumption: The module that is assigned to the CNC process 4 has module number 5.

FI Command	**03_BR_MAP1_4**
Answer	
Line	**Column 1**
1	5

Reference to Literature Additional information regarding module configuration and the construction of the "Moduldef.ini" file can be located in the following Rexroth Indramat documentation:

"Diagnostics and Message System for HMI System ProVi", chapter "Configure Moduldef.ini", DOK-MTC200-DIAG*PROVI*-AW0x-EN.

Read Reference Name of a SPS Variable : MAR

MTCX Device Group

Name **MAR** **M**ap **A**bsolute PCL **R**eference

Explanation The absolute reference name of a symbolic SPS variable is read out.

FI Command Reads the absolute SPS reference name of a SPS variable.

BR_MAR_(1) **(Single Read)**

(1) = Identifier of the SPS variable

Example MAR Read the absolute reference name of the SPS variable with the identifier “abref" at device address 00.

Assumption:

- The SPS variable with the identifier "abref" is of the type "INTEGER"

FI Command	**00_BR_MAR_abref**
Answer	
Line	**Column 1**
1	%M100.0

Device Data of the Module Configuration: MCD

MTCX Device Group

Name **MCD** **M**odule **C**onfiguration: **D**evice Information

Explanation All device data of the module configuration are read-out from the "Moduldef.ini" file that is stored in the "[LW]:\MT-CNC\CONFIG" directory. The device data are in the sections [DeviceAddrX], whereby "X" stands for the configured device addresses.

FI Command Read-out of device data within the module configuration of the MTCX device groups.

BR_MCD1 **(Single Read)**

BC_MCD1 **(Cyclic Read)**

BB_MCD1 **(Break Cyclic Read)**

Note: The MCD1 FI command refers to all devices within the MTCX device group. Therefore, any valid device address can be indicated in the command line (see Example MCD1).

Construction of Answer The following table shows the general construction of the answer of the FI command MCD1 . The number of lines depends on the number of configured devices. Every line consists of four columns for the device address as well as SPS-FB names for the provision of setup diagnostics, warning messages and start requirements.

Line 1	**Column 1**	**...**	**Column 4**

Value Range of the Columns

1 = Device address	[0...15]
2 = SPS-FB name for the setup diagnostics	[max. 9 ASCII characters]
3 = SPS-FB name for the warning messages	[max. 9 ASCII characters]
4 = SPS-FB name for the start requirements	[max. 9 ASCII characters]

Example MCD1

Read all device data of the module configuration

Assumption: The following devices have been configured in the MTCX device group:

- Device address 01 (MTC200-P)
- Device address 03 (MT-CNC)

FI Command	**03_BR_MCD1**			
Answer				
Line	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	01	PVSetup_1	PVWarn_1	PVStart_1
2	03	PVSetup_3	PVWarn_3	PVStart_3

Reference to Literature

Additional information regarding module configuration and the construction of the "Moduldef.ini" file can be located in the following Rexroth Indramat documentation:

"Diagnostics and Message System for HMI System ProVi", chapter "Configure Moduldef.ini", DOK-MTC200-DIAG*PROVI*-AW0x-EN.

Device Data of the Module Configuration: MCM

MTCX Device Group

Name

MCM **M**odule **C**onfiguration: **M**odule Information

Explanation

All module data of a particular device is read out from the "Moduldef.ini'" file. This data is located in the "[LW]:\MT-CNC\CONFIG" directory and contains all module configuration data. The module data is located in sections [DeviceAddrX\ModulY], whereby "X" stands for the device addressed and "Y" for the configured module numbers.

FI Command

Read-out of module data from the module configuration with respect to a device from the MTCX device group.

BR_MCM1 **(Single Read)**

BC_MCM1 **(Cyclic Read)**

BB_MCM1 **(Break Cyclic Read)**

Construction of Answer

The following table shows the general construction of the answer of the FI command MCM1 . 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 SPS-FB (FB = function component) names for general module errors and module messages.

Line 1	**Column 1**	**...**	**Column 4**

Value Range of the Columns

1 = Module number	[0...99]
2 = Module name	[max. 28 ASCII characters]
3 = SPS-FB name for general module errors	[max. 9 ASCII characters]
4 = SPS-FB name for module messages	[max. 9 ASCII characters]

Example MCM1

Read the module data of device 03 from the module configuration:

Assumption: The following modules have been defined:

- Module number 5
- Module number 7

FI Command	03_BR_MCM1			
Answer				
Line	Column 1	Column 2	Column 3	Column 4
1	5	Module 5 – Milling	PVError_5	PVMsg_5
2	7	Module 7 - Drilling	PVError_7	PVMsg_7

Reference to Literature

Additional information regarding module configuration and the construction of the "Moduldef.ini" file can be located in the following Rexroth Indramat documentation:

Diagnostics and Message System for HMI System ProVi, chapter "Configure Moduldef.ini", DOK-MTC200-DIAG*PROVI*-AW0x-EN.

Process Data of the Module Configuration: MCP

MTCX Device Group

Name

MCP **M**odule **C**onfiguration: **P**rocess Information

Explanation

All process data of a particular module is read out from the "Moduldef.ini"' file. This data is located in the "[LW]:\MT-CNC\CONFIG" directory and contains all module configuration data. The module data is located in sections [DeviceAddrX\ModulY\Process], whereby "X" stands for the device addressed and "Y" for the selected module numbers.

FI Command

BR_MCP1_(1) **(Single Read)**

BC_MCP1_(1) **(Cyclic Read)**

BB_MCP1_(1) **(Break Cyclic Read)**

1 = Module number [0...99]

Construction of Answer

The answer of the FI command MCP1 consists of one of up to a maximum number of n=32 lines with 1 column for the number of the CNC process or of the external mechanism.

Value Range of the Column

(1) = Mechanism number [0...31]

Example MCP1

Read the CNC process number of module 5 of device 03 of the module configuration.

Assumption: The following CNC processes have been defined:

- CNC process number 1
- CNC process number 4

FI Command	03_BR_MCS1_5
Answer	
Line	Column 1
1	1
2	4

Reference to Literature

Additional information regarding module configuration and the construction of the "Moduldef.ini" file can be located in the following Rexroth Indramat documentation:

"Diagnostics and Message System for HMI System ProVi", chapter "Configure Moduldef.ini", DOK-MTC200-DIAG*PROVI*-ANW1-EN-P.

SFC Data of the Module Configuration: MCS

MTCX Device Group

Name

MCS **M**odule **C**onfiguration: **S**FC Information

Explanation

All SFC data of a particular module is read out from the "Moduldef.ini"' file. This data is located in the [LW]:\MT-CNC\CONFIG directory and 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 of the SFC data with respect to the module of a device from the module configuration of the MTCX device group.

BR_MCS1_(1) **(Single Read)**

BC_MCS1_(1) **(Cyclic Read)**

BB_MCS1_(1) **(Break Cyclic Read)**

(1) = module number [0...99]

Construction of Answer

The number of lines depends on the number of configured Indrastep Step Chains of a device. Each line contains a column for the name of the Indrastep Step Chains.

Value Range of the Column

1 = Name of the Indrastep Step Chain [Format W.X.Y.Z]

Format W.X.Y.Z	Value range
W	max. 9 ASCII characters
X	max. 9 ASCII characters OPTIONAL !
Y	max. 9 ASCII characters OPTIONAL !
Z	max. 9 ASCII characters OPTIONAL !

Example MCS1

Read the name of the Indrastep Step Chain of module 5 from device 03 of the module configuration.

<u>Assumption:</u> The following Indrastep Step Chains have been defined:

- ISFB_1
- FB_US.ISFB_3
- FB_US.ISFB_3.SW1.ABBA
- FB_US.ISFB_3.SW1.ABBA

FI Command	03_BR_MCS1_5
Answer	
Line	**Column**
1	ISFB_1
2	FB_US.ISFB_3
3	FB_US.ISFB_3.SW1
4	FB_US.ISFB_3.SW1.ABBA

Reference to Literature Additional information regarding module configuration and the construction of the "Moduldef.ini" file can be located in the following Rexroth Indramat documentation:

"Diagnostics and Message System for HMI System ProVi", chapter "Configure Moduldef.ini", DOK-MTC200-DIAG*PROVI*-AW0x-EN.

Maximal Feedrate Override: MFO

MTCX Device Group

Name **MFO** **M**aximal **F**eedrate **O**verride

Explanation The value of the maximal feedrate override of the selected device of the MTCX device group is read out.

FI Command

CR_MFO1_(1) **(Single Read)**

CC_MFO1_(1) **(Cyclic Read)**

CB_MFO1_(1) **(Break Cyclic Read)**

(1) = CNC process number [0...6]

Construction of Answer The following table shows the general construction of the answer of the FI command MFO1. One line with three columns is outputted for the identifier, the current value of the maximal feedrate override and the unit [%].

Line 1	Column 1	....	Column 3

Value Range/Meaning of Columns

1 = Identifier [MAX]

2 = Value of maximum feedrate override [0...100]

3 = Unit [%]

Example MFO1 Read the current value of the maximal feedrate override in CNC process 0 of device address 00.

FI Command	00_CR_MFO1_0		
Answer			
Line	Column 1	Column 2	Column 3
1	[MAX]	100	[%]

Maximal Feedrate: MFR

MTCX Device Group

Name **MFR** **M**aximal **F**eed**R**ate

Explanation The value of the maximal feedrate of the selected device of the MTCX device group is read out.

FI Command

CR_MFR_(1) **(Single Read)**

CC_MFR_(1) **(Cyclic Read)**

CB_MFR_(1) **(Break Cyclic Read)**

(1) = CNC process number [0...6]

Construction of Answer The following table shows the general construction of the answer of the FI command MFR. One line with three columns is outputted for the identifier, the current value of the feedrate override and the unit, according to the settings of the parameters [%].

Line 1	Column 1	....	Column 3

Value Range/Meaning of Columns

1 = Identifier	[F = Feedrate]
2 = Feedrate overrides	[format acc. to settings of the parameters]
3 = Unit	[acc. to settings of axis parameter]

Example MFR Read the value of the feedrate override in CNC process 0 of device address 00.

FI Command	00_CR_MFR_0		
Answer			
Line	**Column 1**	**Column 2**	**Column 3**
1	F	120000.0	[mm/min]

Maxim Rapid Override: MRO

MTCX Device Group

Name **MRO** **M**aximal **R**apid **O**verride

Explanation The value of the maximal rapid override of the selected device of the MTCX device group is read out.

FI Command

CR_MRO1_(1) **(Single Read)**

CC_MRO1_(1) **(Cyclic Read)**

CB_MRO1_(1) **(Break Cyclic Read)**

(1) = CNC process number [0...6]

Construction of Answer The following table shows the general construction of the answer of the FI command MRO1. One line with three columns is outputted for the identifier, the current value of the maximal rapid override and the unit [%].

Line 1	Column 1	....	Column 3

Value Range/Meaning of Columns

1 = Identifier	[RMAX]
2 = Value of maximum rapid override	[0...100]
3 = Unit	[%]

Example MRO1 Read the maximal value of the rapid override in CNC process 0 of device address 00.

FI Command	00_CR_MRO1_0		
Answer			
Line	**Column 1**	**Column 2**	**Column 3**
1	[RMAX]	100	[%]

Maximal Spindle Override: MSO

MTCX Device Group

Name **MSO** **M**aximal **S**pindle **O**verride

Explanation The value of the maximal spindle override of the selected device of the MTCX device group is read out.

FI Command

CR_MSO1_(1)_(2) (Single Read)

CC_MSO1_(1)_(2) (Cyclic Read)

CB_MSO1_(1)_(2) (Break Cyclic Read)

(1) = CNC process number [0...6]

(2) = number of spindle [1...3]

Construction of Answer The following table shows the general construction of the answer of the FI command MSO1. One line with three columns is outputted for the identifier, the value of the maximal spindle override and the unit [%].

Line 1	Column 1	....	Column 3

Value Range/Meaning of Columns

1 = Identifier [SMAX]

2 = Value of maximum rapid override [0...100]

3 = Unit [%]

Example MSO1 Read the maximal value of the spindle override in CNC process 0 of device address 00.

FI Command	00_CR_MSO1_0_1		
Answer			
Line	Column 1	Column 2	Column 3
1	[SMAX]	100	[%]

Maximal Spindle Speed: MSS

MTCX Device Group

Name **MSS** **M**aximal **S**pindle **S**peed

Explanation The value of the maximal spindle speed of the selected device of the MTCX 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) = CNC process number [0...6]

(2) = number of spindle [1...3]

Construction of Answer The following table shows the general construction of the answer of the FI command MSS. One line with three columns is outputted for the identifier, the speed and the unit [1/min].

Line 1	Column 1	....	Column 3

Value Range/Meaning of Columns

1 = Identifier [S = spindle]

2 = Speed [format acc. to settings of the parameters]

3 = Unit [1/min]

Example MSS Read the maximal value of the speed of the 1st spindle in CNC process 0 of device address 00.

FI Command	00_CR_MSS_0_1		
Answer			
Line	**Column 1**	**Column 2**	**Column 3**
1	S	7500.0	[1/min]

Machine Table Data: MTD

MTCX Device Group

Name **MTD** **M**achine **T**able **D**ata

FI Command Outputs the (user) machine table data

CR_MTD1_(1)_(2)_(3)_(4)_(5) **(Single Read)**

CC_MTD1_(1)_(2)_(3)_(4)_(5) **(Cyclic Read)**

(1) = Page number [1...299]

(2) = Run variable 1 [-1000 ... +1000]

(3) = Run variable 2 [-1000 ... +1000]

(4) = Element number [1...1000]

(5) = Name [1...13]

Answer

Data element
10110100
Read from MD page 152 via LV1: 0 and LV2: 1 the 13th element of type UDINT CR_MTD_152_0_1_13_8 150

FI Command Writes the (user) machine table data

CW_MTD1_(1)_(2)_(3)_(4)_(5) **(Single Write)**

(1) = Page number [1...299]

(2) = Run variable 1 [-1000 ... +1000]

(3) = Run variable 2 [-1000 ... +1000]

(4) = Element number [1...1000]

(5) = Identifier code [1...13]

Code	Identifier	Byte	Min. value	Max. value
1	BOOL	1	0	
2	BYTE	1	0	
3	WORD	2	0	
4	DWORD	4	0	
5	STRING	max. 220 bytes		
6	SINT	1	- 128	127
7	INT	2	- 32768	32767
8	DINT	4	- 2147483648	2147483647
9	USINT	1	0	255
10	UINT	2	0	65535
11	UDINT	4	0	4294967295
12	REAL	4	- 3.4 E38	3.4 E38
13	DREAL	8	- 1.7 E308	1.7 E308

Value to be written

Variable value [acc. to the display format of the BOF]

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

Construction of Answer

One line with one column is outputted for the acknowledgement of the FI command issued. The following hereby mean:

(P_ACK) = **P**ositive **ACK**nowledge Value has been successfully transmitted

Status of NC Events: NEV

MTCX Device Group

Name

NEV **N**C **EV**ent

FI Command

Read the status of an NC event of the selected device of the MTCX device group.

CR_NEV_(1){_(2)} (Single Read)

(1) = CNC process number [0...6]

(2) = number of the NC event [0...31] ! Optional !

Note: If the optional parameters is not specified then the status of all NC events is outputted.

Construction of Answer

One line is outputted, whereby the number of columns depends on the number of event statuses that is requested. When the optional parameter has not been specified, the answer consists of one line with 32 columns. If the optional parameter has been specified then the answer consists of one line with one column which contains the status [0] or [1] of the requested NC event.

Example NEV

Read the status of the 17th event in CNC process 0 of device address 00.

FI Command	00_CR_NEV_0_17
Answer	

Line	Column 1
1	0

FI Command Write the status of an NC event of the selected device of the MTCX device group.

CW_NEV_(1)_(2) **(Single Write)**

(1) = CNC process number [0...6]

(2) = event number [0...31]

Value to be written status of NC Event: 0 = delete NC event; 1 = set NC event

Construction of Answer One line is outputted with a column for acknowledgement of whether or not the FI command has been successfully executed.

(P_ACK) = **P**ositive **ACK**nowledge NC event has been deleted or set

Example NEV Set the 17th NC event in CNC process 0 at device address 00.

FI Command	00_CW_NEV_0_17 Value to be written: 1
Answer	
Line	**Column 1**
1	(P_ACK)

Selection of NC Memory: NMM

MTCX Device Group

Name **NMM** **N**C-**M**e**M**ory

Explanation Used in selecting the NC memory for the processing of the NC program. The NC programs are administered on the NC in two NC memories. During the processing of an NC program, e.g. in NC memory A, another NC program package can be transmitted into NC memory B. Both NC memories (A and B) are identically constructed and completely equal; however, only one NC memory can ever be active at any given time.

FI Command **CW_NMM** **(Single Write)**

Value to be written NC memory [1 = memory A; 2 = memory B]

Note: The selection of an NC memory is only possible 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.

Construction of Answer One line with one column is outputted for the acknowledgement of the FI command issued. The following hereby mean:

(P_ACK) = **P**ositive **ACK**nowledge the selected NC memory has been selected

Example NMM Select NC memory B at device 00 for processing the NC program.

FI Command	00_CW_NMM Value to be written: 2
Answer	
Line	Column 1
1	(P_ACK)

Reference to Literature

You can find more details on selecting the NC memory in the Rexroth Indramat documentation:

"MTC200/MT-CNC xxVRS GUI", chapter "Survey of Operations: NC Program Administration",
DOK-MTC200-GBO*GEN*Vxx-AW0x-EN.

Read-Out NC Parameters: NPA

MTCX Device Group

Name

NPA **N**C-**PA**rameter

FI Command

Read-out a parameter line

BR_NPA1_(1)_(2) **(Single Read)**

(1) = parameter record number [1..99]

(2) = parameter number [A00.000..Cxx.120]

Construction of Answer

The following table shows the general construction of the answer of the FI command NPA1. One line with 3 columns is outputted for the identifier, the value and the name.

Line 1	Column 1	Column 2	Column 3

Value Range/Meaning of Columns

1 = Identifier Parameter ID[max. 32 ASCII characters].

2 = Value [ASCII text]

3 = Name [unit, related to the value or empty]

Example NPA1

Return the parameter line from parameter record 10 with parameter number B00.007.

Assumption:

Parameter record 10 has been created and process 00 has been defined. The following information is located here: max. track acceleration 75 mm/sec^2

FI Command		00_BR_NPA1_10_B00.007
Line	Column	Answer
1	1	B00.007
	2	75
	3	mm/sec^2

FI Command

Read-out of several parameter lines from a parameter record.

BR_NPA2_(1)_(2)_(3) **(Single Read)**

(1) = parameter record number [1..99]

(2) = parameter number [from] [A00.000..Cxx.120]

(3) = parameter number [to] [A00.000..Cxx.120]

Construction of Answer

The following table shows the general construction of the answer of the FI command NPA2. As many lines as are requested are outputted – each with three columns – for the identifier, the value and the name.

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

Value Range/Meaning of Columns

1 = identifier [max. 32 ASCII characters]

2 = value [ASCII text]

3 = name [unit, related to the value or empty]

Example NPA2

Return the parameter lines from parameter record 10 of parameter number A00.000 to parameter number A00.001.

Assumption:

Parameter record 10 has been created and contains the following information are this location.

FI Command		00_BR_NPA2_10_A00.000_A00.001
Line	**Column**	**Answer**
1	1	A00.000
	2	Master
	3	
2	1	A00.001
	2	process 1
	3	

FI Command

Read-out of a particular element of a parameter line.

BR_NPA3_(1)_(2)_(3) (Single Read)

(1) = parameter record number [1.99]

(2) = parameter number [A00.000..Cxx.120]

(3) = element number [1..1000]

Construction of Answer

The following table shows the general construction of the answer of the FI command NPA3 . One line with one column for either the name or value or designated name is outputted.

Line 1	Column 1

Value Range/Meaning of Columns

1 = Name/value/designated name [ASCII text]

Example NPA3

Return element 1 of the parameter line from parameter record 10 with parameter number C01.079.

Assumption:

The parameter record has been created and contains the following information are this location:

FI Command		00_BR_NPA3_10_C01.079_19
Line	**Column**	**Answer**
1	1	Required value (here 19) from existing compensation table of axis 1.

FI Command

Read-out of all elements from a parameter line (such as NPA1).

BR_NPA4_(1)_(2) **(Single Read)**

(1) = parameter record number [1..99]

(2) = parameter number [A00.000..Cxx.120]

Construction of Answer

The following table shows the general construction of the answer of the FI command NPA4. One line with 3 columns is outputted for the identifier, the value and the name.

Line 1	Column 1	Column 2	Column 3

Value Range/Meaning of Columns

1 = Identifier [max. 32 ASCII characters]

2 = Value [ASCII text]

3 = Name [unit, related to the value or empty]

Example NPA4

Return the parameter line from parameter record 10 with parameter number A00.000.

Assumption:

The parameter record has been created and contains the following information are this location: Master.

FI Command		00_BR_NPA4_10_A00.000
Line	**Column**	**Answer**
1	1	A00.000
	2	master
	3	

Note: The commands supported in this version are listed using the command 00_NPA1_?.

Activate NC Compiler: NPC

MTCX Device Group

Name

NPC **N**C-**P**ackage **C**ompiling

FI Command

Compiles the selected NC package.

BR_NPC1_(1) **(Single Read)**

(1) = number in NC-package directory [1...99]

Construction of Answer

The following table shows the general construction of the answer of the FI command NPC1 . One line with three columns is outputted for the job ID, FI command and the FI-job error code.

Line 1	Column 1	...	Column 3

Value Range/Meaning of Columns

(1) = Job ID	[01...20] (refer to chapter 7.1 FI Commands for the MPCX Device Group, IFJ).
2 = FI command	[string, in accordance to chapter 6.1, Elements of the FI Command]
3 = FI-job error code	(see chapter 8 Error Codes)

Example NPC

Compile the 2nd NC package.

FI Command	00_BR_NPC1_2		
Answer			
Line	**Column 1**	**Column 2**	**Column 3**
1	01	00_BR_NPC1_2	0

Activate NC Download: NPD

MTCX Device Group

Name

NPD **N**C-**P**ackage **D**ownload

FI Command

Loads the selected NC package **without setup lists** into the specified device.

BW_NPD1_(1)_(2) (Single Write)

(1) = NC memory	[1 = NC memory A; 2 = NC memory B]
(2) = number in NC-package directory	[1...99]

Value to be written

Initialization 1 = trigger NC download

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

Construction of Answer

The answer of the FI command NPD1 consists of three lines, each with one column. The following hereby mean:

Line 1 = Job ID	[01...20] (refer to chapter 7.1 FI Commands for the MPCX Device Group, IFJ).
Line 2 = FI command	[string, in accordance to chapter 6.1, Elements of the FI Command]
Line 3 = FI-job error code	(see chapter 8 Error Codes)

Example NPD1

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

FI Command	00_BW_NPD1_1_2
Value to be written	**1**
Answer	
Line	**Column 1**
1	02
2	00_BW_NPD1_1_2
3	0

Note: If an attempt is made to re-transmit an NC package that is already in the device then the routine "DataTransfer" terminates with error code 1030 (see chapter 8, Error Codes).

FI Command

Loads the selected NC package **with setup lists** into the specified device.

BW_NPD2_(1)_(2) **(Single Write)**

(1) = NC memory [1 = NC memory A; 2 = NC memory B]

(2) = Number in NC-package directory [1...99]

Value to be written

Initialization 1 = trigger NC download

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

Construction of Answer

The answer of the FI command NPD2 consists of three lines, each with one column. The following hereby mean:

Line 1 = Job ID [01...20] (refer to chapter 7.1 FI Commands for the MPCX Device Group, IFJ).

Line 2 = FI command [string, in accordance to chapter 6.1, Elements of the FI Command]

Line 3 = FI-job error code (see chapter 8 Error Codes)

Example NPD2

Load the 3rd NC package (**with setup lists**) in NC memory B of the device with device address 00.

FI Command	**00_BW_NPD2_2_3**
Value to be written	**1**
Answer	
Line	**Column 1**
1	03
2	00_BW_NPD2_2_3
3	0

Note: If an attempt is made to re-transmit an NC package that is already in the device then the routine "DataTransfer" terminates with error code 1030 (see chapter 8, Error Codes).

Comments on NP3 and NP4

These FI commands have been optimized for speed. They are particularly suitable for transmitting small NC programs ((standard value: up to a max. of 100 NC program lines). As the transfer time is less than two seconds for smaller NC programs, then a status prompt is not meaningful. For this reason, the function interface job administration has been dispensed with for these FI commands (see chapter 7.1, FI Commands for the MPCX Device Group, IFJ).

Note: The "DataTransfer" routine dwells for the entire transmission time (dwell time = transmission time). This only applies for these FI commands.

FI Command

Loads the selected NC package **with setup lists** into the specified device.

BW_NPD3_(1)_(2) **(Single Write)**

(1) = NC memory	[1 = NC memory A; 2 = NC memory B]
(2) = number in NC-package directory	[1...99]

Value to be written

Initialization 1 = trigger NC download

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

Construction of Answer

The answer of the FI command NPD3 consists of three lines, each with one column. The following hereby mean:

Line 1 = Job ID	[01...20] (refer to chapter 7.1 FI Commands for the MPCX Device Group, IFJ).
Line 2 = FI command	[string, in accordance to chapter 6.1, Elements of the FI Command]
Line 3 = FI-job error code	(see chapter 8 Error Codes)

Example NPD3

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

FI Command	**00_BW_NPD3_1_2**
Value to be written	**1**
Answer	
Line	**Column 1**
1	02
2	00_BW_NPD3_1_2
3	0

Note: If an attempt is made to re-transmit an NC package that is already in the device then the routine "DataTransfer" terminates with error code 1030 (see chapter 8, Error Codes).

FI Command

Loads the selected NC package **with setup lists** into the specified device.

BW_NPD4_(1)_(2) **(Single Write)**

(1) = NC memory	[1 = NC memory A; 2 = NC memory B]
(2) = Number in NC-package directory	[1...99]

Value to be written

Initialization 1 = trigger NC download

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

Construction of Answer

The answer of the FI command NPD4 consists of three lines, each with one column. The following hereby mean:

Line 1 = Job ID	[01...20] (refer to chapter 7.1 FI Commands for the MPCX Device Group, IFJ).
Line 2 = FI command	[string, in accordance to chapter 6.1, Elements of the FI Command]
Line 3 = FI-job error code	(see chapter 8 Error Codes)

Example NPD4

Load the 3rd NC package (**with setup lists**) in NC memory B of the device with device address 00.

FI Command	**00_BW_NPD4_2_3**
Value to be written	**1**
Answer	
Line	**Column 1**
1	03
2	00_BW_NPD4_2_3
3	0

Note: If an attempt is made to re-transmit an NC package that is already in the device then the routine "DataTransfer" terminates with error code 1030 (see chapter 8, Error Codes).

Read NC Package Directory: NPI

MTCX Device Group

Name **NPI** **N**C-**P**ackage D**I**rectory

Explanation Reads the entries of the NC package directories of the BOF.

FI Command **BR_NPI** **(Single Read)**

Construction of Answer

The following table shows the general construction of the answer of the FI command NPI. The answer consists of up to a maximum of n=99 lines, each with 5 columns.

Line 1...n:	**Column 1**	**...**	**Column 5**

Value Range/Meaning of Columns

1 =	Number in NC package directory	[01...99]
2 =	Name of the NC package	[max. 32 ASCII characters]
3 =	Length of the NC package	[bytes]
4 =	Date of creation/last change to NC package	[DD.MM.YY]
5 =	Time of creation/last change to NC package	[HH:MM:SS]

Example NPI Read the entries in the NC package directory at device address 00.

FI Command		00_BR_NPI
Line	**Column**	**Answer**
1	1	01
	2	KEY1
	3	3579
	4	16.05.99
	5	10:41:08
2	1	10
	2	KEY2
	3	4589
	4	18.05.99
	5	10:12:10

Reference to Literature You can find more details regarding the construction of NC packages in the Rexroth Indramat documentation:

"MTC200/MT-CNC NC Programming Instructions xxVRS", chapter „Sub-Programs", DOK-MTC200-NC**PRO*Vxx-AW0x-EN

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

MTCX Device Group

Name **NPS** **N**C-**P**rogram **S**election

Explanation Used in selecting the NC program located for processing in the active NC memory. The NC programs are administered on the NC in two NC memories. During the processing of an NC program, e.g. in NC memory A, another NC program package can be transmitted into NC memory B. Both NC memories (A and B) are identically constructed and completely equal; however, only one NC memory can ever be active at any given time.

FI Command

CW_NPS_(1) **(Single Write)**

(1) = CNC process number [0...6]

Value to be written Number in NC-package directory [1...99]

Note: Selection of the NC program is only possible when there is a valid NC program package in the active NC memory Otherwise, the request is acknowledged by an error message. The value to be written is passed to the "acValue" parameter as an ASCII value in the "DataTransfer" routine.

Construction of Answer

One line with one column is outputted for the acknowledgement of the FI command issued. The following hereby mean:

(P_ACK) = **P**ositive **ACK**nowledge	the selected NC program has been selected

Example NPS

Select CNC process number 0 for processing NC program 01 in the active NC memory. Assumption:

- There is a valid NC program package in the active NC memory.

FI Command	**00_CW_NPS_0** **Value to be written: 1**
Answer	
Line	**Column 1**
1	(P_ACK)

Reference to Literature

You can find more details on selecting the NC program and the NC memory in the Rexroth Indramat documentation:

"MTC200/MT-CNC xxVRS GUI", chapter "Survey of Operations: NC Program Administration",
DOK-MTC200-GBO*GEN*Vxx-AW0x-EN.

Next Tool Number: NTN

MTCX Device Group

Name

NTN **N**ext **T**ool-**N**umber

Explanation

Returns the next pre-selected tool number of the selected device of the MTCX device group.

FI Command

CR_NTN_(1) **(Single Read)**

CC_NTN_(1) **(Cyclic Read)**

CB_NTN_(1) **(Break Cyclic Read)**

(1) = CNC process number [0...6]

Construction of Answer

One line with two columns is outputted for the identifier [T= Tool] and for the next tool number.

Example NTN

Read the next tool number in CNC process 0 of device address 00.

FI Command	**00_CR_NTN_0**	
Answer		
Line	**Column 1**	**Column 2**
1	T	1

Reading and Writing NC Variables: NVS

MTCX Device Group

Name **NVS** **N**C-**V**ariable **S**ingle

Explanation Read the NC variable of the selected device of the MTCX device group.

FI Command

CR_NVS_(1)_(2){_(3)} **(Single Read)**

CC_NVS_(1)_(2){_(3)} **(Cyclic Read)**

CB_NVS_(1)_(2){_(3)} **(Break Cyclic Read)**

(1) = CNC process number [0...6]

(2) = NC variable number {from} [0...255]

(3) = NC variable number {to} [0...255] !Optional !

Note: If the optional parameter is specified then up to 20 NC variables are outputted.

Construction of Answer One line with a maximum of 20 columns containing the corresponding value of the requested NC variable is outputted.

Note: If the requested NC variable does not exist then [--] is entered in the corresponding column.

Example NVS without optional Parameter Read the value of the NC variable numbered 1 at device address 00 in CNC process 0.

FI Command	00_CR_NVS_0_1
Answer	
Line	Column 1
1	1.111000

Example NVS with optional Parameter

Read the value of the 1st NC variable to the 3rd NC variable at device address 00 in CNC process 0. Assumption: The 2nd NC variable is not defined.

FI Command	00_CR_NVS_0_1_3		
Answer			
Line	Column 1	Column 2	Column 3
1	1.111000	--	23.100000

Explanation

Write an NC variable of the selected device of the MTCX device group.

FI Command

CW_NVS_(1)_(2) **(Single Write)**

(1) = CNC process number [0...6]

(2) = NC variable number [0...255]

Value to be written

NC variable [format acc. to settings of the parameters]

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

Construction of Answer

One line with one column is outputted for the acknowledgement of the FI command issued. The following hereby mean:

(P_ACK) = **P**ositive **ACK**nowledge variable has been written.

Example NVS

Write the value 1.111000 in the 1st NC variable in CNC process 0 at device address 00.

FI Command	00_CW_NVS_0_1 Value to be written: 1.111000
Answer	
Line	Column 1
1	(P_ACK)

Optimal Position Distance from Axes: OPD

MTCX Device Group

Name **OPD** **O**ptimal **P**osition **D**istance

Explanation The optimal position distance of a selected axis of the MTCX 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 optimal position distance of the selected axis of the device specified, related to the code of the axis meaning.

CR_OPD1_(1)_(2) **(Single Read)**

CC_OPD1_(1)_(2) **(Cyclic Read)**

CB_OPD1_(1)_(2) **(Break Cyclic Read)**

(1) = CNC process number [0...6]

(2) = Axis meaning [0...11; 20];

Construction of Answer The following table shows the general construction of the answer of the FI command OPD1. One line with three columns for the name of the axis, the value of the optimal position distance and the unit is outputted in accordance to the settings of the process parameters.

Line 1	Column 1	...	Column 3

Value Range/Meaning of the Columns

1 = Axis name [Xi, Yi, Zi, Ui, Vi, Wi, Ai, Bi, Ci, Si] where i = [,1,2,3]

2 = Optimal position distance [acc. to settings of the process parameter]

3 = Unit [mm, inch]

Note: If the specified axis is not defined in the selected CNC process then the answer in all columns is [--].

Example OPD1 Read the optimal position distance of the Z axis in CNC process 0 of device address 00.

FI Command	00_CR_OPD1_0_2		
Answer			
Line	**Column 1**	**Column 2**	**Column 3**
1	Z	-5.9897	[mm]

FI Command

Output of the optimal position distance of the selected axis of the device specified, related to the physical axis number.

CR_OPD2_(1)	**(Single Read)**
CC_OPD2_(1)	**(Cyclic Read)**
CB_OPD2_(1)	**(Break Cyclic Read)**

(1) = physical axis number [1...32]

Construction of Answer

The following table shows the general construction of the answer of the FI command OPD2. One line with three columns for the name of the axis, the position and the unit is outputted in accordance to the settings of the process parameters.

Line	Column 1	...	Column 3

Value Range/Meaning of the Columns

1 = Axis name	[Xi, Yi, Zi, Ui, Vi, Wi, Ai, Bi, Ci, Si] where i = [,1,2,3]
2 = Optimal position distance	[acc. to settings of the process parameter]
3 = Unit	[mm, inch]

Note: If the specified axis is not defined in the selected CNC process then the answer in all columns is [--].

Example OPD2

Read the optimal position distance of the Z axis (physical axis number = 3) at the device address 00.

FI Command	**00_CR_OPD2_3**		
Answer			
Line	**Column 1**	**Column 2**	**Column 3**
1	Z	-5.9897	[mm]

Process Axis Configuration Parameter: PAC

MTCX Device Group

Name **PAC** **P**rocess **A**xis **C**onfiguration Parameter

Explanation The axis configuration data (parameters) of a process is returned.

FI Command Output of the axis configuration parameters of all CNC processes.

BR_PAC1 **(Single Read)**

Construction of Answer The following table shows the general construction of the answer of the FI command PAC1. The number of lines depends on the number of defined CNC processes. Every line consists of five columns for the CNC process number, the physical axis number, the main axis meaning, the main axis name and the axis type.

Line 1...n:	Column 1	....	Column 5

Value Range/Meaning of Columns

1 = CNC process number	[0...6]
2 = Physical axis number	[1...32]
3 = Main axis meaning	[see chapter 6.2, Data Tables]
4 = Main axis name	[Xi, Yi, Zi, Ui, Vi, Wi, Ai, Bi ,Ci, Si, --] (i=[ ], [1...3])
5 = Axis type	[see chapter 6.2, Data Tables]

Example PAC1 Read all processes of the axis configuration parameters at device address 00.

FI Command		00_BR_PAC1
Line	Column	Answer
1	1	0
	2	1
	3	0
	4	X1
	5	0x81
2	1	1
	2	2
	3	1
	4	Y1
	5	0x82
3	1	2
	2	3
	3	5
	4	--
	5	--

FI Command

Output of the axis configuration parameters of a CNC process.

BR_PAC2_(1) **(Single Read)**

(1) = CNC process number [0...6]

Construction of Answer

The following table shows the general construction of the answer of the FI command PAC2. One line with five columns is outputted for the CNC process number, the physical axis number, the main axis meaning, the main axis name and the axis type.

Line 1	Column 1	....	Column 5

Value Range/Meaning of Columns

1 = CNC process number	[0...6]
2 = Physical axis number	[1...32]
3 = Main axis meaning	[see chapter 6.2, Data Tables]
4 = Main axis name	[Xi, Yi, Zi, Ui, Vi, Wi, Ai, Bi ,Ci, Si, --] (i=[ ], [1...3])
5 = Axis type	[see chapter 6.2, Data Tables]

Example PAC2

Read the axis configuration parameters of process 0 at device address 00.

FI Command		00_BR_PAC2_0
Line	**Column**	**Answer**
1	1	0
	2	1
	3	0
	4	X1
	5	0x81

Programmed Feedrate: PFR

MTCX Device Group

Name

PFR **P**rogrammed **F**eed**R**ate

Explanation

The value of the programmed feedrate of the selected device of the MTCX device group is read out.

FI Command

CR_PFR_(1) **(Single Read)**

CC_PFR_(1) **(Cyclic Read)**

CB_PFR_(1) **(Break Cyclic Read)**

(1) = CNC process number [0...6]

Construction of Answer

The following table shows the general construction of the answer of the FI command PFR. One line with three columns is outputted for the identifier, the current value of the programmed feedrate and the unit, according to the settings of the parameters.

Line 1	Column 1	....	Column 3

Value Range/Meaning of Columns

1 = Identifier	[F = feedrate]
2 = Feedrate	[format acc. to settings of the parameters]
3 = Unit	[acc. to settings of the parameter]

Example PFR

Read the programmed feedrate in CNC process 0 of device address 00.

FI Command	**00_CR_PFR_0**		
Answer			
Line	**Column 1**	**Column 2**	**Column 3**
1	F	120000.0	[mm/min]

Read NC Program Directory: PPD

MTCX Device Group

Name **PPD** **P**art-**P**rogram **D**irectory

FI Command Reads the entries of the NC program directories of the BOF.

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

(1) = number in NC-package directory [1...99]

(2) = CNC process number [0...6]

Construction of Answer The following table shows the general construction of the answer of the FI command PPD. The answer consists of up to a maximum of n=99 lines, each with 5 columns.

Line 1...n:	Column 1	...	Column 5

Value Range/Meaning of Columns

1 = NC program number [00...99]

2 = Name of the program [max. 32 ASCII characters]

3 = Length of the program [bytes]

4 = Date of creation/last change to program [DD.MM.YY]

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

Example PPD Read the entries in the NC program directory of NC package number 1 of CNC process 0 at device address 00.

FI Command		**00_BR_PPD_1_0**
Line	**Column**	**Answer**
1	1	01
	2	TransAM
	3	3579
	4	16.05.99
	5	10:41:08
2	1	10
	2	BMW 3 series
	3	4589
	4	18.05.99
	5	10:12:10

Export NC Program: PPN

MTCX Device Group

Name **PPN** **P**art-**P**rogram **N**C

FI Command Transfers an NC program from the NC program directory into an ASCII file.

BR_PPN_(1)_(2)_(3)_(4) **(Single Read)**

(1) = Number in NC-package directory	[1...99]
(2) = CNC process number	[0...6]
(3) = Number of the NC program	[1...99]
(4) = NC record numbering	[0 = without number; 1 = with number]

Construction of Answer The answer of the FI command PPN consists of one line and column for details of the drive, the directory and the file containing the NC program.

Example PPN Import in a file - without NC record numbering – the NC program with the NC program number 1 of the 2nd NC package of CNC process 0 at device address 00.

FI Command		00_BR_PPN_2_0_1_0
Line	**Column**	**Answer**
1	1	C:\MT-CNC\ANLAGE01\MT_TEMP\T1010001.TMP

Extract of file "C:\MT-CNC\ANLAGE01\MT_TEMP\T1010001.TMP":

```
START

SPF 1 [select active spindle]
T1 BSR .M6
G90 G96 G54 S1 2000 F5000 M03

G00 X60 Y-30
Z-6 [infeed]
G01 X60 Y0 F2000
X5 Y0
Z100
M05 [stop spindle]

T1 BSR .M6
BST .START
END OF PROGRAM
```

Import NC Program: PPN

MTCX Device Group

Name **PPN** **P**art-**P**rogram **N**C

FI Command Transfers an NC program from an ASCII file into the NC program directory.

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

(1) = Number in NC-package directory	[1...99]
(2) = CNC process number	[0...6]
(3) = Number of the NC program	[1...99]
(4) = NC record numbering	[0 = without number; 1 = with number]
(5) = Is the NC package directory entry empty?	[0 = without check (default); 1 = with check] ! Optional !
(6) = Complete specification of the directory	[LW:\..\X.Y]

Note: This FI command has no "value to be written".

Construction of Answer One line with one column is outputted for the acknowledgement of the FI command issued. The following hereby mean:

(P_ACK) = **P**ositive **ACK**nowledge NC program has been exported

Example PPN Export the NC program from file "C:\Data\T1010001.TMP" into NC program number 1 of the 2nd NC package of CNC process 0 at device address 00.

FI Command		00_BW_PPN_2_0_1_0_1_C:\Data\T1010001.TMP
Line	**Column**	**Answer**
1	1	(P_ACK)

Change Name of an NC Program: PPP

MTCX Device Group

Name **PPP** **P**art **P**rogram **P**ackage

Explanation The name of an NC program package of the selected device of the MTCX device group is changed.

FI Command

BA_PPP_(1) **(Single Alternate)**

(1) = NC program package [1...99]

Value to be written Name of the NC program package [max. 32 ASCII characters]

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

Construction of Answer One line with one column is outputted for the acknowledgement of the FI command issued. The following hereby mean:

(BOF_FCT_OK) = **BOF_F**un**CT**ion**_OK** program package has been renamed.

Example PPP

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

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

Reading an NC Record: PPS

MTCX Device Group

Name **PPS** **P**art **P**rogram **S**equence

Explanation An NC record of an NC program from the selected device of the MTCX device group is read out.

FI Command **CR_PPS_(1)_(2)_(3)_(4)** **(Single Read)**

(1) = NC memory [1=memory A, 2=memory B]

(2) = CNC process number [0...6]

(3) = NC program number [0...99]

(4) = NC record number [1...1000]

Construction of Answer One line with one column containing the requested NC record is outputted.

Example PPS Read NC record number 2 from NC program memory A, CNC 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 You can find more details regarding the construction of an NC sequence in the Rexroth Indramat documentation:

“NC Programming Instructions Vxx", chapter „Elements of an NC Record“, DOK-MTC200-NC**PRO*Vxx-AW0x-EN

Programmed Spindle Speed: PSS

MTCX Device Group

Name **PSS** **P**rogrammed **S**pindle **S**peed

Explanation The value of the programmed spindle speed of the selected device of the MTCX device group is read out.

FI Command

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

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

CB_PSS_(1)_(2) **(Break Cyclic Read)**

(1) = CNC process number [0...6]

(2) = number of spindle [1...3]

Construction of Answer The following table shows the general construction of the answer of the FI command PSS. One line with three columns is outputted for the axis name, the speed and the unit [1/min].

Line 1	Column 1	....	Column 3

Value Range/Meaning of Columns

1 = Identifier [S = spindle]

2 = Speed [format acc. to settings of the parameters]

3 = Unit [1/min]

Example PSS Read the speed of the 1st spindle in CNC process 0 of device address 00.

FI Command	00_CR_PSS_0_1		
Answer			
Line	**Column 1**	**Column 2**	**Column 3**
1	S	7500.0	[1/min]

Reference to Literature You can find more details regarding the construction of an NC sequence in the Rexroth Indramat documentation:

“NC Programming Instructions Vxx", chapter „Elements of an NC Record“, DOK-MTC200-NC**PRO*Vxx-AW0x-EN

Process Tool Management Configuration: PTC

MTCX Device Group

Name **PTC** **P**rocess **T**ool Management **C**onfiguration

Explanation Returns the most important process parameter data of the tool management of the selected device of the NTCX device group.

FI Command Read tool management data of all defined CNC processes.

BR_PTC1 **(Single Read)**

BC_PTC1 **(Cyclic Read)**

Construction of Answer The following table shows the general construction of the answer of the FI command PTC1. The number of lines depends on the number of defined CNC processes. Each line consists of 9 columns for the returned values.

Line 1...n:	Column 1	...	Column 9

Value Range/Meaning of the Columns

1 = CNC process number	[0...6]
2 = Process name	
3 = Tool management	[YES, NO]
4 = Tool memory	[[MAGAZINE], [TURRET]]
5 = Endlessly turning tool memory	[YES, NO]
6 = Number of tool memory locations	[0...999]
7 = Number of tool spindles	[0...4]
8 = Number of tool grippers	[0...4]
9 = Axis number of tool axis	[0...20]

Note: If there is no tool management (column 3: NO), then all part-results from column 4 are marked as [--].

Example PTC1 Deliver the process parameter data of the defined processes. This example assumes that there are two processes. One with tool management and one without.

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

FI Command		00_BR_PTC1
Line	**Column**	**Answer**
2	1	1
	2	TRANSFER
	3	NO
	4	--
	5	--
	6	--
	7	--
	8	--
	9	--

FI Command

Read tool management data of a CNC process.

BR_PTC2_(1) **(Single Read)**

BC_PTC2_(1) **(Cyclic Read)**

(1) = CNC process number [0...6]

Construction of Answer

The following table shows the general construction of the answer of the FI command PTC2. One line with 9 columns is outputted for the additional text.

Line 1	Column 1	...	Column 9

Meaning of the Column

1 = CNC process number [0...6]
2 = Process name
3 = Tool management [YES, NO]
4 = Tool memory [[MAGAZINE], [TURRET]]
5 = Endlessly turning tool memory [YES, NO]
6 = Number of tool memory locations [0...999]
7 = Number of tool spindles [0...4]
8 = Number of tool grippers [0...4]
9 = axis number of tool axis [0...20]

Note: If there is no tool management (column 3: NO), then all part-results from column 4 are marked as [--].

Note: If the requested process does not exist then there is no results line.

Example PTC2 Deliver the process parameter data of 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

Formatted Input / Output of SPS Variables: PVF

MTCX Device Group

Name **PVF** **P**LC **V**ariable **F**ormatted

Explanation Formatted reading and writing of SPS variables, arrays and structures.

FI Command Read SPS variables.

CR_PVF_(1) **(Single Read)**

CC_PVF_(1) **(Cyclic Read)**

CB_PVF_(1) **(Break Cyclic Read)**

(1) = identifier of the SPS variable [acc. to declaration part of SPS]

Construction of Answer One line with one column is outputted for simple variables. For array and structure variables, one line per element is outputted, depending on the number of elements.

Line 1...n:	Column 1

n = number of elements.

Note: Only defined SPS variables can be read and written. Addressing a non-declared variable results in an error message. A SPS variable can only be read when its data length does not exceed 240. (refer also to chapter 4.1, Guidelines).

Value Ranges ANSI / ASCII

The value range of the answer depends on the data type of the variable read. The following table informs you of the range in which the results string is to be expected when reading out a simple variable and into which C-data type this string can be converted without loss of information:

Data Type	**Value range**	**Can be Converted to C-Data Type**
BOOL	[0;1]	unsigned char
SINT	[-128...127]	char
INT	[-32768...32767]	short
DINT	[2147483648...2147483647]	long
USINT	[0...255]	unsigned char
UINT	[0...65535]	unsigned short
UDINT	[0...4294967295]	unsigned long
BYTE	[0x00...0xFF]	unsigned char
WORD	[0x0000....0xFFFF]	unsigned short
DWORD	[0x00000000...0xFFFFFFFF]	unsigned long
TIME	[0...4294967295]	unsigned long (msec)
CHAR	[$00...$20,!...~,$7F...$FF]	char
STRING	<String> whereby <String> string is with a maximal of as many characters as are defined for the string in the SPS	Char[xx+1]] +1 e.g. room for the zero byte
REAL	[-3.402823567E+38...3.402823567E+38]	Float

Note: An empty string can be recognized by simple double-inverted commas: “

All simple 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 answer depends on the data type of the variable read. The following table informs you of the value range in which to expect the binary value of a simple variable and how many bytes are included in the binary byte sequence:

Data Type	**Value range**	**Length (bytes)**
BOOL	[00$_H$...01$_H$]	1
SINT	[80 $_H$...7F $_H$] i.e. –128...127	1
INT	[8000 $_H$ (-32768)...7FFF $_H$ (32767)]	2
DINT	[80000000 $_H$ (-2147483648)... 7FFFFFFF $_H$ (2147483647)]	4
USINT	[00 $_H$ (0)...FF $_H$ (255)]	1
UINT	[00 $_H$ (0)...FFFF $_H$ (65535)]	2
UDINT	[0...4294967295]	4
BYTE	[0x00...0xFF]	1
WORD	[0x0000....0xFFFF]	2
DWORD	[0x00000000...0xFFFFFFFF]	4
TIME	[0...4294967295]	4
CHAR	[$00...$20,!...~,$7F...$FF]	1
STRING	<String> whereby <String> string is with a maximal of as many characters as are defined for the string in the SPS	XX+1
REAL	[-3.402823567E+38...3.402823567E+38]	4

Note: Binary array and structure elements are joined together without any spaces between (1-byte alignment).

Example 1 PVF

Read the value of the SPS variables "STK_TXT" in ASCII format from device address 00.

Assumption:

The "STK_TXT" variable is declared as a string in the SPS program.

FI Command		**00_CR_PVF_STK_TXT/1**
Line	**Column**	**Answer**
1	1	Repeat counter

Example 2 PVF

Read the value of the SPS array "BEG_END" in ANSI format from device address 00.

Assumption:

The "BEG_END" variable is declared as BYTE with 2 elements in the SPS program.

FI Command		00_CR_PVF_BEG_END/3
Line	**Column**	**Answer**
1	1	0x00
2		0x1F

Example 3 PVF

Read the value of the SPS structure "MSTRCT" in ASCII format from device address 00.

Assumption:

The "MSTRCT" variable is declared as a structure in the SPS program as follows.

```
TYP STRUCT
        T1      BOOL
        T2      CHAR
        T3      STRING[16]
        T4      TIME
END
```

FI Command		00_CR_PVF_MSTRCT/1
Line	**Column**	**Answer**
1	1	0
2		A
3		ROBOT AXIS X
4		2000

FI Command

Write SPS variables.

CW_PVF_(1) **(Single Write)**

(1) = identifier of the SPS variable [acc. to declaration part of SPS]

Value to be written

Value of data element (see Value Ranges, page 7-21)

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“.

Construction of Answer

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

(P_ACK) = **P**ositive **ACK**nowledge data element has been set

Value Range of the value to be written in ANSI / ASCII Format

The value ranges agree for the most part with the result-value ranges ANSI / ASCII during read access. ANSI umlauts are thereby converted into ASCII umlauts. Only ASCII umlauts are stored in the controls. For deviations to this, please refer to the following note:

Note: Strings are bracketed by two simple inverted commas ’ ’ . e.g. ’drill’

Special characters can be marked in accordance to DIN-1131 by a $ sign.

There are:

- $’’
- $$ $
- $R \r (Carriage Return)
- L n (Linefeed)
- $P \f (Formfeed)
- $T \t <Tab>
- $xx xx refers to a character written as a hexadecimal value. e.g. $20 (space)

Array and structure elements are separated by a space.

Value Range of the Value to be written in Binary Format

The value ranges agree with the binary result-value range during read access. For deviations to this, please refer to the following note:

Example 4 PVF

Write the value of the SPS variable "STK_TXT" at device address 00. The value is output in ANSI format.

Assumption:

The "STK_TXT" variable is declared as a string in the SPS program.

FI Command		00_CW_PVF_STK_TXT/3
Line	**Colum n**	**Answer**
1	1	(P_ACK)

Value to be written

Value of data element ’item counter’

Data code /3

Example 5 PVF

Write into the SPS byte array "BEG_END" at device address 00. The value is output in ANSI format.

Assumption:

The "BEG_END" variable is declared as a BYTE array with 2 elements in the SPS 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

Example 6 PVF

Write the value of element T3 of the SPS 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 SPS program as follows.

```
TYP STRUCT
        T1      BOOL
        T2      CHAR
        T3      STRING[16]
        T4      TIME
END
```

FI Command		**00_CW_PVF_MSTRCT.T3/2**
Line	**Column**	**Answer**
1	1	(P_ACK)

Value to be written

Value of data element	Binary sequence: 43 4F 55 4E 54 45 52 00
Data code	/2

Example 7 PVF

Write the value of the SPS 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 SPS program as follows.

```
TYP STRUCT
        T1      BOOL
        T2      CHAR
        T3      STRING[16]
        T4      TIME
END
```

For the exchange of binary data in a C program, the following 'C' data type can be used:

```
□□□□□□□□□□□□□□□□                □□□□□□□□□□□□□□□□□□□□□
                                □□□□□□□□□□ □□□□□□□ □□□□□ □□□ □□□□
□□□□□□
□□□□□□□□□□□□□□□
□
        □□□□□□□□□□□□□□          □□
        □□□□        □□
        □□□□        □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
        □□□□□□□□□□□□□□          □□
□□□□□□□□□□□□                    □□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□□               □□□□□□□□□□□□□□□□□□□□
```

FI Command		00_CW_PVF_MSTRCT/2
Line	**Column**	**Answer**
1	1	(P_ACK)

Value to be written Address of the C structure.

Value of data element &mstrct

Data code /2

Reading and Writing SPS Variables: PVS

MTCX Device Group

Name **PVS** **P**LC-**V**ariable **S**ingle

Explanation The following types of SPS variable can be read or written:

- BOOL, BYTE, SINT, USINT, CHAR, WORD, INT, UINT, STRING, DWORD, DINT, UDINT, TIME, REAL as well as imported structures and arrays.

FI Command Reading SPS variables.

CR_PVS_(1) **(Single Read)**

CC_PVS_(1) **(Cyclic Read)**

CB_PVS_(1) **(Break Cyclic Read)**

(1) = Identifier of the SPS variable

Note: Addressing a non-declared variable results in an error message. The length of the data must not exceed 240 bytes. (refer also to chapter 4.1, Guidelines).

Example 1 PVS Read the value of the SPS variable with identifier "IB_EXT24" at device address 00 in CNC process 0.

Assumption:

- the SPS variable with the identifier "IB_EXT24" is of the type "BOOL"

FI Command	00_CR_PVS_IB_EXT24
Answer	
Line	**Column 1**
1	1

FI Command Writing a SPS Variable.

CW_PVS_(1) **(Single Write)**

(1) = Identifier of the SPS variable

Value to be written SPS variable [Format acc. to the type in the SPS program]

Note: Only defined SPS variables can be written. Addressing a non-declared variable results in an error message. The length of the data must not exceed 240 bytes. (refer also to chapter 4.1, Guidelines). The value to be written is passed to the "acValue" parameter in the "DataTransfer" routine.

Construction of Answer

The return value of the "DataTransfer" routine is [0] when the write procedure has been successfully completed. In case of an error, more information can requested by the routine "ReadGroupItem" in the form of a general error result line (refer here to chapter 8, Error Codes).

Example 2 PVS

Write the value 1 in the SPS variable with the identifier "IB_EXT24" at device address 00.

Assumption:

- The SPS variable with the identifier "IB_EXT24" is of the type "BOOL"

FI Command	**00_CW_PVS_IB_EXT24**
Value to be written	1

Reading the SPS Variable Declaration: PVT

MTCX Device Group

Name

PVT **P**LC **V**ariable **T**ype

Explanation

A SPS 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 SPS Variables.

FI Command

Reading-out the SPS Variable Type.

BR_PVT_(1)	**(Single Read)**
(1) = Identifier of the SPS variable	[acc. to declaration part of SPS]

Construction of Answer

One line with 2 columns is outputted for each element of the variables.

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

n = number of elements.

Value Range/Meaning of Columns

(1) = Identifier of the SPS variable	[acc. to declaration part of SPS]
2 = Type	[see value range PVF]

Examples:

Reading a Variable

Assumption:

The "TEST" variable is declared as WORD in the SPS program.

FI Command	**00_BR_PVT_TEST**	
Answer		
Line	**Column 1 (Name)**	**Column 2 (Type)**
1	TEST	WORD

Reading a Structure

Assumption:

The "TEST1" variable is declared as STRUCT in the SPS program.

STRUCT

E1 BOOL

E2 INT

E3 SINT

END

FI Command	00_BR_PVT_TEST1	
Answer		
Line	Column 1	Column 2
1	TEST1.E1	BOOL
2	TEST1.E2	INT
3	TEST1.E3	SINT

Reading an Array

Assumption:

The "TEST2" variable is declared as ARRAY in the SPS program.

ARRAY [

0 .. 3

] OF BOOL

....

FI Command	00_BR_PVT_TEST2	
Answer		
Line	Column 1	Column 2
1	TEST2[0]	BOOL
2	TEST2[1]	BOOL
3	TEST2[2]	BOOL
4	TEST2[3]	BOOL

Reading an Array of a Structure

Assumption:

The "TEST3" variable is declared as ARRAY in the SPS program.

ARRAY [

0 .. 1

] OF STRUCT1,

whereby STRUCT1 with

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

Software Installation Data: SID

MTCX Device Group

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

Explanation Information is returned regarding the installation. This information includes the installation path, the software version being used and service pack and release information.

FI Command Reading-in the installation data.

BR_SID1 **(Single Read)**

BC_SID1 **(Cyclic Read)**

Construction of Answer One line with 8 columns is outputted for the additional text.

Line 1	Column 1	...	Column 8

Meaning of the Column

1 = Base directory	[EXE files of the DOS-BOF]
2 = FI installation directory	[FI directory]
3 = Data directory	[in accordance to DOS-BOF]
4 = GBO version	[from INDRAMAT.ini]
5 = IF-DLL mode	[from INDRAMAT.ini]
6 = IF version	[from INDRAMAT.ini from DLL mode 400]
7 = Service pack info	[from INDRAMAT.ini from DLL mode 420]
8 = Release info	[from INDRAMAT.ini from DLL mode 420]

Example SID1 Return the information on the current installation.

FI Command		00_BR_SID1
Line	Column	Answer
1	1	D:\MT-CNC
	2	C:\MT-CNC\IND_DRV
	3	D:\MT-CNC\ANLAGE00
	4	005-18V05
	5	04.20
	6	04V03
	7	--
	8	--

Servo Lag of an Axis: SLA

MTCX Device Group

Name **SLA** **S**ervo**LA**g

Explanation The current servo lag of a selected axis of the MTCX device group is read out. The FI command "SLA1" returns the distance to go of an axis, related to the code of the axis meaning. The FI command "SLA2", on the other hand, returns the distance to go of an axis, related to the physical axis number.

FI Command Output of the servo lag of the selected axis of the device specified, related to the code of the axis meaning.

CR_SLA1_(1)_(2) **(Single Read)**

CC_SLA1_(1)_(2) **(Cyclic Read)**

CB_SLA1_(1)_(2) **(Break Cyclic Read)**

(1) = CNC process number [0...6]

(2) = Axis meaning [0...11; 20]

Construction of Answer The following table shows the general construction of the answer of the FI command SLA1. One line with three columns for the name of the axis, the servo lag and the unit is outputted in accordance to the settings of the process parameters.

Line 1	Column 1	...	Column 3

Value Range/Meaning of Columns

1 = Axis name [acc. to settings of axis parameter]

2 = Servo lag [acc. to settings of the process parameter]

3 = Unit [acc. to settings of the process parameter:] mm, inch]

Note: If the specified axis is not defined in the selected CNC process then the answer in all columns is [--].

Example SLA1 Read the servo lag of the Z axis in CNC process 0 of device address 00.

FI Command	**00_CR_SLA1_0_2**		
Answer			
Line	**Column 1**	**Column 2**	**Column 3**
1	Z	2.9124	[mm]

FI Command

Output of the servo lag of the selected axis of the device specified, related to the physical axis number.

CR_SLA2_(1)	**(Single Read)**
CC_SLA2_(1)	**(Cyclic Read)**
CB_SLA2_(1)	**(Break Cyclic Read)**

(1) = physical axis number [1...32]

Construction of Answer

The following table shows the general construction of the answer of the FI command SLA2. One line with three columns for the name of the axis, the servo lag and the unit is outputted in accordance to the settings of the process parameters.

Line 1	Column 1	...	Column 3

Value Range/Meaning of Columns

1 = Axis name	[acc. to settings of axis parameter]
2 = Servo lag	[acc. to settings of the process parameter]
3 = Unit	[acc. to settings of the process parameter:] mm, inch]

Note: If the specified axis is not defined in the selected CNC process then the answer in all columns is [--].

Example SLA2

Read the servo lag of the Z axis (physical axis number = 3) at the device address 00.

FI Command	**00_CR_SLA2_3**		
Answer			
Line	**Column 1**	**Column 2**	**Column 3**
1	Z	2.9124	[mm]

SPS Long Identification: SLI

MTCX Device Group

Name **SLI** **S**PS **L**ong **I**dentification

Explanation Returns the single data from the SPS long identification.

FI Command Read SPS long identification.

BR_SLI **(Single Read)**

Construction of Answer One line with 15 columns is outputted for the returned values.

Line 1	Column 1	...	Column 15

Value Range/Meaning of the Columns

1 = Device address	[00...15]
2 = Program number	[01...99]
3 = Project name	[max. 8 ASCII characters]
4 = Program name	[max. 8 ASCII characters]
5 = User name	[acc. to password entry]
6 = Program length	[bytes]
7 = Compilation time	[LONG] (coded in long value)
8 = Compilation date	[8 ASCII characters]
9 = Compilation time	[8 ASCII characters]
10 = Download time	[LONG] (coded in long value)
11 = Download date	[8 ASCII characters]1
12 = Download time	[8 ASCII characters]
13 = Version of SPS long identification	[LONG]
14 = RUN Flags	[HEX value]
15 = Compiler info	[LONG]

Example SLI Read the single data from the SPS long identification.

FI Command		00_BR_SLI
Line	Column	Answer
1	1	02
	2	01
	3	--
	4	MOT12
	5	TEST
	6	17672
	7	630163960
	8	15.12.99
	9	17:15:48
	10	630163961
	11	15.12.99
	12	17:15:50
	13	2
	14	0x0000
	15	13

SERCOS Parameters: SPA

MTCX Device Group

Name **SPA** **S**ERCOS **Pa**rameter

Explanation A SERCOS drive parameter is outputted or written. Each parameter consists of 7 elements, whereby any combination of elements can be selected by element coding.

FI Command

BR_SPA1_(1)_(2)_(3)	**(Single Read)**
BC_SPA1_(1)_(2)_(3)	**(Cyclic Read)**
BB_SPA1_(1)_(2)_(3)	**(Break Cyclic Read)**
BW_SPA1_(1)_(2)_(3)	**(Single Write)**
(1) = Drive address	[0...254]
(2) = Parameter No.	in format: X-Y-ZZZZ
(3) = Element coding	[standard or advanced format]

Parameter No.

Format X-Y-ZZZZ	Value range
X	S = standard data P = product data
Y	[0..0,15] = parameter record
Z	[0...4095] = datablock no.

Element Coding Element coding in standard format allows individual elements, such as, e.g. operating date, to be requested. If several elements are to be read out in one request, then the element coding can be added in advanced format, e.g. operating date (0x40) and unit (0x08) produces additive (0x48) → 48

The advanced format 0x80 has priority over 0x40.

Element	Standard Format	Advanced Format	Format	Example
Data status	S	01H	Hexadecimal word	0x0000
Name	N	02H	String	NC cycle time (TNcyc)
Attribute	A	04H	Hexadecimal double word	0x60110001
Unit	U	08H	String	µs
Min. input value	L	10H	Decimal word	2000
Max. input value	H	20H	Decimal word	20000
Operating date	D	40H	see Displaying the Operating **Date**	
Operating date, when no list		80_H		

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 following spaces are allowed.

Hexadecimal

Hexadecimal values are displayed by "0x...", e.g. 0x80. Up to eight positions are allowed. Leading or following spaces are allowed. Leading additional zeros or plus and minus signs are not allowed.

Binary (max. 32 characters)

Leading or following 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 shown:

Format X-Y-ZZZZ	Value range
X	S = standard data P = product data
Y	[0...7] = parameter record
Z	[0...4095] = datablock no.

(see example SPA1/write).

Lists of Variable Length

Lists always begin with two decimal numbered for the actual length and maximal 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 difference 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 maximal length of the list (string) is also transmitted.

Example

Parameter S-0-0030, Operating Date
Standard Format: "DKC2.1-SSE-01V09"
Advanced Format: "16\n16\nDKC2.1-SSE-01V09"

Reference to Literature

Additional information regarding the function of the standard and product-specific 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-xxVRS**-FKBx-EN.

Construction of Answer

The following table shows the general construction of the answer of the FI command SPA1 . Line 1 is outputted both when reading and when writing. Additional lines are only outputted when reading depending on the element coding.

Note: If the element coding has been requested in standard format then the first line is not applicable.

Note: Line 1 is a status line that either contains SERCOS 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].

The number of the drive that reports the SERCOS error is outputted in the second line.

Line	Column 1	Column 2	Column 3	Column 4
1	<SERCOS error>	<Drive no. SERCOS error>	0x0000	0x0000
2	Read: 1. Element corresponding to the element coding.			
...	...			
n	Read: (n-1). Element corresponding to the element coding.			

Example SPA1 / read

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

FI Command	**00_BR_SPA1_3_S-0-0003_48**			
Answer				
Line	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	0x0000	0x0000	0x0000	0x0000
2	µs			
3	2000			

Example SPA1 / write

Write the ID number P-0-0037 in the parameter S-0-0305 of the 3rd drive (element coding 0x40).

Technical Background:

- Real-time status bit 1 is to be assigned the trigger status word of the oscilloscope function of a DIAX04 drive.

FI Command	**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

Active SERCOS Phase Switch-Over: SPH

MTCX Device Group

Name **SPH** **S**ERCOS **Ph**ase

Explanation All drives within a SERCOS ring are in the same communication phase. The phase condition 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 [1...32]

Value to be written Phase [2, 4]

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

Example SPH
Write SERCOS Phase

Switch-over of the first axis (write) after phase 4; phase 2 is active.

FI Command	**00_CW_SPH_1** **Value to be written: 4**	
Answer		
Line	**Column 1**	**Column 2**
1	52	1

Note: Switching-over from phase 2 to phase 4 returns as result of column 1 the value [52]. 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 You can find more details regarding the communication phases in the Rexroth Indramat documentation:

"DIAX04 Drive with Servo Functions", General Instructions on Putting into Operation, DOK-DIAX04-SSE-xxVRS**-FKBx-EN

Selected NC Program: SPP

MTCX Device Group

Name **SPP** **S**elected **P**art-**P**rogram number

FI Command

CR_SPP_(1) **(Single Read)**

CC_SPP_(1) **(Cyclic Read)**

CB_SPP_(1) **(Break Cyclic Read)**

(1) = CNC process number [0...6]

Construction of Answer The answer of the FI command SPP consists of one line with two columns for the identifier of the NC memory and the number of the selected NC program.

Line 1	Column 1	Column 2

Value Range/Meaning of Columns

1 = NC memory [A = NC memory A; B = NC memory B]

2 = Number of selected NC program [acc. to settings of the process parameter]

Example SPP Read the selected NC program in CNC process 0 of device address 00.

FI Command	**00_CR_SPP_0**	
Answer		
Line	**Column 1**	**Column 2**
1	B	55

Read or Write Tool Data Record: TDA

MTCX Device Group

Name **TDA** **T**ool **DA**ta

Explanation A complete tool data record consisting of basic data and defined cutter data is read out of or read into the controls.

FI Command Read-out of the complete tool data record. For this FI command, addressing is via the CNC process number, the tool memory and the location number.

BR_TDA1_(1)_(2)_(3) **(Single Read)**

(1) = CNC process number [0...6]

(2) = tool memory [M = magazine/turret,
S = spindle,
G = gripper]

(3) = location number [1...999]

Construction of Answer The following table shows the general construction of the answer of the FI command BR_TDA1. The number of lines depends on the number of cutters. The first line contains the basic data. The cutter is are listed from line 2 onwards. The basic data consists of 28 basic-data elements and the cutter data comprising 40 cutter-data elements.

Line 1	**Column 1**	**...**	**Column 28**	
Line 2	**Column 1**	**Column 2**	**...**	**Column 40**
...	**...**	**...**	**...**	**...**
Line n+1	**Column 1**	**Column 2**		**Column 40**

n = number of cutters

Example TDA1 Read the complete tool data record

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

1	18	0
	19	0
	20	0.000000
	21	0.000000
	22	0.000000
	23	0.000000
	24	0.000000
	25	0.000000
	26	0.000000
	27	0.000000
	28	0.000000
2	01	1
	02	_
	03	100.000000
	04	5.000000
	05	0.000000
	06	0.000000
	07	0.0000
	08	0.0000
	09	104.8000
	10	40.0000
	11	0.0000
	12	0.0000
	13	0.0000
	14	0.0000
	15	0.0000
	16	0.0000
	17	0.0000
	18	0.0000
	19	-999.0000
	20	999.0000
	21	-999.0000
	22	999.0000
	23	-999.0000
	24	999.0000
	25	-999.0000
	26	999.0000
	27	0.0000
	28	0.0000
	29	0.0000
	30	0.0000
	31	0.000000
	32	0.000000
	33	0.000000
	34	0.000000

FI Command	03_BR_TDA1_0_M_21	
Line	**Column**	**Answer**
2	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, addressing is via the CNC 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) = CNC process number [0...6]

(2) = Tool memory [M = magazine/turret,
S = spindle, G = gripper]

(3) = Location number [1...999]

Values to be written

The values to be written are passed in a table. First, the 3-digit code of the data element must be passed and then the value to be written must be passed. The first position addresses the data record (0 = basic data record, 1 to 9 corresponding cutter data record) and the second and third positions address the actual data element. (refer here also to Value Ranges, page 7-21)

Data Element Code

1st Position	2nd Position	3rd Position
0 = basic data record of 1...9 = cutter data record	**double-digit data element number**	

Note: The character „|“ (= 0x7D) is used as separator between the number of the data element and the value to be written. The individual lines of the table as also separated by a "|". <Element number n> <|> <Value n> <|> <Element number m> <|> <Value m> <|> .....

Example TDA1
Write Tool Data Record

Write the following data elements of the tool data record:

- Element number 002: Name (ID) "drill Z72"
- Element number 008: Number of cutters "1" and
- Element number 107: Length L1 "100"

<u>Assumption:</u>

- CNC process number: 0
- Tool memory: M = magazine and
- position number 2

FI Command	03_BW_TDA1_0_M_2
Values to be written	
002<|>drill Z72<|>008<|>1<|>107<|>100	

Reference to Literature

You can find more details regarding the tool management 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-EN

FI Command

Read-out of the complete tool data record. For this FI command, addressing is via the CNC process number, the tool number and the duplo number.

BR_TDA2_(1)_(2)_(3)   (Single Read)

(1) = CNC process number   [0...6]

(2) = tool number   [1...9999999]

(3) = duplo number   [1...9999]

Construction of Answer

The following table shows the general construction of the answer of 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 are listed from line 2 onwards. The basic data consists of 28 basic-data elements and the cutter data comprising 40 cutter-data elements.

Line 1	Column 1	...	Column 28	
Line 2	**Column 1**	**Column 2**	**...**	**Column 40**
...	**...**	**...**	**...**	**...**
Line n+1	**Column 1**	**Column 2**	**...**	**Column 40**

n = number of cutters

Example TDA2

Read the complete tool data record

FI Command	03_BR_TDA2_0_1_1	
Line	**Column**	**Answer**
1	01	10156
	02	Cutter head D80
	03	M
	04	21
	05	1
	06	1
	07	2
	08	1
	09	-p
	10	0
	11	M 21
	12	M
	13	--
	14	M
	15	--
	16	[cycl]

FI Command	03_BR_TDA2_0_1_1	
Line	**Column**	**Answer**
1	17	[mm]
	18	0
	19	0
	20	0.000000
	21	0.000000
	22	0.000000
	23	0.000000
	24	0.000000
	25	0.000000
	26	0.000000
	27	0.000000
	28	0.000000
2	01	1
	02	_
	03	100.000000
	04	5.000000
	05	0.000000
	06	0.000000
	07	0.0000
	08	0.0000
	09	104.8000
	10	40.0000
	11	0.0000
	12	0.0000
	13	0.0000
	14	0.0000
	15	0.0000
	16	0.0000
	17	0.0000
	18	0.0000
	19	-999.0000
	20	999.0000
	21	-999.0000
	22	999.0000
	23	-999.0000
	24	999.0000
	25	-999.0000
	26	999.0000
	27	0.0000
	28	0.0000
	29	0.0000
	30	0.0000
	31	0.000000
	32	0.000000
	33	0.000000
	34	0.000000
	35	0.000000
	36	0.0000

FI Command	03_BR_TDA2_0_1_1	
Line	**Column**	**Answer**
2	37	0.0000
	38	0.0000
	39	0.0000
	40	0.0000

Reference to Literature

You can find more details regarding the tool management 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-EN

Access to Tool Data Record: TDR

MTCX Device Group

Name

TDR1 **T**ool **D**ata **R**ecord

Explanation

Returns a complete basic data record and/or cutter data record of a tool.

FI Command

Read the basic data record or cutter data record of a tool in the tool memory.

CR_TDR1_(1)_(2)_(3)_(4) **(Single Read)**

CC_TDR1_(1)_(2)_(3)_(4) **(Cyclic Read)**

CB_TDR1_(1)_(2)_(3)_(4) **(Break Cyclic Read)**

(1) = CNC process number	[0...6]
(2) = Tool memory	[M = magazine/turret, S = spindle, G = gripper, P = index address, X = index address]
(3) = Tool position	in magazine/turret: [1...999] in spindle: [1...4] in gripper: [1...4] in change position [1...4] as index address [0...9999999]
(4) = Data record	[0 = basic tool data, 1...9 = cutter data]

Note: The index address of a tool is set by the device. For this reason, during the first access, access can only be made via memory number M, S, G and P. Thereafter, the tool can then also be addressed via the received index address.

Construction of Answer

The following table shows the general construction of the answer of the FI command CR_TDR1. One line with 28 (basic data) or 40 columns (cutter data) is out putted for the returned values.

Line 1...n:	Column 1	...	Column 28/40

Value Range/Meaning of the Columns

1...28 = Requested basic tool data	[max. 28 data elements] (refer to value range basic data)
1..40 = Requested tool cutter data	[max. 40 data elements] (refer to value range cutter data)

Example TDR1

Read the basic tool data record of the 2nd tool in the magazine in CNC process 0.

FI Command		00_CR_TDR1_0_M_2_0
Line	**Column**	**Answer**
1	1	928
	2	miller D20
	3	M
	4	2
	5	2
	6	1
	7	2
	8	1
	9	-p
	10	0
	11	M0
	12	M
	13	0
	14	M
	15	0
	16	~~0~~ [cycl]
	17	~~0~~ [mm]
	18	4
	19	102
	20	0.000000
	21	0.000000
	22	0.000000
	23	0.000000
	24	0.000000
	25	0.000000
	26	0.000000
	27	0.000000
	28	0.000000

Note: A tool identification run must be made before reading the basic data record or cutter data record of a tool!

FI Command

Read basic data record or cutter data record of a tool. Tool is addressed via the tool number and the duplo number.

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) = CNC process number	[0...6]
(2) = tool number	[1...9999999]
(3) = duplo number	[1...9999]
(4) = data record	[0 = basic tool data, 1...9 = cutter data]

Construction of Answer

The following table shows the general construction of the answer of the FI command CR_TDR2. One line with 28 (basic data) or 40 columns (cutter data) is outputted for the returned values.

Line 1...n:	Column 1	...	Column 28/40

Value Range/Meaning of the Columns

1...28 = requested basic tool data	[max. 28 data elements] (refer to value range basic data)
1..40 = requested tool cutter data	[max. 40 data elements] (refer to value range cutter data)

Example TDR2

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

FI Command		00_CR_TDR2_0_2_1_0
Line	**Column**	**Answer**
1	1	928
	2	miller D20
	3	M
	4	2
	5	2
	6	1
	7	2
	8	1
	9	-p
	10	0
	11	M0
	12	M
	13	0
	14	M
	15	0
	16	~~0~~ [cycl]
	17	~~0~~ [mm]
	18	4
	19	102
	20	0.000000
	21	0.000000
	22	0.000000
	23	0.000000
	24	0.000000
	25	0.000000
	26	0.000000
	27	0.000000
	28	0.000000

Tool Insert Finish: TIF

MTCX Device Group

Name **TIF** **T**ool **I**nsert **F**inish

Explanation Complete the insertion of a tool. The reservation of the tool memory location is lifted.

Refer also to: **CR_TII** and **CW_TLD1**

FI Command Finish Insertion.

CR_TIF_(1)_(2)_(3) **(Single Read)**

(1) = CNC process number	[0...6]
(2) = Tool memory	[M = magazine/turret, S = spindle, G = gripper, P = change position]
(3) = Location number tool memory	in magazine/turret: [1...999] in spindle: [1...4] in gripper: [1...4] in change position [1...4]

Construction of Answer One line is outputted with a column for acknowledgement of whether or not the FI command has been successfully executed.

(P_ACK) = **P**ositive **ACK**nowledge data element has been set

Example TIF Finish the insertion of a tool at location 5 in magazine in CNC process 0 of device 00.

FI Command		**00_CR_TIF_0_M_5**
Line	**Column**	**Answer**
1	1	(P_ACK)

Tool Insert Initiate: TII

MTCX Device Group

Name **TII** **T**ool **I**nsert **I**nitiate

Explanation Initiate the insertion of an individual tool. Reserves a location in the tool memory.

After this, the basic data and the cutter data are to be entered by repeated inputting of **CW_TLD1**. After the tool has actually been inserted in the tool memory, the procedure is completed by **CR_TIF**.

FI Command Initiate insertion.

CR_TII_(1)_(2)_(3) **(Single Read)**

(1) = CNC process number	[0...6]
(2) = tool memory	[M = magazine/turret, S = spindle, G = gripper, P = change position]
(3) = location number tool memory	in magazine/turret: [1...999] in spindle: [1...4] in gripper: [1...4] in change position [1...4]

Construction of Answer One line is outputted with a column for acknowledgement of whether or not the FI command has been successfully executed.

(P_ACK) = **P**ositive **ACK**nowledge data element has been set

Example TII Initiate the procedure for inserting tools in tool location at position 5 in CNC process 0 of device 00.

FI Command		00_CR_TII_0_M_5
Line	**Column**	**Answer**
1	1	(P_ACK)

In case of an error: Error is returned by N_ACK-error:

FI Command		00_CR_TII_0_M_5
Line	**Column**	**Answer**
1	1	1 (= N_ACK error class)
1	2	131 (= error number) .
1	3	0x00000000 (= additional information 0)
1	4	tool memory assigned (= error text)

Tool Basic Data List: TLB

MTCX Device Group

Name **TLB** **T**oo**L** **B**asic Data List

Explanation Returns the basic data of the tool list of the selected device of the MTCX device group.

FI Command Read selected basic data of the tool list.

BR_TLB1_(1)_(2)_(3)_(4)_(5) (Single Read)

(1) = CNC process number [0...6]

(2) = Tool memory [M = magazine/revolver, S = spindle, G = gripper]

(3) = Location from [1...999]

(4) = Location to [1...999]

(5) = Data element [1...28]

If more than one element is required as the 5th entry parameter then these are attached to the command with "_" and corresponding numbers.

Construction of Answer The following table shows the general construction of the answer of the FI command BR_TLB1. The number of lines depends on the number of tools. One line with 2 columns is outputted per tool for the returned values. If more than one data element is requested then the number of columns increases correspondingly.

Line 1...n:	Column 1	Column 2	...	Column 29

Value Range/Meaning of the Columns

1 = Tool memory [xxx = magazine/turret, SPx = spindle, GRx = gripper]

2..0,29 = requested basic tool data [max. 28 data elements] (refer to value range Value Ranges

, p. 7-180)

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: Duplo number [1...9999] and
- Element number 007: Correction type [1...5]

For additional elements, refer to Value Ranges

Assumption:

- CNC process number: 0
- Tool memory: 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	TWISTDRILL 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) = CNC process number [0...6]

(2) = data element [1...28]

If more than one element is required as the 2nd entry parameter then these are attached to the command with "_" and corresponding numbers.

Construction of Answer

The following table shows the general construction of the answer of the FI command BR_TLB2. The number of lines depends on the number of tools. One line with 2 columns is outputted per tool for the returned values. If more than one data element is requested then the number of columns increases correspondingly.

Line 1...n:	**Column 1**	**Column 2**	**...**	**Column 29**

Value Range/Meaning of the Columns

1 = Tool memory [xxx = magazine/turret, SPx = spindle, GRx = gripper]

2..29 = requested basic tool data [max. 28 data elements] (refer to value range basic data, p. 7-180)

Example TLB2

Read data elements 2, 5, 6, 7 in CNC process 0.

Explanation of elements:

- Element number 002: Name (ID) [max. 28 ASCII characters]
- Element number 005: Tool number [1..9999999]
- Element number 006: Duplo number [1...9999] and
- Element number 007: Correction type [1...5]

For additional elements, refer to p. 7-180.

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	TWISTDRILL D4.8
	3	0
	4	1
	5	2
6	1	005
	2	DRILL MILLER D8
	3	0
	4	1
	5	2
7	1	006
	2	SLAB MILLING CUTTER D60
	3	0
	4	1
	5	1
8	1	007
	2	--
	3	0
	4	0
	5	0

Tool Data Record Elements: TLD

MTCX Device Group

Name **TLD** **T**oo**L** **D**ata

Explanation Returns elements of the basic data or cutter data of a tool in the tool memory. In a telegram, only basic data or data from one cutter can be returned. If data elements are to be combined from basic data and cutter data then the command CR_TLD3 or CR_TLD4 must be used.

For a complete data record of basic data or cutting data, please refer to CR_TDR.

FI Command Read basic data element(s) or cutter data of a tool.

CR_TLD1_(1)_(2)_(3)_(4)_(5) **(Single Read)**

CC_TLD1_(1)_(2)_(3)_(4)_(5) **(Cyclic Read)**

CB_TLD1_(1)_(2)_(3)_(4)_(5) **(Break Cyclic Read)**

(1) = CNC process number	[0...6]
(2) = Tool memory	[M = magazine/turret, S = spindle, G = gripper, P = change position, X = index address]
(3) = Tool position	in magazine/turret: [1...999] in spindle: [1...4] in gripper: [1...4] in change position [1...4] as index address [0...9999999]
(4) = Data record	[0 = basic tool data, 1...9 = cutter data]
(5) = Data element	The basic data: [1...28] The cutter data: [1...40]

If more than one element is required as the 5th entry parameter then these are attached to the command with "_" and corresponding numbers.

Note: The index address of a tool is set by the device. For this reason, during the first access, access can only be made via memory number M, S, G and P. Thereafter, the tool can then also be addressed via the received index address.

Construction of Answer The following table shows the general construction of the answer of the FI command CR_TLD1. One line with one column is outputted for the delivered value. If more than one data element is requested then the number of columns increases correspondingly.

Line 1...n:	Column 1	...	Up to column 28/40

Value Range/Meaning of the Columns

1...28 = requested basic tool data	[max. 28 data elements] (refer to value range basic data p. 7-180)
1..40 = requested tool cutter data	[max. 40 data elements] (refer to value range cutter data p. 7-183)

Example TLD1

Read the name (basic data 2) of the 4th tool in the magazine in CNC process 0.

FI Command		00_CR_TLD1_0_M_4_0_2
Line	**Column**	**Answer**
1	1	MILLER D24

FI Command

Write single element of basic data or cutter data of a tool in the tool memory.

CW_TLD1_(1)_(2)_(3)_(4)_(5) (Single Write)

(1) = CNC process number	[0...6]
(2) = Tool memory	[M = magazine/turret, S = spindle, G = gripper, P = change position, X = index address]
(3) = Tool memory location	in magazine/turret: [1...999] in spindle: [1...4] in gripper: [1...4] in change position [1...4] as index address [0...9999999]
(4) = Data record	[0 = basic tool data, 1...9 = cutter data]
(5) = Data element	The basic data: [1...28] The cutter data: [1...40]

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 memory number M, S, G and P. Thereafter, the tool can then also be addressed via the received index address.

Value to be written

Value of data element — refer to Value Ranges Basic Data and Cutter Data (p. 7-180)

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

Construction of Answer

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

(P_ACK) = **P**ositive **ACK**nowledge — data element has been set

Example TLD1

Write data elements 4 (warn limit) in CNC process 0 for the tool at the 3rd magazine position in cutter 1.

FI Command		00_CW_TLD1_0_M_3_1_4 Value to be written: 6.5
Line	**Column**	**Answer**
1	1	(P_ACK)

FI Command

Read basic data or cutter data element(s) of a tool. Tool is addressed via the tool number and the duplo number.

⇒ A tool identification number is required beforehand!

ATTENTION

CR_TLD2_(1)_(2)_(3)_(4)_(5)	**(Single Read)**
CC_TLD2_(1)_(2)_(3)_(4)_(5)	**(Cyclic Read)**
CB_TLD2_(1)_(2)_(3)_(4)_(5)	**(Break Cyclic Read)**
(1) = CNC process number	[0...6]
(2) = Tool number	[1...9999999]
(3) = Duplo number	[1...9999]
(4) = Data record	[0 = basic tool data, 1...9 = cutter data]
(5) = Data element	The basic data: [1...28] The cutter data: [1...40]

If more than one element is required as the 5th entry parameter then these are attached to the command with "_" and corresponding numbers.

Construction of Answer

The following table shows the general construction of the answer of the FI command CR_TLD2. One line with one column is outputted for the delivered value. If more than one data element is requested then the number of columns increases correspondingly.

Line 1...n:	Column 1	...	Up to column 28/40

Value Range/Meaning of the Columns

1...28 = requested basic tool data	[max. 28 data elements] (refer to value range basic data, p. 7-180)
1..40 = requested tool cutter data	[max. 40 data elements] (refer to value range cutter data, p. 7-183)

Example TLD2

Read the name (basic data 2) of the 3rd tool/duplo no. 1 in CNC process 0.

FI Command		00_CR_TLD2_0_3_1_0_2
Line	**Column**	**Answer**
1	1	TAPPER M5

FI Command

Write single element of basic data or cutter data of a tool. Tool is addressed via tool number and duplo number.

CW_TLD2_(1)_(2)_(3)_(4)_(5)	**(Single Write)**
(1) = CNC process number	[0...6]
(2) = Tool number	[1...9999999]
(3) = Duplo number	[1...9999]
(4) = Data record	[0 = basic tool data, 1...9 = cutter data]
(5) = Data element	The basic data: [1...28] The cutter data: [1...40]

Value to be written

Value of data element	refer to Value Ranges Basic Data and Cutter Data (p. 7-180)

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

Construction of Answer

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

(P_ACK) = **P**ositive **ACK**nowledge data element has been set

Example TLD2

Write data element 4 (warn limit) in CNC process 0 for the tool number 3/duplo number 1 in cutter 1.

FI Command		**00_CW_TLD1_0_M_3_1_4** **Value to be written: 6.5**
Line	**Column**	**Answer**
1	1	(P_ACK)

Explanation of TLD3 or TLD4

Returns any element of the basic data or cutter data of a tool in any order.

The addressing of an element is expanded by both of the following FI commands to 3 positions. The first position addresses the data record (0 = basic data record, 1-9= cutter data) and the second and third positions address the actual data element.

Addressing Examples

002 Basic data – tool name

103 Cutter 1 – remaining tool life

203 Cutter 2 – remaining tool life

Note: You should always make sure when requesting tool data that the maximal effective data length of 240 bytes is not exceeded. If more than 240 bytes are requested then the error message (NACK) /FI (1014) is returned by the controls.

FI Command

Reading of basic data and cutter data of a tool in the tool memory.

CR_TLD3_(1)_(2)_(3)_(4) **(Single Read)**

CC_TLD3_(1)_(2)_(3)_(4) **(Cyclic Read)**

CB_TLD3_(1)_(2)_(3)_(4) **(Break Cyclic Read)**

(1) = CNC process number [0...6]

(2) = Tool memory [M = magazine/turret, S = spindle, G = gripper, P = change position, X = index address]

(3) = Tool memory location in magazine/turret: [1...999]
in spindle: [1...4]
in gripper: [1...4]
in change position [1...4] as
index address [0...9999999]

(4) = Data element [001...940]

If more than one element is required as the 2nd 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 memory number M, S, G and P. Thereafter, the tool can then also be addressed via the received index address.

Construction of Answer

The following table shows the general construction of the answer of the FI command CR_TLD3. One line with one column is outputted for the delivered value. If more than one data element is requested then the number of columns increases correspondingly.

Line 1...n:	Column 1	...	Column xxx

Value Range/Meaning of the Columns

1...xxx = requested basic tool data and cutter data	refer to Value Ranges Basic Data or Cutter Data (p. 7-180)

Example TLD3

Read the name of the basic tool data of the 4th tool in the magazine and the remaining tool life of cutter 1 in CNC process 0.

FI Command		**00_CR_TLD3_0_M_4_002_103**
Line	**Column**	**Answer**
1	1	MILLER D24
	2	100.00

FI Command

Reading of basic data and cutter data of a tool acc. to tool number and duplo number.

CR_TLD4_(1)_(2)_(3)_(4) **(Single Read)**

CC_TLD4_(1)_(2)_(3)_(4) **(Cyclic Read)**

CB_TLD4_(1)_(2)_(3)_(4) **(Break Cyclic Read)**

(1) = CNC process number	[0...6]
(2) = Tool number	[1...9999999]
(3) = Duplo number	[1...9999]
(4) = Data element	[001...940]

If more than one element is required as the 2nd entry parameter then these are attached to the command with "_" and corresponding numbers.

Construction of Answer

The following table shows the general construction of the answer of the FI command CR_TLD4. One line with one column is outputted for the delivered value. If more than one data element is requested then the number of columns increases correspondingly.

Line 1...n:	Column 1	...	Column xxx

Value Range/Meaning of the Columns

1...xxx = requested basic tool data and cutter data	refer to Value Ranges Basic Data or Cutter Data (p. 7-180)

Example TLD4 Read the name of tool number 3/duplo number 1 and the remaining tool life of cutter 4 in CNC process 0 of device 00.

FI Command		**00_CR_TLD4_0_3_1_002_403**
Line	**Column**	**Answer**
1	1	TAPPER M5
	2	100.00

Tool Edge Data List: TLE

MTCX Device Group

Name **TLE** **T**oo**L** **E**dge Data List

Explanation Returns the cutter data of the tool list.

FI Command Read selected cutter data of the tool list.

BR_TLE1_(1)_(2)_(3)_(4)_(5)_(6) **(Single Read)**

(1) = CNC process number [0...6]

(2) = Tool edge [1...9]

(3) = Tool memory [M = magazine/turret, S = spindle, G = gripper]

(4) = Location from [0...999]

(5) = Location to [0...999]

(6) = Data element [1...40]

If more than one element is required as the 6th entry parameter then these are attached to the command with "_" and corresponding numbers.

Construction of Answer The following table shows the general construction of the answer of the FI command BR_TLE1. The number of lines depends on the number of tools. One line with 2 columns is outputted per tool for the returned values. If more than one data element is requested then the number of columns increases correspondingly.

Line 1...n:	**Column 1**	**Column 2**	**...**	**Column 41**

Value Range/Meaning of the Columns

1 = Tool memory [xxx = magazine/turret, SPx = spindle, GRx = gripper]

2...41 = requested tool cutter data [max. 40 data elements] (refer to value range cutter data, p. 7-183)

Example TLE1

- Element number 002: Cutter status is requested.

Assumption:

- CNC process number: 0
- Edge: 1
- Tool memory: M = Magazine
- Location number from: 1
- Location number to: 3

Read data elements 2 and 3.

FI Command		00_BR_TLE1_0_1_M_1_3_2_3
Line	**Column**	**Answer**
1	1	001
	2	d (tool worn out)
2	1	002
	2	_ (tool ok)
3	1	003
	2	w (fallen below warning limit)

FI Command

Read all cutter data of the tool list.

BR_TLE2_(1)_(2)_(3) **(Single Read)**

(1) = CNC process number [0...6]

(2) = cutter position [0...8]

(3) = data element [1...40]

If more than one element is required as the 3rd entry parameter then these are attached to the command with "_" and corresponding numbers.

Construction of Answer

The following table shows the general construction of the answer of the FI command BR_TLE2. The number of lines depends on the number of cutters. One line with 2 columns is outputted per cutter for the returned values. If more than one data element is requested then the number of columns increases correspondingly.

Line 1...n:	Column 1	Column 2	...	Column 41

Value Range/Meaning of the Columns

1 = Tool memory	[00 = magazine/turret, SP = spindle, GR = gripper]
2..41 = requested basic tool data	[max. 40 data elements] (refer to value range cutter data, p. 7-183)

Example TLE2

- Element number 003: Remaining tool life [0.0000...100.0000]
- Element number 004: Warning limit [0.1...100.00]
- Element number 005: Maximal period of use [0...9999999]
- Element number 009: Length L3 [-9999.9999...9999.9999]

Read in CNC process 0 the data elements 3, 4, 5, 9 for all tools at cutter position 1.

FI Command		00_BR_TLE2_0_1_3_4_5_9
Line	**Column**	**Answer**
1	1	SP1
	2	0.0000
	3	0.0000
	4	0.0000
	5	0.0000
2	1	001
	2	100.0000
	3	5.0000
	4	0.0000
	5	106.8500
3	1	002
	2	100.0000
	3	5.0000
	4	0.0000
	5	132.9600
4	1	003
	2	48.0000
	3	5.0000
	4	100.0000
	5	106.8000
5	1	004
	2	99,8617
	3	5.0000
	4	0.0000
	5	180.0900
6	1	005
	2	100.0000
	3	5.0000
	4	0.0000
	5	78.7000
7	1	006
	2	100.0000
	3	0.0000
	4	0.0000
	5	116.0000
8	1	007
	2	0.0000
	3	0.0000
	4	0.0000
	5	0.0000

Tool Move : TMV

MTCX Device Group

Name **TMV** **T**ool **M**o**V**e

Explanation A complete tool data record consisting of basic data and defined cutter data is moved. This corresponds to the Rexroth Indramat BOF function "Tool Move".

FI Command Move the selected tool data record.

CR_TMV_(1)_(2)_(3)_(4)_(5) **(Single Read)**

(1) = CNC process number [0...6]

(2) = Current tool memory [M = magazine/turret, S = spindle, G = gripper]

(3) = Current location number [1...999]

(4) = Target tool memory [M = magazine/turret, S = spindle, G = gripper]

(5) = Target location number [1...999]

Construction of Answer One line with one column is outputted for the acknowledgement of the FI command issued. The following hereby mean:

(P_ACK) = **P**ositive **ACK**nowledge data record has been moved

Example TMV Move the 24th tool data records in the magazine to the 25th tool data record in the magazine.

Assumption:

- There is a valid tool in magazine 24 in CNC process 0 at device address 00.

FI Command	**00_CR_TMV_0_M_24_M_25**
Answer	
Line	**Column 1**
1	(P_ACK)

Reference to Literature You can find more details regarding the tool management 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-EN.

Torque: TQE

MTCX Device Group

Name **TQE** **T**or**Q**u**E**

Explanation The torque at a selected axis of the MTCX device group is read out. 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 of the torque of the selected axis of the device of the MTCX device group, related to the code of the axis meaning.

CR_TQE1_(1)_(2) **(Single Read)**

CC_TQE1_(1)_(2) **(Cyclic Read)**

CB_TQE1_(1)_(2) **(Break Cyclic Read)**

(1) = CNC process number [0...6]

(2) = Axis meaning [0...11; 20];

Construction of Answer The following table shows the general construction of the answer of the FI command TQE1. One line with three columns is outputted for the name of the axis, the torque and the unit [%].

Line 1	Column 1	...	Column 3

Value Range/Meaning of Columns

1 = Axis name [acc. to settings of axis parameter]

2 = Torque [format acc. to settings of the process parameter]

3 = Unit [%]

Note: If the specified axis is not defined in the selected CNC process then the answer in all columns is [--].

Example TQE1 Read the torque at the Z axis in CNC process 0 of device address 00.

FI Command	00_CR_TQE1_0_2		
Answer			
Line	**Column 1**	**Column 2**	**Column 3**
1	Z	-25.6	[%]

FI Command Output of the torque at the selected axis of the device specified, related to the physical axis number.

CR_TQE2_(1) **(Single Read)**

CC_TQE2_(1) **(Cyclic Read)**

CB_TQE2_(1) **(Break Cyclic Read)**

(1) = Physical axis number [1...32]

Construction of Answer The following table shows the general construction of the answer of the FI command TQE2. One line with three columns is outputted for the name of the axis, the torque and the unit [%].

Line 1	Column 1	...	Column 3

Value Range/Meaning of Columns

1 = Axis name	[acc. to settings of axis parameter]
2 = Torque	[format acc. to settings of the process parameter]
3 = Unit	[%]

Note: If the specified axis is not defined in the selected CNC process then the answer in all columns is [--].

Example TQE2 Read the torque at the Z axis (physical axis number = 3) at device address 00.

FI Command	00_CR_TQE2_3		
Answer			
Line	**Column 1**	**Column 2**	**Column 3**
1	Z	-25.6	[%]

Remove Tool Data Record: TRM

MTCX Device Group

Name **TRM** **T**ool **R**e**M**ove

Explanation A complete tool data record consisting of basic data and defined cutter data is removed from the device. This corresponds to the Rexroth Indramat BOF function "Remove Tool from the Magazine List".

FI Command Remove the selected tool data record.

CR_TRM_(1)_(2)_(3) **(Single Read)**

(1) = CNC process number	[0...6]
(2) = Tool memory	[M = magazine/turret, S = spindle, G = gripper]
(3) = Location number	[1...999]

Construction of Answer One line with one column is outputted for the acknowledgement of the FI command issued. The following hereby mean:

(P_ACK) = **P**ositive **ACK**nowledge data record has been removed

Example TRM Remove the 24^{th} tool data record.

Assumption:

- There is a valid tool in magazine 24 in CNC process 0 at device address 00.

FI Command	00_CR_TRM_0_M_24
Answer	
Line	**Column 1**
1	(P_ACK)

Reference to Literature

You can find more details regarding the tool management 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-EN

Reset Remaining Tool Life of a Tool: TRS

MTCX Device Group

Name

TRS **T**ool **R**e**S**et

Explanation

Resets the percentage remaining tool life of a tool to 100%. The positive or negative execution of the function is reported back via the return value of the routine.

FI Command

Reset remaining tool life of a tool:

CR_TRS_(1)_(2)_(3) **(Single Read)**

(1) = CNC process number	[0...6]
(2) = Tool memory	[M = magazine/turret, S = spindle, G = gripper, P = change position, X = index address]
(3) = Tool position	in magazine/turret: [1...999] in spindle: [1...4] in gripper: [1...4] in position change [1...4] as index address [0...9999999]

Note: The index address of a tool is set by the device. For this reason, during the first access, access can only be made via memory number M, S, G and P. Thereafter, the tool can then also be addressed via the received index address.

Construction of Answer

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

(P_ACK) = **P**ositive **ACK**nowledge tool has been reset

Example TRS

Reset the remaining tool life for the tool located in change position 1 in CNC process 0 of device 00.

FI Command		**00_CR_TRS_0_P_1**
Line	**Column**	**Answer**
1	1	(P_ACK)

Zero Offset Table Data: ZOD

MTCX Device Group

Name **ZOD** **Z**ero **O**ffset **D**ata

Explanation The zero-offset table data can be read and written. The zero-point 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 original location of machine remains securely stored in the CNC controls and is not changed by the zero-point offset.

The following offset types are available in the CNC:

- programmable absolute zero offset G50,
- programmable incremental zero offset G51,
- programmable origin of workpiece G52,
- adjustable zero offsets G54 ... G59 as well as
- adjustable general offset in the zero (origin) table.

Using the zero-point 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.

FI Command Write a zero offset.

CW_ZOD_(1)_(2)_(3)_(4)_(5)	**(Single Write)**
(1) = NC memory	[1 = memory A; 2 = memory B]
(2) = CNC process number	[0...6]
(3) = number of shift bank	[0...9]
(4) = offset type	[offset type code]
(5) = code of the axis meanings	[0...8] axis meanings [9] offset angle "PHI"

Value to be written

Zero offset	[for axes: format acc. to the parameter settings] [offset angle PHI always in format Y.XXXX]

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

Offset Type

Code	Meaning	Explanation
3	General offset	acts additive to all offset types
4	G54	adjustable zero offset
....	....	....
9	G59	adjustable zero offset

Note: The axis meanings are contained in chapter 6.2, Data Tables.

Construction of Answer One line with one column is outputted for the acknowledgement of the FI command issued. The following hereby mean:

(P_ACK) = **P**ositive **ACK**nowledge Value has been written

Example ZOD

Write into zero point database 2 the values of the general offset of axis X 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 are defined.

FI Command	**00_CW_ZOD_1_0_2_3_0** **Value to be written: 0.111**
Answer	
Line	**Column 1**
1	(P_ACK)

Reference to Literature

You can find more details regarding the zero offsets in the Rexroth Indramat documentation:

"MTC200/MT-CNC NC Programming Instructions xxVRS", chapter "Zero Offsets" and chapter "Reading and Writing Zero-Offset Data (NPV) from NC Program OTD", DOK-MTC200-NC**PRO*Vxx-AW0x-EN.

FI Command

The values of the zero offset of all defined axes are outputted for the selected offset (shift) type.

CR_ZOD1_(1)_(2)_(3)_(4) **(Single Read)**

CC_ZOD1_(1)_(2)_(3)_(4) **(Cyclic Read)**

CB_ZOD1_(1)_(2)_(3)_(4) **(Break Cyclic Read)**

(1) = NC memory [1 = memory A; 2 = memory B]

(2) = CNC process number [0...6]

(3) = number of shift bank [0...9]

(4) = offset type [offset type code]

Offset Type

Code	Meaning	Explanation
0	Total	sum of all active offset values
1	G50/G51	programmable absolute/incremental zero offset
2	G52	programmable origin of workpiece
3	General offset	acts additive to all offset types
4	G54	adjustable zero offset
....	....	....
9	G59	adjustable zero offset

Construction of Answer

The following table shows the general construction of the answer of the FI command ZOD1. The answer consists of one of a maximum of n=10 lines, each with three columns for the name of the axis, value of zero offset and the unit.

Line 1...n:	Column 1	...	Column 3

Value Range/Meaning of Columns

1 = Axis names	[acc. to settings of the axis parameter; PHI] [Xi, Yi, Zi, Ui, Vi, Wi, Ai, Bi ,Ci, Si, --] (i=[ ], [1...3])
2 = Value	format acc. to the parameter settings] [offset angle PHI always in format Y.XXXX]
3 = Unit	[mm, inch] [offset angle PHI: deg]

Example ZOD1

Read into the zero point database 2 the values of the general offset of all defined axes 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_ZOD1_1_0_2_3**		
Answer			
Line	**Column 1**	**Column 2**	**Column 3**
1	X	0.111	[mm]
2	Y	0.000	[mm]
3	*Z	0.000	[mm]
4	PHI	0.0000	[deg]

Reference to Literature

You can find more details regarding the zero offsets in the Rexroth Indramat documentation:

"MTC200/MT-CNC NC Programming Instructions xxVRS",

chapter "Zero Offsets" and chapter "Reading and Writing the Zero-Offset Data (NPV) of NC Program OTD",

DOK-MTC200-NC**PRO*Vxx-AW0x-EN.

FI Command

Output of all zero offset values for the axes selected in a list.

CR_ZOD2_(1)_(2)_(3)_(4)_(5)	**(Single Read)**
CC_ZOD2_(1)_(2)_(3)_(4)_(5)	**(Cyclic Read)**
CB_ZOD2_(1)_(2)_(3)_(4)_(5)	**(Break Cyclic Read)**
(1) = NC memory	[1 = memory A; 2 = memory B]
(2) = CNC process number	[0...6]
(3) = Number of shift bank	[0...9]
(4) = Offset type	[offset type code]
(5) = Selection list for a max. of 10 elements	[0...8] axis meanings [9] offset angle "PHI"

Offset Type

Code	Meaning	Explanation
0	total	sum of all active offset values
1	G50/G51	programmable absolute/incremental zero offset
2	G52	programmable origin of workpiece
3	general offset	acts additive to all offset types
4	G54	adjustable zero offset
....	....	....
9	G59	adjustable zero offset

Note: The axis meanings are contained in chapter 6.2, Data Tables.

Construction of Answer

The following table shows the general construction of the answer of the FI command ZOD2. The answer consists of one of a maximum of n=10 lines, each with three columns for the axis meaning, value of zero offset and the unit. The number of lines depends on the number of list elements.

Line 1...n:	Column 1	...	Column 3

Value Range/Meaning of Columns

1 = Axis names	[acc. to settings of the axis parameter; PHI] [Xi, Yi, Zi, Ui, Vi, Wi, Ai, Bi ,Ci, Si, --] (i=[ ], [1...3])
2 = Value	format acc. to the 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 preceded by "*", e.g. "*Z", then this access is only assigned to the process at certain times (GAX/FAX).

Example ZOD2

Write into zero point database 2 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

You can find more details regarding the zero offsets in the Rexroth Indramat documentation:

"MTC200/MT-CNC NC Programming Instructions xxVRS",

chapter "Zero Offsets" and chapter "Reading and Writing the Zero-Offset Data (NPV) of NC Program OTD",

DOK-MTC200-NC**PRO*Vxx-AW0x-EN.

Value Ranges

Basic Data

MTCX Device Group

Element Number	Name of the File Element	Value range	Writable?
1	index address	0...9999999	no
2	name (ID)	max. 28 ASCII characters	yes
3	memory	[M = magazine/turret, S = spindle, G = gripper]	no
4	position	0...999	no
5	tool number	1...9999999	yes
6	duplo number	1...9999	yes
7	correction type	1...5	yes
8	number of cutters	1...9	yes
9	tool status	32 status bits with 0/1 (see following table)	yes
10	unassigned half-location	0...4	yes
11	old tool position	memory [M/S/G] location [0..999]	no
12	memory of the next replacement tool	[M = magazine/turret, S = spindle, G = gripper]	no
13	location of the next replacement tool	0...999	no
14	memory of the previous replacement tool	[M = magazine/turret, S = spindle, G = gripper]	no
15	location of the previous replacement tool	0..999	no
16	time unit	0 = min, 1 = cycle	yes
17	unit of length	0 = mm, 1 = inch	yes
18	tool code	0...9	yes
19	display type	0...999	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

Tool Status Bits

Bit	Symbol	Value	Group Name	Group Information	Adjustable	Comment
1	!	1 0	Present?	tool not available tool available	no	tool is missing
2	?	1 0		tool not required tool required	no	tool required for processing
3	\t	1 0	error correction type	correction type faulty	no	correction type does not accord with the requirements
4	e	1 0	error number of cutters	wrong number of cutters correct number	no	number of cutters does not accord with the requirements
5	\f	1 0	error cutter	cutter faulty cutter not faulty	no	cutter data does not accord with the requirements
6	$	1 0	error tool code	tool code faulty tool code not faulty	no	
7	*				no	reserved
8	*				no	reserved
9	B	1 0	location blocked	location blocked location not blocked	yes	e.g. location is damaged
10		1 0		upper half-location blocked. not blocked	no	blocked for fpc tool located in gripper or spindle
11		1 0		lower half-location blocked. not blocked	no	blocked for fpc tool located in gripper or spindle
12		1 0	location reservation	upper half-location reserved not reserved	yes	e.g. for a tool to be attached
13		1 0		lower half-location reserved not reserved	yes	e.g. for a tool to be attached
14		1 0	location assignment	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	condition of wear	tool is worn tool is not worn	no	the tool can no longer be used (replace)
18	w	1 0		warn limit reached warn limit not reached	no	the remaining tool life is near its end (replace)
19	-p	1 0	name of sister	processing tool no processing tool	no	there is a processing tool for every sister tool group

20	s	1 0		replacement tool no replacement tool	no	a replacement tool is a tool still to be used, not a processing tool
21	C	1 0	fixed position coding	fixed position coding, tool no fixed position coding, tool	yes	the tool always remains at the same position in the magazine
22	L	1 0	tool condition	tool blocked tool not blocked	yes	by user or application, e.g. edge is broken
23	*				no	reserved
24	*				no	reserved
25	1	1 0	ANW 1	user tool status bit 2	yes	any meaning
26	2	1 0	ANW 2	user tool status bit 2	yes	any meaning
Bit	**Symbol**	**Value**	**Group Name**	**Group Information**	**Adjustable**	**Comment**
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

Fig. 7-1: Tool Status Bits

Cutter Data

MTCX Device Group

Element Number	Name of the Data Element	Value Range	Writable?
1	cutter position	0...8	yes
2	Cutter status	16 status bits with 0/1 (see following table)	yes
3	remaining tool life	0.0...100.00	yes
4	warn limit	0.1...100.00	yes
5	max. life time	0...9999999	yes
6	time used	0...9999.999	no
7	length L1	-9999.9999...+9999.9999 or -999.99999...+999.99999	yes
8	length L2	-9999.9999...+9999.9999 or -999.99999...+999.99999	yes
9	length L3	-9999.9999...+9999.9999 or -999.99999..+999.99999	yes
10	radius R	-9999.9999...+9999.9999 or -999.99999...+999.99999	yes
11	wear L1	-9999.9999...+9999.9999 or -999.99999..+999.99999	yes
12	wear L2	-9999.9999...+9999.9999 or -999.99999...+999.99999	yes
13	wear L3	-9999.9999...+9999.9999 or -999.99999...+999.99999	yes
14	wear R	-9999.9999...+9999.9999 or -999.99999...+999.99999	yes
15	offset L1	-9999.9999...+9999.9999 or -999.99999...+999.99999	yes
16	offset L2	-9999.9999...+9999.9999 or -999.99999...+999.99999	yes
17	offset L3	-9999.9999...+9999.9999 or -999.99999...+999.99999	yes
18	offset R	-9999.9999...+9999.9999 or -999.99999...+999.99999	yes
19	L1_min	-9999.9999...+9999.9999 or -999.99999...+999.99999	no
20	L1_max	-9999.9999...+9999.9999 or -999.99999...+999.99999	no
21	L2_min	-9999.9999...+9999.9999 or -999.99999...+999.99999	no
22	L2_max	-9999.9999...+9999.9999 or -999.99999...+999.99999	no
23	L3_min	-9999.9999...+9999.9999 or -999.99999...+999.99999	no
24	L3_max	-9999.9999...+9999.9999 or -999.99999...+999.99999	no
25	R_min	-9999.9999...+9999.9999 or -999.99999...+999.99999	no
26	R_max	-9999.9999...+9999.9999 or -999.99999...+999.99999	no
27	wear factor L1	-9999.9999...+9999.9999 or -999.99999...+999.99999	yes
28	wear factor L2	-9999.9999...+9999.9999 or -999.99999...+999.99999	yes
29	wear factor L3	-9999.9999...+9999.9999 or -999.99999...+999.99999	yes
30	wear factor R	-9999.9999...+9999.9999 or -999.99999...+999.99999	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	-9999.9999...+9999.9999 or -999.99999...+999.99999	yes
37	user data 7	-9999.9999...+9999.9999 or -999.99999...+999.99999	yes
38	user data 8	-9999.9999...+9999.9999 or -999.99999...+999.99999	yes
39	user data 9	-9999.9999...+9999.9999 or -999.99999...+999.99999	yes
40	user data 10	-9999.9999...+9999.9999 or -999.99999...+999.99999	yes

Cutter Status Bits

Bit	Symbol	Value	Group Name	Group Information	Adjustable	Comment
1	e	1 0	wrong cutter position	wrong cutter position correct position	no	
2	1	1 0	L1 faulty	L1 faulty not faulty	no	
3	2	1 0	L2 faulty	L2 faulty not faulty	no	
4	3	1 0	L3 faulty	L3 faulty not faulty	no	
5	r	1 0	R faulty	R faulty not faulty	no	
6	*				no	reserved
7	*				no	reserved
8	*				no	reserved
9	d	1 0	condition of wear	cutter worn cutter not worn	no	the cutter can no longer be used (replace)
10	w	1 0		warn limit reached warn limit not reached	no	the remaining tool life is near its end (replace)
11	*				no	reserved
12	*				no	reserved
13	A	1 0	ANW 1	user cutter status bit 1	yes	any meaning
14	B	1 0	ANW 2	user cutter status bit 2	yes	any meaning
15	C	1 0	ANW 3	user cutter status bit 3	yes	any meaning
16	D	1 0	ANW 4	user cutter status bit 4	yes	any meaning

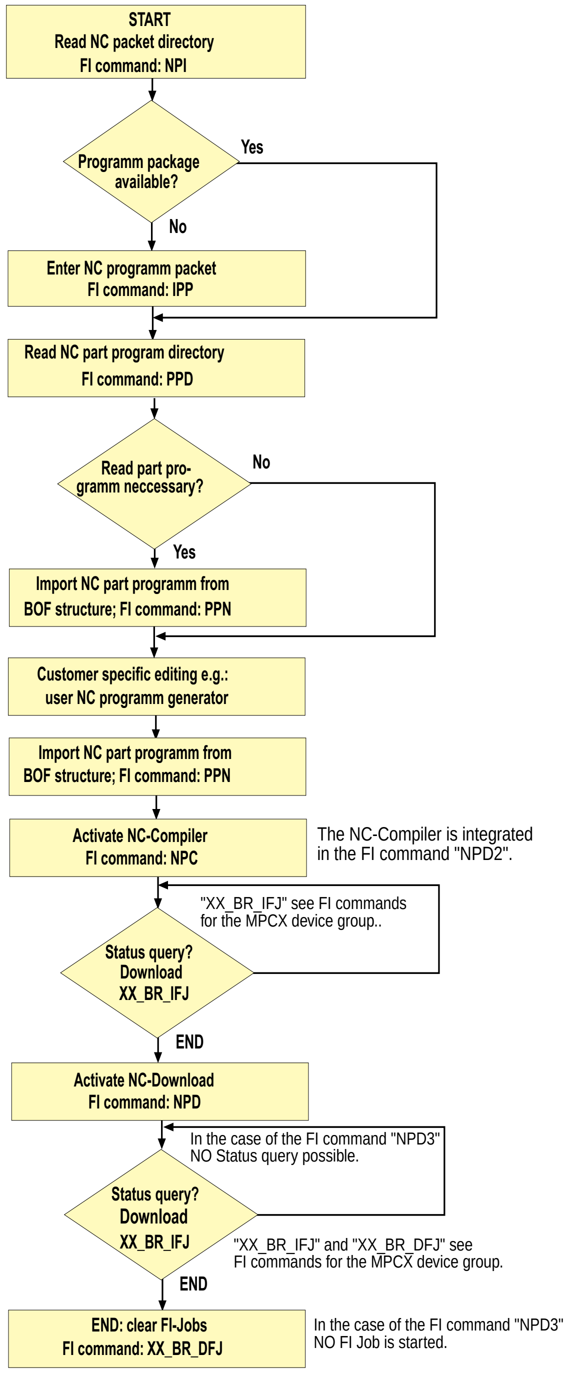
Fig. 7-2: Cutter Status Bits

Flow Diagram for Command Groups

NC Download Commands: IPP, NPC, NPD, NPI, PPD, PPN

MTCX Device Group

The following diagram shows by way of an example the sequence (flow) required for a complete NC download.



NC-Download.EPS

Fig. 7-3: Structural Construction during an NC Download

Edit Tool Data Records: TDA, TRM

MTCX Device Group

The following diagram shows by way of an example the sequence (flow) required for editing complete tool data records.

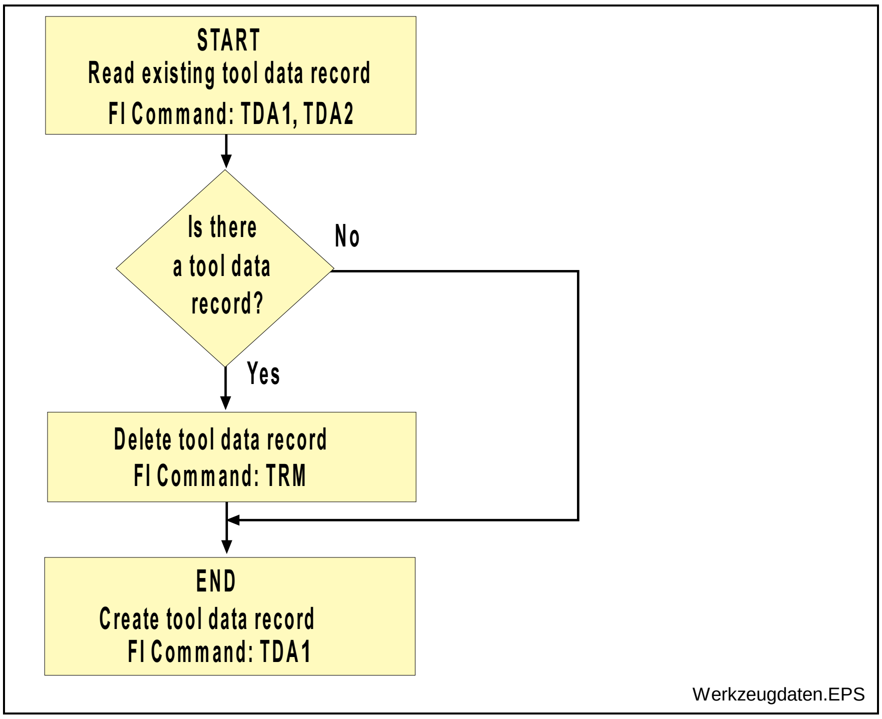


Fig. 7-4: Structural Construction for Editing Tool Data Records

Using the Tool Command in Practice

In addition to exact knowledge of the individual commands, the multitude of tool commands also requires additional information for practical usage.

This chapter therefore deals with this theme from the point of view of the user.

Fundamentals when Replacing a Tool

The controls support two different strategies:

1. The tool is transported to its previous location after use. The location remains reserved for the tool.
2. The tool is transported to another, unassigned location after use. Only the controls know which tool is actually located where.

Point two is meaningful when a machine is equipped with a replacement gripper that fetches the tool from its tool location before actually being used and then queues it. After use, the old tool location may now be occupied by a tool that has already been put down and therefore the next free location must be occupied.

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.

Comment No additional command required.

BR_TLB Returns one or more elements of the basic tool data of several tools from the tool memory.

Comment No additional command required.

BR_TLE Returns one or more elements of the tool cutter data of several tools from the tool memory.

Comment No additional command required.

CR_TDR Returns a complete basic data record or cutter data record of a tool in the tool memory.

Comment 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.

Comment No additional command required.

Block Tool Location

CR_TII The specified tool location is temporarily blocked from automatic assignment by the controls.

Requirement: The tool location must be free (unassigned).

Comment No additional command required.

Release Tool Location

CR_TIF The indicated tool location is released after a temporary block.

Comment No additional command required.

Remove Tool

Note: Removing a tool means deleting the tool from the tool database. The actual tool itself must be removed previously by the user.

CR_TRM The tool data at this tool location is deleted from the database.

Comment 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 effected.

CW_TLD Writes a single element of the basic tool data or cutter data in the tool memory.

Comment 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.

Comment This command must also be called out repeatedly to replace a tool of the same type.

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.

Comment This command must be carried out in the following order:

- CR_TRM remove old tool.
- BW_TDA write complete new tool data record.

Comment 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.

Requirement: The target location must be free (unassigned).

Comment No additional command required.

Read Active Tool Number

CR_ATN The number of the active tool is read out.

Comment No additional command required.

Read Active Cutter Number

CR_AEN The number of the active cutter is read out.

Comment No additional command required.

Read Long Identification

CR_DIS4 The directory entry of the valid tool list is read out. It is uploaded after every download by CW_TDF.

Comment 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%.

Comment No additional command required.

Initiate Download

CW_TDI The controls are prepared for the download of tool data.

Comment No additional command required.

Downloading Tool Data

CW_TDD The tool data for one or more tools is downloaded.

Comment This command must be carried out in the following order:

- CW_TDI initiate Download
- CW_TDD Write complete basic data record or cutter data record By executing CW_TDD repeatedly, all basic data and cutter data 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.

Comment No additional command required.

7.2 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:

Group	Device Type	Address
MSCX	SERCANS-A, SERCANS-P	[00]

Determining the Current SERCANS Error: ASE

MSCX Device Group

Name **ASE** **A**ctual **S**ystem **E**rror

Explanation The current system error is read out, whereby the answer 0x0000 indicates that the SERCANS card is functioning correctly.

FI Command

CR_ASE **(Single Read)**

CC_ASE **(Cyclic Read)**

CB_ASE **(Cyclic Read)**

Construction of Answer The following table shows the general construction of the answer of the FI command ASE. In line 1, column 4, the number of the drive is outputted that reports the current system error. Not all current system errors can be assigned directly to a drive. In this case, the single result "Drive No." is set to 0x0000.

Line 1	Column 1	...	Column 4

Value Range/Meaning of Columns

1 = 0x0000

2 = 0x0000

3 = Current system error

4 = Drive No.

Example ASE Read-out of the current system error returns LWL-Ring interrupted.

FI Command		00_CR_ASE
Line	**Column**	**Answer**
1	1	0x0000
	2	0x0000
	3	0x8009
	4	0x0000

Reference to Literature You can find more details regarding the communication phases in the Rexroth Indramat documentation:

"DIAX04 Drive with Servo Functions", General Instructions on Putting into Operation, DOK-DIAX04-SSE-xxVRS**-FK0x-EN

Clearing a SERCANS Error: CSE

MSCX Device Group

Name **CSE** **Cl**ear **S**ERCANS **E**rror

Explanation An error reported by the SERCANS card is cleared.

FI Command **CW_CSE** **(Single Read)**

Value to be written: The contents of the value parameter is not evaluated.

Construction of Answer The following table shows the general construction of the answer of the FI command CSE. In line 1, column 4, the number of the drive is outputted that reports the current system error. Not all current system errors can be directly assigned to a drive. In this case, the single result "Drive No." is set to 0x0000.

Line 1	Column 1	...	Column 4

Value Range/Meaning of Columns

1 = 0x0000

2 = 0x0000

3 = Current system error

4 = Drive No.

Example CSE The current system error is cleared.

FI Command		00_CW_CSE
Line	**Column**	**Answer**
1	1	0x0000
	2	0x0000
	3	0x0000
	4	0x0000

Reference to Literature You can find more details regarding SERCANS errors in the Rexroth Indramat documentation:

"SERCANS /SERCVME SERCOS interface components with universal µP interface or VMEbus", description of application, system structure and axis structure.

Device Type and Accompanying Components: DTY

MSCX Device Group

Name **DTY** **D**evice **TY**pe

Explanation The device type as well as the accompanying components of the selected device addresses are outputted.

FI Command **CR_DTY1** **(Single Read)**

Construction of Answer The following table shows the general construction of the answer of the FI command DTY1 . A line with three columns for the device type is outputted as well as the names of the first device component and the name of the second device component.

Line 1	Column 1	...	Column 3

Value Range/Meaning of Columns

1 =	Device type:	(see chapter 6.1 Elements of the FI Command, Identifier)
2 =	Component type1	IND_DEV.INI entry: Componenttype1=
3 =	Component type 2	IND_DEV.INI entry: Component-type2=

Example DTY1 Output the device type and the accompanying components of device address 00.

FI Command	00_CR_DTY1		
Answer			
Line	Column 1	Column 2	Column 3
1	SERCANS-A	NONE	NONE

Software Installation Data: SID

MSCX Device Group

Name **SID** Software Installation Data:

Explanation Information is returned regarding the installation. This information includes the installation path, the software version being used and service pack and release information.

FI Command Reading-in the installation data.

BR_SID1 **(Single Read)**

BC_SID1 **(Cyclic Read)**

Construction of Answer One line with 8 columns is outputted for the additional text.

Line 1	Column 1	...	Column 8

Meaning of the Column

1 = Base directory	[EXE files of the DOS-BOF]
2 = FI installation directory	[FI directory]
3 = Data directory	[in accordance to DOS-BOF]
4 = GBO version	[from INDRAMAT.ini]
5 = IF-DLL mode	[from INDRAMAT.ini]
6 = IF version	[from INDRAMAT.ini from DLL mode 400]
7 = Service pack info	[from INDRAMAT.ini from DLL mode 420]
8 = Release info	[from INDRAMAT.ini from DLL mode 420]

Example SID1 Return the information on the current installation.

FI Command		00_BR_SID1
Line	**Column**	**Answer**
1	1	D:\MT-CNC
	2	C:\MT-CNC\IND_DRV
	3	D:\MT-CNC\ANLAGE00
	4	005-18V05
	5	04.20
	6	04V03
	7	--
	8	--

SERCOS Parameters: SPA

MSCX Device Group

Name **SPA** **S**ERCOS **Pa**rameter

Explanation One SERCOS parameter of a drive or a SERCANS parameter is read out or is written. Each parameter consists of 7 elements, whereby any combination of elements can be selected by element coding.

FI Command

BR_SPA1_(1)_(2)_(3)	**(Single Read)**
BC_SPA1_(1)_(2)_(3)	**(Cyclic Read)**
BB_SPA1_(1)_(2)_(3)	**(Break Cyclic Read)**
BW_SPA1_(1)_(2)_(3)	**(Single Write)**
(1) = Drive address	[0...254]
(2) = Parameter No.	in format: X-Y-ZZZZ
(3) = Element coding	[standard or advanced format]

Parameter No.

Format X-Y-ZZZZ	**Value range**
X	S = standard data P = product data Y = SERCANS parameter
Y	[0...15] = parameter record
Z	[0...4095] = datablock no.

Element Coding Element coding in standard format allows individual elements, such as, e.g. operating date, to be requested. If several elements are to be read out in one request, then the element coding can be additive in advanced format, e.g. operating date (0x40) and unit (0x08) produces additive (0x48) → 48

The advanced format 0x80 has priority over 0x40.

Element	**Standard Format**	**Advanced Format**	**Format**	**Example**
Data status	S	01H	Hexadecimal word	0x0000
Name	N	02H	(string)	NC cycle time (TNcyc)
Attribute	A	04H	Hexadecimal double word	0x60110001
Unit	U	08H	String	µs
Min. input value	L	10H	Decimal word	2000
Max. input value	H	20H	Decimal word	20000
Operating date	D	40H	(see Displaying the Operating Date)	
Operating date, when no list		80H		

Displaying the Operating Date

The display of the operating date depends on the parameter number requested.

Decimal

Decimal values are given as floating points, e.g. 1.5. Leading spaces, zeros, plus and minus signs as well as following spaces are allowed.

Hexadecimal

Hexadecimal values are displayed by "0x...", e.g. 0x80. A maximum of 8 positions is allowed. Leading or following spaces are allowed. Leading additional zeros or plus and minus signs are not allowed.

Binary (max. 32 characters)

Leading or following 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 shown:

Format X-Y-ZZZZ	Value range
X	S = standard data P = product data
Y	[0..0,7] = parameter record
Z	[0...4095] = datablock no.

(see example SPA1/write).

Lists of Variable Length

Lists always begin with two decimal numbered for the actual length and maximal 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 difference 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 maximal length of the list (string) is also transmitted.

Example

Parameter S-0-0030, Operating Date
Standard Format: "DKC2.1-SSE-01V09"
Advanced Format: "16\n16\nDKC2.1-SSE-01V09"

Note: When requesting SERCANS parameters, can be anywhere within the range [0..254].

Reference to Literature

Additional information regarding the function of the standard and product-specific 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-xxVRS**-FKBx-EN

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-EN

Construction of Answer

The following table shows the general construction of the answer of the FI command SPA1 . Line 1 is outputted both when reading and when writing. Additional lines are only outputted when reading depending on the element coding.

Note: If the element coding has been requested in standard format then the first line is not applicable.

Note: Line 1 is a status line that either contains SERCOS/SERCANS errors or displays the successful processing of the FI command. If the command has been processed successfully, then columns 1 and 3 contain the value [0x0000].

In the first line, column 2 or column 4, the number of the drive is outputted that reports the SERCOS error or the global SERCANS error. Not all global SERCANS errors can be directly assigned to a drive. In this case, the single result "Drive No." is set to 0x0000.

Line	Column 1	Column 2	Column 3	Column 4
1	<SERCOS error>	<Drive no. SERCOS error>	<Global SERCANS error>	<Drive No.. Global SERCANS error>
2	(read) 1. Element corresponding to the element coding.			
...	...			
n	(read) -1...n: Element corresponding to the element coding.			

Rexroth Indramat

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

FI Command	**00_BR_SPA1_3_S-0-0003_48**			
Answer				
Line	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	0x0000	0x0000	0x0000	0x0000
2	µs			
3	2000			

Example SPA1 / write Write the ID number P-0-0037 in the parameter S-0-0305 of the 3rd drive (element coding 0x40).

Technical Background:

- Real-time status bit 1 is to be assigned the trigger status word of the oscilloscope function of a DIAX04 drive.

FI Command	**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

Active SERCOS Phase Switch-Over: SPH

MSCX Device Group

Name **SPH** **S**ERCOS **Ph**ase

Explanation All drives within a SERCOS ring are in the same communication phase. The phase condition 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 conditions allowed are shown by the numbers [0...4].

Reference to Literature You can find more details regarding the communication phases in the Rexroth Indramat documentation:

"DIAX04 Drive with Servo Functions", General Instructions on Putting into Operation, DOK-DIAX04-SSE-xxVRS**-FK0x-EN-P

Construction of Answer The following table shows the general construction of the answer of the FI command SPH . In the first line, column 2 or column 4, the number of the drive is outputted that reports the SERCOS error or the global SERCANS error. Not all global SERCANS errors can be directly assigned to a drive. In this case, the single result "Drive No." is set to 0x0000.

Line	Column 1	Column 2	Column 3	Column 4
1	<SERCOS error>	<Drive No.. SERCOS error>	<Global SERCANS error>	<Drive no. that has caused the global SERCANS error>
2	Read: write current phase: previous phase			

Example SPH Switch-over (write) after phase 4; phase 2 is active.

FI Command	**00_CW_SPH with value 4**			
Answer				
Line	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	0x0000	0x0000	0x0000	0x0000
2	2			

7.3 FI Commands for the MISX Device Group

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

Group	Accompanying Types	Address
MISX	ISP200-P, ISP200-R	[00...63]

Note: Please note that the device address must be set before the respective FI command, e.g. 00_BR_ASM1 (refer also here to chapter 6.1 Elements of the FI Command).

Active System Error Messages: ASM

MISX Device Group

Name **ASM** **A**ctive **S**ystem **M**essages

Explanation The active system error messages that effect the functioning of the entire electrical device are outputted Depending on the FI command, the device address, device name, message number, type of message, short text and additional text are all outputted.

FI Command Output of all existing current system error messages of all active devices from the MISX device group.

BR_ASM1 **(Single Read)**

BC_ASM1 **(Cyclic Read)**

BB_ASM1 **(Break Cyclic Read)**

Note: The ASM1 FI command refers to all devices within this device group. Therefore, any valid device address can be indicated in the command line (see Example ASM1).

Construction of Answer The following table shows the general construction of the answer of the FI command ASM1 . The number of lines (1 .. n=15) depends on the number of defined devices. Every 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 1...n:	Column 1	...	Column 7

Value Range/Meaning of Columns

1 =	Device address	[00...15]
2 =	Device name	max. 32 ASCII characters
3 =	Message number	[0...150]
4 =	Type of message	[F = Fault/Error, D = Diagnosis]
5 =	Short text	max. 54 ASCII characters
6 =	Additional Text	[x= exists, -- = does not exist]
7 =	2 bytes of additional info for the message number	is required to resolve the information „@“ (see ASM5)

Example ASM1 Read the current system error messages of all defined devices of the MISX device group.

Assumption: the following three devices are defined:

- Device address 01:
- Device address 07 and
- Device address 10:

FI Command		07_BR_ASM1
Line	**Column**	**Answer**
1	1	01
	2	Drill center
	3	71
	4	F
	5	SPS 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 of existing current system error message of the selected device from the MISX device group.

BR_ASM2 **(Single Read)**

BC_ASM2 **(Cyclic Read)**

BB_ASM2 **(Break Cyclic Read)**

Construction of Answer

The following table shows the general construction of the answer of the FI command ASM2 . The answer consists of a line of 7 columns for the device address, device name, message number, message status, short text and indication of whether there is an additional text for this error message.

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

Value Range/Meaning of Columns

1 =	Device address	[00...15]
2 =	Device name	max. 32 ASCII characters
3 =	Message number	[0...150]
4 =	Type of message	[F = Fault/Error, D = Diagnosis]
5 =	Short text	max. 54 ASCII characters
6 =	Additional Text	[x= exists, -- = does not exist]
7 =	2 bytes of additional info for the message number	is required to resolve the information „@“ (see ASM5)

Example ASM2

Read the current system error messages of device address 01.

Assumption: the following three devices are defined:

- Device address 01:
- Device address 07 and
- Device address 10:

FI Command		01_BR_ASM2
Line	**Column**	**Answer**
1	1	01
	2	Drill center
	3	71
	4	F
	5	SPS battery voltage too low.
	6	X
	7	0

FI Command

Output of all current system error messages of the device listed from the MISX 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 MISX devices [00_01_02_ ... _15]

Construction of Answer

The following table shows the general construction of the answer of the FI command ASM3 . The number of lines (1 .. n=15) depends on the number of defined listed MISX devices. Every 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 1...n:	Column 1	...	Column 7

Value Range/Meaning of Columns

1 =	Device address	[00...15]
2 =	Device name	max. 32 ASCII characters
3 =	Message number	[0...150]
4 =	Type of message	[F = Fault/Error, D = Diagnosis]
5 =	Short text	max. 54 ASCII characters
6 =	Additional Text	[x= exists, -- = does not exist]
7 =	2 bytes of additional info for the message number	is required to resolve the information „@“ (see ASM5)

Example ASM3

Read the current system error messages of the selected MISX devices.

Assumption: The following device types have been 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	SPS battery voltage too low.
	6	X
	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 of all current system error messages of all defined devices (in accordance to the system configuration) from the MISX device group.

BR_ASM4_(1)	**(Single Read)**
BC_ASM4_(1)	**(Cyclic Read)**
BB_ASM4_(1)	**(Break Cyclic Read)**
(1) = device group	MISX

Construction of Answer

The following table shows the general construction of the answer of the FI command ASM4. The number of lines (1 .. n=15) depends on the number of defined MISX devices. Every 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 1...n:	**Column 1**	**...**	**Column 7**

Value Range/Meaning of Columns

1 =	Device address	[00...15]
2 =	Device name	max. 32 ASCII characters
3 =	Message number	[0...150]
4 =	Type of message	[F = Fault/Error, D = Diagnosis]
5 =	Short text	max. 54 ASCII characters
6 =	Additional Text	[x= exists, -- = does not exist]
7 =	2 bytes of additional info for the message number	is required to resolve the information „@“ (see ASM5)

Example ASM4

Read the current system error messages of all defined devices of the MISX device group.

Assumption: The following devices have been defined:

- Device address 01 and
- Device address 10:

FI Command		01_BR_ASM4_MISX
Line	**Column**	**Answer**
1	1	01
	2	Drill center
	3	71
	4	F
	5	SPS battery voltage too low.
	6	X
	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 of the additional text for the currently existing error message, related to the device and the message number.

BR_ASM5_(1)_(2) **(Single Read)**

(1) = message number [0...150]

(2) = 2 bytes of additional info for the message number

Construction of Answer

The following table shows the general construction of the answer of the FI command ASM5. The answer consists of a line with 5 columns for the device address, device name, message number and additional text.

Line 1...n:	Column 1	...	Column 5

Rexroth Indramat

Value Range/Meaning of Columns

1 =	Device address	[00...15]
2 =	Device name	max. 32 ASCII characters
3 =	Message number	[0...150]
4 =	Type of message	[F = Fault/Error, D = Diagnosis]
6 =	Additional Text	[max. 14 lines with a max. 78 characters/line]

Example ASM5

Read the additional text of the system error with message number 74 of device address 01.

FI Command		**03_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

Additional 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-EN

Trigger Control Reset: CRT

MISX Device Group

Name

CRT **C**ontrol **R**ese**T**

Explanation

The control reset allows the selected device to be reset after a system error. If there is no system error at the selected device then the job is ignored.

ATTENTION

Carrying out a reset completely re-initializes the device.
During initialization, communication is thereby temporarily interrupted (inherent to design).

FI Command

CW_CRT **(Single Write)**

Value to be written

Trigger reset 0

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

Error Codes

The return value of the "DataTransfer" routine is [0] when the write procedure has been successfully completed. In case of an error, more information can requested by the routine "ReadGroupItem" in the form of a general error result line (refer here to chapter 8, Error Codes).

Example CRT

trigger a control reset on the selected device.

FI Command	**00_CW_CRT**
Value to be written	0

Reference to Literature

You can find more details on resetting the device in the Rexroth Indramat documentation:

"SPS Programming Instructions xxVRS", Chapter "Putting into Operation / Start", DOK-CONTRL-SPS*PRO*Vxx-AW0x-EN

.

Long ID of the SPS Data Record: DIS

MISX Device Group

Name

DIS **D**ata **I**dentification **S**tring

Explanation

Reads the long ID (directory entries) of the SPS program. Included in the directory entries are the number of the entry in the directory, the name, length and date and time of creation and/or details of the last time the respective data record was changed.

FI Command

CR_DIS2 **(Single Read)**

Construction of Answer

The following table shows the general construction of the answer of the FI command DIS2 . The answer consists of a line with five columns.

Line 1	Column 1	...	Column 5

Value Range/Meaning of Columns

1 =	Number in SPS directory	[01...99]
2 =	Name of the SPS program	max. 8 ASCII characters
3 =	Length of the SPS program	[bytes]
4 =	Date of creation/last change to SPS program	[DD.MM.YY]
5 =	Date of creation/last change to SPS program	[HH:MM:SS]

Note: If there is no valid NC package in the selected NC memory of the indicated device then column 1 contains an empty string and columns 2 to 5 are not applicable (redundant).

Example DIS2

Read the directory entries of the SPS program at address 00. Assumption:

- there is a valid SPS program in the selected device.

FI Command		00_CR_DIS2
Line	**Column**	**Answer**
1	1	01
	2	KEY1
	3	20018
	4	10.05.99
	5	12:42:00

Reference to Literature

You can find more details regarding the SPS Programming System in the Rexroth Indramat documentation:

"SPS Programming Instructions xxVRS" Application Description" DOK-CONTRL-SPS*PRO*Vxx-AW0x-EN

Device Type and Accompanying Components: DTY

MISX Device Group

Name

DTY **D**evice **TY**pe

Explanation

The device type as well as the accompanying components of the selected device addresses are outputted.

FI Command

CR_DTY1 **(Single Read)**

Construction of Answer

The following table shows the general construction of the answer of the FI command DTY1. A line with three columns for the device type is outputted as well as the names of the first device component and the name of the second device component.

Line 1	Column 1	...	Column 3

Value Range/Meaning of Columns

1 =	Device type:	(see chapter 6.1 Elements of the FI Command, Identifier)
2 =	Component type1	IND_DEV.INI entry: Componenttype1=
3 =	Component type 2	IND_DEV.INI entry: Componenttype2=

Example DTY1

Output the device type and the accompanying components of device address 00.

FI Command	00_CR_DTY1		
Answer			
Line	**Column 1**	**Column 2**	**Column 3**
1	ISP200-P	MTS-P	NONE

Read Reference Name of a SPS Variable : MAR

MISX Device Group

Name **MAR** **M**ap **A**bsolute PCL **R**eference

Explanation The absolute reference name of a symbolic SPS variable is read out.

FI Command Reads the absolute SPS reference name of a SPS variable.

BR_MAR_(1) **(Single Read)**

(1) = Identifier of the SPS variable

Example MAR Read the absolute reference name of the SPS variable with the identifier “abref" at device address 00.

Assumption:

- the SPS variable with the identifier "abref" is of the type "INTEGER"

FI Command	**00_BR_MAR_abref**
Answer	
Line	**Column 1**
1	%M100.0

Device Data of the Module Configuration: MCD

MISX Device Group

Name **MCD** **M**odule **C**onfiguration: **D**evice Information

Explanation All device data of the module configuration are read-out from the "Moduldef.ini" file that is stored in the "[LW]:\MT-CNC\CONFIG" directory. The device data are in the sections [DeviceAddrX], whereby "X" stands for the configured device addresses.

FI Command Read-out of device data within the module configuration of the MISX device groups.

BR_MCD1 **(Single Read)**

BC_MCD1 **(Cyclic Read)**

BB_MCD1 **(Break Cyclic Read)**

Note: The MCD1 FI command refers to all devices within the MISX device group. Therefore, any valid device address can be indicated in the command line (see Example MCD1).

Construction of Answer The following table shows the general construction of the answer of the FI command MCD1 . The number of lines depends on the number of configured devices. Every line consists of four columns for the device address as well as SPS-FB (FB = function component) names for the provision of setup diagnostics, warning messages and start requirements.

Value Range of the Columns

1 = Device address	[0...15]
2 = SPS-FB name for the setup diagnostics	max. 9 ASCII characters
3 = SPS-FB name for the warning messages	max. 9 ASCII characters
4 = SPS-FB name for the start requirements	max. 9 ASCII characters

Example MCD1

Read all device data of the module configuration

Assumption: The following devices have been configured in the MISX device group:

- Device address 01 (ISP200-P)
- Device address 03 (ISP200-R)

FI Command	03_BR_MCD1			
Answer				
Line	Column 1	Column 2	Column 3	Column 4
1	01	PVSetup_1	PVWarn_1	PVStart_1
2	03	PVSetup_3	PVWarn_3	PVStart_3

Reference to Literature

Additional information regarding module configuration and the construction of the "Moduldef.ini" file can be located in the following Rexroth Indramat documentation:

"Diagnostics and Message System for HMI System ProVi", chapter "Configure Moduldef.ini", DOK-MTC200-DIAG*PROVI*-AW0x-EN

Device Data of the Module Configuration: MCM

MISX Device Group

Name

MCM **M**odule **C**onfiguration: **M**odule Information

Explanation

All module data of a particular device is read out from the "Moduldef.ini"' file. This data is located in the "[LW]:\MT-CNC\CONFIG" directory and contains all module configuration data. The module data is located in sections [DeviceAddrX\ModulY], whereby "X" stands for the device addressed and "Y" for the configured module numbers.

FI Command

Read-out of module data from the module configuration with respect to a device from the MISX device group.

BR_MCM1 **(Single Read)**

BC_MCM1 **(Cyclic Read)**

BB_MCM1 **(Break Cyclic Read)**

Construction of Answer

The following table shows the general construction of the answer of the FI command MCM1. 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 SPS-FB names for general module errors and module messages.

Line 1	Column 1	...	Column 4

Value Range of the Columns

1 = Module number	[0...99]
2 = Module name	[max. 28 ASCII characters]
3 = SPS-FB name for general module errors	[max. 9 ASCII characters]
4 = SPS-FB name for module messages	[max. 9 ASCII characters]

Example MCM1

Read the module data of device 03 from the module configuration:

Assumption: The following modules have been defined:

- Module number 5
- Module number 7

FI Command	03_BR_MCM1			
Answer				
Line	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	5	Module 5 – Milling	PVError_5	PVMsg_5
2	7	Module 7 - Drilling	PVError_7	PVMsg_7

Reference to Literature

Additional information regarding module configuration and the construction of the "Moduldef.ini" file can be located in the following Rexroth Indramat documentation:

Diagnostics and Message System for HMI System ProVi, chapter "Configure Moduldef.ini", DOK-MTC200-DIAG*PROVI*-AW0x-EN

SFC Data of the Module Configuration: MCS

MISX Device Group

Name

MCS **M**odule **C**onfiguration: **S**FC Information

Explanation

All SFC data of a particular module is read out from the "Moduldef.ini"' file. This data is located in the [LW]:\MT-CNC\CONFIG directory and 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 of the SFC data with respect to the module of a device from the module configuration of the MISX device group.

BR_MCS1_(1) **(Single Read)**

BC_MCS1_(1) **(Cyclic Read)**

BB_MCS1_(1) **(Break Cyclic Read)**

1 = module number [0...99]

Construction of Answer

The number of lines depends on the number of configured Indrastep Step Chains of a device. Each line contains a column for the name of the Indrastep Step Chains.

Value Range of the Column

1 = Name of the Indrastep Step Chain [Format W.X.Y.Z]

Format W.X.Y.Z	Value range
W	max. 9 ASCII characters
X	max. 9 ASCII characters OPTIONAL !
Y	max. 9 ASCII characters OPTIONAL !
Z	max. 9 ASCII characters OPTIONAL !

Example MCS1

Read the name of the Indrastep Step Chain of module 5 from device 03 of the module configuration.

Assumption: The following Indrastep Step Chains have been defined:

- ISFB_1
- FB_US.ISFB_3
- FB_US.ISFB_3.SW1.ABBA
- FB_US.ISFB_3.SW1.ABBA

FI Command	03_BR_MCS1_5
Answer	
Line	**Column 1**
1	ISFB_1
2	FB_US.ISFB_3
3	FB_US.ISFB_3.SW1
4	FB_US.ISFB_3.SW1.ABBA

Formatted Input / Output of SPS Variables: PVF

MISX Device Group

Name

PVF **P**LC **V**ariable **F**ormatted

Explanation

Formatted reading and writing of SPS variables, arrays and structures.

FI Command

Read SPS variables.

CR_PVF1_(1) **(Single Read)**

CC_ PVF1_(1) **(Cyclic Read)**

CB_ PVF1_(1) **(Break Cyclic Read)**

(1) = Identifier of the SPS variable [acc. To declaration part of SPS]

Construction of Answer

One line with one column is outputted for simple variables. For array and structure variables, one line per element is outputted, depending on the number of elements.

Line 1...n:	Column 1

n = number of elements.

Note: Only defined SPS variables can be read and written. Addressing a non-declared variable results in an error message. A SPS variable can only be read when its data length does not exceed 240 (see also chapter 4.1, Guidelines).

Value Ranges ANSI / ASCII

The value range of the answer depends on the data type of the variable read. The following table informs you of the range in which the results string is to be expected when reading out a simple variable and into which C-data type this string can be converted without loss of information:

Data Type	Value range	Can be converted to C-data type
BOOL	[0;1]	unsigned char
SINT	[-128...127]	char
INT	[-32768...32767]	short
DINT	[2147483648...2147483647]	long
USINT	[0...255]	unsigned char
UINT	[0...65535]	unsigned short
UDINT	[0...4294967295]	unsigned long
BYTE	[0x00...0xFF]	unsigned char
WORD	[0x0000....0xFFFF]	unsigned short
DWORD	[0x00000000...0xFFFFFFFF]	unsigned long
TIME	[0...4294967295]	unsigned long (msec)
CHAR	[$00...$20,!...~,$7F...$FF]	char
STRING	<String> whereby <String> string is with a maximal of as many characters as are defined for the string in the SPS	Char[xx+1]] +1 e.g. room for the zero byte
REAL	[-3.402823567E+38...3.402823567E+38]	Float

Note: An empty string can be recognized by simple double-inverted commas: “

All simple 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 answer depends on the data type of the variable read. The following table informs you of the value range in which to expect the binary value of a simple variable and how many bytes are included in the binary byte sequence:

Data Type	Value range	Length (bytes)
BOOL	[00_H...01_H]	1
SINT	[80_H...$7F_H$] i.e. –128...127	1
INT	[8000_H (-32768)...$7FFF_H$ (32767)]	2
DINT	[80000000_H (-2147483648)... $7FFFFFFF_H$ (2147483647)]	4
USINT	[00_H (0)...FF_H (255)]	1
UINT	[00_H (0)...$FFFF_H$ (65535)]	2
UDINT	[0...4294967295]	4
BYTE	[0x00...0xFF]	1
WORD	[0x0000....0xFFFF]	2
DWORD	[0x00000000...0xFFFFFFFF]	4
TIME	[0...4294967295]	4
CHAR	[$00...$20,!...~,$7F...$FF]	1
STRING	<String> whereby <String> string is with a maximal of as many characters as are defined for the string in the SPS	XX+1
REAL	[-3.402823567E+38...3.402823567E+38]	4

Note: Binary array and structure elements are joined together without any spaces between (1-byte alignment).

Example 1 PVF1

Read the value of the SPS variables "STK_TXT" in ASCII format from device address 00.

Assumption:

The "STK_TXT" variable is declared as a string in the SPS program.

FI Command		00_CR_PVT1_STK_TXT
Line	**Colum n**	**Answer**
1	1	Repeat counter

Example 2 PVF1

Read the value of the SPS array "BEG_END" in ANSI format from device address 00.

Assumption:

The "BEG_END" variable is declared as BYTE with 2 elements in the SPS program.

FI Command		00_CR_PVT1_BEG_END/3
Line	Column	Answer
1	1	0x00
2		0x1F

Example 3 PVF1

Read the value of the SPS structure "MSTRCT" in ASCII format from device address 00.

Assumption:

The "MSTRCT" variable is declared as a structure in the SPS program as follows.

```
TYP STRUCT
        T1      BOOL
        T2      CHAR
        T3      STRING[16]
        T4      TIME
END
```

FI Command		00_CR_PVT1_MSTRCT/1
Line	Column	Answer
1	1	0
2		A
3		ROBOT AXIS X
4		2000

FI Command

Write SPS variables.

CW_PVF1_(1) **(Single Read)**

(1) = Identifier of the SPS variable [acc. to declaration part of SPS]

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“.

Construction of Answer

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

(P_ACK) = **P**ositive **ACK**nowledge Data element has been set

Value Range of the value to be written in ANSI / ASCII Format

The value ranges agree for the most part with the result-value ranges ANSI / ASCII during read access. For deviations to this, please refer to the following note:

Note: Strings are bracketed by two simple inverted commas ’ ’ . e.g. ’drill’

Special characters can be marked in accordance to DIN-1131 by a $ sign.

There are:

- $’’
- $$ $
- $R \r (carriage return)
- $L \n (linefeed)
- $P \f (formfeed)
- $T \t (tab)
- $xx xx refers to a character written as a hexadecimal value. e.g. $20 (space)

Array and structure elements are separated by a space.

Value Range of the Value to be written in Binary Format

The value ranges agree with the binary result-value range during read access. For deviations to this, please refer to the following note:

Example 4 PVF1

Write the value of the SPS variable "STK_TXT" to device address 00. The value is output in ASCII format.

Assumption:

The "STK_TXT" variable is declared as a string in the SPS program.

FI Command		00_CW_PVT1_STK_TXT
Line	**Column**	**Answer**
1	1	(P_ACK)

Value to be written

Value of data element	’item counter’
Data code	1

Example 5 PVF1

Write the value of the SPS array "BEG_END" at device address 00. The value is output in ANSI format.

Assumption:

The "BEG_END" variable is declared as BYTE with 2 elements in the SPS program.

FI Command		00_CW_PVT1_BEG_END
Line	**Column**	**Answer**
1	1	(P_ACK)

Value to be written

Value of data element	0x20 0x3f
Data code	3

Example 6 PVF1

Write the value of element T3 of the SPS structure "MSTRCT" at device address 00. The string “COUNTER“ is output in binary format.

<u>Assumption:</u>

The "MSTRCT" variable is declared as a structure in the SPS program as follows.

```
TYP STRUCT
        T1      BOOL
        T2      CHAR
        T3      STRING[16]
        T4      TIME
END
```

FI Command		00_CW_PVT1_MSTRCT.T3
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

Example 7 PVF1 Write the value of the SPS structure "MSTRCT" from the structure mstrct previously stored in the C program at device address 00.

<u>Assumption:</u>

The "MSTRCT" variable is declared as a structure in the SPS program as follows.

```
TYP STRUCT
        T1      BOOL
        T2      CHAR
        T3      STRING[16]
        T4      TIME
END
```

For the exchange of binary data in a C program, the following 'C' data type can be used:

FI Command		00_CW_PVT1_MSTRCT
Line	**Column**	**Answer**
1	1	(P_ACK)

Value to be written Address of the C structure.

Value of data element	&mstrct
Data code	2

Reading and Writing SPS Variables: PVS

MISX Device Group

Name **PVS** **PLC**-**V**ariable **S**ingle

Explanation The following types of SPS variable can be read or written:

- BOOL, BYTE, SINT, USINT, CHAR, WORD, INT, UINT, STRING, DWORD, DINT, UDINT, TIME, REAL as well as imported structures and arrays.

FI Command Reading SPS variables.

CR_PVS_(1)	**(Single Read)**
CC_PVS_(1)	**(Cyclic Read)**
CB_PVS_(1)	**(Break Cyclic Read)**

(1) = Identifier of the SPS variable

4

Example PVS Read the value of the SPS variable with identifier "IB_EXT24" at device address 00 in CNC process 0.

Assumption:

- the SPS variable with the identifier "IB_EXT24" is of the type "BOOL"

FI Command	**00_CR_PVS_IB_EXT24**
Answer	
Line	**Column 1**
1	1

FI Command Writing a SPS Variable.

CW_PVS_(1)	**(Single Write)**

(1) = Identifier of the SPS variable

Value to be written

SPS variable	[Format acc. to the type in the SPS program]

Note: Only defined SPS variables can be written. Addressing a non-declared variable results in an error message. The length of the data must not exceed 240 bytes (see also chapter 4.1, Guidelines). The value to be written is passed to the "acValue" parameter in the "DataTransfer" routine.

Construction of Answer The return value of the "DataTransfer" routine is [0] when the write procedure has been successfully completed. In case of an error, more information can requested by the routine "ReadGroupItem" in the form of a general error result line (refer here to chapter 8, Error Codes).

Example PVS Write the value 1 in the SPS variable with the identifier "IB_EXT24" at device address 00.

Assumption:

- the SPS variable with the identifier "IB_EXT24" is of the type "BOOL"

FI Command	**00_CW_PVS_IB_EXT24**
Value to be written	1

Reading the SPS Variable Declaration: PVT

MISX Device Group

Name **PVT** **P**LC **V**ariable **T**ype

Explanation A SPS 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 SPS Variables.

FI Command Reading-out the SPS Variable Type.

BR _PVT_(1)	**(Single Read)**
(1) = Identifier of the SPS variable	[acc. to declaration part of SPS]

Construction of Answer One line with 2 columns is outputted for each element of the variables.

Line 1...n:	Column 1	Column 2

n = number of elements.

Value Range/Meaning of Columns

(1) = Identifier of the SPS variable	[acc. to declaration part of SPS]
2 = Type	[see value range PVF]

Examples:

Reading a simple variable Assumption:

The variable TEST of type WORD is declared in the SPS (device 0).

FI Command	**00_BR_PVT_TEST**	
Answer		
Line	**Column 1 (Name)**	**Column 2 (Name)**
1	TEST	WORD

Reading a Structure

Assumption:

The variable TEST1 of type STRUCKT is declared in the SPS (device 0).

STRUCT

E1 BOOL

E2 INT

E3 SINT

END

FI Command	00_BR_PVT_TEST1	
Answer		
Line	Column 1	Column 2
1	TEST1.E1	BOOL
2	TEST1.E2	INT
3	TEST1.E3	SINT

Reading an Array

Assumption:

The variable TEST2 of type ARRAY is declared in the SPS (device 0).

ARRAY [

0 .. 3

] OF BOOL

....

FI Command	00_BR_PVT_TEST2	
Answer		
Line	Column 1	Column 2
1	TEST2[0]	BOOL
2	TEST2[1]	BOOL
3	TEST2[2]	BOOL
4	TEST2[3]	BOOL

Reading an Array of a Structure

Assumption:

The variable TEST3 of type ARRAY is declared in the SPS (device 0).

ARRAY [

0 .. 1

] OF STRUCT1,

whereby STRUCT1 with

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

Comment:

The data types are outputted according to IEC1131.

See also command PVF.

Software Installation Data: SID

MISX Device Group

Name **SID** Software Installation Data:

Explanation Information is returned regarding the installation. This information includes the installation path, the software version being used and service pack and release information.

FI Command Reading-in the installation data.

BR_SID1 **(Single Read)**

BC_SID1 **(Cyclic Read)**

Construction of Answer One line with 8 columns is outputted for the additional text.

Line 1	Column 1	...	Column 8

Meaning of the Column

1 = Base directory	[EXE files of the DOS-BOF]
2 = FI installation directory	[FI directory]
3 = Data directory	[in accordance to DOS-BOF]
4 = GBO version	[from INDRAMAT.ini]
5 = IF-DLL mode	[from INDRAMAT.ini]
6 = IF version	[from INDRAMAT.ini from DLL mode 400]
7 = Service pack info	[from INDRAMAT.ini from DLL mode 420]
8 = Release info	[from INDRAMAT.ini from DLL mode 420]

Example SID1 Return the information on the current installation.

FI Command		00_BR_SID1
Line	**Column**	**Answer**
1	1	D:\MT-CNC
	2	C:\MT-CNC\IND_DRV
	3	D:\MT-CNC\ANLAGE00
	4	005-18V05
	5	04.20
	6	04V03
	7	--
	8	--

SPS Long Identification: SLI

MISX Device Group

Name

SLI **SPS L**ong **I**dentification

Explanation Returns the single data from the SPS long identification.

FI Command Read SPS long identification.

BR_SLI **(Single Read)**

Construction of Answer One line with 15 columns is outputted for the returned values.

Line 1	Column 1	…	Column 15

Value Range/Meaning of the Columns

1 = Device address	[00...15]
2 = Program number	[01...99]
3 = Project name	[max. 8 ASCII characters]
4 = Program name	[max. 8 ASCII characters]
5 = User name	[acc. to password entry]
6 = Program length	[bytes]
7 = Compilation time	[LONG] (coded in long value)
8 = Compilation date	[8 ASCII characters]
9 = Compilation time	[8 ASCII characters]
10 = Download time	[LONG] (coded in long value)
11 = Download date	[8 ASCII characters]1
12 = Download time	[8 ASCII characters]
13 = Version of SPS long identification	[LONG]
14 = RUN Flags	[HEX value]
15 = Compiler info	[LONG]

Example SLI Read the single data from the SPS long identification.

FI Command		**00_BR_SLI**
Line	**Column**	**Answer**
1	1	02
	2	01
	3	--
	4	MOT12
	5	TEST
	6	17672
	7	630163960
	8	15.12.99
	9	17:15:48
	10	630163961
	11	15.12.99
	12	17:15:50
	13	2
	14	0x0000
	15	13

7.5 FI Commands for the MTAX Device Group

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

Group	Accompanying Types	Address
MTAX	MTA200-P (ANDRON controller)	[00...15]

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).

Active Mechanism Messages: AMM

MTAX Device Group

Name

AMM **A**ctive **M**echanism **M**essages

Explanation

Messages regarding active mechanism errors and mechanism diagnostics are outputted. 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 outputted.

FI Command

Output of all active mechanism messages currently pending.

BR_AMM7 **(Single Read)**

BC_AMM7 **(Cyclic Read)**

BB_AMM7 **(Break Cyclic Read)**

Note: The AMM7 FI command refers to all devices within the MTAX device group. You should therefore make sure that only MTA devices are addressed via the system address.

Construction of Answer

The following table shows the general construction of the answer of the FI command AMM7. The answer consists of 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 pasted into the first line. The maximum content for a result may not exceed 56 Kbytes.

Line 1...n:	Column 1	...	Column 11

Value Range/Meaning of the Columns

1 = Device address	[00...15]
2 = Device name	[max. 32 ASCII characters]
3 = Mechanism number	[0, default value always 0]
4 = Mechanism name	[max. 28 ASCII characters, default value always the MTA process]
5 = Message type	[F = Fault/Error, D = Diagnosis]
6 = Message source	[CNC, SPS, default value always "CNC"]
7 = Message group	[1...9999]
8 = Message number	[0...32768]
9 = Message text	[max. 1024 ASCII characters]
10 = Additional text	[X = exists, - - = does not exist, Default value - - does not exist (Compatibility to Rexroth Indramat controls)]
11 = 2 bytes of additional info for the message number	[is required to resolve the information "@", default value "0" (Compatibility to Rexroth Indramat controls)]

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
2	1	03
	2	Crankshaft grinding machine
	3	0
	4	MTA process
	5	F
	6	CNC
	7	1
	8	6
	9	Cycle point error
	10	- -
	11	0
3	1	03
	2	Crankshaft grinding machine
	3	0

	4	MTA process
	5	F
	6	CNC
	7	1
	8	19
	9	Emergency-OFF with immediate stop
	10	- -
	11	0

Current (Actual) Position of an Axis: APO

MTAX Device Group

Name **APO** **A**ctual Axis **PO**sition

Explanation The current (actual) position of a selected axis of the MTAX device group is read out. The FI command "APO2" returns the position of an axis, related to the physical axis number.

FI Command Output of the position of the selected axis of the device specified, related to the physical axis number.

CR_APO2_(1)_(2) **(Single Read)**

CC_APO2_(1)_(2) **(Cyclic Read)**

CB_APO2_(1)_(2) **(Break Cyclic Read)**

(1) = Physical axis number [1...16]

(2) = Coordinate system [1 = Machine coordinates
 2 = Program coordinates]

Construction of Answer The following table shows the general construction of the answer of the FI command APO2 . One line with three columns for the name of the axis, the position and the unit is outputted in accordance to the settings of the process parameters.

Line 1	Column 1	...	Column 3

Value Range/Meaning of Columns

1 = Axis name [acc. to settings of axis parameter]

2 = Position [acc. to settings of axis parameter]

3 = Unit [acc. to settings of axis process parameter] mm, inch]

8

Example APO2 Read the current position of the Z axis (physical axis number = 3) in machine coordinates are the device address 00.

FI Command	**00_CR_APO2_3_1**		
Answer			
Line	**Column 1**	**Column 2**	**Column 3**
1	Z	-5.9897	mm

Reference to Literature Additional 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-EN

Active System Error Messages: ASM

MTAX Device Group

Name **ASM** **A**ctive **S**ystem **M**essages

Explanation The active system error messages that effect the functioning of the entire electrical device are outputted Depending on the FI command, the device address, device name, message number, type of message, short text and additional text are all outputted. Access to system error messages only refers to the SPS part (ISP200).

FI Command Output of the current system error messages pending of all active devices from the MTAX device group.

BR_ASM1 **(Single Read)**

BC_ASM1 **(Cyclic Read)**

BB_ASM1 **(Break Cyclic Read)**

Note: The ASM1 FI command refers to all devices within this device group. Therefore, any valid device address can be indicated in the command line (see Example ASM1).

Construction of Answer The following table shows the general construction of the answer of the FI command ASM1 . The number of lines (1 .. n=15) depends on the number of defined devices. Every 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 1...n:	Column 1	...	Column 7

Value Range/Meaning of Columns

1 =	Device address	[00...15]
2 =	Device name	[max. 32 ASCII characters]
3 =	Message number	[0...150]
4 =	Type of message	[F = Fault/Error, D = Diagnosis]
5 =	Short text	[max. 54 ASCII characters]
6 =	Additional Text	[x= exists, -- = does not exist]
7 =	2 bytes of additional info for the message number	is required to resolve the information „@“ (see ASM5)

Example ASM1

Read the current system error messages of all defined devices of the MTAX device group. Assumption: the following three devices are defined:

- Device address 01:
- Device address 07 and
- Device address 10:

FI Command		07_BR_ASM1
Line	**Column**	**Answer**
1	1	01
	2	Drill center
	3	71
	4	F
	5	SPS battery voltage too low.
	6	X
	7	0
2	1	07
	2	Milling center 1
	3	74
	4	F
	5	SLM time monitoring
	6	X
	7	0
3	1	10
	2	Milling center 2
	3	1
	4	D
	5	Error has been corrected.
	6	X
	7	0

FI Command

Output of existing current system error message of the selected device from the MTAX device group.

BR_ASM2 **(Single Read)**

BC_ASM2 **(Cyclic Read)**

BB_ASM2 **(Break Cyclic Read)**

Construction of Answer

The following table shows the general construction of the answer of the FI command ASM2. The answer consists of a line of 7 columns for the device address, device name, message number, message status, short text and indication of whether there is an additional text for this error message.

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

Value Range/Meaning of Columns

1 =	Device address	[00...15]
2 =	Device name	[max. 32 ASCII characters]
3 =	Message number	[0...150]
4 =	Type of message	[F = Fault/Error, D = Diagnosis]
5 =	Short text	[max. 54 ASCII characters]
6 =	Additional Text	[x= exists, -- = does not exist]
7 =	2 bytes of additional info for the message number	is required to resolve the information „@“ (see ASM5)

Example ASM2

Read the current system error messages of device address 01.

Assumption: the following three devices are defined:

- Device address 01:
- Device address 07 and
- Device address 10:

FI Command		01_BR_ASM2
Line	**Column**	**Answer**
1	1	01
	2	Drill center
	3	71
	4	F
	5	SPS battery voltage too low.
	6	X
	7	0

Reference to Literature

Additional 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-EN

FI Command

Output of all current system error messages of the device listed from the MTAX 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 MTAX devices [00_01_02_ ... _15]

Construction of Answer

The following table shows the general construction of the answer of the FI command ASM3. The number of lines (1 .. n=15) depends on the number of listed MTAX devices. Every 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 1...n:	Column 1	...	Column 7

Value Range/Meaning of Columns

1 =	Device address	[00...15]
2 =	Device name	[max. 32 ASCII characters]
3 =	Message number	[0...150]
4 =	Type of message	[F = Fault/Error, D = Diagnosis]
5 =	Short text	[max. 54 ASCII characters]
6 =	Additional Text	[x= exists, -- = does not exist]
7 =	2 bytes of additional info for the message number	is required to resolve the information „@“ (see ASM5)

Example ASM3

Read the current system error messages of the selected MTAX devices.

Assumption: The following device types have been 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	SPS battery voltage too low.
	6	X
	7	0
2	1	10
	2	Milling center 2
	3	1
	4	D
	5	Error has been corrected.
	6	X
	7	0

FI Command

Output of all current system error messages of all defined devices (in accordance to the system configuration) from the MTAX device group.

BR_ASM4_(1)	**(Single Read)**
BC_ASM4_(1)	**(Cyclic Read)**
BB_ASM4_(1)	**(Break Cyclic Read)**
(1) = device group	[MTRX, MTCX, MISX, MTAX]

Construction of Answer

The following table shows the general construction of the answer of the FI command ASM4. The number of lines (1 .. n=15) depends on the number of defined MTAX devices. Every 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 1...n:	Column 1	...	Column 7

Value Range/Meaning of Columns

1 =	Device address	[00...15]
2 =	Device name	[max. 32 ASCII characters]
3 =	Message number	[0...150]
4 =	Type of message	[F = Fault/Error, D = Diagnosis]
5 =	Short text	[max. 54 ASCII characters]
6 =	Additional Text	[x= exists, -- = does not exist]
7 =	2 bytes of additional info for the message number	is required to resolve the information „@“ (see ASM5)

Example ASM4

Read the current system error messages of all defined devices of the MTAX device group. Assumption: The following devices have been defined:

- Device address 01 and
- Device address 10:

FI Command		01_BR_ASM4_MTAX
Line	**Column**	**Answer**
1	1	01
	2	Drill center
	3	71
	4	F
	5	SPS battery voltage too low.
	6	X
	7	0
2	1	10
	2	Milling center 2
	3	1
	4	D
	5	Error has been corrected.
	6	X
	7	0

FI Command

Output of the additional text for the currently existing error message, related to the device and the message number.

BR_ASM5_(1)_(2) **(Single Read)**

(1) = message number [0...150]

(2) = 2 bytes of additional info for the message number

Construction of Answer

The following table shows the general construction of the answer of the FI command ASM5. The answer consists of a line with 5 columns for device addresses, device names, message number and additional text.

Line 1...n:	Column 1	...	Column 5

Value Range/Meaning of Columns

1 =	Device address	[00...15]
2 =	Device name	[max. 32 ASCII characters]
3 =	Message number	[0...150]
4 =	Type of message	[F = Fault/Error, D = Diagnosis]
6 =	Additional Text	[max. 14 lines with a max. 78 characters/line]

Example ASM5

Read the additional text of the system error with message number 74 of device address 01.

FI Command		03_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

Additional 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-EN

Reading and Writing CMOS RAM ASCII Parameters: CMA

MTAX Device Group

Name **CMA** **CM**OS RAM **A**SCII Parameter

Explanation CMOS RAM ASCII parameters can be read and written.

FI Command Reading of CMOS RAM ASCII parameters.

CR_CMA_(1) **(Single Read)**

(1) = CMOS RAM ASCII parameter numbers [0..79]

Construction of Answer One line with one column is outputted for the value of the selected CMOS RAM ASCII parameter.

Example Read CMA Parameter Read the value of the CMOS RAM ASCII parameter with the number 0 at device address 00.

FI Command	00_CR_CMA_0
Answer	
Line 1	Column 1
1	Waiting for tool change

FI Command Writing of CMOS RAM ASCII parameters.

CW_CMA_(1) **(Single Write)**

(1) = CMOS RAM ASCII parameter numbers [0..79]

Value to be written

Value of the parameter [ASCII characters]

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

Construction of Answer The return value of the "DataTransfer" routine is [0] when the write procedure has been successfully completed. In case of an error, more information can requested by the routine "ReadGroupItem" in the form of a general error result line (refer here also to chapter 8, Error Codes).

Example Write CMA Parameter Write "Waiting for tool change" in the CMOS RAM ASCII parameter numbered 0 at device address 00.

FI Command	00_CW_CMA_0
Value to be written	Waiting for tool change

Reading and Writing CMOS RAM Floating Point Parameters: CMF

MTAX Device Group

Name **CMF** **CM**OS RAM **F**loating Point Parameter

Explanation CMOS RAM Floating Point parameters can be read and written.

FI Command Reading of CMOS RAM Floating Point parameters.

CR_CMF_(1) **(Single Read)**

(1) = CMOS RAM Floating Point parameter numbers [0..79]

Construction of Answer One line with one column is outputted for the value of the selected CMOS RAM Floating Point parameter.

Example Read CMF Parameters Read the value of the CMOS RAM Floating Point parameter numbered 1 at device address 00.

FI Command	**00_CR_CMF_1**
Answer	
Line 1	**Column 1**
1	4711.0123

FI Command Writing of CMOS RAM Floating Point parameters.

CW_CMF_(1) **(Single Write)**

(1) = CMOS RAM Floating Point parameter numbers [0..79]

Value to be written

Value of the parameter [Type: floating point]

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

Construction of Answer The return value of the "DataTransfer" routine is [0] when the write procedure has been successfully completed. In case of an error, more information can requested by the routine "ReadGroupItem" in the form of a general error result line (refer here also to chapter 8, Error Codes).

Example Write CMF Parameter Write the value [4711.0123] in the CMOS RAM Floating Point parameter numbered 1 at device address 00.

FI Command	**00_CW_CMF_1**
Value to be written	4711.0123

Reading and Writing CMOS RAM Integer Parameters: CMI

MTAX Device Group

Name **CMI** **CM**OS RAM **I**nteger Parameter

Explanation CMOS RAM Integer parameters can be read and written.

FI Command Reading of CMOS RAM Integer parameters.

CR_CMI_(1) **(Single Read)**

(1) = CMOS RAM Integer parameter numbers [0..79]

Construction of Answer One line with one column is outputted for the value of the selected CMOS RAM Integer parameter.

Example Read CMI Parameters Read the value of the CMOS RAM Integer parameter numbered 2 at device address 00.

FI Command	**00_CR_CMI_2**
Answer	
Line 1	**Column 1**
1	120270

FI Command Writing of CMOS RAM Integer parameters.

CW_CMI_(1) **(Single Write)**

(1) = CMOS RAM Integer parameter numbers [0..79]

Value to be written

Value of the parameter [Type: integer]

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

Construction of Answer The return value of the "DataTransfer" routine is [0] when the write procedure has been successfully completed. In case of an error, more information can requested by the routine "ReadGroupItem" in the form of a general error result line (refer here also to chapter 8, Error Codes).

Example Write CMI Parameter Write the value [120270] in the CMOS RAM Integer parameter numbered 2 at device address 00.

FI Command	**00_CW_CMI_2**
Value to be written	120270

Trigger Control Reset: CRT

MTAX Device Group

Name **CRT** **C**ontrol **R**ese**T**

Explanation The control reset allows the selected device to be reset after a system error. If there is no system error at the selected device then the job is ignored.

ATTENTION

Carrying out a reset completely re-initializes the device.
During initialization, communication is thereby temporarily interrupted (inherent to design).

FI Command **CW_CRT** **(Single Write)**

Value to be written Trigger reset 0

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

Construction of Answer The return value of the "DataTransfer" routine is [0] when the write procedure has been successfully completed. In case of an error, more information can requested by the routine "ReadGroupItem" in the form of a general error result line (refer here to chapter 8, Error Codes).

Example CRT Trigger a control reset on the selected device.

FI Command	**00_CW_CRT**
Value to be written	0

Reference to Literature You can find more details on resetting the device in the Rexroth Indramat documentation:

"SPS Programming Instructions xxVRS", Chapter "Putting into Operation / Start", DOK-CONTRL-SPS*PRO*Vxx-AW0x-EN

Device Configuration Parameters: DCP

MTAX Device Group

Name **DCP** **D**evice **C**onfiguration **P**arameter

Explanation The device configuration parameters that are entered in the active machine parameter record as well as in the "IND_DEV.INI" file are outputted. The configuration parameters of the device include the device address, the device name, device type, mechanism number, mechanism name, and the process type.

FI Command Output of the configuration parameters of all defined devices.

BR_DCP1 **(Single Read)**

Note: The DCP1 FI command refers to all devices within this device group. Therefore, any valid device address can be indicated in the command line (see Example DCP1).

Construction of Answer The following table shows the general construction of the answer of the FI command DCP1 . The answer consists of a maximum of n=512 lines (n=16 devices x 32 mechanisms = 512), each with 7 lines.

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

Note: If no active machine parameter record exists in the device, then columns [1...7] of the respective device are not applicable.

Value Range/Meaning of Columns

1 =	Device address	[00...15]
2 =	Device name.	[max. 32 ASCII characters]
3 =	Device type:	[MTCNC, MTC200-P, MTC200-R, MTVNC, MTRA-P, MTRA-R]
4 =	Mechanism number	[0...31]
5 =	Mechanism name	[max. 28 ASCII characters]
6 =	Process type	[1= internal, 2 = external process]
7 =	Process type	[1 = CNC process, 2 = SPS process]

Example DCP1

Read the device configuration parameters of all defined devices.

Assumption: Three device have been defined

- Device address 00 (MTCNC)
- Device address 01 (MTC200-P) and
- Device address 02 (MTC200-P)

FI Command		00_BR_DCP1
Line	**Column**	**Answer**
1	1	00
	2	Rotary transfer machine
	3	MTCNC
	4	1
	5	Master
	6	1
	7	2
2	1	01
	2	0
	3	MTC200-P
	4	0
	5	Milling machine 01
	6	1
	7	1
3	1	02
	2	0
	3	MTC200-P
	4	1
	5	Milling machine 02
	6	1
	7	1

Reference to Literature

Additional information regarding process parameters and their functions as well as value ranges are located in the Rexroth Indramat documentation:

"CNC/SPS Interface Description xxVRS Application Description, chapter "External Mechanisms"
DOK-MTC200-SPS*GWY*Vxx-AW0x-EN.

FI Command

Output of the configuration settings of the selected device.

BR_DCP2 **(Single Read)**

Construction of Answer

The following table shows the general construction of the answer of the FI command DCP2 . The answer consists of a line with 7 columns.

Line 1	Column 1	...	Column 7

Note: If no active machine parameter record exists in the device, then columns [1...7] of the respective device are not applicable.

Value Range/Meaning of Columns

1 =	Device address	[00...15]
2 =	Device name.	[max. 32 ASCII characters]
3 =	Device type:	[MTCNC, MTC200-P, MTC200-R, MTVNC, MTRA-P, MTRA-R]
4 =	Mechanism number	[0...31]
5 =	Mechanism name	[max. 28 ASCII characters]
6 =	Process type	[1= internal, 2 = external process]
7 =	Process type	[1 = CNC process, 2 = SPS process]

Example DCP2

Read the device configuration parameter of the selected device (device address 01).

<u>Assumption:</u> 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

Additional information regarding process parameters and their functions as well as value ranges are located in the Rexroth Indramat documentation:

"CNC/SPS Interface Description xxVRS Application Description,
chapter "External Mechanisms"
DOK-MTC200-SPS*GWY*Vxx-AW0x-EN

Long Identification of the SPS Data Record: DIS

MTAX Device Group

Name **DIS** **D**ata **I**dentification **S**tring

Explanation Reads the long ID (directory entries) of the SPS program. Included in the directory entries are the number of the entry in the directory, the name, length and date and time of creation and/or details of the last time the respective data record was changed.

FI Command **CR_DIS2** **(Single Read)**

Construction of Answer The following table shows the general construction of the answer of the FI command DIS2 . The answer consists of a line with five columns.

Line 1	Column 1	...	Column 5

Value Range/Meaning of Columns

1 =	Number in SPS directory	[01...99]
2 =	Name of the SPS program	[max. 8 ASCII characters]
3 =	Length of the SPS program	[bytes]
4 =	Date of creation/last change to SPS program	[DD.MM.YY]
5 =	Date of creation/last change to SPS program	[HH:MM:SS]

Note: If there is no valid SPS program in the selected device, then additional information is provided in the "FI_ERROR_CLASS_NACK" (see chapter "Error Codes", General Error Result Line).

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

- there is a valid SPS program in the selected device.

FI Command		00_CR_DIS2
Line	**Column**	**Answer**
1	1	01
	2	KEY1
	3	20018
	4	10.05.99
	5	12:42:00

Reference to Literature You can find more details regarding the SPS Programming System in the Rexroth Indramat documentation:

"SPS Programming Instructions xxVRS" Application Description” DOK-CONTRL-SPS*PRO*Vxx-AW0x-EN

Distance to Go of Axis Movement: DTG

MTAX Device Group

Name **DTG** **D**istance **T**o **G**o

Explanation The distance to go of the movement of a selected axis is output. The FI command "DTG2" returns the distance to go of an axis, related to the physical axis number.

FI Command

CR_DTG2_(1)_(2) **(Single Read)**

CC_DTG2_(1)_(2) **(Cyclic Read)**

CB_DTG2_(1)_(2) **(Break Cyclic Read)**

(1) = Physical axis number [1...16]

(2) = Coordinate system [1 = Machine coordinates
2 = Program coordinates]

Construction of Answer The following table shows the general construction of the answer of the FI command DTG2 . One line with three columns for the name of the axis, the distance to go and the unit is outputted in accordance to the settings of the process parameters.

Line 1	Column 1	...	Column 3

Value Range/Meaning of Columns

1 = Axis name [acc. to settings of axis parameter]

2 = Distance to go [acc. to settings of axis parameter]

3 = Unit [acc. to settings of axis process parameter] mm, inch]

Note: If the selected axis is not defined then additional information is provided in the "FI_ERROR_CLASS_NACK" (see chapter 8.1 General Error Result Line).

Example DTG2 Read the distance to go of the movement of the Z axis (physical axis number = 3) in machine coordinates at the device address 00.

FI Command	00_CR_DTG2_3_1		
Answer			
Line	Column 1	Column 2	Column 3
1	Z	-5.9897	mm

Reference to Literature Additional 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-EN

Device Type and Accompanying Components: DTY

MTAX Device Group

Name **DTY** **D**evice **TY**pe

Explanation The device type as well as the accompanying components of the selected device addresses are outputted.

FI Command **CR_DTY1** **(Single Read)**

Construction of Answer The following table shows the general construction of the answer of the FI command DTY1. A line with three columns for the device type is outputted as well as the names of the first device component and the name of the second device component.

Line 1	Column 1	...	Column 3

Value Range/Meaning of Columns

1 =	Device type:	(see chapter 6.1 Elements of the FI Command, Identifier)
2 =	Component type1	IND_DEV.INI entry: Componenttype1=
3 =	Component type 2	IND_DEV.INI entry: Componenttype2=

Example DTY1 Output the device type and the accompanying components of device address 00.

FI Command	00_CR_DTY1		
Answer			
Line	Column 1	Column 2	Column 3
1	MTA200-P	MTS-P	MTC-P

Module Assignment of a Process: MAP

MTAX Device Group

Name **MAP** **M**odule **A**ssign of **P**rocess

Explanation The module to which a particular process is assigned is read out from the "Moduldef.ini'" file. This data is located in the [LW]:\MT-CNC\CONFIG directory and contains all module configuration data. The process data is located in three sections:

[DeviceAddrX\ModulY\Process]

whereby "X" stands for the device addressed and "Y" for the configuration of the module numbers.

FI Command Determine the module to which the process belongs. Information is read out from the module configuration of the MTAX device group.

BR_MAP1_(1) **(Single Read)**

BC_MAP1_(1) **(Cyclic Read)**

BB_MAP1_(1) **(Break Cyclic Read)**

1 = Mechanism number [0]

Construction of Answer The following table shows the general construction of the answer of the FI command MAP1. One line with one column is outputted for module number that has been determined.

Value Range of the Column 1 = module number [0...99]

Example MAP1 Read the module number that is assigned to the CNC process number 0 from the module configuration.

Assumption:

The module that is assigned to the CNC process 0 has module number 5.

FI Command	03_BR_MAP1_0
Answer	
Line	**Column 1**
1	5

Reference to Literature Additional information regarding module configuration and the construction of the "Moduldef.ini" file can be located in the following Rexroth Indramat documentation:

"Diagnostics and Message System for HMI System ProVi", chapter "Configure Moduldef.ini", DOK-MTC200-DIAG*PROVI*-AW0x-EN

Read Reference Name of a SPS Variable : MAR

MTAX Device Group

Name **MAR** **M**ap **A**bsolute PCL-**R**eference

Explanation The absolute reference name of a symbolic SPS variable is read out.

FI Command Reads the absolute SPS reference name of a SPS variable.

BR_MAR_(1) **(Single Read)**

(1) = Identifier of the SPS variable

Example MAR Read the absolute reference name of the SPS variable with the identifier “abref" at device address 00.

<u>Assumption:</u>

- the SPS variable with the identifier "abref" is of the type "INTEGER"

FI Command	**00_BR_MAR_abref**
Answer	
Line	**Column 1**
1	%M100.0

Device Data of the Module Configuration: MCD

MTAX Device Group

Name
MCD **M**odule **C**onfiguration: **D**evice Information

Explanation
All device data of the module configuration are read-out from the "Moduldef.ini" file that is stored in the "[LW]:\MT-CNC\CONFIG" directory. The device data are in the sections [DeviceAddrX], whereby "X" stands for the configured device addresses.

FI Command
Read-out of device data within the module configuration of the MTAX device groups.

BR_MCD1 **(Single Read)**

BC_MCD1 **(Cyclic Read)**

BB_MCD1 **(Break Cyclic Read)**

Note: The MCD1 FI command refers to all devices within the MTAX device group. Therefore, any valid device address can be indicated in the command line (see Example MCD1).

Construction of Answer
The following table shows the general construction of the answer of the FI command MCD1. The number of lines depends on the number of configured devices. Every line consists of four columns for the device address as well as SPS-FB (FB = function component) names for the provision of setup diagnostics, warning messages and start requirements.

Line 1	Column 1	...	Column 4

Value Range of the Columns

1 = device address	[0...15]
2 = SPS-FB name for the setup diagnostics	[max. 9 ASCII characters]
3 = SPS-FB name for the warning messages	[max. 9 ASCII characters]
4 = SPS-FB name for the start requirements	[max. 9 ASCII characters]

Example MCD1
Read all device data of the module configuration

Assumption: The following devices have been configured in the MTAX device group:

- Device address 01 (MTC200-P)
- Device address 03 (MT-CNC)

FI Command	03_BR_MCD1			
Answer				
Line	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	01	PVSetup_1	PVWarn_1	PVStart_1
2	03	PVSetup_3	PVWarn_3	PVStart_3

Reference to Literature
Additional information regarding module configuration and the construction of the "Moduldef.ini" file can be located in the following Rexroth Indramat documentation:

"Diagnostics and Message System for HMI System ProVi", chapter "Configure Moduldef.ini", DOK-MTC200-DIAG*PROVI*-AW0x-EN

Device Data of the Module Configuration: MCM

MTAX Device Group

Name **MCM** **M**odule **C**onfiguration: **M**odule Information

Explanation All module data of a particular device is read out from the "Moduldef.ini'" file. This data is located in the "[LW]:\MT-CNC\CONFIG" directory and contains all module configuration data. The module data is located in sections [DeviceAddrX\ModulY], whereby "X" stands for the device addressed and "Y" for the configured module numbers.

FI Command Read-out of module data from the module configuration with respect to a device from the MTAX device group.

BR_MCM1 **(Single Read)**

BC_MCM1 **(Cyclic Read)**

BB_MCM1 **(Break Cyclic Read)**

Construction of Answer The following table shows the general construction of the answer of the FI command MCM1 . 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 SPS-FB names for general module errors and module messages.

Line 1	Column 1	...	Column 4

Value Range of the Columns

1 = Module number	[0...99]
2 = Module name	[max. 28 ASCII characters]
3 = SPS-FB name for general module errors	[max. 9 ASCII characters]
4 = SPS-FB name for module messages	[max. 9 ASCII characters]

Example MCM1 Read the module data of device 03 from the module configuration:

Assumption: The following modules have been defined:

- Module number 5
- Module number 7

FI Command	**03_BR_MCM1**			
Answer				
Line	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	5	Module 5 – Milling	PVError_5	PVMsg_5
2	7	Module 7 - Drilling	PVError_7	PVMsg_7

Reference to Literature Additional information regarding module configuration and the construction of the "Moduldef.ini" file can be located in the following Rexroth Indramat documentation:

Diagnostics and Message System for HMI System ProVi, chapter "Configure Moduldef.ini", DOK-MTC200-DIAG*PROVI*-AW0x-EN

Process Data of the Module Configuration: MCP

MTAX Device Group

Name **MCP** **M**odule **C**onfiguration: **P**rocess Information

Explanation All process data of a particular module is read out from the "Moduldef.ini"' file. This data is located in the "[LW]:\MT-CNC\CONFIG" directory and contains all module configuration data. The module data is located in sections [DeviceAddrX\ModulY\Process], whereby "X" stands for the device addressed and "Y" for the selected module numbers.

BR_MCP1_(1) **(Single Read)**

BC_MCP1_(1) **(Cyclic Read)**

BB_MCP1_(1) **(Break Cyclic Read)**

1 = module number [0...99]

Construction of Answer The answer of the FI command MCP1 consists of one of up to a maximum number of n=32 lines with 1 column for the number of the CNC process or of the external mechanism.

Value Range of the Column 1 = Mechanism number [0]

Example MCP1 Read the CNC process number of module 5 of device 03 of the module configuration.

Assumption: The following CNC processes have been defined:

- CNC process number 0

FI Command	**03_BR_MCS1_5**
Answer	
Line	**Column 1**
1	0

Reference to Literature Additional information regarding module configuration and the construction of the "Moduldef.ini" file can be located in the following Rexroth Indramat documentation:

"Diagnostics and Message System for HMI System ProVi", chapter "Configure Moduldef.ini", DOK-MTC200-DIAG*PROVI*-AW0x-EN

SFC Data of the Module Configuration: MCS

MTAX Device Group

Name **MCS** **M**odule **C**onfiguration: **S**FC Information

Explanation All SFC data of a particular module is read out from the "Moduldef.ini"' file. This data is located in the [LW]:\MT-CNC\CONFIG directory and 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 of the SFC data with respect to the module of a device from the module configuration of the MTAX device group.

BR_MCS1_(1) **(Single Read)**

BC_MCS1_(1) **(Cyclic Read)**

BB_MCS1_(1) **(Break Cyclic Read)**

1 = module number [0...99]

Construction of Answer The number of lines depends on the number of configured Indrastep Step Chains of a device. Each line contains a column for the name of the Indrastep Step Chains.

Value Range of the Column 1 = Name of the Indrastep Step Chain [Format W.X.Y.Z]

Format W.X.Y.Z	Value range
W	max. 9 ASCII characters
X	max. 9 ASCII characters OPTIONAL !
Y	max. 9 ASCII characters OPTIONAL !
Z	max. 9 ASCII characters OPTIONAL !

Example MCS1 Read the name of the Indrastep Step Chain of module 5 from device 03 of the module configuration.

<u>Assumption:</u>

The following Indrastep Step Chains have been defined:

- ISFB_1
- FB_US.ISFB_3
- FB_US.ISFB_3.SW1.ABBA
- FB_US.ISFB_3.SW1.ABBA

FI Command	03_BR_MCS1_5
Answer	
Line	**Column 1**
1	ISFB_1
2	FB_US.ISFB_3
3	FB_US.ISFB_3.SW1
4	FB_US.ISFB_3.SW1.ABBA

Reference to Literature Additional information regarding module configuration and the construction of the "Moduldef.ini" file can be located in the following Rexroth Indramat documentation:

"Diagnostics and Message System for HMI System ProVi", chapter "Configure Moduldef.ini", DOK-MTC200-DIAG*PROVI*-AW0x-EN

Formatted Input / Output of SPS Variables: PVF

MTAX Device Group

Name **PVF** **P**LC **V**ariable **F**ormatted

Explanation Formatted reading and writing of SPS variables, arrays and structures.

FI Command Read SPS variables.

CR_PVF1_(1) **(Single Read)**

CC_ PVF1_(1) **(Cyclic Read)**

CB_ PVF1_(1) **(Break Cyclic Read)**

(1) = Identifier of the SPS variable [acc. to declaration part of SPS]

Construction of Answer One line with one column is outputted for simple variables. For array and structure variables, one line per element is outputted, depending on the number of elements.

Line 1...n:	Column 1

n = number of elements.

Note: Only defined SPS variables can be read and written. Addressing a non-declared variable results in an error message. A SPS variable can only be read when its data length does not exceed 240 (see also chapter 4.1, Guidelines).

Value Ranges ANSI / ASCII

The value range of the answer depends on the data type of the variables read. The following table informs you of the range in which the results string is to be expected when reading out a simple variable and into which C-data type this string can be converted without loss of information:

Data Type	Value range	Can be converted to C-data type
BOOL	[0;1]	unsigned char
SINT	[-128...127]	char
INT	[-32768...32767]	short
DINT	[2147483648...2147483647]	long
USINT	[0...255]	unsigned char
UINT	[0...65535]	unsigned short
UDINT	[0...4294967295]	unsigned long
BYTE	[0x00...0xFF]	unsigned char
WORD	[0x0000....0xFFFF]	unsigned short
DWORD	[0x00000000...0xFFFFFFFF]	unsigned long
TIME	[0...4294967295]	unsigned long (msec)
CHAR	[$00...$20,!...~,$7F...$FF]	char
STRING	<String> whereby <String> string is with a maximal of as many characters as are defined for the string in the SPS	Char[xx+1]] +1 e.g. room for the zero byte
REAL	[-3.402823567E+38...3.402823567E+38]	Float

Note: An empty string can be recognized by simple double-inverted commas: “

All simple 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 answer depends on the data type of the variables read. The following table informs you of the value range in which to expect the binary value of a simple variable and how many bytes are included in the binary byte sequence:

Data Type	Value range	Length (bytes)
BOOL	[00_H...01_H]	1
SINT	[80_H...$7F_H$] i.e. –128...127	1
INT	[8000_H (-32768)...$7FFF_H$ (32767)]	2
DINT	[80000000_H (-2147483648)... $7FFFFFFF_H$ (2147483647)]	4
USINT	[00_H (0)...FF_H (255)]	1
UINT	[00_H (0)...$FFFF_H$ (65535)]	2
UDINT	[0...4294967295]	4
BYTE	[0x00...0xFF]	1
WORD	[0x0000....0xFFFF]	2
DWORD	[0x00000000...0xFFFFFFFF]	4
TIME	[0...4294967295]	4
CHAR	[$00...$20,!...~,$7F...$FF]	1
STRING	<String> whereby <String> string is with a maximal of as many characters as are defined for the string in the SPS	XX+1
REAL	[-3.402823567E+38...3.402823567E+38]	4

Note: Binary array and structure elements are joined together without any spaces between (1-byte alignment).

Example 1 PVF1

Read the value of the SPS variables "STK_TXT" in ASCII format from device address 00.

Assumption:

The "STK_TXT" variable is declared as a string in the SPS program.

FI Command		00_CR_PVT1_STK_TXT
Line	**Column**	**Answer**
1	1	Repeat counter

Example 2 PVF1

Read the value of the SPS array "BEG_END" in ANSI format from device address 00.

Assumption:

The "BEG_END" variable is declared as BYTE with 2 elements in the SPS program.

FI Command		00_CR_PVT1_BEG_END/3
Line	**Column**	**Answer**
1	1	0x00
2		0x1F

Example 3 PVF1

Read the value of the SPS structure "MSTRCT" in ASCII format from device address 00.

Assumption:

The "MSTRCT" variable is declared as a structure in the SPS program as follows.

TYP STRUCT

T1 BOOL

T2 CHAR

T3 STRING[16]

T4 TIME

END

FI Command		00_CR_PVT1_MSTRCT/1
Line	**Column**	**Answer**
1	1	0
2		A
3		ROBOT AXIS X
4		2000

FI Command

Write SPS variables.

CW_PVF1_(1) **(Single Read)**

(1) = Identifier of the SPS variable [acc. to declaration part of SPS]

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“.

Construction of Answer

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

(P_ACK) = **P**ositive **ACK**nowledge Data element has been set

Value Range of the value to be written in ANSI / ASCII Format

The value ranges agree for the most part with the result-value ranges ANSI / ASCII during read access. For deviations to this, please refer to the following note:

Note: Strings are bracketed by two simple inverted commas ’ ’ . e.g. ’drill’

Special characters can be marked in accordance to DIN-1131 by a $ sign.

There are:

- $’’
- $$ $
- $R \r (Carriage Return)
- L n (Linefeed)
- $P \f (Formfeed)
- $T \t <Tab>
- $xx xx refers to a character written as a hexadecimal value. e.g. $20 (space)

Array and structure elements are separated by a space.

Value Range of the Value to be written in Binary Format

The value ranges agree with the binary result-value range during read access. For deviations to this, please refer to the following note:

Example 4 PVF1

Write the value of the SPS variable "STK_TXT" to device address 00. The value is output in ASCII format.

Assumption:

The "STK_TXT" variable is declared as a string in the SPS program.

FI Command		00_CW_PVT1_STK_TXT
Line	Column	Answer
1	1	(P_ACK)

Value to be written

Value of data element ’item counter’

Data code 1

Example 5 PVF1

Write the value of the SPS array "BEG_END" at device address 00. The value is output in ANSI format.

Assumption:

The "BEG_END" variable is declared as BYTE with 2 elements in the SPS program.

FI Command		00_CW_PVT1_BEG_END
Line	Column	Answer
1	1	(P_ACK)

Value to be written

Value of data element 0x20 0x3f

Data code 3

Example 6 PVF1

Write the value of element T3 of the SPS 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 SPS program as follows.

TYP STRUCT

T1 BOOL

T2 CHAR

T3 STRING[16]

T4 TIME

END

FI Command		00_CW_PVT1_MSTRCT.T3
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

Example 7 PVF1

Write the value of the SPS 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 SPS program as follows.

TYP STRUCT

T1 BOOL

T2 CHAR

T3 STRING[16]

T4 TIME

END

For the exchange of binary data in a C program, the following 'C' data type can be used:

[illegible]

FI Command		00_CW_PVT1_MSTRCT
Line	**Column**	**Answer**
1	1	(P_ACK)

Value to be written Address of the C structure.

Value of data element	&mstrct
Data code	2

Reading and Writing SPS Variables: PVS

MTAX Device Group

Name **PVS** **P**LC-**V**ariable **S**ingle

Explanation The following types of SPS variable can be read or written:

- BOOL, BYTE, SINT, USINT, CHAR, WORD, INT, UINT, STRING, DWORD, DINT, UDINT, TIME, REAL as well as imported structures and arrays.

FI Command Reading SPS variables.

CR_PVS_(1)	**(Single Read)**
CC_PVS_(1)	**(Cyclic Read)**
CB_PVS_(1)	**(Break Cyclic Read)**

(1) = Identifier of the SPS variable

Note: Addressing a non-declared variable results in an error message. The length of the data must not exceed 240 bytes (see also chapter 4.1, Guidelines).

Example PVS Read the value of the SPS variable with identifier "IB_EXT24" at device address 00 in CNC process 0.

Assumption:

- the SPS variable with the identifier "IB_EXT24" is of the type "BOOL"

FI Command	**00_CR_PVS_IB_EXT24**
Answer	
Line	**Column 1**
1	1

FI Command Writing a SPS Variable.

CW_PVS_(1)	**(Single Write)**

(1) = Identifier of the SPS variable

Value to be written

SPS variable	[Format acc. to the type in the SPS program]

Note: Only defined SPS variables can be written. Addressing a non-declared variable results in an error message. The data length must not exceed 240 bytes. (refer also to chapter 4.1, Guidelines). The value to be written is passed to the "acValue" parameter in the "DataTransfer" routine.

Construction of Answer The return value of the "DataTransfer" routine is [0] when the write procedure has been successfully completed. In case of an error, more information can requested by the routine "ReadGroupItem" in the form of a general error result line (refer here to chapter 8, Error Codes).

Example PVS Write the value 1 in the SPS variable with the identifier "IB_EXT24" at device address 00.

Assumption:

- the SPS variable with the identifier "IB_EXT24" is of the type "BOOL"

FI Command	**00_CW_PVS_IB_EXT24**
Value to be written	1

Reading the SPS Variable Declaration: PVT

MTAX Device Group

Name **PVT** **P**LC **V**ariable **T**ype

Explanation A SPS 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 SPS Variables.

FI Command Reading-out the SPS Variable Type.

BR _PVT_(1) **(Single Read)**

(1) = Identifier of the SPS variable [acc. to declaration part of SPS]

Construction of Answer One line with 2 columns is outputted for each element of the variables.

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

n = number of elements.

Value Range/Meaning of Columns

(1) = Identifier of the SPS variable [acc. to declaration part of SPS]

2 = Type [see value range PVF]

Examples:

Reading a simple variable Assumption:

The variable TEST of type WORD is declared in the SPS (device 0).

FI Command	**00_BR_PVT_TEST**	
Answer		
Line	**Column 1 (Name)**	**Column 2 (Name)**
1	TEST	WORD

Reading a Structure Assumption:

The variable TEST1 of type STRUCKT is declared in the SPS (device 0).

STRUCT

E1 BOOL

E2 INT

E3 SINT

END

FI Command	00_BR_PVT_TEST1	
Answer		
Line	**Column 1**	**Column 2**
1	TEST1.E1	BOOL
2	TEST1.E2	INT
3	TEST1.E3	SINT

Reading an Array

Assumption:

The variable TEST2 of type ARRAY is declared in the SPS (device 0).

ARRAY [
0 .. 3
] OF BOOL
....

FI Command	00_BR_PVT_TEST2	
Answer		
Line	**Column 1**	**Column 2**
1	TEST2[0]	BOOL
2	TEST2[1]	BOOL
3	TEST2[2]	BOOL
4	TEST2[3]	BOOL

Reading an Array of a Structure

Assumption:

The variable TEST3 of type ARRAY is declared in the SPS (device 0).

ARRAY [
0 .. 1
] OF STRUCT1,
whereby STRUCT1 with
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

Comment:

The data types are outputted according to IEC1131.

See also command PVF.

Software Installation Data: SID

MTAX Device Group

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

Explanation Information is returned regarding the installation. This information includes the installation path, the software version being used and service pack and release information.

FI Command Reading-in the installation data.

BR_SID1	**(Single Read)**
BC_SID1	**(Cyclic Read)**

Construction of Answer One line with 8 columns is outputted for the additional text.

Line 1	Column 1	...	Column 8

Meaning of the Column

1 = Base directory	[EXE files of the DOS-BOF]
2 = FI installation directory	[FI directory]
3 = Data directory	[in accordance to DOS-BOF]
4 = GBO version	[from INDRAMAT.ini]
5 = IF-DLL mode	[from INDRAMAT.ini]
6 = IF version	[from INDRAMAT.ini from DLL mode 400]
7 = Service pack info	[from INDRAMAT.ini from DLL mode 420]
8 = Release info	[from INDRAMAT.ini from DLL mode 420]

Example SID1 Return the information on the current installation.

FI Command		00_BR_SID1
Line	**Column**	**Answer**
1	1	D:\MT-CNC
	2	C:\MT-CNC\IND_DRV
	3	D:\MT-CNC\ANLAGE00
	4	005-18V05
	5	04.20
	6	04V03
	7	--
	8	--

SPS Long Identification: SLI

MTAX Device Group

Name **SLI** **SPS L**ong **I**dentification

Explanation Returns the single data from the SPS long identification.

FI Command Read SPS long identification.

BR_SLI **(Single Read)**

Construction of Answer One line with 15 columns is outputted for the returned values.

Line 1	Column 1	...	Column 15

Value Range/Meaning of the Columns

1 = Device address	[00...15]
2 = Program number	[01...99]
3 = Project name	[max. 8 ASCII characters]
4 = Program name	[max. 8 ASCII characters]
5 = User name	[acc. to password entry]
6 = Program length	[bytes]
7 = Compilation time	[LONG] (coded in long value)
8 = Compilation date	[8 ASCII characters]
9 = Compilation time	[8 ASCII characters]
10 = Download time	[LONG] (coded in long value)
11 = Download date	[8 ASCII characters]1
12 = Download time	[8 ASCII characters]
13 = Version of SPS long identification	[LONG]
14 = RUN Flags	[HEX value]
15 = Compiler info	[LONG]

Example SLI Read the single data from the SPS long identification.

FI Command		00_BR_SLI
Line	**Column**	**Answer**
1	1	02
	2	01
	3	--
	4	MOT12
	5	TEST
	6	17672
	7	630163960
	8	15.12.99
	9	17:15:48
	10	630163961
	11	15.12.99
	12	17:15:50
	13	2
	14	0x0000
	15	13

7.6 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 as well as possible addresses are listed:

Group	Accompanying Types	Address
MSYX	SYNAX200-P, SYNAX200-R	[00...15]

Note: The parameters of the MSYX device group are gathered together in chapter 6.2, Data Tables.

Determining the Current (Actual) System Error: ASE

MSYX Device Group

Name **ASE** **A**ctual **S**ystem **E**rror

Explanation The current system error is read out, whereby the answer 0x0000 indicates that the Synax device is functioning correctly.

FI Command **CR_ASE** **(Single Read)**

Construction of Answer The following table shows the general construction of the answer of the FI command ASE. In line 1, column 4, the number of the drive is outputted that reports the current system error. Not all current system errors can be directly assigned to a drive. In this case, the single result "Drive No." is set to 0x0000.

Line 1	Column 1	...	Column 4

Value Range/Meaning of Columns

1 = 0x0000

2 = 0x0000

3 = Current system error

4 = Drive No.

Example ASE Read-out of the current system error returns LWL-Ring interrupted.

FI Command		00_CR_ASE
Line	**Column**	**Answer**
1	1	0x0000
	2	0x0000
	3	0x8009
	4	0x0000

Reference to Literature You can find more details regarding the communication phases in the Indramat documentation:

"DIAX04 Drive with Servo Functions", General Instructions on Putting into Operation,
DOK-DIAX04-SSE-xxVRS**-FK0x-EN

Clearing a Current System Error. CSE

MSYX Device Group

Name **CSE** **C**lear **S**ystem **E**rror

Explanation An error reported by the Synax device is again cleared.

FI Command **CW_CSE** **(Single Write)**

Value to be written: The contents of the value parameter is not evaluated.

Construction of Answer The following table shows the general construction of the answer of the FI command CSE. In line 1, column 4, the number of the drive is outputted that reports the current system error. Not all current system errors can be directly assigned to a drive. In this case, the single result "Drive No." is set to 0x0000.

Line 1	Column 1	...	Column 4

Value Range/Meaning of Columns

1 = 0x0000

2 = 0x0000

3 = Current system error

4 = Drive No.

Example CSE The current system error is cleared.

FI Command		00_CW_CSE
Line	**Column**	**Answer**
1	1	0x0000
	2	0x0000
	3	0x0000
	4	0x0000

Reference to Literature You can find more details regarding the current system error in the Indramat documentation:

"SERCANS /SERCVME SERCOS Interface Assemblies with Universal µP Interface or VMEbus", Application description, System Structure and Axis Structure.

Device Type and Accompanying Components: DTY

MSYX Device Group

Name **DTY** **D**evice **TY**pe

Explanation The device type as well as the accompanying components of the selected device addresses are outputted.

FI Command

CR_DTY1 **(Single Read)**

Construction of Answer The following table shows the general construction of the answer of the FI command DTY1. A line with three columns for the device type is outputted as well as the names of the first device component and the name of the second device component.

Line 1	Column 1	...	Column 3

Value Range/Meaning of Columns

1 =	Device type:	(see chapter 6.1 “Elements of the FI Command”, Identifier)
2 =	Component type1	IND_DEV.INI entry: Componenttype1=
3 =	Component type 2	IND_DEV.INI entry: Component-type2=

Example DTY1 Output the device type and the accompanying components of device address 00.

FI Command	00_CR_DTY1		
Answer			
Line	Column 1	Column 2	Column 3
1	SYNAX200-R	NONE	PPC-R

Software Installation Data: SID

MSYX Device Group

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

Explanation Information is returned regarding the installation. This information includes the installation path, the software version being used and service pack and release information.

FI Command Reading-in the installation data.

BR_SID1 **(Single Read)**

BC_SID1 **(Cyclic Read)**

Construction of Answer One line with 8 columns is outputted for the additional text.

Line 1	Column 1	...	Column 8

Meaning of the Column

1 = Base directory	[EXE files of the DOS-BOF]
2 = FI installation directory	[FI directory]
3 = Data directory	[in accordance to DOS-BOF]
4 = GBO version	[from INDRAMAT.ini]
5 = IF-DLL mode	[from INDRAMAT.ini]
6 = IF version	[from INDRAMAT.ini from DLL mode 400]
7 = Service pack info	[from INDRAMAT.ini from DLL mode 420]
8 = Release info	[from INDRAMAT.ini from DLL mode 420]

Example SID1 Return the information on the current installation.

FI Command		**00_BR_SID1**
Line	**Column**	**Answer**
1	1	D:\MT-CNC
	2	C:\MT-CNC\IND_DRV
	3	D:\MT-CNC\ANLAGE00
	4	005-18V05
	5	04.20
	6	04V03
	7	--
	8	--

SERCOS Parameters: SPA

MSYX Device Group

Name **SPA** **S**ERCOS **Pa**rameter

Explanation A SERCOS drive parameter is outputted or written. Each parameter consists of 7 elements, whereby any combination of elements can be selected by element coding.

FI Command

BR_SPA1_(1)_(2)_(3)	**(Single Read)**
BC_SPA1_(1)_(2)_(3)	**(Cyclic Read)**
BB_SPA1_(1)_(2)_(3)	**(Break Cyclic Read)**
BW_SPA1_(1)_(2)_(3)	**(Single Write)**
(1) = Drive address	[0...254]
(2) = Parameter No.	in format: X-Y-ZZZZ
(3) = Element coding	[standard or advanced format]

Parameter No.

Format X-Y-ZZZZ	Value range
X	A = control parameter specific to the axis C = general control parameter S = standard data P = product data
Y	[0...15] = parameter record
Z	[0...4095] = datablock no.

Element Coding Element coding in standard format allows individual elements, such as, e.g. operating date, to be requested. If several elements are to be read out in one request, then the element coding can be additive in advanced format, e.g. operating date (0x40) and unit (0x08) produces additive (0x48) → 48

The advanced format 0x80 has priority over 0x40.

Element	Standard Format	Advanced Format	Format	Example
Data status	S	01H	Hexadecimal word	0x0000
Name	N	02H	String	NC cycle time (TNcyc)
Attribute	A	04H	Hexadecimal double word	0x60110001
Unit	U	08H	String	µs
Min. input value	L	10H	Decimal word	2000
Max. input value	H	20H	Decimal word	20000
Operating date	D	40H	(see Displaying the Operating Date)	
Operating date, when no list		80_H		

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 following spaces are allowed.

Hexadecimal

Hexadecimal values are displayed by "0x...", e.g. 0x80. up to eight positions are allowed. Leading and following spaces are allowed. Leading additional zeros or plus and minus signs are not allowed.

Binary (max. 32 characters)

Leading or following 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 shown:

Format X-Y-ZZZZ	Value range
X	S = standard data P = product data
Y	[0...7] = parameter record
Z	[0...4095] = datablock no.

(see example SPA1/write).

Lists of Variable Length

Lists always begin with two decimal numbered for the actual length and maximal 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 difference 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 maximal length of the list (string) is also transmitted.

Example

Parameter S-0-0030, Operating Date
Standard Format: "DKC2.1-SSE-01V09"
Advanced Format: "16\n16\nDKC2.1-SSE-01V09"

Reference to Literature

Additional information regarding the function of the standard and product-specific SERCOS parameters (S and P) is contained in the Indramat documentation:

"DIAX04 Drive with Servo Functions", Appendix A Description of Parameters,
DOK-DIAX04-SSE-xxVRS**-FK0x-EN

Construction of Answer

The following table shows the general construction of the answer of the FI command SPA1 . Line 1 is outputted both when reading and when writing. Additional lines are only outputted when reading depending on the element coding.

Note: If the element coding has been requested in standard format then the first line is not applicable.

Note: Line 1 is a status line that either contains SERCOS 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].

The number of the drive that reports the SERCOS error is outputted in the second line.

Line	Column 1	Column 2	Column 3	Column 4
1	<SERCOS error>	<Drive no. SERCOS error>	0x0000	0x0000
2	Read: 1. Element corresponding to the element coding.			
...	...			
n	Read: -1...n: Element corresponding to the element coding.			

Example SPA1 / read

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

FI Command	00_BR_SPA1_3_S-0-0003_48			
Answer				
Line	**Column 1**	**Column 2**	**Column 3**	**Column 4**
1	0x0000	0x0000	0x0000	0x0000
2	µs			
3	2000			

Example SPA1 / write

Write the ID number P-0-0037 in the parameter S-0-0305 of the 3rd drive (element coding 0x40).

Technical Background:

- Real-time status bit 1 is to be assigned the trigger status word of the oscilloscope function of a DIAX04 drive.

FI Command	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

Active SERCOS Phase Switch-Over: SPH

MSYX Device Group

Name **SPH** **S**ERCOS **Ph**ase

Explanation All drives within a SERCOS ring are in the same communication phase. The phase condition 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 [1...32]

Value to be written

Phase [2, 4]

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

Example SPH Write SERCOS Phase

Switch-over of the first axis (write) after phase 4; phase 2 is active.

FI Command	**00_CW_SPH_1** **Value to be written: 4**	
Answer		
Line	**Column 1**	**Column 2**
1	52	1

Note: Switching-over from phase 2 to phase 4 returns as result of column 1 the value [52]. 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

You can find more details regarding the communication phases in the Indramat documentation:

"DIAX04 Drive with Servo Functions", General Instructions on Putting into Operation,
DOK-DIAX04-SSE-xxVRS**-FK0x-EN

8 Error Codes

8.1 General Error Result Line

If the "DataTransfer" routine returns an error code then the requested data are 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.

The following table shows the general construction of the error result line. One line consisting of 5 columns is outputted for the class of error, error code, expanded additional information, error text and additional text

Line 1	Column 1	...	Column 5

Meaning of the Column

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 MAXZEILEN in the "VERSION.DAT" file found. Remedy: Delete "VERSION.DAT" file

Code	Error text	Name and Meaning of Error
207	BOF_MUTEX_ERROR	Error generating a MUTEX object.
208	BOF_FILE_MAPPING_ERROR	Error generating a file mapping.
209	BOF_MEMORY_ERROR	Memory allocation 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	Preset 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 exists.
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 SPS 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 preset time.
238	BOF_SYS_MSG_SET_ERROR	Access to SYS-MSG channel not possible in the preset time. (SYS-Message is issued).
239	BOF_SYS_MSG_GET_ERROR	Access to SYS-MSG channel not possible in the preset time (SYS-Message is fetched).
240	BOF_DATA_TIME_ERROR	A data element in the shared memory area was not released in the preset 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 preset 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 "IfDllMode=" of file "INDRAMAT.INI".
248	BOF_LOG_GRP_COUNT_ERROR	Invalid group error 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 preset 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 preset 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 → NT job processing.
260	BOF_DOS_NT_JOB_STR_ERROR	Invalid DOS → NT command string.
261	BOF_SYS_MSG_RANGE_ERROR	SYS-MSG message number is outside of the valid number range.
262	BOF_DOS_NT_JOB_INFO_ERROR	DOS → NT command information is too long
263	BOF_DOS_NT_SYS_MSG_Q_ERROR	An uneven SYS-MSG message number (acknowledgement) was passed by the DOS → NT command SYSC_xxx.
264	BOF_DOS_NT_FKT_NOT_FOUND_ERROR	DOS → 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 → NT commands in the "BOFINTFC.DAT" file.
266	BOF_DOS_NT_DLL_NOT_FOUND_ERROR	DLL for the DOS → NT commands not found.
267	BOF_DOS_NT_FKT_NOT_IN_DLL_ERROR	DOS → 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, or the entry is invalid in the "IND_DEV.INI" file.
282	BOF_MEMORY_CLASS_ERROR	No valid memory class for DOS → 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, or the entry is invalid in the "IND_DEV.INI" file.
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 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 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 or an invalid CYCLIC entry in the "BOFINTFC.DAT" file.
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 → DOS job channel not possible in the preset time.
304	BOF_NT_DOS_COMMAND_LENGTH_ERROR	NT → DOS command string is too long.
305	BOF_NT_DOS_COMMAND_INFO_LENGTH_ ERROR	NT → DOS command info string is too long.
306	BOF_NT_DOS_TIMEOUT_ERROR	NT → DOS job could not be executed in the preset time.
307	BOF_NT_DOS_FKT_NOT_FOUND_ERROR	An NT → 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.
309	BOF_NT_DOS_DLL_NOT_FOUND_ERROR	DLL for the NT → DOS commands not found.

Code	Error Text	Name and Meaning of Error
310	BOF_NT_DOS_FKT_NOT_IN_DLL_ERROR	NT → DOS processing function not found in the specified DLL.
311	BOF_NT_DOS_JOB_STR_ERROR	Invalid NT → DOS command string.
312	BOF_NT_DOS_JOB_INFO_ERROR	NT → DOS command information is too long
313	BOF_NT_DOS_DPR_TIMEOUT_ERROR	Access to NT-DOS-DPR memory not possible in the preset time.
314	BOF_NT_DOS_NO_COMMAND_ERROR	No NT → DOS command string.
315	BOF_NT_DOS_BOF_INDEX_ERROR	Invalid DOS-BOF INDEX when issuing an NT → 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 preset time.
319	BOF_DEVICE_TYP_GROUP_ERROR	Selected device address does not exist in this device group.
320	BOF_INVALID_PROCESS_NUMBER_ERROR	Invalid CNC 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 were returned on requesting the device type ID.
325	BOF_DEVICE_SPS_IDENT_ERROR	Invalid data were delivered by the interface on requesting the long ID of the SPS MAP file.
326	BOF_INVALID_AXIS_NUMBER_ERROR	Invalid axis number received [1...32].
327	BOF_NO_GBOVERSION_ERROR	There is no "GBOVERSION=" entry, or the entry is invalid in the "INDRAMAT.INI" file.
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_ ERROR	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 administration 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 preset time.

Code	Error Text	Name and Meaning of Error
342	BOF_FI_ERROR_STRING_TO_LONG	String for the general FI-ERROR ANSWER-TELEGRAM (general error result line) is too long.
343	BOF_DOS_MANAGERPROG_NOT_READY_ ERROR	DOS-NT manager program not running.
344	BOF_NT_DOS_ORDER_TO_LONG	NT → 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	Diagnostics text file "STERRxx.YYY" not found.
352	BOF_FILE_STERR_FILE_CLOSE_ERROR	Diagnostics file "STERRxx.YYY" could not be closed.
353	BOF_FILE_STERR_FILE_OPEN_ERROR	Diagnostics text file "STERRxx.YYY" could not be opened.
354	BOF_FILE_STERR_FILE_POSITION_ERROR	File positioning in diagnostics text file "STERRxx.YYY" could not be carried out.
355	BOF_FILE_STERR_FILE_READ_ERROR	Read function of the diagnostics text file "STERRxx.YYY" could not be carried out.
356	BOF_FILE_STERR_FILE_NOT_FOUND_ ERROR	Diagnostics 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_ LONG	Additional information passed for the TASK-THREAD triggering is too long.
360	BOF_TASK_THREAD_TRIGGER_TIMEOUT_ ERROR	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_ ERROR	A result data type that is not valid (e.g. 00_BR_AMM1/2) was requested for an FI-command (BR_...).
363	BOF_FILE_TEXT_FILE_NOT_FOUND_ ERROR	Corresponding TEXTxx.YY file does not exist.
364	BOF_FILE_TIND_FILE_NOT_FOUND_ ERROR	Corresponding 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 of the text file "TEXTxx.YY" could not be carried out.

Code	Error Text	Name and Meaning of Error
370	BOF_DIAGNOSTIC_NUMBER_TO_LARGE_ ERROR	Message number for CNC/SPS 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.
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	Diagnostics text file SYSANW.YY not found.
376	BOF_FILE_SYSANW_FILE_CLOSE_ERROR	Diagnostics text file SYSANW.YY could not be closed.
377	BOF_FILE_SYSANW_FILE_OPEN_ERROR	Diagnostics 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 of the diagnostics 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	Corresponding TXERR.YY file not found.
382	BOF_FILE_TXERI_FILE_NOT_FOUND_ERROR	Corresponding TXERI.YY file not found.
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 does not exist for the "IfDllMode=" set in the "INDRAMAT.INI" file.
383	BOF_FILE_TXERI_FILE_OPEN_ERROR	TXERI.YY could not be opened.
388	BOF_NO_PARAMETER_SET_IN_CONTROL	No valid parameter record in the controls.
389	BOF_ANDRON_COMMANDLINE_ERROR	No valid command line for the ANDRON 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_ ERROR	Access to DOS → NT job channel not possible in the preset time.
395	BOF_PROCESS_NAME_ERROR	A syntax error has been recognised 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_ ERROR	No terminate event found for the registered TASK.
399	BOF_PARENT_WIN_ALREADY_EXIST_ERROR	PARENT-WINDOW name already exists in the task file.

Code	Error Text	Name and Meaning of Error
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_ANDRON_INST_PATH	No ANDRON installation path found.
403	BOF_SYSANW_FILTER_FILE_CREATE_ERROR	Filter file SYSSTW.XX for SYSANW.XX (only SHORT MESSAGES!) could not be created.
404	BOF_FILTER_FILE_DIRECTORY_CREATE_ ERROR	The temporary sub-directory TEMPDATA could not be created for the data files of the small devices.
405	BOF_DELETE_FILE_ERROR	Data file (small devices) can not be deleted.
406	BOF_TXERR_FILTER_FILE_CREATE_ERROR	Filter file TXEST.XX for TXERR.XX (only SHORT MESSAGES!) could not be created.
407	BOF_STERR_FILTER_FILE_CREATE_ERROR	Filter file STESTyy.XX for STERRyy.XX (only SHORT MESSAGES!) could not be created.
408	BOF_TXERR_FILTER_FILE_ NOT_FOUND_ERROR	Filter file TXEST.XX for TXERR.XX (only SHORT MESSAGES!) does not exist in the sub-directory TEMPDATA.
409	BOF_TXERR_FILTER_FILE_OPEN_ERROR	Filter file TXEST.XX for TXERR.XX (only SHORT MESSAGES!) could not be opened in the temporary sub-directory TEMPDATA.
410	BOF_TXEST_INDEX_FILE_CREATE_ERROR	INDEX file TXEST.XX (only SHORT MESSAGES!) 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_ERROR	Filter file SYSSTW.XX for SYSANW.XX (only SHORT MESSAGES!) does not exist in the temporary sub-directory TEMPDATA.
416	BOF_SYSSTW_INDEX_FILE_CREATE_ERROR	Index file SYSSTW.XX (only SHORT MESSAGES!) could not be created.
417	BOF_SYSANW_FILTER_FILE_OPEN_ERROR	Filter file SYSSTW.XX for SYSANW.XX (only SHORT MESSAGES!) could not be opened in the temporary sub-directory TEMPDATA.
418	BOF_STERR_FILTER_FILE_ NOT_FOUND_ERROR	Filter file STESTyy.XX for STERRyy.XX (only SHORT MESSAGES!) does not exist in the temporary sub-directory TEMPDATA.
419	BOF_STESTYY_INDEX_FILE_CREATE_ERROR	The index file for STESTyy.XX (only SHORT MESSAGES!) could not be created.
420	BOF_STERR_FILTER_FILE_OPEN_ERROR	Filter file STESTYY.XX for STERRYY.XX (only SHORT MESSAGES!) could not be opened in the temporary sub-directory TEMPDATA.
421	BOF_WRONG_TELEGRAMM_CODE_ERROR	An incorrect TELEGRAM CODE has been returned to the controls.
422	BOF_TXEST_INDEX_FILE_NOT_FOUND_ ERROR	Index file TXESI.XX could not be found.
423	BOF_TXEST_INDEX_FILE_OPEN_ERROR	Index file TXESI.XX could not be opened.

Code	Error Text	Name and Meaning of Error
424	BOF_TXEST_INDEX_FILE_READ_ERROR	Index file TXESI.XX could not be read.
425	BOF_SYSSTW_INDEX_FILE_NOT_FOUND_ ERROR	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_ ERROR	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 can not be read.
431	BOF_DEVICE_TYPE_VALUE_TO_LARGE	DEVICE-TYPE number is too large.
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 (only SHORT MESSAGES!) could not be created (TXESK.XX).
434	BOF_TXEST_KENNUNG_FILE_OPEN_ERROR	The ID FILE for TXEST.XX (only SHORT MESSAGES!) could not be opened (TXESK.XX).
435	BOF_TXEST_KENNUNG_FILE_READ_ERROR	The ID FILE for TXEST.XX (only SHORT MESSAGES!) could not be read (TXESK.XX).
436	BOF_SYSSTW_KENNUNG_FILE_CREATE_ ERROR	The ID FILE for SYSSTW.XX (only SHORT MESSAGES!) could not be created (SYSSKW.XX).
437	BOF_SYSSTW_KENNUNG_FILE_OPEN_ERROR	The ID FILE for SYSSTW.XX (only SHORT MESSAGES!) could not be opened (SYSSKW.XX).
438	BOF_SYSSTW_KENNUNG_FILE_READ_ERROR	The ID FILE for SYSSTW.XX (only SHORT MESSAGES!) could not be read (SYSSKW.XX).
439	BOF_STESK_KENNUNG_FILE_CREATE_ ERROR	The ID FILE for STEST.xx (only SHORT MESSAGES!) could not be created (STESKxx.YY).
440	BOF_STESK_KENNUNG_FILE_OPEN_ERROR	The ID FILE for STESTxx.YY (only SHORT MESSAGES!) could not be opened (STESKxx.YY).
441	BOF_STESK_KENNUNG_FILE_READ_ERROR	The ID FILE for STESTxx.YY (only SHORT MESSAGES!) 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 were returned by the interface on requesting the device component types.
445	BOF_DEVICE_DAT_FILE_NOT_FOUND_ERROR	Corresponding DEVICE-DAT file not found for the BOF configuration.
446	BOF_MAIN_MENU_ITEM_ERROR	Invalid BOF 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	Corresponding 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.

Code	Error Text	Name and Meaning of Error
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_ ERROR	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 info for the SYS message is too long.
456	BOF_DEVICE_STATUS_INFO_ERROR	More than one “critical” condition is administered in the DEVICE-STATUS INFO (SYSTEM-MAP) e.g.: parameter download.
457	BOF_SYS_MSG_HOOK_LIST_TIMEOUT_ERROR	The SYS-MSG-HOOK-LIST can not be accessed within the preset time.
458	BOF_PROCESS_LOGOUT_TIMEOUT_NETINTFC	NETINTFC has not logged out from the TASK LIST within the preset WAIT TIME.
459	BOF_PROCESS_LOGOUT_TIMEOUT_DESKTOP	DESKTOP has not logged out from the TASK LIST within the preset WAIT TIME.
460	BOF_PROCESS_LOGOUT_TIMEOUT_ CONTROLDATA	CONTROL DATA has not logged out from the TASK LIST within the preset WAIT TIME.
461	BOF_PROCESS_LOGOUT_TIMEOUT_ LOGDBCOM	LOGDBCOM has not logged out from the TASK LIST within the preset WAIT TIME.
462	BOF_PROCESS_LOGOUT_TIMEOUT_MPI	MPI has not logged out from the TASK LIST within the preset WAIT TIME.
463	BOF_PROCESS_LOGOUT_TIMEOUT_BOFINTFC	BOFINTFC has not logged out from the TASK LIST within the preset WAIT TIME.
464	BOF_IF_DLL_MODE_TO_SMALL	IF-DLL MODE set too small for the function to be executed.
465	BOF_WATCH_LIST_OVERRUN_ERROR	NO WATCHLIST available (overrun) for the selected device.
466	BOF_INVALID_WATCH_LIST_NUMBER_ERROR	INVALID WATCHLISTNUMBER for the selected DEVICE.
467	BOF_NO_SYSTEM_ERRORTEXT_ADM	There is NO administration system for access to the SYSTEM ERROR TEXTS (SYSANW.XX)
468	BOF_NO_TX_ERRORTEXT_ADM	There is NO administration system for access to the TRANSMISSION ERROR TEXTS (TXERR.XX)
469	BOF_NO_MECH_ERRORTEXT_ADM	There is NO administration system for access to the MECHANISM ERROR TEXTS (STERRyy.XX)
470	BOF_INVALID_PLC_TYPE	An invalid SPS type was recognised for the selected device.

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.
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 CNC 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	No tool management 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 controls.
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 can not 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	Part-result is invalid.
1042	BOF_FUNC_INVALID_PARAM	Invalid parameter for function
1043	BOF_PIPE_NO_FREE_PIPE	All pipes already assigned.

Code	Error Text	Name and Meaning of Error
1501	BOF_FUNC_NAME_LIMIT150	Name of interface 'B' functions is too large.
1502	EXEPTION	Internal error.
1503	EXEPTION	Internal error.
1504	EXEPTION	Internal error.
1505	EXEPTION	Internal error.
1506	EXEPTION	Internal error.
1507	EXEPTION	Internal error.
1508	EXEPTION	Internal error.
1509	EXEPTION	Internal error.
1510	EXEPTION	Internal error.
1511	EXEPTION	Internal error.
1512	BOF_FUNC_EOF_STRING_150	FI command incomplete.

8.4 Error Codes 2000 to 2999

Code	Meaning
2001	No channel free.
2002	Channel already open.
2003	Channel can not be closed.
2004	Channel not open.
2005	Re-initialization error.
2006	Channel can not 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) can not 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.

Code	Meaning
2052	Communication process (COM task) does not answer.
2053	"LOGINTFC.EXE" file not found.
2059	Error message from the LOG process.

Code	Meaning
2150	"LOGINTFC.DAT" file can not be opened.
2154	File Version Mismatch.
2155	"LOGINTFC.DAT" file is too large.
2156	Internal configuration error.
2160	Invalid command string.
2161	Telegram code not implemented.
2162	Parameters are outside of the limit value.
2163	Invalid parameter syntax.
2164	Unknown SPS variable.
2165	Not enough parameters transmitted.
2166	SPS map file can not be opened.
2167	SPS variable type not implemented.
2168	Reference error of SPS variable.
2169	Date can not be edited.
2170	Checksum error.
2171	Undefined telegram code.
2172	Missing processing rule.
2173	Too much data for the answer telegram.
2174	Unknown additional diagnostics information.
2175	Unknown unit.
2176	SPS variable is larger than 240 bytes.

Code	Meaning
2201	Input string "Date-Time" not in format: "DD.MM.YY hh:mm:ss".
2202	Effective data length of SIS telegram is too large.

Code	Meaning
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. (refer also to 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. (refer also to FI command: NPA1_/?)
	invalid Parametervalue or No.:[<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.
	[Nr.] missing Startparameter	The command has not been passed on in its entirety.
	Illegal Startparameter value [incorrect transmisson]	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. (refer also to FI command: NPA1_/?)
	ERROR : Invalid startparameter - ProcNo out of Range	When requesting one or more process parameters, an invalid definition range has been detected. Requests can only be made that remain within the range of the CNC process number [0...6].
	ERROR : Invalid startparameter - ProcNo out of Range	When requesting one or more axis parameters, an invalid definition range has been detected. Requests can only be made within the range of the axis numbers 1 to 20 or 32.
4001	ERROR : invalid function	The FI command contains an invalid parameter.
4002	NO_PARAMETER_DATA_FOUND	The 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).
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	An 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.
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.

Code	Error Text	Name and Meaning of Error
		=> Carry out update installation. • There is an error in the file. The file has been accidentally edited or illegally copied. Data recognition has thereby been invalid. => 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. => Contact our customer service department.
4006	Device Address out of Range	A system outside of the definition range has been selected in the command.
4007	Buffererror 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. => Contact our customer service department.
4012	Create_DLL_Error detected!	The result buffer could not be initialised. 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. When necessary, re-transmit the parameter(s) to the controls. If the error remains, or the parameter(s) can not 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 initialised 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 (in case of TSRPG25I.EXE) if an empty parameter name has been transmitted. This means that no parameter record is as yet in the controls. No error is returned.
4100	Couldn‘t open ParameterIndexFile:[File Error=xxxx]	An error has been recognised when attempting to open the parameter directory file. The following could all trigger this error: • Versions do not agree => Parameters have to be converted. (see Converting Parameters, page 8-18) • The parameter directory file has been accidentally destroyed. • The drive is faulty.
4102	ParameterIndexFile has wrong structure	An error has been detected when reading-out the parameter directory file which indicates that the data in the file are not

Code	Error Text	Name and Meaning of Error
		in the correct format. Check this by running Converting Parameters (p. 8-18). If the error continues to occur after this then you must contact our customer service department.
4103	to many Indizies found - File has wrong structure ?:	An error has been detected when creating the directory data. More directories have been recognised than allowed by the definition range. Probably the parameter directory data is from an earlier version. Carry out a Converting Parameters (p. 8-18). If the error remains, please contact our customer service department.
4104	invalid Parametervalue detected	An invalid range was detected when initializing (booting up the GUI). Contact our customer service department.
4105	Can’t create Parameterindex buffer: [filename]	No data could be provided in the memory. Close other applications to free up enough memory for the compilation of the data.
4108	Couldn’t find 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 has failed when starting the GUI. Contact our customer service department.
4111	invalid Parameter value Cxx.053 [Cxx.053 <Value>]	Initialization has 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 be once more transmitted to the controls. If the problem remains, 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 of the definition range. Otherwise, please contact our customer service department.
4201	invalid Parametertype	A parameter request has been made with a non-defined parameter type. Check the entry and/or request
4202	Buffersize not enough	The result of the parameter request can not be transmitted as the transmission range is not large enough. In case of 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	couldn’t find Dir 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 run on InterFace-Version: Version	During the command request, the program has detected that it can not be run on this version. Contact our customer service department.

Code	Error Text	Name and Meaning of Error
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[ParameterNummer	The parameter number is outside of 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" (in case of 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.
4223	WriteError by Config-SCR-File = [Fehlernummer]	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 [Dateiname]	This message text only appears in the starting phase with the setting "/U0" (in case of TSRPG25I.EXE). The attempt to rename the original file could not be executed. Check the properties for the corresponding parameter file and also the free remaining space on the storage medium.
4225	SaveError – Couldn't rename tmp -> dat => copy old to DAT[Dateiname]	This message text only appears in the starting phase with the setting "/U0" (in case of TSRPG25I.EXE). The attempt to recopy the newly created file could not be executed. Check the properties for the corresponding parameter file and also the free remaining 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>]	The specified file could not be stored acc. to the additional info. Check the amount of free memory on the storage medium and the access properties of the corresponding directory. Otherwise, please contact our customer service department.
4229	Can't create File = [file name <additional info>]	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]	This error message is only displayed internally when in debug mode.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 VerzLine [paramter directory line]	The specified parameter directory line could not be updated.
4235	Can't actualize Date or Length in Verzline	When updating the parameter director line, the date or the

Code	Error Text	Name and Meaning of Error
		length could not be 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 Parameternumber [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 continues to occur after this then you must contact our customer service department.
4240	Invalid Parametervalue =[parameter line]	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". A parameter conversion of parameters is executed by calling the DOS program "COPAxxyy.EXE" in standard installation directory "C:\MT-CNC\".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	Error Text	Name and Meaning of Error
5001	IF500_ERR_INVALID_ALIAS	Alias used is not defined (is not yet used)
5002	IF500_ERR_INVALID_LOGICALDEVICE	Invalid device address
5003	IF500_ERR_DEVICESYNTAX	Syntax error in the device address
5010	IF500_NO_REQUEST_ACTIVE	No request active
5011	IF500_INVALID_COMMANDSTRING	Invalid command string
5012	IF500_ERR_NO_COMMAND_BUFFER	No answer buffer specified
5013	IF500_ERR_NO_REGISTER_MODE	Incorrect cyclic login mode (internal error)
5014	IF500_ERR_NO_REFRESH_MODE	Incorrect cyclic update mode (internal error)
5050	IF500_ERR_REMOTE_CONNECT_FAILED	No access to remote PC possible
5051	IF500_ERR_REMOTE_DISCONNECTED	Remote connection has been disconnected
5052	IF500_ERR_REMOTE_NO_PROXY	Network interface can not be initialised
5101	IF500_ERR_UNEXPECTED	General unexpected error (internal error)
5102	IF500_ERR_OUT_OF_MEMORY	Memory error
5401	NET_EFALSE	Unspecific error
5402	NET_EINVPARAM	Invalid parameter passed to function
5403	NET_ETIMEOUT	Transfer timeout, remote PC not ready, or network connection down
5404	NET_ESND_ERROR	Send failed; error sending to a remote PC
5405	NET_ENOMEM	Memory shortage; in remote access of the interface
5406	NET_EINVCONN	Invalid connection to a remote PC
5407	NET_ESERVDIS	Service disabled
5408	NET_EABORT	Connection to remote partner aborted
5409	NET_EINVHOOK	Invalid parameter hook ID; Sys Message Handling
5410	NET_EINVDEVICE	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 processor number not OK.
6006	BOF_C_PROG_FAULT	Transmitted program number not OK.
6007	BOF_C_FILE_NOT_DEL	File can not 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 can not be positioned to N0000
6017	BOF_C_NCPROG_NOT_READ	File can not 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

Code	Meaning and Notes Regarding Diagnostics and Troubleshooting

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 on the BOF/GBO GUI have been deleted or corrupted.

Note: As for all error codes, additional information regarding the error can be requested via the "Error Codes" (p. 8-1). The error information informs the user in plain text regarding the cause of the error.

7000	Syntax error in NC program The "Error Codes" (p. 8-1) contains additional information.
7002	File with incorrect information. The "Error Codes" (p. 8-1) contains the file name and the line.
7005	File not found. The "Error Codes" (p. 8-1) contains the file name.
7006	File can not be created. The "Error Codes" (p. 8-1) contains the file name.
7008	File can not be read. The "Error Codes" (p. 8-1) contains the file name.
7009	Error in connecting the function interface. No connection can be made to the device (controls) by the function interface.
7015	Too many axes defined. More than 9 axes are being used in the CNC process.
7016	Invalid number of parameters. The number of parameters has been exceeded in the "NCPRG.CFG" file.
7017	Axis name is invalid. The axis name in the 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	Maximal axis speed is invalid. The value of axis parameter "CXX.016" is invalid.
7020	Maximal axis acceleration is invalid. The value of axis parameter "CXX.018" is invalid.
7021	Lowest run time of an NC record = [2.5...30ms]. The counter value of the parameter "METB" in the NC options of the BOF/GBO is outside of 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 positions after the decimal point are allowed. The process parameter "BXX.002" is invalid.
7024	Invalid counter value. The counter value of the parameter "VFBT" or “BBTC” in the NC options of the BOF/GBO is invalid.
7025	Only 0 (mm) or 1 (inch) allowed! The process parameter "BXX.001" is invalid.

Code	Meaning and Notes Regarding Diagnostics and Troubleshooting
7026	Counter value outside of the allowed range. Axis parameter "CXX.006" is smaller than 0.1.
7027	Internal record number is invalid. The record numbers in the NC program file are in the wrong order.
7028	Record number in the file is invalid. The "Error Codes" (p. 8-1) contains the names of the file in which the record numbers are not correct.
7070	Counter value outside of the allowed range (1..10). The counter value of the parameter "BBTC" in the NC options of the BOF/GBO is outside of the allowed range.
7077	Counter value outside of the allowed range (1..25). The counter value of the parameter "VFBT" in the NC options of the BOF/GBO is outside of the allowed range.
7083	Invalid parameter. The "Error Codes" (p. 8-1) contains the 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 (answer telegram)
8002	INVALIDARG	Incorrect request string
8003	REQUEST_NOT_FILLED	Internal run error
8004	GET_ATTRIBUT_FAILED	Incorrect attribute contained in answer telegram
8005	WALK	Internal run error
8006	EXTRACT_COMMON_INFO_FAILED	Error in transmitting parameter (answer telegram)
8007	WRONG_DATA_SIZE	Undefined data length in the answer
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 IfDllMode.
8011	ERROR_BYTE_INFO	Error reading out the error byte information
8012	CANT_OPEN_MODULDEF_INI	The "Moduldef.ini" file can not 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. keyword error or message missing).
8024	DEVICE_ADR_FALSE	Device address not in the valid range.
8025	MODULE_NO_FALSE	Module number not within valid range (0-99)
8026	KEY_WORD_FALSE	Wrong keyword (e.g. No ModulY in section names [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 of 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

Code	Error Text	Name and Meaning of Error
8039	NO_TOOLMANAGEMENT	The tool management is not activated for the process
8040	WRONG_TOOL_NUMBER	Wrong 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 is 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 Codes10000 and above

Code	Meaning
10001	The WIN-HMI component is not installed.
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	WIN-HMI has not been started in the same process

8.11 Error Codes 35000 and above

Code	Error Text	Name and Meaning of Error
35500	PARA_NOT_DEFINED	CMOS parameter not yet defined. Remedy: write CMOS parameter.
35501	VALUE_TYPE_INVALID	Invalid coding type.
35502	ERROR_VERSION_MISMATCH	Command does not yet exist for set IfDllMode.

8.12 Error Codes100000 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 size of record
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

Code	Error Text	Name and Meaning of Error
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 bytes.
100115	FS_ERROR_T04	Without "iT04" ID
100116	FS_ERROR_FILE_NOT_OPEN	File can not 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 Codes110000 and above

Code	Error Text	Name and Meaning of Error
110001	BOF_MAP_VERSION_FUNC_ERROR	Incorrect Dll mode set
110002	BOF_MAP_FILE_VERSION_ERROR	Incorrect file version number
110003	BOF_MAP_LANGKENNUNG_VERSION_ERROR	When the long ID version is invalid
110004	BOF_MAP_LANGKENNUNG_INVALID_ERROR	When the long ID is invalid
110005	BOF_MAP_LANGKENNUNG_PARAM_ERROR	Missing parameter in SplittLangKennung
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_KENNUNGS_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 device 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	13...46 → variable not found.
110022	BOF_MAP_LANGKENNUNG_DIFFERENT_TO_ MAP12	MAP file long ID is different from SPS 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	No map access has been generated
110026	BOF_MAP_LANGKENNUNG_INVALID_NO_MAP_ ERROR	Long ID is not valid and no SPS Map access has yet been initialised.

Code	Error Text	Name and Meaning of Error
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 administration information has not been created
110030	BOF_MAP_DOWNLOAD_STATUS	Access to map during a download
110050	BOF_MAP_COMMON_FILETOOL_ERROR	Basic number cErrorGruppe_filetool
110100	BOF_MAP_COMMON_MAP_BAS_C_ERROR	Basic number ErrorGruppe_map_bas_c
110150	BOF_MAP_COMMON_LKENN_ERROR	Basic number cErrorGruppe_filetool
110200	BOF_MAP_COMMON_GROUP_ERROR	Basic number general error
110263 110296	BOF_MAP_VARIABLE_NO_FOUND_ERROR_ BASE+13 .... BOF_MAP_VARIABLE_NO_FOUND_ERROR_ BASE+46	SPS variable does not exist; to clearly identify the error, the error number is added to the BASE.

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	Counter value has been fallen below
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" A CLC internal error that has effected the exchange of data has occurred during parameter transmission (see C-0-0041).
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 error
0x2001	Name does not exist.
0x2002	Name transmitted too short.
0x2003	Name transmitted too long.
0x2004	Name can not be changed.
0x2005	Name currently write-protected.
0x3002	Attribute transmitted too short.
0x3003	Attribute transmitted too long.
0x3004	Attribute can not be changed.
0x3005	Attribute currently write-protected.
0x4001	Unit does not exist.
0x4002	Unit transmitted too short.
0x4003	Unit transmitted too long.
0x4004	Unit can not be changed.
0x4005	Unit currently write-protected.
0x5001	Minimal input value does not exist.
0x5002	Minimal input value transmitted too short.
0x5003	Minimal input value transmitted too long.
0x5004	Minimal input value can not be changed.
0x5005	Minimal input value currently write-protected.
0x6001	Maximal input value does not exist.
0x6002	Maximal input value transmitted too short.
0x6003	Maximal input value transmitted too long.

Code	Error Messages in Serial Protocol
0x6004	Maximal input value can not be changed.
0x6005	Maximal 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)
0x700C	"Date outside of counter range" The transmitted value is smaller than zero or larger than the modulo value (S-0-0103); in case of modulo axis.
0x700D	"Length of date can not be currently changed" The length of the date can not be changed in the current mode.
0x700E	"Length of the date can not be currently changed" The length of the date is permanently write-protected.
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 controls as it must currently communicate with the same drive (higher priority).
0x800C	Unauthorised 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 error codes have been defined:

Code	Error Messages in Serial Protocol
0x8006	HS timeout
0x8007	Doubled AT breakdown.
0x8008	Lightwave ring not closed.
0x8009	Lightwave 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 can not be currently activated.
0xD001	Drive error (status class 1, S-0-0011).
0xD004	Command can not be executed in drive.
0xF001	"Configuration error". When configuring the command channel or actual channel, an error has occurred: a) There are too many command values or actual values configured b) The configured command values 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 controls no longer access 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.

9 Answers to Frequently Asked Questions: FAQ

9.1 Function Interface FAQs

This chapter provides FAQs (Frequently Asked Questions) regarding the Rexroth Indramat function interface that we have gathered from customer feedback.

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", then it is easy to see what has actually issued the message box.

Answer To do this, open the Windows NT Task Manager e.g. using keyboard combination: <Ctrl> + <Shift> + <Esc>

Mark the message box entry in the "applications" card and click with the right-hand mouse button.

Note: The keyboard combination <Ctrl>+<F10> does nor function here for the right mouse button!

Select the “Switch to Process” command in the context menu that opens for the marked 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 only serves for issuing single requests that read or write. Group requests are issued via the routines for cyclic reading via pipes.

Question 3 Why does the login for my application to the function interface take so long?

3-2 During the initialization phase of the function interface, numerous security checks are made (refer here also to the chapter, Function Interface Structure with Configuration Data, p. 3-8).

9.2 Windows NT FAQs

This chapter contains FAQs regarding Windows NT that we have gathered from customer feedback.

Question 1 How can I automatically login with my name and password (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	Type:	Contents	Info
AutoAdminLogon	REG_SZ	1	Switch on/off Autologin
DefaultUserName	REG_SZ	<user name>	User name
DefaultPassword	REG_SZ	<password>	User password (a password must exist)
DefaultDomainName	REG_SZ	<domain name>	Login must be made on another computer

Note: A message box no longer appears. If you want to log in using another name then you must keep the <SHIFT> key pressed during the starting procedure. You will then be prompted to enter your name and password.

Note: If no password is entered in the registry then AutoLogin only functions once and Windows then resets “AutoAdminLogon” to "0". The password must also be entered. Please note that the password can then be viewed by everyone in the registry !

10 Glossary

System

All processes that are controlled by an MTC200 or MT-CNC are termed systems. Control by families MTC200 or MT-CNC therefore represents a system.

ANSI

American **N**ational **S**tandards **I**nstitute, American standards institute which developed the ANSI emulation (refer also to: ANSI Code).

ANSI Code.

Standard code standardized by ANSI which allows pictures, animations and texts to be generated as well as sounds to be generated from the PC loudspeaker as a sequence of ANSI control frequencies. Method of designing a GUI mostly used in mailboxes. Often, the ASCII code is also referred to as the ANSI code. These characters are generated in a document by pressing the <AltGr> key and the respective code.

ASCII

American **S**tandard **C**ode for **I**nformation **I**nterchange; more widespread code, particularly on home and personal computers. Used for displaying numbers, letters and special characters. Designed as a 7-bit code with a character store of 128 characters or as an 8-bit code with a character store of 256 characters including upper case and lower case letters. The unassigned eighth and ninth bits (in byte format) are used as parity bits.

Operating Date

The operating date is data block element 7 of a parameter. The value of the parameter is stored by it.

BOF

Bedien**O**ber**F**läche (Engl.: Graphical User Interface, GUI) (see GBO).

BTV20

The BTV20 is a machine operating terminal in which one or more NC controls can be integrated with SPS or one or more stand-alone SPSs. The number of components that can be integrated depends on their configuration. In contrast to the BTV30, the BTV20 provides a user-oriented function keyboard with the following characteristics:

- Faceplate made of 4mm aluminum with scarfed edges.
- fully flushed, chemically resistant polyester foil with lifted stamping.
- integrated EMC-compatible glass plate for protection of display.
- integrated machine keys with intermediate plate avoids doubled operation and unintentional triggering of keys.
- Key switch for locking the security functions.

BTV30

The BTV30 has all of the functions and operating elements of an entire industrial PC. In addition to the 10.4" flat color display and a complete ASCII keyboard with cursor block, keyboard mouse and Windows keys, this also contains a standard diskette drive located behind a lockable cover and a connection for an external keyboard. The 10 PC function keys are located under the display and the 8 machine function keys are located to the right of the display. These keys are either led to the outside by a bush or, in the case of an integrated SPS component, are directly connected to the SPS. Genuine key elements are embedded in the stable faceplate made of PC/ABS allowing fatigue-free programming, even for longer periods of time. The display is protected by a stable, EMC-tested glass plate. When the diskette cover is closed, the front of the BTV30 complies to protective system IP65 and is resistant to all known coolant and lubricant.

Client

A client is a computer system or process that requests the services of another computer system or process. It is also a workplace computer that can use the services and resources (e.g. printer, scanner, plotter) of a server or other clients. It usually has a lot less access authorization than the server.

DDE

Dynamic **D**ata **E**xchange, is a standard defined by Microsoft for data exchange between programs under MS-Windows from Version 3.0. DDE allows files or parts of files to be linked between two applications that support the DDE standard. A difference is made here between a source application (server) and a target application (client) whereby the target application maps an entire copy or part copy of the server file. If the data in the source file is changed then this information is transmitted to the target application via the link and is dynamically updated there. However, DDE communication can simply be used to exchange commands and instructions between two programs.

DLL

Dynamic-**L**ink **L**ibrary, is a library linked to a program when it is running. DLLs are special data for Windows, from which, e.g. functions, dialog boxes or symbols are loaded from applications. They simplify programming and save hard drive space when, e.g. the same functions are required by several applications. A dynamic library provides several advantages: It only needs to be loaded on demand and does not use any RAM up to this point.

dual port RAM

This is a memory area between the two connected users: The actual control and the user interface (GUI) (PC). This memory area, used by both users, only allows limited data traffic.

If the control, e.g. wants to transmit a message to the GUI (PC) then it first sends this to the Dual Port RAM. A cyclic mechanism running at the PC recognizes this new information and fetches it, acknowledges it for the control and then passes it on for further processing (display on the GUI).

When reversed, and the GUI wants to transmit a message to the control, then it also first stores this in the Dual Port RAM. A cyclic mechanism (analog in the control), recognizes and fetches this new information, acknowledges it for the PC and passes it on for further 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). From the point of view of an application (client), the device is **not** at the PC on which the application is running, but at a PC within the PC network.

GBO

Graphical User Interface (GUI). Referred to throughout this document also as GBO, the graphical user interface provides a wide range of possibilities to the machine manufacturer and the end user for the configuration and operation of the System200 control family, as well as for the display of data. Due to the structure, the assignment of keys and the communication mechanism, various things can be controlled from the calling of screen masks and functions right up to user guidance via the SPS application.

Device

A device is a piece of control hardware, a drive device or an I/O device.

Device Address

The device address corresponds to the system address within the Indramat BOF/GBO. This means, e.g. that device 00 corresponds to system 0. Please observe, however, that the BOF/GBO always requires a device 00.

Device Type:

The device type indicates which Rexroth Indramat device this is, e.g. MTC200-P, MTVNC, ISP200-R, etc.

FarDevice

To configure the PC network, a list of the FarDevices is required as well as that of the PCs. For a linear and unique addressing of the devices in the PC network, the FarDevice address has been introduced. This ensures that every available device within the PC network receives an additional address (FarDevice address). A device that is available within the PC network has a FarDevice address and is termed a FarDevice. The list of FarDevices is created based on the devices connected previously at every PC

Local Device

This term depends on the point of view (refer also to Remote Device). From the point of view of an application, the device is at the PC on which the application is running.

MCI

A function within the MTC200 or MT-CNC GUIs for the visualization and operation of linked controls within the MTC200 or MT-CNC family (systems). It is an optional extra to the graphical user interface GBO 17VRS (Release V02).

MPI

(**MPI** = **M**ulti-**P**rotocol **I**nterface). The Rexroth Indramat MPI provides a standardized user interface for the communication interfaces Profibus-FMS, MMS-Ethernet (MAP), TCP/IP and FIPWAY on PC assemblies under the Windows-NT 4.0 operating system. The MPI provides the interfaces for the realization of the client and server applications. The MPI communication driver makes the connection between the MPIs and the function interface. This ensures the connection to all protocols supported by MPI, and that can be configured via the function interface, at the Rexroth Indramat devices.

MT-CNC

The MT-CNC is the Rexroth Indramat control family comprising the controls MTC02 and MTC03, and including all accompanying components. The BTV1.3 is used as visualization device.

MTC200

MTC200 is a new control generation based on the PC. The MTC200 system integrates the entire functions of a CNC and SPS control, including the complete drive technology. Components of this system are, e.g. MTC-P, MTC-R, MTS-P, MTS-R.

Up to seven independent CNC processes can be controlled by the MTC200. These seven CNC processes can be divided amongst a maximum of 32 axes. The MTC200 is thereby multi-axis capable as well as multi-process capable.

MTC-P

The MTC-P is a powerful CNC control in ISA-bus circuit card format for insertion into an industrial PC and belongs to the MTC200 family. It consists of a basic unit with the processor system of the CNC and an integrated axis processor at which a maximum of 8 drives can be attached via a SERCOS interface. By the expansion by a max. of three axis processor modules, a total of up to 32 drives can be controlled at the highest expansion level. These can then be divided between a maximum of seven processes. Together with the SPS control MTS-P01.1, this unit forms a compact and flexible solution for a tool machine control.

MTC200-P

See MTC-P.

MTC-R

The RECO-based CNC component MTC-R comprises a complete CNC processor, compatible to MTC-P, and an axis processor module for controlling up to 8 digital drives via the SERCOS interface. Up to 3 additional axis processor modules can be slotted in via the PC/104 bus in order to achieve the maximum capacity of 32 drives. While an additional PC/104 module can be inserted in the regular-width component (for the control of up to 16 drives), a double-width housing is available for additional axis processors.

The MTC-R can not function on its own; it always requires an MTS-R as an offshore adaptive control. Both components are connected via a local bus for communication between MTC-R and MTS-R. Both components are then together slotted into an RMB02.2 or RMB02.4 component carrier. When required, and as described for the MTS-R, additional I/O components can be addressed for the local I/O level.

MTC200-R

See MTC-R.

MTS-R

The RECO-SPS is a powerful small-sized SPS that is compatible to the SPS in the MTC200 control system. The housing conforms to IP20. It can be used as a stand-alone SPS and together with an MTC-R as a slave SPS. There is an RS 232/RS 485 programming interface available for the connection of several SPS controls together, the connection to a programming device or to a PC. There is a free serial interface (RS 232/RS 422) available for connection to a printer, a write/read memory or a visualization device.

The MTS-R01.1 occupies one module slot in the RMB02 module carrier and two in the MTS-R02.1. The ISP200-R is thereby able to drive the bus for up to an additional 15 I/O modules. There is an internal local bus with an adapter board for communication with the MTC-R NC control that is part of the system.

Optionally, the MTS-R01 and the MTS-R02 can be equipped with the open field bus interfaces INTERBUS or PROFIBUS-DP. Decentralized I/O periphery devices, each with up to 4096 inputs and 4096 outputs, can be connected via these optional interfaces. The MTS-R02 can also be additionally equipped with a serial interface module (2 x RS 232 und 2 x RS 422).

PC Network

The PC network is the connection of several PCs on the level of the function interface. The PC network comprises the PCs that are used in the controlling of a machine (most importantly the visualization, operation and programming).

Process

The process is the combination of functions and axes from a control standpoint that is applicable to the MTC200 and MT-CNC controls into a processing unit within a control system. Every MT-CNC (MTC02/03) or MTC-P / MTC-R (MTC200) has a maximum of 7 processes.

RECO

The RECO is a modular I/O system for the fast exchange of signals with the SPS The module carriers for two or four I/O modules can be mounted on a standardized hat rail. Analog and digital inputs and outputs as well as serial interfaces are available.

Registry

See Registry Database.

Registry Database

In Windows NT, the registry database replaces most of the INI files in Win3.x (these files still exist in Windows NT, but are mostly used only by 16-bit applications). Information regarding the configuration is logged in the registry both by Windows NT itself as well as by all 32-bit programs.

Registry Editor

The entries in the registry database "Registry" are changed using the registry editor. The editor is located in the Windows system directory and is called REGEDT32.EXE (enter -> "Start" -> "Run" REGEDT32).

RS232

Serial interface with a 9-pin or 25-pin connection that conforms to the V.4 standard and that has been developed by the EIA for communication with devices; maximum 19200 bits/second. It is often used for connections between computers and modems.

Server

A server is a computer that contains applications and documents that can be accessed by other computers connected to it (clients). The term also indicates a program that provides certain services that can be accessed using programs that have been specially adapted on the server itself.

Shared Memory

An area in the computer's RAM that can be accessed by several processes (applications).

System200

The System200 from Rexroth Indramat is a comprehensive and scalable control and drive system for the entire field of mechanical engineering and system construction. Various software packages (WIN-HMI, MPI, function interface, etc) for one and the same PC hardware platform (MTC200), various visualization devices (BTV20, BTV30, etc), application-optimized drives (DIAX04, ECODRIVE, etc) and periphery connections (Profibus-DP, Profibus-FMS, SERCOS interface, etc) can thereby be chosen.

Thread

Threads are objects within processes that execute program instructions. They allow various actions to be carried out simultaneously within the same process and allow a process to execute different parts of a program simultaneously on different processors.

WIN-HMI

(**WIN-HMI** = **WIN**dows based - **H**uman **M**achine **I**nterface). The WIN-HMI software package is a unified GUI for automatic production.

11 List of Figures

12 Index

D

E

F

G

I

K

L

M

N

O

P

R

Z

13 Service & Support

13.1 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 07:00 am -6:00 pm
- by fax: **+49 (0) 9352 40 49 41**
- by e-mail: **service@indramat.de**

13.2 Service-Hotline

Außerhalb der Helpdesk-Zeiten ist der Service direkt ansprechbar unter

+49 (0) 171 333 88 26
oder **+49 (0) 172 660 04 06**

After helpdesk hours, contact our service department directly at

+49 (0) 171 333 88 26
or **+49 (0) 172 660 04 06**

13.3 Internet

Weitere Hinweise zu Service, Reparatur und Training finden Sie im Internet unter

www.indramat.de

Außerhalb Deutschlands nehmen Sie bitte zuerst Kontakt mit Ihrem lokalen Ansprechpartner auf. Die Adressen sind im Anhang aufgeführt.

Additional notes about service, repairs and training are available on the Internet at

www.indramat.de

Please contact the sales & service offices in your area first. Refer to the addresses on the following pages.

13.4 Vor der Kontaktaufnahme... - Before contacting us...

Wir können Ihnen schnell und effizient helfen wenn Sie folgende Informationen bereithalten:

1. detaillierte Beschreibung der Störung und der Umstände.
2. Angaben auf dem Typenschild der betreffenden Produkte, insbesondere Typenschlüssel und Seriennummern.
3. Tel.-/Faxnummern und e-Mail-Adresse, unter denen Sie für Rückfragen zu erreichen sind.

For quick and efficient help, please have the following information ready:

1. Detailed description of the failure and circumstances.
2. Information on the type plate of the affected products, especially type codes and serial numbers.
3. Your phone/fax numbers and e-mail address, so we can contact you in case of questions.

13.5 Kundenbetreuungsstellen - Sales & Service Facilities

(white)	Verkaufsniederlass	(white)	sales agencies
(grey)	Niederlassungen	(grey)	offices providing service

Deutschland – Germany

vom Ausland: (0) nach Landeskennziffer weglassen!
from abroad: don't dial (0) after country code!

Vertriebsgebiet Mitte Germany Centre Rexroth Indramat GmbH Bgm.-Dr.-Nebel-Str. 2 97816 Lohr am Main **Kompetenz-Zentrum Europa** Tel.: +49 (0)9352 40-0 Fax: +49 (0)9352 40-4885	SERVICE **CALL ENTRY CENTER** **MO – FR** **von 07:00 - 18:00 Uhr** **from 7 am – 6 pm** **Tel. +49 (0) 9352 40 50 60** service@indramat.de	SERVICE **HOTLINE** **MO – FR** **von 17:00 - 07:00 Uhr** **from 5 pm - 7 am** + SA / SO **Tel.: +49 (0)172 660 04 06** **oder / or** **Tel.: +49 (0)171 333 88 26**	SERVICE **ERSATZTEILE / SPARES** verlängerte Ansprechzeit - extended office time - ♦ nur an Werktagen - only on working days - ♦ von 07:00 - 18:00 Uhr - from 7 am - 6 pm - **Tel. +49 (0) 9352 40 42 22**
Vertriebsgebiet Süd Germany South Rexroth Indramat GmbH Ridlerstraße 75 80339 München Tel.: +49 (0)89 540138-30 Fax: +49 (0)89 540138-10 indramat.mue@t-online.de	Gebiet Südwest Germany South-West Mannesmann Rexroth AG Vertrieb Deutschland – VD-BI Geschäftsbereich Rexroth Indramat Regionalzentrum Südwest Ringstrasse 70 / Postfach 1144 70736 Fellbach / 70701 Fellbach Tel.: +49 (0)711 57 61–100 Fax: +49 (0)711 57 61–125	Vertriebsgebiet Ost Germany East Rexroth Indramat GmbH Beckerstraße 31 09120 Chemnitz Tel.: +49 (0)371 35 55-0 Fax: +49 (0)371 35 55-333	Vertriebsgebiet Nord Germany North Mannesmann Rexroth AG Vertriebsniederlassung Region Nord Gesch.ber. Rexroth Indramat Walsroder Str. 93 30853 Langenhagen Tel.: +49 (0) 511 72 66 57-0 Fax: +49 (0) 511 72 66 57-95
Vertriebsgebiet West Germany West Mannesmann Rexroth AG Vertrieb Deutschland Regionalzentrum West Borsigstrasse 15 40880 Ratingen Tel.: +49 (0)2102 409-0 Fax: +49 (0)2102 409-406	Vertriebsgebiet Mitte Germany Centre Mannesmann Rexroth AG Gesch.ber. Rexroth Indramat Lilistraße 14-18 63067 Offenbach Tel.: +49 (0) 69 82 00 90-0 Fax: +49 (0) 69 82 00 90-80	Vertriebsgebiet Ost Germany East Mannesmann Rexroth AG GB Rexroth Indramat GmbH Holzhäuser Str. 122 04299 Leipzig Tel.: +49 (0)341 86 77-0 Fax: +49 (0)341 86 77-219	Vertriebsgebiet Nord Germany North Rexroth Indramat GmbH Kieler Straße 212 22525 Hamburg Tel.: +49 (0) 40 81 955 966 Fax: +49 (0) 40 85 418 978

Europa – Europe

vom Ausland: (0) nach Landeskennziffer weglassen, Italien: 0 nach Landeskennziffer mitwählen
from abroad: don't dial (0) after country code, Italy: dial 0 after country code

Austria - Österreich Mannesmann Rexroth Ges.m.b.H. Gesch.ber. Rexroth Indramat Hägelingasse 3 1140 Wien Tel.: +43 (0)1 9852540-400 Fax: +43 (0)1 9852540-93	Austria - Österreich Mannesmann Rexroth G.m.b.H. Gesch.ber. Rexroth Indramat Industriepark 18 4061 Pasching Tel.: +43 (0)7221 605-0 Fax: +43 (0)7221 605-21	Belgium - Belgien Mannesmann Rexroth N.V.-S.A. Gesch.ber. Rexroth Indramat Industrielaan 8 1740 Ternat Tel.: +32 (0)2 5830719 Fax: +32 (0)2 5830731 indramat@rexroth.be	Denmark - Dänemark BEC AS Zinkvej 6 8900 Randers Tel.: +45 (0)87 11 90 60 Fax: +45 (0)87 11 90 61
Czech Republic - Tschechien Mannesmann-Rexroth, spol.s.r.o. Hviezdoslavova 5 627 00 Brno Tel.: +420 (0)5 48 126 358 Fax: +420 (0)5 48 126 112	England Mannesmann Rexroth Ltd. Rexroth Indramat Division Broadway Lane, South Cerney Cirencester, Glos GL7 5UH Tel.: +44 (0)1285 863000 Fax: +44 (0)1285 863030	Finland - Finnland Rexroth Mecman Oy Rexroth Indramat division Ansatie 6 017 40 Vantaa Tel.: +358 (0)9 84 91-11 Fax: +358 (0)9 84 91-13 60	France - Frankreich Mannesmann Rexroth S.A. Division Rexroth Indramat Parc des Barbanniers 4, Place du Village 92632 Gennevilliers Cedex Tel.: +33 (0)141 47 54 30 Fax: +33 (0)147 94 69 41 Hotline: +33 (0)608 33 43 28
France - Frankreich Mannesmann Rexroth S.A. Division Rexroth Indramat 270, Avenue de Lardenne 31100 Toulouse Tel.: +33 (0)5 61 49 95 19 Fax: +33 (0)5 61 31 00 41	France - Frankreich Mannesmann Rexroth S.A. Division Rexroth Indramat 91, Bd. Irène Joliot-Curie 69634 Vénissieux – Cedex Tel.: +33 (0)4 78 78 53 65 Fax: +33 (0)4 78 78 53 62	Hungary - Ungarn Mannesmann Rexroth Kft. Angol utca 34 1149 Budapest Tel.: +36 (1) 364 00 02 Fax: +36 (1) 383 19 80	Italy - Italien Mannesmann Rexroth S.p.A. Divisione Rexroth Indramat Via G. Di Vittoria, 1 20063 Cernusco S/N.MI Tel.: +39 02 2 365 270 Fax: +39 02 700 408 252378
Italy - Italien Mannesmann Rexroth S.p.A. Divisione Rexroth Indramat Via Borgomanero, 11 10145 Torino Tel.: +39 011 7 50 38 11 Fax: +39 011 7 71 01 90	Italy - Italien Mannesmann Rexroth S.p.A. Divisione Rexroth Indramat Via del Progresso, 16 (Zona Ind.) 35020 Padova Tel.: +39 049 8 70 13 70 Fax: +39 049 8 70 13 77	Italy - Italien Mannesmann Rexroth S.p.A. Divisione Rexroth Indramat Via Mascia, 1 80053 Castellamare di Stabia NA Tel.: +39 081 8 71 57 00 Fax: +39 081 8 71 68 85	Italy - Italien Mannesmann Rexroth S.p.A. Divisione Rexroth Indramat Viale Oriani, 38/A 40137 Bologna Tel.: +39 051 34 14 14 Fax: +39 051 34 14 22
Netherlands - Niederlande/Holland 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 indramat@hydraudyne.nl	Netherlands - Niederlande/Holland Rexroth Hydrocare B.V. Kruisbroeksestraat 1 (P.O. Box 32) 5281 RV Boxtel Tel.: +31 (0)411 65 19 51 Fax: +31 (0)411 67 78 14	Norway - Norwegen Rexroth Mecman 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	Poland - Polen Mannesmann Rexroth Sp.zo.o. Biuro Poznan ul. Dabrowskiego 81/85 60-529 Poznan Tel.: +48 061 847 67 99 Fax: +48 061 847 64 02
Rumania - Rumänien Mannesmann Rexroth Sp.zo.o. Str. Drobety nr. 4-10, app. 14 70258 Bucuresti, Sector 2 Tel.: +40 (0)1 210 48 25	Russia - Russland Tschudnenko E.B. Arsenia 22 153000 Ivanovo Tel.: +7 093 223 96 33	Spain - Spanien Mannesmann 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	Spain - Spanien Goimendi S.A. División Rexroth Indramat Parque Empresarial Zuatzu C/ Francisco Montagne no.2 20018 San Sebastian Tel.: +34 9 43 31 84 21

+40 (0)1 210 29 50 Fax: +40 (0)1 210 29 52	oder/or +7 093 223 95 48 Fax: +7 093 223 46 01	Fax: +34 9 37 47 94 01	- service: +34 9 43 31 84 56 Fax: +34 9 43 31 84 27 - service: +34 9 43 31 84 60 satindramat-goimendi@adegi.es
Sweden - Schweden Rexroth Mecman Svenska AB Rexroth Indramat Division Varuvägen 7 125 81 Stockholm Tel.: +46 (0)8 727 92 00 Fax: +46 (0)8 647 32 77	Slowenia - Slowenien DOMEL elektromotorji in gospodinjski aparati d.d. Otoki 21 4228 Zelezniki Tel.: +386 4 51 17 100 Fax: +386 4 51 17 225	Switzerland East - Schweiz Ost Mannesmann Rexroth Schweiz AG Gesch.ber. Rexroth Indramat Gewerbestraße 3 8500 Frauenfeld Tel.: +41 (0)52 720 21 00 Fax: +41 (0)52 720 21 11	Switzerland West - Schweiz West Mannesmann 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
Turkey - Türkei Mannesmann Rexroth Hidropar A..S. Fevzi Cakmak Cad No. 3 34630 Sefaköy Istanbul Tel.: +90 212 541 60 70 Fax: +90 212 599 34 07			

Africa, Asia, Australia – incl. Pacific Rim

vom Ausland:	(x) nach Landeskennziffer weglassen!
from abroad:	don't dial (x) after country code!

Australia - Australien AIMS - Australian Industrial Machinery Services Pty. Ltd. Unit 3/45 Horne ST Campbellfield , VIC 3061 Melbourne Tel.: +61 (0)3 93 59 02 28 Fax: +61 (0)3 93 59 02 86	Australia - Australien Mannesmann Rexroth Pty. Ltd. No. 7, Endeavour Way Braeside Victoria, 31 95 Melbourne Tel.: +61 (0)3 95 80 39 33 Fax: +61 (0)3 95 80 17 33 mel@rexroth.com.au	China Shanghai Mannesmann Rexroth Hydraulics & Automation Ltd. Wai Gaoqiao Free Trade Zone No.122, Fu Te Dong Yi Road Shanghai 200131 - P.R.China Tel.: +86 21 58 66 30 30 Fax: +86 21 58 66 55 23	China Mannesmann Rexroth (China) Ldt. 15/F China World Trade Center 1, Jianguomenwai Avenue Beijing 100004, P.R.China Tel.: +86 10 65 05 03 80 Fax: +86 10 65 05 03 79
China Mannesmann Rexroth (China) Ldt. A-5F., 123 Lian Shan Street Sha He Kou District Dalian 116 023, P.R.China Tel.: +86 411 46 78 930 Fax: +86 411 46 78 932	China Mannesmann Rexroth (China) Ldt. Guangzhou Repres. Office Room 1014-1016, Metro Plaza, Tian He District, 183 Tian He Bei Rd Guangzhou 510075, P.R.China Tel.: +86 20 8755-0030 +86 20 8755-0011 Fax: +86 20 8755-2387	Hongkong Rexroth (China) Ldt. 1/F., 19 Cheung Shun Street Cheung Sha Wan, Kowloon, Hongkong Tel.: +852 22 62 51 00 Fax: +852 27 41 33 44	India - Indien Mannesmann Rexroth (India) Ltd. Rexroth Indramat Division Plot. A-58, TTC Industrial Area Thane Turbhe Midc Road Mahape Village Navi Mumbai - 400 701 Tel.: +91 (0)22 7 61 46 22 Fax: +91 (0)22 7 68 15 31
India - Indien Mannesmann Rexroth (India) Ltd. Rexroth Indramat Division Plot. 96, Phase III Peenya Industrial Area Bangalore - 560058 Tel.: +91 (0)80 8 39 73 74 Fax: +91 (0)80 8 39 43 45	Indonesia - Indonesien PT. Rexroth Wijayakusuma Jl. Raya Bekasi Km 21 Pulogadung Jakarta Timur 13920 Tel.: +62 21 4 61 04 87 +62 21 4 61 04 88 Fax: +62 21 4 60 01 52	Japan Rexroth Automation Co., Ltd. Service Center Japan Yutakagaoka 1810, Meito-ku, NAGOYA 465-0035, Japan Tel.: +81 (0)52 777 88 41 +81 (0)52 777 88 53 +81 (0)52 777 88 79 Fax: +81 (0)52 777 89 01	Japan Rexroth Automation Co., Ltd. Rexroth Indramat Division 1F, I.R. Building Nakamachidai 4-26-44, Tsuzuki-ku YOKOHAMA 224-0041, Japan Tel.: +81 (0)45 942 72 10 Fax: +81 (0)45 942 03 41
Korea Mannesmann Rexroth-Korea Ltd. Rexroth Indramat Division 1500-12 Dadae-Dong- Saha-Ku Pusan, 604-050 Republic of South Korea Tel.: +82 (0)51 26 00 741 Fax: +82 (0)51 26 00 747 gyhan@rexrothkorea.co.kr	Malaysia Mannesmann Rexroth SDN.BHD. Head Office No. 3, Block B, Jalan SS 13/5 Subang Jaya Industrial Estate 47500 Petaling Jaya - Selangor Tel.: +60 (0) 3 73 44 870 Fax: +60 (0) 3 73 44 864	South Africa - Südafrika TECTRA Automation (Pty) Ltd. 28 Banfield Road,Industria North RSA - Maraisburg 1700 Tel.: +27 (0)11 673 20 80 Fax: +27 (0)11 673 72 69	Taiwan Rexroth Uchida Co., Ltd. No.17, Lane 136, Cheng Bei 1 Rd., Yungkang, Tainan Hsien Taiwan, R.O.C. Tel.: +886 (0)6 25 36 565 Fax: +886 (0)6 25 34 754
Thailand NC Advance Technologies Co. Ltd.			

59/76 Moo 9 Soi Ramintra 34 Ramintra Road, Tharang, Bangkhen Bangkok 10220 Tel.: +66 2 943 70 62 +66 2 943 71 21 Fax: +66 2 509 23 62 sonkawin@hotmail.com			

Nordamerika – North America

USA Hauptniederlassung - Headquarters Mannesmann Rexroth Corporation Rexroth Indramat Division 5150 Prairie Stone Parkway Hoffman Estates, IL 60192-3707 **Competence Centre America** Tel.: +1 847 6 45 36 00 Fax: +1 847 6 45 62 01 service@indramat.com	USA Central Region - Mitte Mannesmann Rexroth Corporation Rexroth Indramat Division Central Region Technical Center Auburn Hills, MI 48326 Tel.: +1 248 3 93 33 30 Fax: +1 248 3 93 29 06	USA Southeast Region - Südwest Mannesmann Rexroth Corporation Rexroth Indramat Division Southeastern Technical Center 3625 Swiftwater Park Drive Suwanee, Georgia 30174 Tel.: +1 770 9 32 32 00 Fax: +1 770 9 32 19 03	USA SERVICE-HOTLINE - 7 days x 24hrs - **+1-800-860-1055**
USA Northeast Region – Nordost Mannesmann 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	USA Northeast Region – Nordost Mannesmann 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	Canada East - Kanada Ost Basic Technologies Corporation Burlington Division 3426 Mainway Drive Burlington, Ontario Canada L7M 1A8 Tel.: +1 905 335 55 11 Fax: +1 905 335-41 84	Canada West - Kanada West Basic Automation Burnaby 5345 Goring St. Burnaby, British Columbia Canada V7J 1R1 Tel. +1 604 205-5777 Fax +1 604 205-6944 dave.gunby@basic.ca

Südamerika – South America

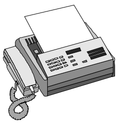
Argentina - Argentinien Mannesmann Rexroth S.A.I.C. Division Rexroth Indramat Acassusso 48 41/7 RA - 1605 Munro (Buenos Aires) Tel.: +54 (0)11 4756 01 40 Fax: +54 (0)11 4762 6862 mannesmann@mannesmannsaic.com.ar	Argentina - Argentinien NAKASE Servicio Tecnico CNC Calle 49, No. 5764/66 RA - 1653 Villa Balester Prov. - Buenos Aires Tel.: +54 (0) 11 4768 36 43 Fax: +54 (0) 11 4768 24 13 nakase@usa.net nakase@infovia.com.ar	Brazil - Brasilien Mannesmann Rexroth Automação Ltda. Divisão Rexroth Indramat Rua Georg Rexroth, 609 Vila Padre Anchieta BR - 09951-270 Diadema-SP [Caixa Postal 377] [BR-09901-970 Diadema-SP] Tel.: +55 (0)11 4075 90 60 +55 (0)11 4075 90 70 Fax: +55 (0)11 4075 90 50 awittwer@rexroth.com.br	Brazil - Brasilien Mannesmann Rexroth Automação Ltda. Divisão Rexroth Indramat R. Dr.Humberto Pinheiro Vieira, 100 Distrito Industrial BR - 89220-390 Joinville - SC [Caixa Postal 1273] Tel./Fax: +55 (0)47 473 58 33 Mobil: +55 (0)47 974 66 45 prochnow@zaz.com.br
Mexico Mannesmann Rexroth Mexico S.A. de C.V. Calle Neptuno 72 Unidad Ind. Vallejo MEX - 07700 Mexico, D.F. 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			

Notizen - Notes

14 Revisions to this Document

Despite careful creation and proofreading of this document, we cannot guarantee that it is absolutely free of mistakes. It can also be possible that the most recent modifications and/or supplements of the product described here could not be included in the document. If you notice any incorrect or missing specifications in this description, or if you have any suggestions about improving this publication, do not hesitate to tell us about it on this form. Fax a copy of this form to the address below - and you will help us to keep this document up to date.

Thank you very much for your cooperation.



To:
Rexroth Indramat GmbH
+49 (0) 93 52/40-44 65

From:
Company: ________________
Dept.: ________________
Name: ________________

Details of Document:

Title: ______________________________

Type of Document: ______________________________

Where is the position of the document that you have problems with? (Chapter, Page, Fig., Table)

① Chapter:_____ Page:______ Fig.: ______ Table: ____

② Chapter:_____ Page:______ Fig.: ______ Table: ____

③ Chapter:_____ Page:______ Fig.: ______ Table: ____

What is the problem? (please describe exactly what is wrong with the document or what is missing in your opinion).

Notes

289539

Printed in Germany

Rexroth Indramat