Rexroth
Bosch Group

Rexroth Rho 4
DLL-Library

1070072176
Edition 07

Software manual

Title Rexroth Rho 4
DLL−Library

Type of Documentation Software manual

Document Typecode DOK-RHO*4*-DLL−BIBLIO*-PR07-EN-P

Purpose of Documentation The present manual informs about:

- Programming of the rho4 on library functions (Windows, BAPS)

Record of Revisions

Description	Release Date	Notes
DOK-RHO*4*-DLL−BIBLIO*-PR06-EN-P	10.2003	Valid from VO07
DOK-RHO*4*-DLL−BIBLIO*-PR07-EN-P	01.2005	Valid from VO08

Published by Bosch Rexroth AG
Postfach 11 62
D-64701 Erbach
Berliner Straße 25
D-64711 Erbach
Tel.: +49 (0) 60 62/78-0
Fax: +49 (0) 60 62/78-4 28
Abt.: BRC/ESH (KW)

Overview of all manuals

Overview of all manuals

Manual	Contents	
	6	Software
	7	File management
	8	Scope of the rho4.1 Software

Manual	Contents	
Control functions	2	Survey of special functions
	3	Accurate position switching
	4	Setting the machine position
	5	Calling operating system functions
	6	Parameterization of the belt characte-ristic
	7	Selecting a point-file
	8	Mirroring
	9	Belt type
	10	System date and time
	11	System counter
	12	WC main range
	13	Setting the belt counter
	14	Recording of reference path
	15	Flying measurement (rho4.1 only)
	16	MOVE_FILE
	17	Setting the block preparation
	18	Exception−Handling
	19	Belt counter current value
	20	Automatic velocity adjustment for PTP movements
	21	Belt-synchronous working area belt kind 4
	22	Current belt speed
	23	Changing the belt simulation speed
	24	General functions
	25	Process-oriented functions
	26	BAPS3 keywords
Machine parameters	2	General information
	3	Application of the machine parameters
	4	General system parameters
	5	Speeds
	6	Positions
	7	Kinematic parameters

Overview of all manuals

Manual	Contents	
	8	Measuring system parameters
	9	Belt parameters
	10	Drive parameters Servodyn-GC
	11	Drive parameter Servodyn-D
	12	Table of parameters

Manual	Contents	
BAPS3 Programming manual	2	Program structure
	3	Constants
	4	Variables
	5	Program control
	6	Value assignments and combinations
	7	Functions
	8	Movement statement
	9	Write/read functions
	10	BAPS3 keywords
BAPS3 Short description	2	Program structure
	3	Constants and variables
	4	Program structure
	5	Value assignments and combinations
	6	Standard functions
	7	Movements and speeds
	8	Belt synchronous
	9	Workspace limitation
	10	Write/read functions
	11	Special functions
	12	Library functions
	13	Fix files
	14	BAPS3 keywords
Signal descriptions	2	rho4 interface description
	3	Signal description of PCL inputs
	4	Signal description of PCL outputs
Status messages and warnings	2	rho4 status messages
	3	Warnings
	4	CANopen error codes
ROPS4/Online	2	General information
	3	Activation and functions of Online
	4	The function key box

Overview of all manuals

Manual	Contents	
	5	Function key assignment
	6	The marker box
	7	File ROPS4WIN.ini
	8	Selection of a file
	9	TCP/IP settings for ROPS4

Manual	Contents	
DLL library	2	Library functions
	3	Calling library functions in BAPS
	4	Block structure of the rho4.1
	5	Library server
	6	Application development
	7	rho4 library functions
	8	Variable access per DLL
PHG2000	2	Hand-held programming unit PHG2000
	3	PHG2000 system variables
	4	Selection of PHG functions
	5	Info function of the PHG
	6	Controlling the PHG2000 output
	7	Define/Teach
	8	SRCAN functions
	9	File and User Memory Functions
	10	File list
	11	Process info
	12	Restoring the PGH display
	13	Variable assignment of PHG keys
	14	Select point file and point name
	15	BDT editor
Connection conditions Rho 4.1/IPC 40.2	2	System Overview
	3	Security Functions
	4	Installation
	5	Eelectrical Connections
	6	Interface Ports & Connectors
	7	Display- and Operating Components
	8	Maintenance and Replacement
	9	Software
	10	Ordering Informations

Overview of all manuals

Manual	Contents	
DDE-Server	2	Introduction
	3	Hardware and Software
	4	Operation
	5	Items of Server 4
	6	Scope of function

Overview of all manuals

Notes:

Contents

page

Contents

Safety Instructions

1 Safety Instructions

Please read this manual before you startup the rho4.
Store this manual in a place to which all users have access at any time.

1.1 Intended use

This instruction manual presents a comprehensive set of instructions and information required for the standard operation of the described products. The described products are used for the purpose of operating with a robot control rho4.

The products described

- have been developed, manufactured, tested and documented in compliance with the safety standards. These products normally pose no danger to persons or property if they are used in accordance with the handling stipulations and safety notes prescribed for their configuration, mounting, and proper operation.
- comply with the requirements of
  - the EMC Directives (89/336/EEC, 93/68/EEC and 93/44/EEC)
  - the Low-Voltage Directive (73/23/EEC)
  - the harmonized standards EN 50081-2 and EN 50082-2
- are designed for operation in industrial environments, i.e.
  - no direct connection to public low-voltage power supply,
  - connection to the medium- or high-voltage system via a transformer.

  The following applies for application within a personal residence, in business areas, on retail premises or in a small-industry setting:
  - Installation in a control cabinet or housing with high shield attenuation.
  - Cables that exit the screened area must be provided with filtering or screening measures.
  - The user will be required to obtain a single operating license issued by the appropriate national authority or approval body. In Germany, this is the Federal Institute for Posts and Telecommunications, and/or its local branch offices.

☞ **This is a Class A device. In a residential area, this device may cause radio interference. In such case, the user may be required to introduce suitable countermeasures, and to bear the cost of the same.**

The faultless, safe functioning of the product requires proper transport, storage, erection and installation as well as careful operation.

Safety Instructions

1.2 Qualified personnel

The requirements as to qualified personnel depend on the qualification profiles described by ZVEI (central association of the electrical industry) and VDMA (association of German machine and plant builders) in:
Weiterbildung in der Automatisierungstechnik
edited by: ZVEI and VDMA
MaschinenbauVerlag
Postfach 71 08 64
D-60498 Frankfurt.

The present manual is designed for RC technicans. They need special knowledge on handling and programming robots.

Interventions in the hardware and software of our products, unless described otherwise in this manual, are reserved to specialized Rexroth personnel.

Tampering with the hardware or software, ignoring warning signs attached to the components, or non-compliance with the warning notes given in this manual may result in serious bodily injury or damage to property.

Only electrotechnicians as recognized under IEV 826-09-01 (modified) who are familiar with the contents of this manual may install and service the products described.

Such personnel are

- those who, being well trained and experienced in their field and familiar with the relevant norms, are able to analyze the jobs being carried out and recognize any hazards which may have arisen.
- those who have acquired the same amount of expert knowledge through years of experience that would normally be acquired through formal technical training.

With regard to the foregoing, please note our comprehensive range of training courses. Please visit our website at
http://www.boschrexroth.com
for the latest information concerning training courses, teachware and training systems. Personal information is available from our Didactic Center Erbach,
Telephone: (+49) (0) 60 62 78-600.

Safety Instructions

1.3 Safety markings on products

Warning of dangerous electrical voltage!

Warning of danger caused by batteries!

Electrostatically sensitive components!

Warning of hazardous light emissions (optical fiber cable emissions)!

Disconnect mains power before opening!

Lug for connecting PE conductor only!

Functional earthing or low-noise earth only!

Connection of shield conductor only

Safety Instructions

1.4 Safety instructions in this manual

DANGEROUS ELECTRICAL VOLTAGE
This symbol is used to warn of a **dangerous electrical voltage.** The failure to observe the instructions in this manual in whole or in part may result in **personal injury**.

DANGER
This symbol is used wherever insufficient or lacking compliance with instructions may result in **personal injury**.

CAUTION
This symbol is used wherever insufficient or lacking compliance with instructions may result in **damage to equipment or data files**.

☞ This symbol is used to draw the user’s attention to special circumstances.

★ This symbol is used if user activities are required.

Safety Instructions

1.5 Safety instructions for the described product

DANGER
Danger of life through inadequate EMERGENCY-STOP devices!
EMERGENCY-STOP devices must be active and within reach in all system modes. Releasing an EMERGENCY-STOP device must not result in an uncontrolled restart of the system!
First check the EMERGENCY-STOP circuit, then switch the system on!

DANGER
Danger for persons and equipment!
Test every new program before starting up a system!

DANGER
Retrofits or modifications may adversely affect the safety of the products described!
The consequences may include severe injury, damage to equipment, or environmental hazards. Possible retrofits or modifications to the system using third-party equipment therefore have to be approved by Rexroth.

DANGER
Do not look directly into the LEDs in the optical fiber connection. Due to their high output, this may result in eye injuries.
When the inverter is switched on, do not look into the LED or the open end of a short connected lead.

DANGEROUS ELECTRICAL VOLTAGE
Unless described otherwise, maintenance works must be performed on inactive systems! The system must be protected against unauthorized or accidental reclosing.

Measuring or test activities on the live system are reserved to qualified electrical personnel!

Safety Instructions

CAUTION
Danger to the module!
Do not insert or remove the module while the controller is switched ON! This may destroy the module. Prior to inserting or removing the module, switch OFF or remove the power supply module of the controller, external power supply and signal voltage!

CAUTION
use only spare parts approved by Rexroth!

CAUTION
Danger to the module!
All ESD protection measures must be observed when using the module! Prevent electrostatic discharges!

The following protective measures must be observed for modules and components sensitive to electrostatic discharge (ESD)!

- Personnel responsible for storage, transport, and handling must have training in ESD protection.
- ESD-sensitive components must be stored and transported in the prescribed protective packaging.
- ESD-sensitive components may only be handled at special ESD-workplaces.
- Personnel, working surfaces, as well as all equipment and tools which may come into contact with ESD-sensitive components must have the same potential (e.g. by grounding).
- Wear an approved grounding bracelet. The grounding bracelet must be connected with the working surface through a cable with an integrated 1 MΩ resistor.
- ESD-sensitive components may by no means come into contact with chargeable objects, including most plastic materials.
- When ESD-sensitive components are installed in or removed from equipment, the equipment must be de-energized.

Safety Instructions

1.6 Documentation, software release and trademarks

Documentation

The present manual provides information about programming of the rho4 on DLL library functions.

Overview of available documentation	Part no.	
	German	English
Rho 4.0 Connectivity Manual	1070 072 364	1070 072 365
Rho 4.0 System description	1070 072 366	1070 072 367
Rho 4.1/IPC 40.2 Connectivity Manual	R911308219	R911308220
Rho 4.1/BT155, Rho 4.1/BT155T, Rho 4.1/BT205 Connectivity manual	1070 072 362	1070 072 363
Rho 4.1, Rho 4.1/IPC300 Connectivity manual	1070 072 360	1070 072361
Control panels BF2xxT/BF3xxT, connection	1070 073 814	1070 073 824
Rho 4.1 System description	1070 072 434	1070 072 185
ROPS4/Online	1070 072 423	1070 072 180
BAPS plus	1070 072 422	1070 072 187
BAPS3 Short description	1070 072 412	1070 072 177
BAPS3 Programming manual	1070 072 413	1070 072 178
Control functions	1070 072 420	1070 072 179
Signal descriptions	1070 072 415	1070 072 182
Status messages and warnings	1070 072 417	1070 072 181
Machine parameters	1070 072 414	1070 072 175
PHG2000	1070 072 421	1070 072 183
DDE-Server 4	1070 072 433	1070 072 184
DLL-Library	1070 072 418	1070 072 176
Rho 4 available documentation on CD ROM	1070 086 145	1070 086 145

☞ **In this manual the floppy disk drive always uses drive letter A:, and the hard disk drive always uses drive letter C:.**

Special keys or key combinations are shown enclosed in pointed brackets:

- Named keys: e.g., <Enter>, <PgUp>, <Del>
- Key combinations (pressed simultaneously): e.g., <Ctrl> + <PgUp>

Safety Instructions

Release

☞ **This manual refers to the following versions:**
Hardware version: rho4
Software release: ROPS4

Trademarks

All trademarks of software installed on Rexroth products upon delivery are the property of the respective manufacturer.

Upon delivery, all installed software is copyright-protected. The software may only be reproduced with the approval of Rexroth or in accordance with the license agreement of the respective manufacturer.

MS-DOS® and Windows™ are registered trademarks of Microsoft Corporation.

PROFIBUS® is a registered trademark of the PROFIBUS Nutzerorganisation e.V. (user organization).

MOBY® is a registered trademark of Siemens AG.

AS-I® is a registered trademark of AS-International Association.

SERCOS interface™ is a registered trademark of Interessengemeinschaft SERCOS interface e.V. (Joint VDW/ZVEI Working Committee).

INTERBUS-S® is a registered trade mark of Phoenix Contact.

DeviceNet® is a registered trade mark (TM) of ODVA (Open DeviceNet Vendor Association, Inc.).

2 Library functions

In the rho4 library functions, C-functions are made available for the OEM. They serve the communication with the basic operating system and permit access to operating system variables. As it is obvious from the block structure of the rh04, see chapter 4, these functions supply the connecting link between OEM programs and the operating system core of the rho4.1. For the integration of the OEM, the function library rho4fkt.lib is available in the control.

The library rho4fkt.lib consists of different files. The files contain each a function group for a specific topic as e.g. movement or information on kinematics.

rho4 library functions can be called in BAPS programs. The integration in BAPS is described in chapter 3.

Most of the these library functions can be called from BAPS with rho4fkt.lib as well as under Windows with rho4fkt.dll.

☞ **Functions which can be only called under Windows contain the comment 'Can only be called as Windows-DLL!'.**

☞ **Functions which can be only called in BAPS contain the comment 'Can only be called in BAPS!'.**

The responsibility for the correct call and the correct parameter supply is with the OEM. The rho4 library functions are not controlled by the control core. There is no plausibility check. Especially, the interactions with other processes resp. travel instructions can lead to illegal system conditions.

Library functions

Notes:

3 Calling library functions in BAPS

Each function of the rho4 library receives a function name of its own and a function-specific record type for the parameters to be transferred. The function names and record types are available for the user in different include files *.inc sorted in groups. The names of the include files are contained in the table of the library functions.

The defined names are exchangeable. The OEM can use its own function names instead of the preset names by changing the function declaration in the corresponding include file *.inc.

Function declaration

The declaration of the rho4 library functions is carried out, similar to the special functions, via a function declaration part called RHO_FKT.

```
RHO_FKT:<fct_number>=<fct_name> (<record_type>: <record_parameter>)
```

Example:

```
RHO_FKT:2031=rKGAxPos(TKGAxPos: PKGAxPos)
```

All functions of the rho4 library are characterized by a fix associated function number. The function name and the name of the record parameter can be changed.

☞ **The record type is strictly prescribed and must not be changed by the OEM**

Function call in the BAPS program

An rho4 library function call is carried out by the following line in the BAPS program

```
<return_code>=<fct_name> (<record_variable>)
```

Example:

```
Return_Code=rKGAxPos (PKGAxPos)
```

Each function supplies a return code which contains error messages and warnings. The return code is a variable of the type INTEGER that is to be defined in the user program. The function delivers 0 as return code in case of an error-free processing.

Calling library functions in BAPS

As error messages and warnings, three types of return codes are possible:

- Codes from 0 to −4999 are return messages of the rho4 library function. A description of the possible return codes can be found in the according include file *.inc.
- Codes from −5000 to −5999 are error messages of the TCP/IP communication. A description of these codes can be found in the header file rT.h.
- Codes higher than 10000 are error messages of the MS-C-Library. A description of these codes can be found in the header file rmain.h.

The transfer of the function parameters is effected as BAPS record variable. The declaration of the record variable must take place in the user program. As type, only the record type of the function from the associated include file *.inc can be used. This file also contains an exact description of the individual components of the record types.

Example record type:

```
TYP: TKGAxPos=RECORD

   .

   <function specific parameters>

   .

RECORD_END
```

An include file contains all record types of a function group.

BAPS example

The rho4 function rKGAxPos supplies all axis positions of a kinematic. The call in the BAPS program is described in the table below.

```
PROGRAM example

;;INCLUDE rmain.inc                ;Contains general constants and types of the
                                   ;rho4

;;INCLUDE rK.inc                   ;Contains constants and record types of the
                                   ;rho4 functions

INTEGER: return_code

TKGAxPos: PKGAxPos                 ;Declaration of the record variables TKGAxPos
                                   ;is contained in rK.inc

BEGIN

   PKGAxPos.ChannelId=0            ;must be in BAPS = 0

   PKGAxPos.KinNr=1                ;Kinematic 1
```

Calling library functions in BAPS

```
   PKGAxPos.Coordsys = WC_COORD_SYS ;World coordinates

   return_code=rKGAxPos(PKGAxPos)   ;rho4 library function call rKGAxPos is
                                    ;declared in rK.inc

   IF return_code<>0 THEN WRITE PHG,'error'

                     ELSE WRITE PHG,
                          PKGAxPos.KinAxPos[1], PKGAxPos.KinAxPos[2]
                                        ;PHG output of the two first axis
                                        ;position values of kinematic 1

PROGRAM_END
```

In order to declare the transfer parameter PKGAxPos, the program requires the BAPS type TKGAxPos. It is contained in the include file rK.inc. rK.inc also contains the function declaration of the rho4 function rKGAxPos as well as the possible return codes.

The function call with request of the return code can also be programmed in a simplified way. The functionality is the same as that in the example above. Here, it is only a simplified way of writing.

IF rKGAxPos (PKGAxPos) < 0

THEN <error treatment>

For functions which send back several different error messages or warnings, the call can also be carried out in a CASE-instruction.

The meaning of the return codes is contained in the corresponding include files *.inc.

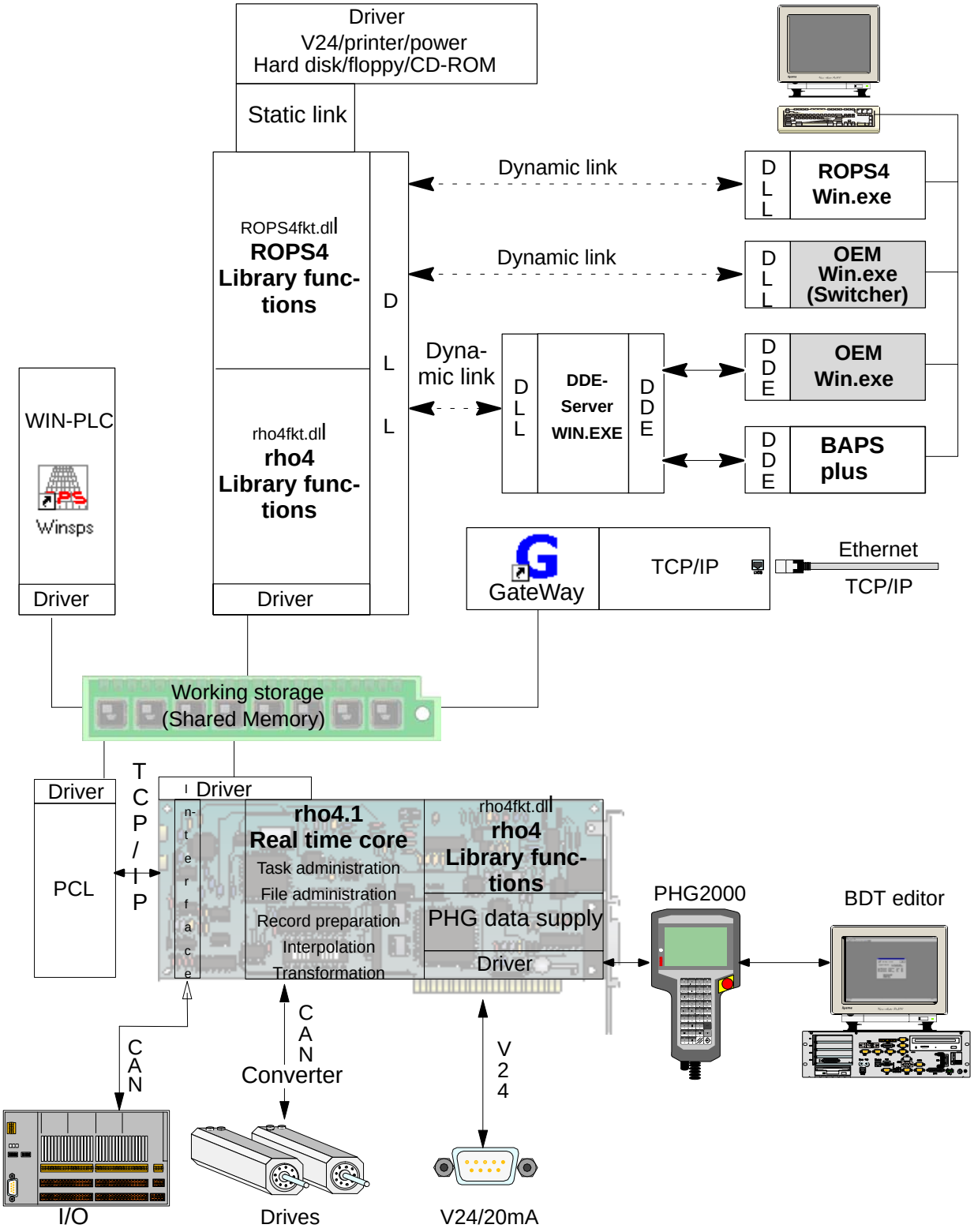
```
PROGRAM Pcase

;;INCLUDE rmain.inc                ;Contains general constants and types of the
                                   ;rho4

;;INCLUDE rk.inc                   ;Contains constants and record types of the
                                   ;rho4 functions

CONST:

   wc_kin_nr=1

   wc_coord_sys=-3

TKGAxPos: PKGAxPos                 ;Declaration of the record variables TKGAxPos
                                   ;is contained in rK.inc

BEGIN

   PKGAxPos.ChannelId=0            ;must be in BAPS = 0

   READ PHG, PKGAxPos.KinNr        ;Read in number of kinematic

   READ PHG, PKGAxPos.Coordsys     ;Read in coordinate system
```

Calling library functions in BAPS

```
   CASE rKGAxPos(PKGAxPos) EQUAL 0: ;rho4 library function call rKGAxPos is
                                    ;declared in rk.inc

      BEGIN

         WRITE PHG, PKGAxPos.KinAxPos[1],PKGAxPos.KinAxPos[2]
                                   ;Normal case PKGAxPos has been occupied in
                                   ;RHO_FKT correctly PHG output of the two first
                                   ;axis position values of kinematic 1

      END

      EQUAL wc_kin_nr: WRITE PHG,'invalid number of kinematic'
                                   ;<error treatment>

      EQUAL wc_coord_sys: WRITE PHG,'invalid coordinate system'
                                   ;<error treatment>

      DEFAULT WRITE PHG,'unknown return code'
                                   ;function returns unknown return code

   CASE_END

PROGRAM_END
```

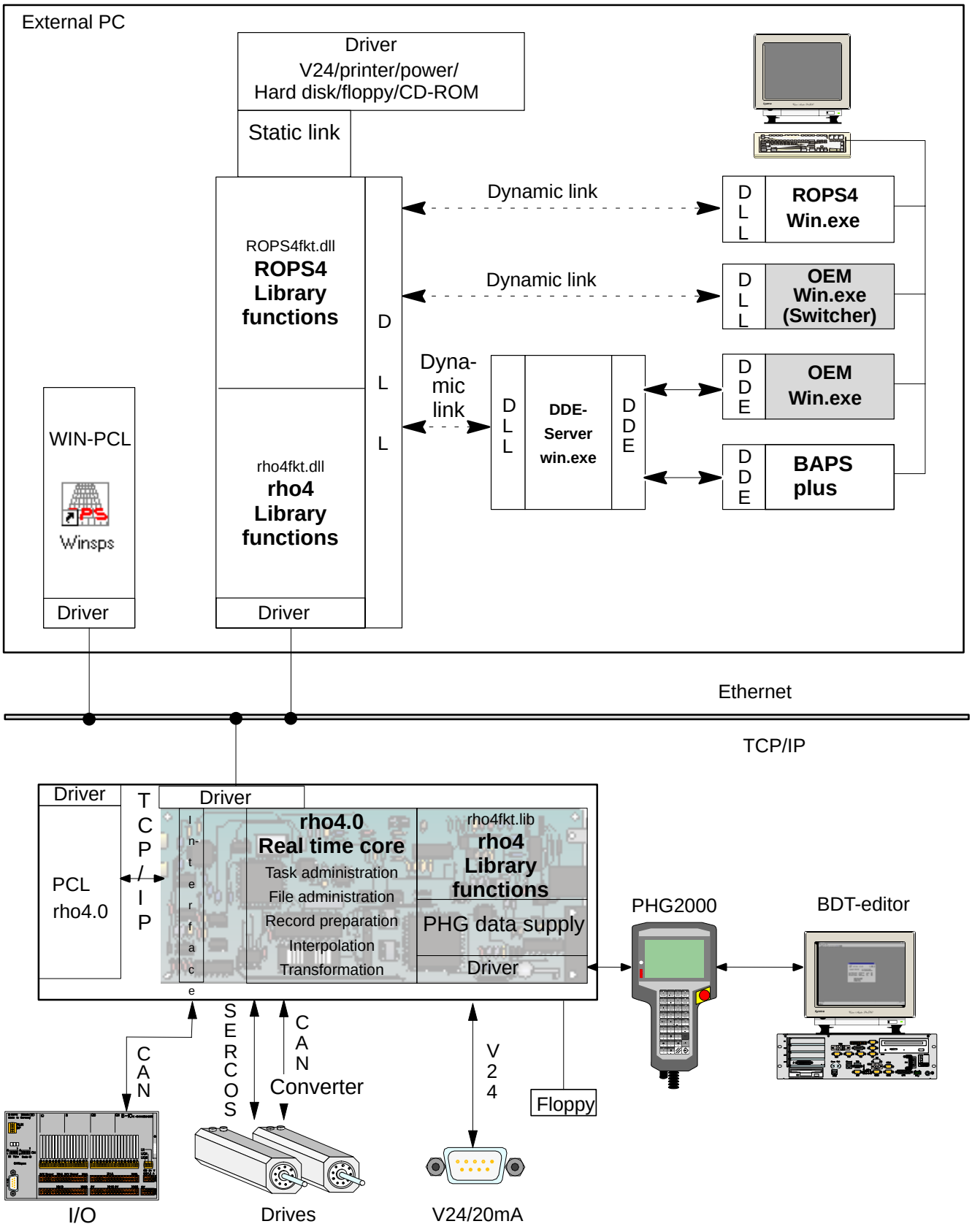
4 Block structure

4.1 Block structure of the rho4.1



Block structure

4.2 Block structure of the rho4.0



5 Library Server

The rho4 is equipped with four Servers for the library functions called rho_Funktionen_1 to rho_Funktionen_4. All Servers have the same scope of function and can be active simultaneously.

For example, Server1 (rho_Funktionen_1) approaches the reference points, while Server2 (rho_Funktionen_2) supplies the axis positions of the moved kinematics. Simultaneously, Server3 supplies the system statuses of the rho4 (rho_Funktionen_3).

An active Server is locked for other function calls during service processing. For example, the referencing can take several seconds. During this time, the Server is locked for other function calls.

The four Servers rho_Funktionen_1 to rho_Funktionen_4 are distinguished by their port numbers.

☞ **Also see documentation of rho4 system description.**

Library Server

Notes:

6 Application development

For the development of a Windows application or a BAPS program, include files are available.

The proto-typing of the functions and the declaration of the associated types are to be found in the files r*.h.

The BAPS record types, constants and parameters etc. are to be found in the files r*.inc.

The following name conventions have been defined:

r*.....	rho-name of the library functions
T*.....	Type of the library function (c-struct)
P*.....	Parameters of the library function
C*.....	Constants of the number of the library function
R*.....	Constants of the return codes of the library function

The language of the header-files (r*.h) is English. All tools as e.g. Compiler, Linker, Make or the library of the Developer-Studios are in English. The common use of the library functions and of these tools makes it necessary to use the English language.

The corresponding include files for the BAPS types r*_En.inc have a German and an English description.

The names of the BAPS record types, constants and parameters etc. are in English.

C/C++ applications

The include files with the extension .h are for C/C++ programs. The sequence of the Includes is as follows:

#include <windows.h>	is required for all Windows applications ☞ **Is only valid for C, not for C++**
#include “rimp.h”	Prototypes for Win-dll functions
#include “rmain.h”	rho4-library: main-include-file
#include “r*.h”	rho4-library: class-include-file

The library functions are joined into two .dll’s, the rho4Fkt.dll and the ROPS4Fkt.dll.

☞ **The scope of supply of the library functions includes an example program in C and three example programs in C++.**

Application development

BAPS programs

Files with the extension .inc are for BAPS programs. The sequence of the include files is as follows:

#include ”rmain.inc”	rho4-library: main-include-file
#include ”r*_En.inc”	rho4-library: class-include-file

7 rho4 library functions

In the following, you will find a short description after each function. The detailed documentation as well as the description of the corresponding parameters is contained in the include files.

The library functions are assembled in groups. All the function names of a group have the same prefix.

In the group movement e.g., all functions begin with the prefix rM as e.g. rhoMove.

Each function has got a number. The functions of a group have got the same group number.

The rho4 function library contains the following groups:

Group name	Abbreviation	Group number
Movement	[rM] (rhoMove)	1000
Kinematics info	[rK] (rhoKinematic)	2000
Process	[rP] (rhoProcess)	3000
System statuses	[rS] (rhoSystem)	4000
Errors / warnings	[rE] (rhoError)	5000
reserved		6000
Interface	[rI] (rhoInterface)	7000
File mangement	[rF] (rhoFile)	8000
Machine parameters	[rMPG] (rhoMachine-ParametersGet)	9000
Machine parameters	[rMPS] (rhoMachine-ParameterSet)	10000
TCP Server / Client	[rT] (rhoTCP)	11000
reserved		12000
reserved		13000
Archive	[rA] (rhoArchive)	14000
reserved		15000
Extended machine parameter	[rMxG] (rho machine-Parameter Get)	19000
Extended machine parameter	[rMxS] (rho machine-ParameterSet)	20000

rho4 library functions

7.1 Movement rhoMove [rM] 1000

☞ **Can only be called as Windows-dll !**

The group contains functions for moving the axes. For the integration of the .dll under Windows, the file rM.h is available.

☞ **The functionality of these routines is already contained in the BAPS language scope. The function rMAxPos corresponds to the BAPS function MOVE, the function rMRefPAx corresponds to the BAPS function REF_PNT. Therefore, the functions of this group are not available in BAPS.**

DANGER
Interactions with other processes or move instructions can lead to illegal system statuses.

The function names are composed of different abbreviations. The following abbreviations mean:

r	rho	M	Move
Kin	Kinematic	Pos	Position
Ax	Axis	Inc	Increments
RefP	Reference point	All	All

Function name	Ft. No.	Comment
rMKinPos	1000	moves the kinematic to the indicated point
rMAxPos()	1001	moves the axis to the indicated position
rMKinInc()	1010	moves the kinematic by the number of increments
rMAxInc()	1011	moves the axis by the number of increments
rMRefPAllKin	1020	references all kinematics
rMRefPKin	1021	references the indicated kinematic
rMRefPAx	1022	references the indicated axes

rho4 library functions

7.2 Kinematics info rhoKin [rK] 2000

The functions of this group supply information on the kinematics, on belt and axis data as well as on the WC-system and the currently used tool. The required record types for BAPS and a detailed documentation are contained in the include file rK_En.inc. For the integration of the .dll under Windows, the file rK.h is available.

The function number is necessary for the declaration in BAPS.

The function names are composed of different abbreviations. The following abbreviations mean:

r	rho	S	Set
G	Get	Ax	Axes
All	All	Blt	Belt
Pos	Position	Nam	Name
EPos	End position	IPos	In-position-flag
Lag	Lag	Sgl	Single
K	Kinematic		

rho4 library functions

Function name	Ft. No.	Comment
rKGAllInfo()	2000	supplies the data of all kinematics
rKGInfo()	2010	supplies the data of one kinematic
rKxGAxData()	2020	supplies all axis, belt, tool, and WC-system data of one kinematic, extended by the reference-flags of the kinematics
rKGAxData()	2030	supplies all axis, belt, tool, and WC-system data of one kinematic
rKGAxPos()	2031	supplies all axis positions of one kinematic
rKGAxNam()	2032	supplies all axis names of one kinematic
rKGAxEPos()	2033	supplies all end points of one kinematic
rKGAxIPos()	2034	supplies all inposition flags of one kinematic
rKGAxLag()	2035	supplies the lag of all axes of one kinematic
rKGTool	2036	supplies all tool data of one kinematic
rKGWcsystem	2037	supplies all data of the WC-system of one kinematic
rKSTool	2038	selection of a tool for one kinematic. Can only be called as Windows-DLL
rKGAxRefp()	2039	supplies all inposition-flags of a kinematics
rKGBltData	2040	supplies the data of all belts of one kinematic
rKGBltPos	2041	supplies the positions of all belts of one kinematic
rKGBltNam	2042	supplies the name of all belts of one kinematic

rho4 library functions

Function name	Ft. No.	Comment
rKGSglAxData()	2050	supplies all data of one axis
rKGSglAxPos()	2051	supplies the position of one axis
rKGSglAxNam()	2052	supplies the name of one axis
rKGSglAxEPos()	2053	supplies the end point of one axis
rKGSglAxIPos()	2054	supplies the inposition-flag of one axis
rKGSglAxLag()	2055	supplies the lag of one axis
rKGSglAxRefp()	2056	supplies the inposition-flag of an axis
rKGSglAxDat()	2057	supplies all axis-data, extended by the reference-flag of the axis
rKGSglBltDat()	2060	supplies all data of a belt
rKGSglBltPos()	2061	supplies the position of a belt
rKGSglBltNam()	2062	supplies the name of a belt
rKxGWcSystem()	2071	supplies name and coordinates of the active WC−System (workpiece−coord.−syst.)
rKSWcSystem()	2072	selects a WC−System for a kinematic
rKCWpFrame()	2073	calculates of 3 points the coordinates of the appropriate workpiece−coord.−system)
rKTrfLocOri()	2074	transforms a point from the local WC−System to the OC (original coordinate system)
rKTrfOriLoc()	2075	transforms a point from the OC (original coordinate system) to the local WC (World coordinate system)

rho4 library functions

7.3 Process rhoProcess [rP] 3000

The functions of this group supply information on the running processes. Equally contained are functions for the control of processes.

The required record types for BAPS as well as a detailed documentation are contained in the include file rP_En.inc.

For the integration of the .dll under Windows, the file rP.h is available.

The function number is required for the declaration in BAPS.

The function names are composed of different abbreviations. The following abbreviations mean:

r	rho	P	Process
G	Get	All	All
To	Total	Prio	Priority
Sta	Status		

Function name	**Ft. No.**	**Comment**
rPGTotSta()	3001	supplies all information for a process
rPGTyp()	3002	supplies the process type permanent, normal, etc.
rPGPrio()	3003	supplies the process priority
rPGSta()	3004	supplies the process status ready, wait, etc.
rPGError()	3005	supplies the process error
rPGActLin()	3006	supplies the current qll-line of the process
rPGActIncLin()	3007	supplies the current Include file-qll-line number
rPGActKinNo()	3008	supplies the last occupied kinematic of the process
rPGQuaSub()	3009	supplies the number of subprocesses of the process
rPGExtName()	3010	supplies the name of the external program
rPGExtLevel()	3011	supplies the level depth of the external program
rPGExtdata()	3012	supplies the names and level depths of the external programs

rho4 library functions

Function name	Ft. No.	Comment
rPList()	3020	supplies the entire process list as momentary record, can only be called as Windows-dll.
rPListFirst()	3021	initializes the process list, delivers back the number of all processes
rPListNext()	3022	supplies names, current qll-line and status of a process at each call
rPListExit()	3023	enables the process list created before
rPFirstProc()	3030	supplies all data of the first process
rPNextProc()	3031	supplies all data of the next process
rPStop()	3040	stops the indicated process, can only be called as Windows−dll.
rPSelect()	3050	selects the indicated process, can only be called as Windows−dll.
rPDebug()	3100	Debugging a rho4−process, can only be called as Windows−dll.

rho4 library functions

7.4 System statuses rhoSystem [rS] 4000

The group contains functions for reading and setting system statuses. The required record types for BAPS as well as a detailed documentation are contained in the include file rS_En.inc.

For the integration of the .dll under Windows, the file rS.h is available. The function number is required for the declaration in BAPS.

DANGER
Interactions with other processes can lead to illegal system statuses.

The function names are composed of different abbreviations. The following abbreviations mean:

r	rho	S	System
G	Get	S	Set
Hw	Hardware	Fac	Factor
x	extended		

Function name	Ft. No.	Comment
rSGAVDFac()	4030	supplies the global A-factor / V-factor / D-factor
rSGAFac()	4031	supplies the global A-factor
rSGVFac()	4032	supplies the global V-factor
rSGDFac()	4033	supplies the global D-factor
rSSAVDFac()	4040	sets the global A-factor / V-factor / D-factor
rSSAFac()	4041	sets the global A-factor
rSSVFac()	4042	sets the global V-factor
rSSDFac()	4043	sets the global D-factor
rSGrho4Fkt()	4050	supplies the version and a list of all functions of the rho4-library
rSGrho4Dll()	4051	supplies the version and a list of all functions of the Windows-library (only Win-dll)
rSxGrho4Fkt()	4055	supplies the version and a list of all functions of the rho4 library
rSxGrho4Dll()	4056	supplies the version and a list of all functions of the Windows-library (only Win-dll)
rSGSrCAN()	4060	supplies the data from the SRCAN-module
rSGSerialNb()	4061	supplies the serial number stored in the SRCAN-module

rho4 library functions

Function name	Ft. No.	Comment
rSSSrCAN()	4070	writes data into the SRCAN-module
rSSSerialNb()	4071	writes the serial number stored in the machine parameter P314 into the SRCAN-module
rSGVersion()	4080	supplies information on the control version
rSRestart()	4090	re-starts the control
rSShutDown()	4091	force the rho4 to shutdown
rSGServDa()	4104	data transfer control −−> DSS
rSSServDa()	4105	data transfer DSS −−> control
rSGTimeDay()	4110	supplies the time and day (only Win-dll)
rSSTimeDay()	4111	set the time and day (only Win-dll)
rSGAxisPar()	4120	data transfer axis−control −−> BAPS
rSSAxisPar()	4121	data transfer BAPS −−> axis−control
rSStateUSMEM()	4200	determines the archive status of the user memory
rSSaveUSMEM()	4201	saves the user memory
rSGPHGBuf()	4300	supplies the contents of the PHG display (only Win-dll)
rSGHwConfig()	4310	supplies the rho4 hardware configuration
rSPclversion()	4311	supplies the rho4 PLC−version
rSGPclState()	4312	supplies the rho4 PLC−state

rho4 library functions

7.5 Errors/warnings rhoError [rE] 5000

These functions supply the errors and warnings of the rho4 existing at the moment.

The required record types for BAPS as well as a detailed documentation are contained in the include file rE_En.inc.

For the integration of the .dll under Windows, the file rE.h is available.

The function number is required for the declaration in BAPS.

The function names are composed of different abbreviations. The following abbreviations mean:

r	rho	E	Error
G	Get	All	All

Function name	**Ft. No.**	**Comment**
rEGQuantity()	5010	supplies the number of errors and warnings
rEGSglInfo()	5020	supplies info for errors or warnings including kinematic/axis reference of the current number
rEGSupInfo()	5021	generates, updates and deletes (optional) a DLL−image of the rho4 message buffer and/or supplies state messages of the drives as well as extended infos, can only be called as Windows-dll
rEReset()	5030	clears all errors and warnings without a reset at the interface
rEGErrText()	5040	converts the Return-Code from a rho−function to an ASCII−Text, can only be called as Windows-dll

rho4 library functions

7.6 PLC interface rhoInterface [rI] 7000

These functions are determined for reading PLC outputs, and setting and reading PLC inputs.

The required record types for BAPS as well as a detailed documentation are contained in the include file rI_En.inc.
For the integration of the .dll under Windows, the file rI.h is available.
The function number is required for the declaration in BAPS.

For the function of setting PLC inputs (RC-outputs), the following fields are allowed:

E8.0 to E15.7	PHG2000-outputs
E36.0 to E37.7	RC outputs IO logic
E72.0 to E79.7	RC outputs Library function 7000
E114.0 to E175.7	RC outputs Byte / user outputs
E176.0 to E207.7	System communication outputs

DANGER
Interactions with other processes or PLC functions can lead to illegal system statuses.

The function names are composed of different abbreviations. The following abbreviations mean:

r	rho	I	Interface
G	Get	S	Set
n	Number	O	Outputs
I	Inputs		

rho4 library functions

Function name	**Ft. No.**	**Comment**
rIGnOByt()	7000	reads several PLC-output-bytes
rIGOByt()	7001	reads one PLC-output-byte
rIGOBit()	7002	reads one PLC-output (bit)
rIGnIByt()	7010	reads several PLC-input-bytes
rIGIByt()	7011	reads one PLC-input-byte
rIGIBit()	7012	reads one PLC-input (bit)
rISnlByt()	7020	setting of several PLC-input-bytes (RC-output-bytes)
rISlByt()	7021	setting of one PLC-input-bytes (RC-output-bytes)
rISlBit()	7022	setting of one PLC-input-bit (RC-output-bit)

rho4 library functions

7.7 File system rhoFile [rF] 8000

The class 8000 (rhoFile [rF]) of the rho4 library functions contains functions for the file handling:

Function name	FctNo.	Comment
rFCopy()	8010	Copying of files
rFxCopy()	8015	Deleting of files
rFRemove()	8021	Copying of files (incl. Remote-PC)
rFRename()	8022	Renaming of files
rFxRemove()	8025	Deleting of files (incl. Remote-PC)
rFxRename()	8026	Renaming of files (incl. Remote-PC)
rFReadBlk()	8030	Reads a block of n bytes from a rho4−file into a buffer, can only be called as Windows-dll.
rFDir()	8040	File-List of the rho4-files in user memory
rFStat()	8050	Reading of file status
rFChmod()	8051	Modifying of file access rights
rFxStat()	8055	Reading of file status (incl. Remote-PC)
rFMpFloppy()	8070	Copies the machine parameter file of the rho4.0 to the rho40-Fds (Floppy Disc)
rFxMpFloppy()	8071	Copies the extended machine parameter file of the rho4.0 to the rho40-Fds (Floppy Disc)
rFMem()	8080	Suplies the current state of the rho4-user−memory: memory size, used memory, free memory, number of files

The functions can be called both from BAPS and under WINDOWS (rho4fkt.dll). The constant and type definitions are to be read in the header files rf_En.inc or rf.h.

The rF−functions 8010, 8021, 8022, 8050 are kept for compatibility reasons. They can only be used with the rho4.1 when additionally the file handling exclusively takes place between rho4.1 user memory and the 'c:' of the rho4.1-PC. The OEM-Windows application must run on the rho4.1-Windows part.

The rFx functions 8015, 8025, 8026, 8055 contain the scope of the rF-functions and have extended functionalities for the cases rho4.0; File handling Remote-PC <−> user memory or user memory <−> user memory. They have compared with the rf functions a call list extended with portnumber and IP address (see detailed description in the header files rf_En.inc or rf.h).

☞ **It is recommended to use the rFx functions in general from the software version VO04H.**

For the file handling between the rho4-user memory and a remote PC, a rho4Fkt server is required on the remote Pc with corresponding Ini file and a rho4fkt-Dll. These three files are part of the rho delivery scope.

They are deposited during the software installation automatically in the Bosch structure, for the rho4.1 with the rho4 software installation and for the rho4.0 with the ROPS installation:

c:\Bosch\rho4\winexe\rho4FktSvr.exe (rho4Fkt-Server)

c:\Bosch\rho4\origin\rho4FktSvr.ini (Initialization file)

c:\Bosch\rho4\rho4fkt\ rho4fkt.dll

Server and Ini files can be deposited there in any subdirectory (but both in the same one) on the remote PC. The Dll must be copied into the directory ’c:\winnt\system32’ of the remote PC.

For file access from the Windows world to the rho4, two channels are basically required:

(1) for the call: a channel of the rho4Fkt.dll (e. g. rho_Functions_3 (PortNo. 6093))

(2) for the execution: a WinServer (e .g. Win_Server_3, PortNo. 6003)

To avoid access conflicts, the whole configuration is to be designed in such a way that always a Win-Server corresponds to one channel of the rho4Fkt.dll (e.g. PortNo. 6002 to 6092).

In the rho4.0, a file access via rhoFile library functions is possible from BAPS or a remote PC. Files of the user memory and the rho4.0-Fds (Floppy-disk) can be treated. In the rho4.0, the IP address of the remote PC contained in the rFx function transfer parameters must be programmed.

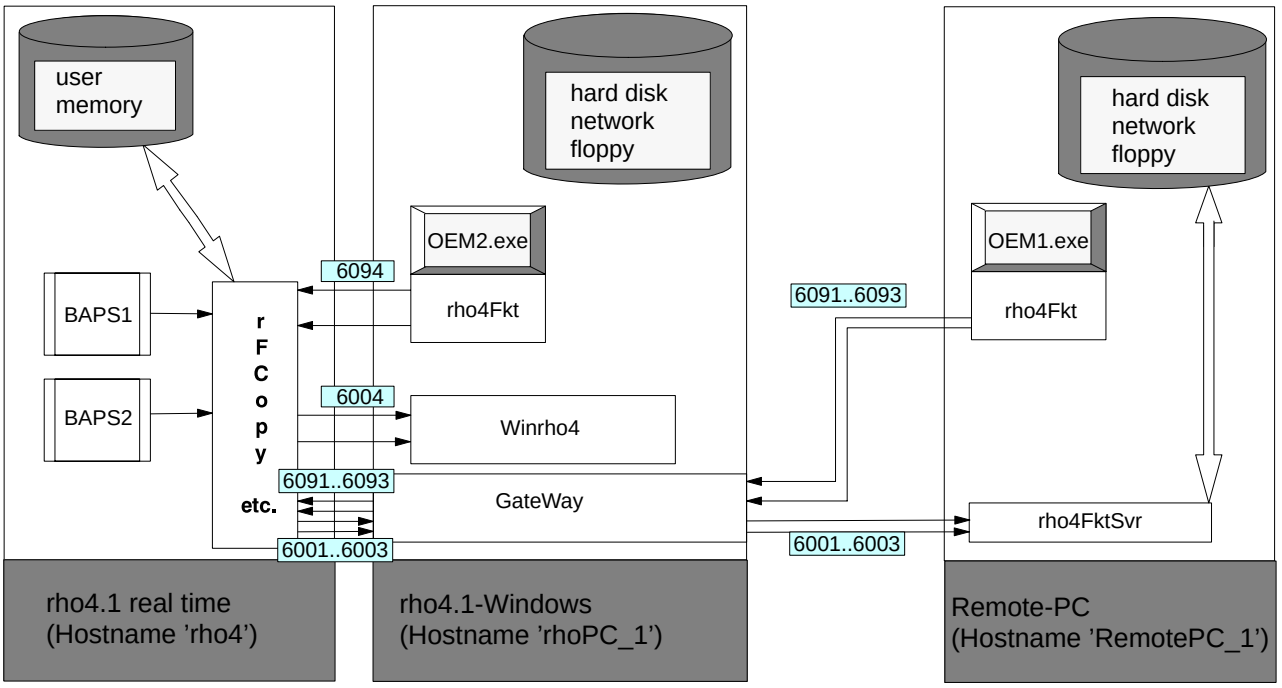
In the rho4.1, a file access via rhoFile library functions is possible from BAPS, from the rho4.1 Windows part or a remote PC.

In the rho4.1, the ServerPortNo contained in the rFx function transfer parameters must be programmed. It is the port number of the server (Win-Server or Rho4Fkt server) which serves the rFx function output. The physical position of the target file is then fixed via the files ’Gateway.ini’, ’Whinrho4.ini’ and ’Rho4FktSvr.ini’.

rho4 library functions

7.7.1 Example rho4.1

With Remote-PC



To make things clear, the example from the figure in chapter 3 is explained in detail. The hardware/software configuration is the following:

(1) Remote-PC with

(a) rho4FktSvr.exe

(b) rho4Fkt.dll

(c) Windows application 'OEM1.exe' (switcher) calling the rhoFile functions.

(2) rho4.1 with

(a) Winrho4.exe

(b) rhoFkt.dll

(c) Windows application 'OEM2.exe' (switcher) calling the rhoFile functions

(d) Gateway.exe

OEM1.exe works on the control with the servers rho_Functions_1, rho_Functions_2 and rho_Functions_3 (PortNo. 6091, 6092, 6093). OEM2.exe works on the control with the server rho_Functions_4 (PortNo. 6094).

Via the system channels Win_Server_1, Win_Server_2, Win_Server_3 (ServerPortNo 6001, 6002, 6003) the files are exchanged between rho4.1 user memory and the remote PC (e. g. drive c: of the remote PC).

Via the system channel Win_Server_4 (ServerPortNo 6004) the files are exchanged between rho4.1 user memory and the rho4.1 Windows part (e.g. drive c: of the rho4.1).

☞ **By calling the rF−functions, the ServerPortNo 6001 to 6004 must be programmed as transfer parameter.**

Compare also the example application switcher.exe supplied in the delivery (including source code, specially swi_rf.c). By calling the function rTClientCon() for the initialization of the connection the hostname of the server PC must be entered there. In the example, this is for OEM1.exe the hostname of the rho4.1-Windows page (rhoPC_1) and for OEM2.exe the Hostname of the rho4 real time part (rho4).

For this example the following Ini-files must be configured:

- Gateway.ini
- Winrho4.ini
- Rho4FktSvr.ini

Gateway.ini

```
;; ----------------Initialization file for GateWay.exe ------------------
[GateWay]
                  ;; Version = VO04E
Version            =  408
                  ;; no connection
ConnectionNo       =  7
                  ;; Connection for ROPS4-Coupling
1                 =  Coupling
                  ;; Connection for rho4 library function
2                 =  rho_Function_1
3                 =  rho_Function_2
4                 =  rho_Function_3
                  ;; Connection for win-Server
5                 =  Win_Server_1
6                 =  Win_Server_2
7                 =  Win_Server_3
```

rho4 library functions

```
;;-- Section Coupling ----------------------------------------
[Coupling]
                 ;; Symbolic name for rho4 in the ”hosts” file
ServerAlias       = rho4
                ;; rho4 port number for  ROPS4-Coupling
ServerPortNo      = 6010
                ;; GateWay port number for ROPS4-Coupling
GateWayPortNo     = 6010
                ;; Messages length, don’t change this value
Msglen            = 256


;;-- Section rho_Function_1 ----------------------------------------
[rho_Function_1]
                 ;; Symbolic name for rho4 in the ”hosts” file
ServerAlias       = rho4
                 ;; rho4 port number for rho_Function_1
ServerPortNo      = 6091
                 ;; GateWay port number for rho_Function_1
GateWayPortNo     = 6091
                 ;; Messages length, don’t change this value
Msglen            = 512


;;-- Section rho_Function_2 ----------------------------------------
[rho_Function_2]
                   ;; Symbolic name for rho4 in the ”hosts” file
ServerAlias         = rho4
                   ;; rho4 port number for rho_Function_2
ServerPortNo        = 6092
                   ;; GateWay port number for rho_Function_2
GateWayPortNo       = 6092
                   ;; Messages length, don’t change this value
Msglen              = 512


;;-- Section rho_Function_3 ----------------------------------------
[rho_Function_3]
                   ;; Symbolic name for rho4 in the ”hosts” file
ServerAlias         = rho4
                   ;; rho4 port number for rho_Function_2
ServerPortNo        = 6093
                   ;; GateWay port number for rho_Function_2
GateWayPortNo       = 6093
                   ;; Messages length, don’t change this value
Msglen              = 512
```

rho4 library functions

```
;;-- Section Win_Server_1 ----------------------------------------
[Win_Server_1]
                    ;; Symbolic name for Remote-PC in the ”hosts” file
ServerAlias          = RemotePC_1
                    ;; rho4 port number for system server 1
ServerPortNo         = 6001
GateWayPortNo        = 6001
Msglen               = 4096


;;-- Section Win_Server_2 ----------------------------------------
[Win_Server_2]
                    ;; Symbolic name for Remote-PC in the ”hosts” file
ServerAlias          = RemotePC_1
                    ;; rho4 port number for system server 2
ServerPortNo         = 6002
GateWayPortNo        = 6002
Msglen               = 4096


;; ------------------------------------------------------------------
[Win_Server_3]
                    ;; Symbolic name for Remote-PC in the ”hosts” file
ServerAlias          = RemotePC_1
                    ;; rho4 port number for system server 3
ServerPortNo         = 6003
GateWayPortNo        = 6003
Msglen               = 4096
```

Winrho4.ini

```
;; ------------------------------------------------------------------
;; --------- Initialization file for Winrho4.exe ---------------------
;; All entries are in comment with the exception of ”[rho4Svr] Version =
;; 405”
;; Changes in the Winrho4.ini get only effective if the Winrho4.exe
;; is started again.
;; ------------------------------------------------------------------

[rho4Svr]
                  ;; VO04E
Version            = 408
                  ;; Number of servers
ServerNo           = 1
                  ;; Name of system servers
1                  = Win_Server_4
UPSControl         = 1
```

rho4 library functions

```
;; ---------------------------------------------------------------------
[Win_Server_4]
                ;; Symbolic name for rho4 in the ”hosts” file
ServerAlias      = rho4
                 ;; rho4 port number for system server 4
ServerPortNo     = 6004
;; ---------------------------------------------------------------------
      .
      .
      etc.
```

Rho4FktSvr.ini

```
;; ---------------------------------------------------------------------
;; --------- Initialization file for Rho4FktSvr.exe ------------------
;; Changes in the Rho4FktSvr.ini get only effective if the Rho4FktSvr.exe
;; is started again.
;; ---------------------------------------------------------------------
[rho4FktSvr]
                 ;; VO04E
Version           = 408


ServerNo             = 3
                 ;; Name of system servers
1                 = Win_Server_1
2                 = Win_Server_2
3                 = Win_Server_3

;; ---------------------------------------------------------------------
[Win_Server_1]
                 ;; rho4 port number for system server 1
ServerPortNo       = 6001
;; ---------------------------------------------------------------------
[Win_Server_2]
                 ;; rho4 port number for system server 2
ServerPortNo     = 6002
;; ---------------------------------------------------------------------
[Win_Server_3]
                 ;; rho4 port number for system server 3
ServerPortNo     = 6003
;; ---------------------------------------------------------------------
```

Programm examples

FCopy function calls by means of C- or BAPS program sections:

OEM2.EXE: copy file from drive c: of the rhoPC_1 −> user memory

rho4 library functions

```
PFxCopy.ChannelId    = rhoTCPClientConnect();            // connect to rho4
PFxCopy.ServerPortNo = 6004;                            // =rhoPC_1 by
                                                        ;// winrho4.ini
PFxCopy.IpAddr_RemPC = ”                                ”  ;// (32 blanks)
                                                           ;// redundant
                                                        ;// with rho4.1
strcpy(PFxCopy.SrcName, ”c:\\Sub1\\Test.dat”);           // name of
                                                        ;// source-file
strcpy(PFxCopy.DstName, ”TestAws.dat”);                  // name of
                                                        ;// dest.-file
PFxCopy.Overwrite    = 1;
lRet                 = rFxCopy (&PFxCopy);               // function call
rhoTCPClientDisconnect (PFxCopy.ChannelId);              // close channel
```

OEM2.EXE:
Copy file from user memory −> drive c: of the rhoPC_1

```
PFxCopy.ChannelId    = rhoTCPClientConnect();            // connect to rho4
PFxCopy.ServerPortNo = 6004;                            // =rhoPC_1 by
                                                        ;// winrho4.ini
PFxCopy.IpAddr_RemPC = ”                                ” ;// (32 blanks)
                                                        ;// redundant
                                                        ;// with rho4.1
strcpy(PFxCopy.SrcName, ”TestAws.dat”);                  // name of
                                                        ;// source-file
strcpy(PFxCopy.DstName, ”c:\\Sub1\\Test.dat”);           // name of
                                                        ;// dest.-file
PFxCopy.Overwrite    = 1;
lRet                 = rFxCopy (&PFxCopy);               // function call
rhoTCPClientDisconnect (PFxCopy.ChannelId);              // close channel
```

OEM1.EXE:
Copy file from drive c: of the RemotePC_1 −> user memory

```
PFxCopy.ChannelId     = rhoTCPClientConnect();           // connect to rho4
PFxCopy.ServerPortNo  = 6002;                           // =RemotePC_1 by
                                                        ;// gateway.ini
PFxCopy.IpAddr_RemPC = ”                                ” ;// (32 blanks)
                                                        ;// redundant
                                                        ;// with rho4.1
strcpy(PFxCopy.SrcName, ”c:\\Sub1\\Test.dat”);           // name of
                                                        ;// source-file
strcpy(PFxCopy.DstName, ”TestAws.dat”);                  // name of
                                                        ;// dest.-file
PFxCopy.Overwrite     = 1;
lRet                  = rFxCopy (&PFxCopy);              // function call
rhoTCPClientDisconnect (PFxCopy.ChannelId);              // close channel
```

OEM1.EXE:
Copy file from drive c: of the RemotePC_1 −> user memory

rho4 library functions

```
PFxCopy.ChannelId    = rhoTCPClientConnect();            // connect to rho4
PFxCopy.ServerPortNo = 6002;                            // =RemotePC_1 by
                                                       ;// gateway.ini
PFxCopy.IpAddr_RemPC = ”                                ” ;// (32 blanks)
                                                        ;// redundant
                                                        ;// with rho4.1
strcpy(PFxCopy.SrcName, ”c:\\Sub1\\Test.dat”);           // name of
                                                        ;// source-file
strcpy(PFxCopy.DstName, ”TestAws.dat”);                  // name of
                                                        ;// dest.-file
PFxCopy.Overwrite    = 1;
Ret = rFxCopy (&PFxCopy);                                // function call
rhoTCPClientDisconnect (PFxCopy.ChannelId);              // close channel
```

OEM1.EXE:
Copy file from user memory −> drive c: of the RemotePC_1

```
PFxCopy.ChannelId     = rhoTCPClientConnect();           // connect to rho4
PFxCopy.ServerPortNo  = 6002;                            // = RemotePC_1 by
                                                        ;// gateway.ini
PFxCopy.IpAddr_RemPC = ”                                ” ;// (32blanks)
                                                         ;// redundant
                                                         ;// with rho4.1
strcpy(PFxCopy.SrcName, ”TestAws.dat”);                  // name of
                                                        ;// source-file
strcpy(PFxCopy.DstName, ”c:\\Sub1\\Test.dat”);           // name of
                                                        ;// dest.-file
PFxCopy.Overwrite     = 1;
lRet                  = rFxCopy (&PFxCopy);              // function call
rhoTCPClientDisconnect (PFxCopy.ChannelId);              // close channel
```

rho4 library functions

OEM1.EXE:
Copy file from user memory −> user memory

```
PFxCopy.ChannelId     = rhoTCPClientConnect();           // connect to rho4
PFxCopy.ServerPortNo  = -1;                               // virtual server
PFxCopy.IpAddr_RemPC = ”                                ” ;// (32 blanks)
                                                          ;// redundant
                                                          ;// with rho4.1
strcpy(PFxCopy.SrcName, ”TestAws1.dat”);                   // name of
                                                          ;// source-file
strcpy(PFxCopy.DstName, ”TestAws2.dat”);                   // name of
                                                          ;// dest.-file
PFxCopy.Overwrite     = 1;
lRet = rFxCopy (&PFxCopy);                                 // function call
rhoTCPClientDisconnect (PFxCopy.ChannelId);                // close channel
```

OEM2.EXE:
Copy file from user memory −> user memory

```
PFxCopy.ChannelId        = rhoTCPClientConnect();        // connect to rho4
PFxCopy.ServerPortNo = -1;                               // virtual server
PFxCopy.IpAddr_RemPC = ”                                ” ;// (32 blanks)
                                                          ;// redundant
                                                          ;// with rho4.1
strcpy(PFxCopy.SrcName, ”TestAws1.dat”);                   // name of
                                                          ;// source-file
strcpy(PFxCopy.DstName, ”TestAws2.dat”);                   // name of
                                                          ;// dest.-file
PFxCopy.Overwrite        = 1;
lRet = rFxCopy (&PFxCopy);                                 // function call
rhoTCPClientDisconnect (PFxCopy.ChannelId);                // close channel
```

BAPS:
Copy file from user memory −> user memory

```
PFxCopy.ChannelId    = 0                                ; must be in BAPS=0
PFxCopy.ServerPortNo = -1                                 ; virtual server
PFxCopy.IpAddr_RemPC = ”                                 ”; (32 blanks) in
                                                          ; rho4.1 without
                                                          ; signification
PFxCopy.SrcName      = ’Test1.qll’                        ; Name of source
                                                          ; file
PFxCopy.DstName     = ’Test2.qll’                         ; Name of target
                                                          ; file
PFxCopy.Overwrite    = 1
FILL_STRING (PFxCopy.SrcName)              ; Fills up the field with BLANKs
FILL_STRING (PFxCopy.DstName)              ; Fills up the field with BLANKs
RetCode = rFxCopy (PFxCopy)               ; Call of the rho4fkt
```

BAPS:
Copy file from user memory −> drive c: of the rhoPC_1

rho4 library functions

```
PFxCopy.ChannelId    = 0                                  ; must be in BAPS=0
PFxCopy.ServerPortNo = 6004                                ; rhoPC_1 wg.
                                                           ; winrho4.ini
PFxCopy.IpAddr_RemPC = ''                                 '' ; (32 blanks) in
                                                           ; rho4.1 without
                                                           ; signification
PFxCopy.SrcName      = 'Test1.qll'                         ; Name of source
                                                           ; file
PFxCopy.DstName      = 'c:\Sub2\Test3.qll'                 ; Name of target
                                                           ; file
PFxCopy.Overwrite    = 1
FILL_STRING (PFxCopy.SrcName)              ; Fills up the field with BLANKs
FILL_STRING (PFxCopy.DstName)              ; Fills up the field with BLANKs
RetCode              = rFxCopy (PFxCopy)   ; Call of the rho4fkt
```

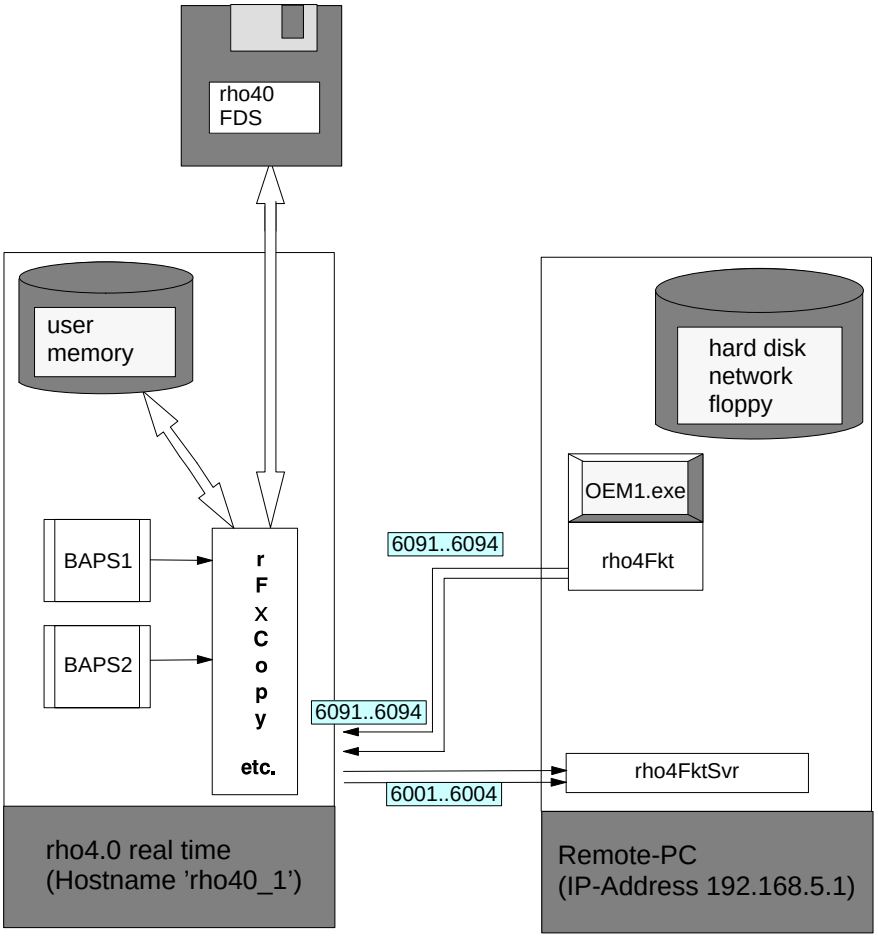
BAPS:
Copy file from drive c: of the RemotePC_1 −> user memory

```
PFxCopy.ChannelId    = 0                                  ; must be in BAPS=0
PFxCopy.ServerPortNo = 6003                                ; RemotePC_1 wg.
                                                           ; gateway.ini
PFxCopy.IpAddr_RemPC = ''                                 '' ; (32 blanks) in
                                                           ; rho4.1 without
                                                           ; signification
PFxCopy.SrcName      = 'c:\Sub3\Test1.qll'                 ; Name of
                                                           ; source file
PFxCopy.DstName      = 'Test1.qll'                         ; Name of
                                                           ; target file
PFxCopy.Overwrite    = 1
FILL_STRING (PFxCopy.SrcName)              ; Fills up the field with BLANKs
FILL_STRING (PFxCopy.DstName)              ; Fills up the field with BLANKs
RetCode              = rFxCopy (PFxCopy)   ; Call of the rho4fkt
```

rho4 library functions

7.7.2 Example rho4.0

With Remote-PC



To make things clear, the example from the figure is explained in detail. The hardware/software configuration is the following:

(1) Remote-PC with

(a) rho4FktSvr.exe

(b) rho4Fkt.dll

(c) Windows application 'OEM1.exe' (switcher) calling the rhoFile functions.

(2) rho4.0 with

(a) external floppy disk (Fds)

OEM1.exe works on the control with the servers rho_Functions_1 up to rho_Functions_4 (= PortNo. 6091, 6092, 6093, 6094).

Via the system channels Win_Server_1 to Win_Server_4 (= ServerPortNo 6001, 6002, 6003, 6004) the files are exchanged between rho4.0 user memory and the remote PC (e. g. drive c: of the remote PC).

☞ **By calling the rFx functions, the IP address of the remote PC (e.g. 192.168.5.1) must be programmed.**

Compare also the example application switcher.exe supplied in the delivery (including source code, specially swi_rf.c). By calling the function rTClientCon() for the initialization of the connection the hostname of the server PC must be entered there. In the example, this is for OEM1.exe the hostname of the rho4.0-Windows page (rho40_1). The timeout indicated there refers to the entire file operation, this is to be taken into account especially when copying large files.

For this example, the Ini-file 'Rho4FktSvr.ini' must be configured as follows:

Rho4FktSvr.ini

```
;; ------------------------------------------------------------------
;; --------- Initialization file for Rho4FktSvr.exe ------------------
;; Changes in the Rho4FktSvr.ini get only effective if the Rho4FktSvr.exe
;; is started again.
;; ------------------------------------------------------------------
[rho4Svr]
        ;; VO04H
Version = 408


ServerNo= 4
            ;; Name of system servers
1       = Win_Server_1
2       = Win_Server_2
3       = Win_Server_3
4       = Win_Server_4
```

rho4 library functions

```
;; ------------------------------------------------------------------
[Win_Server_1]
                ;; rho4 port number for system server 1
ServerPortNo     = 6001
;; ------------------------------------------------------------------
[Win_Server_2]
                 ;; rho4 port number for system server 2
ServerPortNo     = 6002
;; ------------------------------------------------------------------
[Win_Server_3]
                 ;; rho4 port number for system server 3
ServerPortNo     = 6003
;; ------------------------------------------------------------------
;[Win_Server_4]
                 ;; rho4 port number for system server 4
ServerPortNo     = 6004
;; ------------------------------------------------------------------
```

Programm examples

rFxCopy function calls by means of C- or BAPS program sections:

OEM1.EXE:
Copy file from drive c: of the RemotePC_1 −> user memory

```
PFxCopy.ChannelId    = rhoTCPClientConnect();          // connect to rho4
PFxCopy.ServerPortNo = 0;                              // next free PortNo
PFxCopy.IpAddr_RemPC = ”192.168.5.1”;                  // IP-address of
                                                      ;// remote-PC
strcpy(PFxCopy.SrcName, ”c:\\Sub1\\Test.dat”)         ;// name of
                                                      ;//source-file
strcpy(PFxCopy.DstName, ”TestAws.dat”);                // name of
                                                       ;// destination-file
PFxCopy.Overwrite    = 1;
lRet = rFxCopy (&PFxCopy);                             // function call
rhoTCPClientDisconnect (PFxCopy.ChannelId);            // close channel
```

OEM1.EXE:
Copy file from user memory −> drive c: of the RemotePC_1

```
PFxCopy.ChannelId      = rhoTCPClientConnect();        // connect to rho4
PFxCopy.ServerPortNo   = 0;                            // next free PortNo
PFxCopy.IpAddr_RemPC   = ”192.168.5.1”;                // IP-address of
                                                      ;// remote-PC
strcpy(PFxCopy.SrcName, ”TestAws.dat”);                // name of
                                                      ;// source-file
strcpy(PFxCopy.DstName, ”c:\\Sub1\\Test.dat”);         // name of
                                                      ;// destination-file
PFxCopy.Overwrite      = 1;
lRet = rFxCopy (&PFxCopy);                             // function call
rhoTCPClientDisconnect (PFxCopy.ChannelId);            // close channel
```

OEM1.EXE:
Copy file from drive c: of the RemotePC_1 −> external Floppy-Disk (Fds)

rho4 library functions

```
PFxCopy.ChannelId     = rhoTCPClientConnect();          // connect to rho4
PFxCopy.ServerPortNo  = 0;                              // next free PortNo
PFxCopy.IpAddr_RemPC  = "192.168.5.1";                  // IP-address of
                                                       ;// remote-PC
strcpy(PFxCopy.SrcName, "c:\\Sub1\\Test.dat");          // name of
                                                       ;// source-file
strcpy(PFxCopy.DstName, "/fd0/TestAws.dat");            // name of
                                                       ;// destination-file
PFxCopy.Overwrite     = 1;
lRet                  = rFxCopy (&PFxCopy);             // function call
rhoTCPClientDisconnect (PFxCopy.ChannelId);             // close channel
```

OEM1.EXE:
Copy file from external Floppy-Disk (Fds) −> drive c: of the RemotePC_1

```
PFxCopy.ChannelId     = rhoTCPClientConnect();          // connect to rho4
PFxCopy.ServerPortNo  = 0;                             // next free PortNo
PFxCopy.IpAddr_RemPC  = "192.168.5.1";                 // IP-address of
                                                       ;// remote-PC
strcpy(PFxCopy.SrcName, "/fd0/TestAws.dat");           // name of
                                                       ;// source-file
strcpy(PFxCopy.DstName, "c:\\Sub1\\Test.dat");         // name of
                                                       ;// destination-file
PFxCopy.Overwrite     = 1;
lRet = rFxCopy (&PFxCopy);                             // function call
rhoTCPClientDisconnect (PFxCopy.ChannelId);            // close channel
```

OEM1.EXE:
Copy file from external Floppy-Disk (Fds) −> user memory

```
PFxCopy.ChannelId     = rhoTCPClientConnect();          // connect to rho4
PFxCopy.ServerPortNo  = -1;                             // virtual server
PFxCopy.IpAddr_RemPC  = "                              ";// (32 blanks)
                                                        ;// redundant
                                                         ;// with virt.
                                                        ;// server
strcpy(PFxCopy.SrcName, "/fd0/TestAws.dat");            // name of
                                                        ;// source-file
strcpy(PFxCopy.DstName, "Test.dat");                    // name of
                                                        ;// dest.-file
PFxCopy.Overwrite     = 1;
lRet                  = rFxCopy (&PFxCopy);             // function call
rhoTCPClientDisconnect (PFxCopy.ChannelId);            // close channel
```

rho4 library functions

OEM1.EXE:
Copy file from user memory −> external Floppy-Disk (Fds)

```
PFxCopy.ChannelId     = rhoTCPClientConnect();          // connect to rho4
PFxCopy.ServerPortNo  = -1;                            // virtual server
PFxCopy.IpAddr_RemPC = ”                               ”;// (32 blanks)
                                                        ;// redundant
                                                         ;// with virt.
                                                        ;//server
strcpy(PFxCopy.SrcName, ”Test.dat”);                    // name of
                                                        ;// source-file
strcpy(PFxCopy.DstName, ”/fd0/TestAws.dat”);            // name of
                                                        ;// dest.-file
PFxCopy.Overwrite     = 1;
lRet                  = rFxCopy (&PFxCopy);             // function call
rhoTCPClientDisconnect (PFxCopy.ChannelId);            // close channel
```

OEM1.EXE:
Copy file from user memory −> user memory

```
PFxCopy.ChannelId     = rhoTCPClientConnect();          // connect to rho4
 PFxCopy.ServerPortNo  = -1;                           // virtual server
PFxCopy.IpAddr_RemPC = ”                               ”;// (32 blanks)
                                                        ;// redundant
                                                         ;// with virt.
                                                        ;// server
strcpy(PFxCopy.SrcName, ”TestAws1.dat”);                // name of
                                                        ;// source-file
strcpy(PFxCopy.DstName, ”TestAws2.dat”);                // name of
                                                        ;// dest.-file
PFxCopy.Overwrite     = 1;
lRet                  = rFxCopy (&PFxCopy);              // function call
rhoTCPClientDisconnect (PFxCopy.ChannelId);            // close channel
```

BAPS:
Copy file from user memory −> user memory

```
PFxCopy.ChannelId     = 0                              ; must be in BAPS=0
PFxCopy.ServerPortNo  = -1                              ; virtual server
PFxCopy.IpAddr_RemPC = ”                               ”; (32 blanks)
                                                        ;without meaning
                                                        ; in virt. server
PFxCopy.SrcName       = ’Test1.qll’                     ; Name of the
                                                        ; source file
PFxCopy.DstName       = ’Test2.qll’                     ; Name of the
                                                        ; target file
PFxCopy.Overwrite     = 1
FILL_STRING (PFxCopy.SrcName)              ; Fills up the field with BLANKs
FILL_STRING (PFxCopy.DstName)             ; Fills up the field with BLANKs
RetCode = rFxCopy (PFxCopy)              ; Call of the rho4fkt
```

BAPS:
Copy file from user memory −> external Floppy-Disk (Fds)

rho4 library functions

```
PFxCopy.ChannelId     = 0                                  ; must be in BAPS=0
PFxCopy.ServerPortNo  = -1                                  ; virtual server
PFxCopy.IpAddr_RemPC = ”                                   ”; (32 blanks)
                                                            ; without meaning
                                                            ; in virt. server
PFxCopy.SrcName       = 'Test1.qll'                         ; Name of the
                                                            ; source file
PFxCopy.DstName       = '/fd0/Test2.qll'                    ; Name of the
                                                            ; target file
PFxCopy.Overwrite     = 1
FILL_STRING (PFxCopy.SrcName)              ; Fills up the field with BLANKs
FILL_STRING (PFxCopy.DstName)             ; Fills up the field with BLANKs
RetCode = rFxCopy (PFxCopy)              ; Call of the rho4fkt
```

BAPS:
Copy file from user memory −> drive c: of the Remote-PC

```
PFxCopy.ChannelId     = 0                              ; must be in BAPS=0
PFxCopy.ServerPortNo  = 0                              ; next free PortNo
PFxCopy.IpAddr_RemPC  = ”192.168.5.1”;                 ; IP-Adresse of the
                                                       ; Remote-PC
PFxCopy.SrcName       = 'Test1.qll'                    ; Name of the
                                                       ; source file
PFxCopy.DstName       = 'c:\Sub2\Test3.qll'            ; Name of the
                                                       ; target file
PFxCopy.Overwrite     = 1
FILL_STRING (PFxCopy.SrcName)              ; Fills up the field with BLANKs
FILL_STRING (PFxCopy.DstName)             ; Fills up the field with BLANKs
RetCode               = rFxCopy (PFxCopy); Call of the rho4fkt
```

BAPS:
Copy file from drive c: of the RemotePC_1 −> user memory

```
PFxCopy.ChannelId     = 0                               ; must be in BAPS=0
PFxCopy.ServerPortNo  = 0                               ; next free PortNo
PFxCopy.IpAddr_RemPC  = ”192.168.5.1”;                  ; IP-Adresse of the
                                                        ; Remote-PC
PFxCopy.SrcName       = 'c:\Sub3\Test1.qll'             ; Name of the
                                                        ; source file
PFxCopy.DstName       = 'Test1.qll'                     ; Name of the
                                                        ; target file
PFxCopy.Overwrite     = 1
FILL_STRING (PFxCopy.SrcName)              ; Fills up the field with BLANKs
FILL_STRING (PFxCopy.DstName)              ; Fills up the field with BLANKs
RetCode = rFxCopy (PFxCopy)               ; Call of the rho4fkt
```

BAPS:
Copy file from external Floppy-Disk (Fds) −> external Floppy-Disk (Fds)

rho4 library functions

```
PFxCopy.ChannelId     = 0                                  ; must be in BAPS=0
PFxCopy.ServerPortNo  = -1                                 ; virtual server
PFxCopy.IpAddr_RemPC = ”                                    ”; (32 blanks)
                                                           ; without meaning
                                                           ; in virt. server
PFxCopy.SrcName       = ’/fd0/Test1.qll’                   ; Name of the
                                                           ; source file
PFxCopy.DstName      b= ’/fd0/Test2.qll’                   ; Name of the
                                                           ; target file
PFxCopy.Overwrite    = 1
FILL_STRING (PFxCopy.SrcName)                  ; Fills up the field with BLANKs
FILL_STRING (PFxCopy.DstName)                   ; Fills up the field with BLANKs
RetCode = rFxCopy (PFxCopy)                    ; Call of the rho4fkt
```

The required compound types for BAPS and a detailed documentation are to be found in the insert file rF_En.inc.

rho4 library functions

7.8 Machine parameters rhoMpGet [rMPG] 9000

Contains functions for reading machine parameters, also see manual on machine parameters.

The required record types for BAPS as well as a detailed documentation are contained in the include file rMPG_En.inc.

For the integration of the .dll under Windows, the file rMPG.h is available.

The function number is required for the declaration in BAPS.

DANGER
Interactions with other machine parameters can lead to illegal system statuses.

The function names are composed of different abbreviations. The following abbreviations mean:

r	rho	MP	Machine parameters
G	Get	All	All
P	Parameters		

System parameters

Function name	Ft. No.	Comment
rMPGAllP0001	9001	Number of kinematic
rMPGAllP0002	9002	Machine configuration
rMPGAllP0004	9004	Parity for INTEGER-inputs at the interface
rMPGAllP0005	9005	CLOCK starting time
rMPGAllP0006	9006	Running time control P1 - P2 logic
rMPGAllP0007	9007	Partition of the user memory
rMPGAllP0008	9008	Strobe times for INTEGER user outputs
rMPGAllP0009	9009	Strobe times for system outputs
rMPGAllP0010	9010	Selection of the national language
rMPGAllP0011	9011	Number of sensor inputs
rMPGAllP0016	9016	IRDATA stack size
rMPGAllP0017	9017	reserved
rMPGAllP0018	9018	reserved
rMPGAllP0019	9019	Definition of the PHG key groups
rMPGAllP0020	9020	I/O assembly group configuration

rho4 library functions

Function name	Ft. No.	Comment
rMPGAllP0021	9021	Address range soft-PLC
rMPGAllP0022	9022	Global A-, D-factor range limit
rMPGAllP0023	9023	Global V-factor range limit
rMPGAllP0024	9024	Deletion of user memory
rMPGAllP0025	9025	Reset of A/D/V-factors
rMPGAllP0027	9027	Strobe INTEGER inputs
rMPGAllP0028	9028	Display of options
rMPGAllP0030	9030	I/O configuration CAN
rMPGAllP0031	9031	Addresses of CAN inputs
rMPGAllP0032	9032	Addresses of CAN outputs
rMPGAllP0034	9034	Number of characters in output buffer
rMPGAllP0035	9035	PHG mode
rMPGAllP0036	9036	MF-I/O
rMPGAllP0037	9037	Electronic type plate
rMPGAllP0038	9038	Sercos Data
rMPGAllP0039	9039	Workspace monitoring

Speeds

Function name	Ft. No.	Comment
rMPGAxP0101	9101	Nominal lag
rMPGKinP0102	9102	Maximum belt speed
rMPGAxP0103	9103	Maximum axis speed PTP
rMPGAxP0104	9104	Slope-acceleration PTP
rMPGAxP0105	9105	Slope-point PTP JC
rMPGKinP0106	9106	Slope-point belt operation
rMPGAxP0107	9107	Slope-point Jog WC
rMPGAxP0108	9108	Referencing speed
rMPGAxP0109	9109	First reduced referencing speed
rMPGAxP0110	9110	Second reduced referencing speed
rMPGAxP0111	9111	Jog-speed WC slow
rMPGAxP0112	9112	Jog-speed WC fast
rMPGAxP0113	9113	Jog-speed JC slow
rMPGAxP0114	9114	Jog-speed JC fast
rMPGKinP0115	9115	Incremental steps WC
rMPGKinP0116	9116	Incremental steps WC
rMPGAxP0117	9117	A/D-slope Jog WC

rho4 library functions

Function name	Ft. No.	Comment
rMPGKinP0118	9118	Range limits for AFACTOR and DFACTOR
rMPGKinP0119	9119	Range limits for VFACTOR
rMPGKinP0120	9120	Switch-on status of slope-type (program or block-slope)
rMPGKinP0121	9121	Slope-shape (ramp, or $\sin^2$ similar)
rMPGAxP0122	9122	Times of acceleration or delay JC PTP
rMPGAxP0123	9123	Times of acceleration or delay Jog WC
rMPGKinP0124	9124	Times of acceleration or delay for belt operation
rMPGAxP0125	9125	Switch-off time of the interpolator-slope-monitoring
rMPGAxP0126	9126	Switch-off time of the standstill monitoring
rMPGAxP0127	9127	Inpos-range of the standstill monitoring
rMPGAxP0128	9128	A/D-slope JC
rMPGAxP0129	9129	Slope-point Jog JC
rMPGAxP0130	9130	Times of acceleration or delay for Jog JC

rho4 library functions

Speeds (data of kinematics)

Function name	Ft. No.	Comment
rMPGKinP0101	9151	Nominal lag of the kinematic
rMPGKinP0103	9153	Maximum axis speeds PTP of the kinematic
rMPGKinP0104	9154	Slope-acceleration PTP of the kinematic
rMPGKinP0105	9155	Slope-point PTP (of the kinematic)
rMPGKinP0107	9157	Slope-point Jog of the kinematic in WC
rMPGKinP0108	9158	Referencing speed of the kinematic
rMPGKinP0109	9159	First reduced referencing speed of the kinematic
rMPGKinP0110	9160	Second reduced referencing speed of the kinematic
rMPGKinP0111	9161	Jog-speed WC slow of the kinematic
rMPGKinP0112	9162	Jog-speed WC fast of the kinematic
rMPGKinP0113	9163	Jog-speed JC slow of the kinematic
rMPGKinP0114	9164	Jog-speed JC fast of the kinematic
rMPGKinP0117	9167	A/D-slope Jog of the kinematic WC
rMPGKinP0122	9172	Times of acceleration or delay of the kinematic JC PTP
rMPGKinP0123	9173	Times of acceleration or delay Jog of the kinematic WC
rMPGKinP0125	9175	Switch-off time of the interpolator-slope-monitoring of the kinematic
rMPGKinP0126	9176	Switch-off time of the standstill monitoring of the kinematic
rMPGKinP0127	9177	Inpos-range of the standstill monitoring of the kinematic
rMPGKinP0128	9178	A/D-slope of the kinematic JC in degrees/s^2 or mm/s^2
rMPGKinP0129	9179	Slope-point Jog of the kinematic JC in degrees/s^2 or mm/s^2
rMPGKinP0130	9180	Times of acceleration or delay for Jog of the kinematic JC

rho4 library functions

Positions

Function name	Ft. No.	Comment
rMPGAxP0201	9201	In-position-range
rMPGAxP0202	9202	Positive software limit switches for WC
rMPGAxP0203	9203	Negative software limit switches for WC
rMPGAxP0204	9204	Positive software limit switches for JC
rMPGAxP0205	9205	Negative software limit switches for JC
rMPGAxP0206	9206	Software-limit-switch-tolerance
rMPGAxP0207	9207	Actual value of reference point
rMPGAxP0208	9208	Reference point offset
rMPGAxP0213	9213	Passing distances in JC

Positions, data of kinematics

Function name	Ft. No.	Comment
rMPGKinP0212	9212	Presetting of passing distances and factors of the kinematic
rMPGKinP0214	9214	Type of passing of a kinematic
rMPGKinP0201	9251	In-position-range of the kinematic in mm or degrees
rMPGKinP0202	9252	Positive software limit switches of the kinematic for WC
rMPGKinP0203	9253	Negative software limit switches of the kinematic for WC
rMPGKinP0204	9254	Positive software limit switches of the kinematic for JC
rMPGKinP0205	9255	Negative software limit switches of the kinematic for JC
rMPGKinP0206	9256	Software-limit-switch-tolerance of the kinematic
rMPGKinP0207	9257	Actual value of reference point of the kinematic
rMPGKinP0208	9258	Reference point offset of the kinematic
rMPGKinP0213	9263	Passing distances in JC of the kinematic

rho4 library functions

Kinematic parameters

Function name	Ft. No.	Comment
rMPGKinP0301	9301	Kinematic name
rMPGKinP0302	9302	Number of axes of the kinematic
rMPGAxP0303	9303	Axis type
rMPGAxP0304	9304	Axis name
rMPGAxP0305	9305	Name of coordinates
rMPGKinP0306	9306	Selection of robot type and transformation
rMPGKinP0307	9307	Axis lengths, offset angles
rMPGKinP0308	9308	Axis coupling factors
rMPGKinP0309	9309	Flange coordinates/mounting precision
rMPGKinP0310	9310	Displacement of the world coordinate system
rMPGAxP0311	9311	Modulo value of the endless axis
rMPGKinP0313	9313	Allocation of robot axes to world coordinates
rMPGKinP0303	9353	Axis types of the kinematic
rMPGKinP0304	9354	Axis names of the kinematic
rMPGKinP0305	9355	Coordinate names of the kinematic
rMPGKinP0311	9361	Modulo value of the endless axes of the kinematic
rMPGKinP0314	9314	Serial number of electronic type plate

Measuring system parameters

Function number	Ft. No.	Comment
rMPGAxP0401	9401	Placement of the measuring system cards
rMPGAxP0402	9402	Direction of referencing
rMPGAxP0403	9403	Efficiency of reference point switch
rMPGP0404	9404	Number of the analog outputs
rMPGP0405	9405	Assignment of the analog outputs
rMPGP0406	9406	Number of the analog inputs
rMPGP0407	9407	Assignment of the analog inputs
rMPGBltP0401	9431	Parameter of a belt in P401

rho4 library functions

Measuring system parameters, data of kinematics

Function number	Ft. No.	Comment
rMPGKinP0401	9451	Placement of the measuring system cards of a kinematic
rMPGKinP0402	9452	Direction of referencing of the kinematic
rMPGKinP0403	9453	Efficiency of reference point switch of the kinematic
rMPGKiBP0401	9481	Belt parameter of a kinematic in P401

Belt parameters

Function number	Ft. No.	Comment
rMPGKinP0501	9501	Number of belts
rMPGBltP0502	9502	Direction of belt counter
rMPGBltP0503	9503	Belt coupling factor
rMPGBltP0504	9504	Maximum travel distance in belt direction
rMPGBltP0505	9505	Limit values for belt counters
rMPGBltP0506	9506	Belt names
rMPGBltP0507	9507	Belt pre-wait times
rMPGBltP0508	9508	Belt simulator speed

Belt parameters, data of kinematics

Function number	Ft. No.	Comment
rMPGKinP0502	9552	Direction of belt counter of the kinematic
rMPGKinP0503	9553	Belt coupling factors of the kinematic
rMPGKinP0504	9554	Maximum travel distance in belt direction of the kinematic
rMPGKinP0505	9555	Limit values for belt counters of the kinematic
rMPGKinP0506	9556	Belt names of the kinematic
rMPGKinP0507	9557	Belt pre-wait times of the kinematic
rMPGKinP0508	9558	Belt simulator speeds of the kinematic

Drive parameters, Servodyn-GC

Function number	Ft. No.	Comment
rMPGAxP0600	9600	Supplies the parameters P600 for one axis
rMPGKinP0600	9650	Supplies the parameters P600 for one kinematic

rho4 library functions

Drive parameters, Servodyn-D

Function number	Ft. No.	Comment
rMPGAxP0700	9700	Supplies the parameters P700 for one axis
MPGKinP0700	9750	Supplies the parameters P700 for one kinematic

rho4 library functions

7.9 Machine parameters rhoMpSet [rMPS] 10000

Contains functions for setting machine parameters. The necessary record types for BAPS, as well as a detailed documentation are to be found in the include file rMPS_En.inc, see manual of machine parameters.

For the integration of the .dll under Windows, the file rMPS.h is available.

The function number is required for the declaration in BAPS.

DANGER
Interactions with other machine parameters can lead to illegal system statuses.

The changing of machine parameters via library functions is admissible for all parameters. The set machine parameters are active after the next run-up.

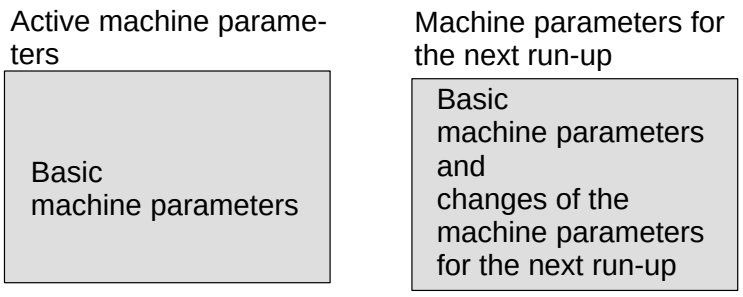
Efficiency of the machine parameters

In the rho4 two machine parameter sets are active parallel to each other. After run-up of the control, the two machine parameter sets are identical. There is a basic set and a further set of machine parameters in which changes via the library functions rMPS...() are considered. Until the next run-up of the control, thes basic machine parameters are active. By a controlled run-down of the control, the changed machine parameters are saved, in addition to other system-relevant files, and they are active after the next run-up. After run-up of the control, the machine parameters saved before, are now the basic machine parameters. A copy is made and is available for further changes.

When loading machine parameters via the ROPS-coupling or via the rhoArchiveFunktion rADownload(), changes of the machine parameters carried out via library functions rMPS...() are overwritten.

When saving machine parameters via the ROPS-coupling or via the rhoArchiveFunktion rAUpload(), changes of the machine parameters carried out via library functions rMPS...() are saved.

Machine parameter sets



rho4 library functions

The function names are composed of different abbreviations. The following abbreviations mean

r	rho	MP	Machine parameters
S	Set	All	All
P	Parameters		

rho4 library functions

System parameters

Function name	Ft. No.	Comment
rMPSAllP0001	10001	Number of the kinematics
rMPSAllP0002	10002	Machine configuration
rMPSAllP0004	10004	Parity for INTEGER inputs at the interface
rMPSAllP0005	10005	CLOCK starting time
rMPSAllP0006	10006	Running time contol P1-P2 logic
rMPSAllP0007	10007	Partition of the user memory
rMPSAllP0008	10008	Strobe times for INTEGER user outputs
rMPSAllP0009	10009	Strobe times for system outputs
rMPSAllP0010	10010	Selection of the national language
rMPSAllP0011	10011	Sensor inputs
rMPSAllP0016	10016	IRDATA stack size
rMPSAllP0017	10017	reserved
rMPSAllP0018	10018	reserved
rMPSAllP0019	10019	Definition of the PHG-key groups
rMPSAllP0020	10020	I/O assembly group configuration
rMPSAllP0021	10021	Address range soft-PLC
rMPSAllP0022	10022	Global A and D-factor range limit
rMPSAllP0023	10023	Global V-factor range limit
rMPSAllP0024	10024	Deletion of the user memory
rMPSAllP0025	10025	Reset of A/D/V-factors
rMPSAllP0027	10027	Strobe INTEGER-inputs
rMPSAllP0030	10030	I/O-configuration CAN
rMPSAllP0031	10031	Addresses of CAN inputs
rMPSAllP0032	10032	Addresses of CAN outputs
rMPSAllP0034	10034	Number of characters in output buffer
rMPSAllP0035	10035	PHG-mode
rMPSAllP0036	10036	MF-I/O
rMPSAllP0037	10037	Electronic type plate
rMPSAllP0038	10038	Sercos Data
rMPSAllP0039	10039	Sets data of workspace monitoring

rho4 library functions

Speeds

Function name	Ft. No.	Comment
rMPSAxP0101	10101	Nominal lag
rMPSKinP0102	10102	Maximum belt speed
rMPSAxP0103	10103	Maximum axis speed PTP
rMPSAxP0104	10104	Slope-acceleration PTP
rMPSAxP0105	10105	Slope-point PTP JC
rMPSKinP0106	10106	Slope-point belt operation
rMPSAxP0107	10107	Slope-point Jog WC
rMPSAxP0108	10108	Referencing speed
rMPSAxP0109	10109	First reduced referencing speed
rMPSAxP0110	10110	Second reduced referencing speed
rMPSAxP0111	10111	Jog-speed WC slow
rMPSAxP0112	10112	Jog-speed WC fast
rMPSAxP0113	10113	Jog-speed JC slow
rMPSAxP0114	10114	Jog-speed JC fast
rMPSKinP0115	10115	Incremental steps WC
rMPSKinP0116	10116	Incremental steps WC
rMPSAxP0117	10117	A/D-slope Jog WC
rMPSKinP0118	10118	Range limits for AFACTOR and DFACTOR
rMPSKinP0119	10119	Range limits for VFACTOR
rMPSKinP0120	10120	Switch-on status of slope-type (program or block-slope)
rMPSKinP0121	10121	Slope-shape (ramp or $\sin^2$-similar)
rMPSAxP0122	10122	Times of acceleration or delay JC PTP
rMPSAxP0123	10123	Times of acceleration or delay Jog in WC
rMPSKinP0124	10124	Times of acceleration or delay for belt operation
rMPSAxP0125	10125	Switch-off time of the interpolator-slope-monitoring
rMPSAxP0126	10126	Switch-off time of the standstill monitoring
rMPSAxP0127	10127	Inpos-range of the standstill monitoring
rMPSAxP0128	10128	A/D-slope JC
rMPSAxP0129	10129	Slope-point Jog in JC
rMPSAxP0130	10130	Times of acceleration or delay Jog in JC

rho4 library functions

Speeds, data of kinematics

Function name	Ft. No.	Comment
rMPSKinP0101	10151	Nominal lag of the kinematic
rMPSKinP0103	10153	Maximum axis speeds PTP of the kinematic
rMPSKinP0104	10154	Slope-acceleration PTP of the kinematic
rMPSKinP0105	10155	Slope-point PTP of the kinematic JC
rMPSKinP0107	10157	Slope-point Jog of the kinematic WC
rMPSKinP0108	10158	Referencing speed of the kinematic
rMPSKinP0109	10159	First reduced referencing speed
rMPSKinP0110	10160	Second reduced referencing speed
rMPSKinP0111	10161	Jog-speed WC slow of the kinematic
rMPSKinP0112	10162	Jog-speed WC fast of the kinematic
rMPSKinP0113	10163	Jog-speed JC slow of the kinematic
rMPSKinP0114	10164	Jog-speed JC fast of the kinematic
rMPSKinP0117	10167	A/D-slope Jog of the kinematic WC
rMPSKinP0122	10172	Times of acceleration or delay of the kinematic JC PTP
rMPSKinP0123	10173	Times of acceleration or delay Jog of the kinematic WC
rMPSKinP0125	10175	Switch-off time of the interpolator-slope-monitoring of the kinematic
rMPSKinP0126	10176	Switch-off time of the standstill monitoring of the kinematic
rMPSKinP0127	10177	Inpos-range of the standstill monitoring of the kinematic
rMPSKinP0128	10178	A/D-slope of the kinematic JC
rMPSKinP0129	10179	Slope-point Jog of the kinematic JC
rMPSKinP0130	10180	Times of acceleration or delay for Jog of the kinematic in JC

rho4 library functions

Positions

Function name	Ft. No.	Comment
rMPSAxP0201	10201	In-position range
rMPSAxP0202	10202	Positive software limit switches for WC
rMPSAxP0203	10203	Negative software limit switches for WC
rMPSAxP0204	10204	Positive software limit switches for JC
rMPSAxP0205	10205	Negative software limit switches for JC
rMPSAxP0206	10206	Software-limit-switch-tolerance
rMPSAxP0207	10207	Actual value of reference point
rMPSAxP0208	10208	Reference point offset
rMPSAxP0212	10212	Presetting of passing distances and factors
rMPSAxP0213	10213	Passing distances in JC
rMPSAxp0214	10214	Passing criteria

Positions ,data of kinematics

Function name	Ft. No.	Comment
rMPSKinP0212	10212	Presetting of passing distances and factors of the kinematic
rMPSKinP0201	10251	In-position-range of the kinematic
rMPSKinP0202	10252	Positive software limit switches of the kinematic for WC
rMPSKinP0203	10253	Negative software limit switches of the kinematic for WC
rMPSKinP0204	10254	Positive software limit switches of the kinematic for JC
rMPSKinP0205	10255	Negative software limit switches of the kinematic for JC
rMPSKinP0206	10256	Software-limit-switch-tolerance of the kinematic
rMPSKinP0207	10257	Actual value of reference point of the kinematic
rMPSKinP0208	10258	Reference point offset of the kinematic
rMPSKinP0213	10263	Passing distances JC of the kinematic

rho4 library functions

Kinematic parameters

Function name	Ft. No.	Comment
rMPSKinP0301	10301	Name of kinematic
rMPSKinP0302	10302	Number of axes of the kinematic
rMPSAxP0303	10303	Axis type
rMPSAxP0304	10304	Axis name
rMPSAxP0305	10305	Name of coordinates
rMPSKinP0306	10306	Selection of robot type and transformation
rMPSKinP0307	10307	Axis lengths, offset angles
rMPSKinP0308	10308	Axis coupling factors
rMPSKinP0309	10309	Flange coordinates/mounting accuracies
rMPSKinP0310	10310	Displacement of the world coordinate system
rMPSAxP0311	10311	Modulo value of the endless axis
rMPSKinP0313	10313	Allocation of robot axes to world coordinates
rMPSKinP0303	10353	Axis types of the kinematic
rMPSKinP0304	10354	Axis names of the kinematic
rMPSKinP0305	10355	Coordinate names of the kinematic
rMPSKinP0311	10361	Modulo value of the endless axes of the kinematic
rMPSKinP0313	10363	Serial number of electronic type plate

Measuring system parameters

Function name	Ft. No.	Comment
rMPSAxP0401	10401	Placement of the measuring system cards
rMPSAxP0402	10402	Direction of referencing
rMPSAxP0403	10403	Efficiency of reference point switch
rMPSP0404	10404	Number of the analog outputs
rMPSP0405	10405	Assignment of the analog outputs
rMPSP0406	10406	Number of the analog inputs
rMPSP0407	10407	Assignment of the analog inputs
rMPSBltP0401	10431	Parameter of a belt in P401

rho4 library functions

Measuring system parameters, data of kinematics

Function name	Ft. No.	Comment
rMPSKinP0401	10451	Placement of the measuring system cards of the kinematic
rMPSKinP0402	10452	Direction of referencing of the kinematic
rMPSKinP0403	10453	Efficiency of reference point switch of the kinematic
rMPSKiBP0401	10481	Belt parameter of a kinematic in P401

Belt parameters

Function name	Ft. No.	Comment
rMPSKinP0501	10501	Number of belts
rMPSBltP0502	10502	Direction of belt counter
rMPSBltP0503	10503	Belt coupling factor
rMPSBltP0504	10504	Maximum travel distance in belt direction
rMPSBltP0505	10505	Limit values for belt counter
rMPSBltP0506	10506	Belt names
rMPSBltP0507	10507	Belt pre-wait time
rMPSBltP0508	10508	Belt simulator speed

Belt parameters, data of kinematics

Function name	Ft. No.	Comment
rMPSKinP0502	10552	Direction of belt counter of the kinematic
rMPSKinP0503	10553	Belt coupling factors of the kinematic
rMPSKinP0504	10554	Maximum travel distance of the kinematic in belt direction
rMPSKinP0505	10555	Limit values for belt counter of the kinematic
rMPSKinP0506	10556	Belt names of the kinematic
rMPSKinP0507	10557	Belt pre-wait time of the kinematic
rMPSKinP0508	10558	Belt simulator speed of the kinematic

Drive parameters, Servodyn-GC

Function number	Ft. No.	Comment
rMPSAxP0600	10600	Sets the parameters P600 for one axis
rMPSKinP0600	10650	Sets the parameters P600 for one kinematic

rho4 library functions

Drive parameters, Servodyn-D

Function name	Ft. No.	Comment
rMPSAxP0700	10700	Sets the parameters P700 for one axis
rMPSKinP0700	10750	Sets the parameters P700 for one kinematic

rho4 library functions

7.10 TCP-Server/Client rhoTCP [rT] 11000

Functions for a Client/Server connection via TCP between the Windows PC and the control.

The required record types for BAPS as well as a detailed documentation are contained in the include file rT_En.inc.

For the integration of the .dll under Windows, the file rT.h is available.

The function number is required for the declaration in BAPS.

The function names are composed of different abbreviations. The following abbreviations mean:

r	rho	T	TCP
Con	Connect	Dis	Disconnect
Ac	Accept		

	Fct No.	**Comment**
rTClientCon()	11000	Creates a connection with the indicated IP port address
rTClientDis()	11001	Closes a connection
rTServerAc()	11010	Generates a server
rTServerDis()	11011	Closes a server
rTWrite() *	11020	Writes data onto the connection
rTRead() *	11021	Reads data from the connection
rTSTimeOut	11022	Sets the TimeOut for the connection. Can only be called as Windows-Dll

* The TCP/IP communication is limited to a max. block length of 4kByte.

☞ **See also the documentation of the rho4 system description.**

rho4 library functions

7.11 Archiving rhoArchiving [rA] 14000

☞ **Can only be called as Windows-dll!**

These functions are also available in the DDE-Server.

These functions serve the archiving of the user memory on the hard disk or on the diskette.

The required C-types as well as a detailed documentation are contained in the include file rA.h.

The function names are composed of different abbreviations. The following abbreviations mean:

r rho A Archive

	Fct No.	Comment
rAComInit()	14000	Initialization of the serial interface
rAComExit()	14001	Closes the serial interface
rATCPCon()	14002	Creates a connection to the indicated IP port address
rATCPDis()	14003	Closes the TCP-connection
rAUpload()	14010	Copies a file from the user memory to the hard disk
rADownload()	14020	Copies a file from the hard disk to the user memory
rAList()	14030	Lists all files in the user memory
rARename()	14040	Renames a file in the user memory
rADelete()	14050	Deletes a file in the user memory

☞ **See also the documentation of the rho4 system description.**

rho4 library functions

7.12 Machine parameter rhoMxGet [rMxG] 19000

Contains functions for reading machine parameters that have been extended in the course of new developments.

The required record types for BAPS as well as a detailed documentation are contained in the include file rMxG.inc.

The file rMxG.h is available for the integration of the .dll under Windows.

The function number is necessary for the declaration in BAPS.

DANGER
Interactions with other machine parameters can lead to inadmissible system states.

The function names are composed of different abbreviations. The following abbreviations mean:

r	rho	MP	Machine parameter
G	Get	All	All
P	Parameter	Kin	Kinematic
x	extended	Ax	Axis

System parameters

Function name	Fct No.	Comment
rMxGAllP0021	19021	Address Soft−PLC
rMxGAllP0030	19030	I/O configuration CAN bus
rMxGAllP0031	19031	Address CAN inputs
rMxGAllP0032	19032	Address CAN outputs
rMxGAllP0036	19036	Multifunction−I/O

Kinematic parameters

Function name	Fct No.	Comment
rMxGKinP0310	19310	Displacement of the world coordinate system of kinematics with 6 degrees of freedom

rho4 library functions

Measuring system parameters

Function name	Fct No.	Comment
rMxGAxP0401	19401	Placement of the measuring system cards
rMxGKinP0401	19451	Placement of the measuring system cards of a kinematic

rho4 library functions

7.13 Machine parameter rhoMxSet [rMxS] 20000

Contains functions for setting machine parameters that have been extended in the course of new developments.

The required record types for BAPS as well as a detailed documentation are contained in the include file rMxS.inc.

The file rMxS.h is available for the integration of the .dll under Windows.

The function number is required for the declaration in BAPS.

DANGER
Interactions with other machine parameters can lead to inadmissible system states.

The change of machine parameters via library functions is allowed for all parameters. The set machine parameters are effective after the next start.

Effectiveness of the machine parameters

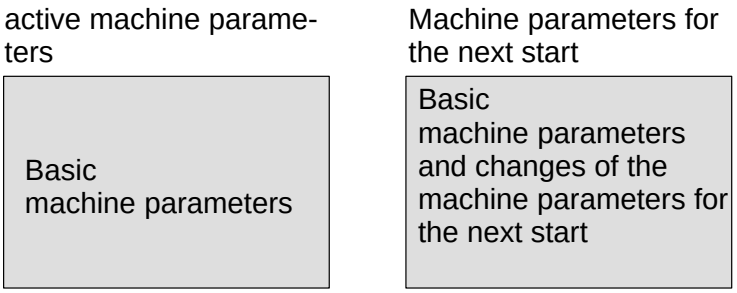
In the rho4, two machine parameter sets are kept parallel. After the start of the control unit, both machine parameter sets are identical. There are a basic set and another set of machine parameters in which changes via library functions rMPS...() are taken into account. Until the next start of the control unit, the basic machine parameters are effective. With the regular shut-down of the control unit, the changed machine parameters, besides other system relevant files, are saved and will be active at the next start. After the start of the control unit, the machine parameters previously saved become basic maachine parameters. A backup is created and is available for other changes.

When loading machine parameters via the ROPS coupling or via the rhoArchiveFunction rADownload(), changes of the machine parameters performed via library functions rMPS...() are overwritten.

When saving machine parameters via the ROPS coupling or via the rhoArchiveFunction rADownload(), machine parameters changed via library functions rMPS...() are saved.

rho4 library functions

Machine parameter sets



The function names are composed of different abbreviations. The following abbreviations mean:

r	rho	MP	Machine parameter
S	Set	All	All
P	Parameter	Kin	Kinematic
x	extended	Ax	Axis

System parameters

Function name	Fct No.	Comment
rMxSAllP0021	20021	Address Soft−PLC
rMxSAllP0030	20030	I/O configuration CAN bus
rMxSAllP0031	20031	Address CAN inputs
rMxSAllP0032	20032	Address CAN outputs
rMxSAllP0036	20036	Multifunction I/O

Speeds

Function name	Fct No.	Comment
rMxSAxP0103	20103	modify maximum axis speed PTP without reboot of the control
rMxSAxP0104	20104	modify slope-acceleration PTP without reboot of the control
rMxSKinP0103	20153	modify maximum axis speed PTP of a kinematics without reboot of the control
rMxSKinP0104	20154	modify slope acceleration PTP of a kinematics without reboot of the control

Kinematic parameters

Function name	Fct No.	Comment
rMxSKinP0310	20310	Displacement of the world coordinate system of kinematics with 6 degrees of freedom

rho4 library functions

Measuring system parameters

Function name	Fct No.	Comment
rMxSAxP0401	20401	Placement of the measuring system cards
rMxGKinP0401	20451	Placement of the measuring system cards of a kinematic

8 Variable access per DLL

In the library ‘rho4 library functions’, C functions are made available for the OEM enabling the access to user variables. The class 16000 contains functions for reading and writing user variables in BAPS programs that can be on the rho04 as well as on the local disk of the PC.

For rho4 accesses, it is allowed to change variables of programs running as active rho4 process.

☞ **There can be changed (unintended) movements when the variable is used in movement instructions.**

8.1 Variable access

8.1.1 Access to SYM-, IRD- and PKT file

Symbol information

In order to be able to access to variables symbolically, the rho4fkt-DLL requires information from the SYM file. This file must be present on the PC and the rho4fkt-DLL must be informed where it is.

In a ASCII string, the path and file name are given without extension. In this string, all ‘\’ characters are doubled.

Example

C:\\Bosch\\rho4\\Example\\Baps\\BapsVar4

The file ‘BapsVar4.SYM’ is searched in ‘C:\Bosch\rho4\Example\Baps’ on the local hard disk of the PC and used for evaluation of the symbol tables. For the access to point variables (DEF- and Teach points), the file ‘BapsVar4.PKT’ is expected in the same PC file, since PKT files also contain symbol information.

Variable values

The content of static BAPS variables is in the IRD file and in the case of DEF and teach points in the PKT file.

Variable accesses are allowed for:

- files in the RAM memory of the rho4 (RC access)
- files saved from the rho4 on the PC (PC access)

Variable access per DLL

On which file the rho4fkt-DLL is to access, is indicated in the corresponding transfer parameters. If an access is to be done on an IRD file on the PC, this file must be in the same path as the SYM file. This also applies to accesses on the PKT file.

Allowed variables

In principle, the rho4fkt-DLL can access to all user variables, the contents of which are filed in the IRD- or PKT file, i.e. variables that are defined in the main program.

The PKT file (Level 0) contains teach points and points declared with the ‘DEF’. All other static variables are filed in the IRD file (Level 1), see also page 8−5, Variable access via relative address.

Dynamic user variables existing only at the running time of a process on the IRDATA stack cannot be read or described. Transfer parameters to subprograms or variables that are not defined in subprograms belong e.g. to this kind of variables.

The rho4fkt-DLL has no access on ‘system variables’. These are variables that are always present in every process and do not need to be specially declared by the programmer.

The following belong to the system variables:

POS, @POS, @MPOS, LIMIT_MIN, LIMIT_MAX, V, VFIX, T, TFIX, A, AFIX, V_PTP, VFIX, VFACTOR, AFACTOR, WC_SYSTEM, DFACTOR, R_PTP and R

The current values of these variables are not filed in the IRD file but treated separately by the operating system of the control unit.

Entry of variable names

The variable name is transferred to the rho function as ASCII string. Capital and small letters are considered in the same way. The entry of wildcards is not allowed. Name extensions such as kinematic names or components of point variables or records are separated from the real variable name through a point. Records can be transferred completely or as record components up to the 3rd level, e.g Var1.PalPos.Z_C.

Arrays and records can be transferred completely when the data amount does not exceed 3.75 K-bytes (3840 bytes). Individual array elements are requested by entering the index number in brackets. In the case of arrays, it is possible to give a zone by using a hyphen ‘-’. In the case of multidimensional arrays, it is allowed to give a zone only in the last array dimension.

Variable access per DLL

Examples for access on one-dimensional arrays:

Syntax	Description
ARRAY[1..100] REAL: Real_array	Declaration in BAPS
Real_array	Access on the complete array
Real_array[75]	Access on an array element, here access on the 75th array element
Real_array[21-30]	Access on an array zone, here access on the array elements 21 to 30 (10 elements = 40 bytes)

Variable access per DLL

Examples for access on multidimensional arrays:

Syntax	**Description**
ARRAY[1..10] ARRAY[1..20] INTEGER: Int_array	Declaration in BAPS
Int_array	Access on the complete array
Int_array[5][18]	Access on an array element
Int_array[5][12-17]	Access on an array zone, here access on the array elements 12 to 17 in Int_array[5] (6 elements = 24 bytes)

Examples for access on records:

Syntax	**Description**
TV_DATA = RECORD INTEGER : VG REAL : VD POINT :VP RECORD_END	Type declaration in BAPS
TV_DATA: Data1	Variable declaration in BAPS
Data1	Access on the complete record
Data1.VD	Access on a record component
Data1.VP	Access on several elements of a record component (type POINT)
Data1.VP.X_K	Access on a record component

Safety inquiry (CommonID)

The rho4fkt-DLL can guarantee the correctness of the supplied variable contents only when all required informations come from files that have been created in the same compilation operation. To ensure this, a socalled ‘CommonID monitoring is performed. (CommonID is an identification number that is noted during compilation in every file and indicates at what time this file has been last modified or created).

If CommonID of the IRD-, PKT- and SYM file do not correspond, an error message appears in the return code of the rho function.

The CommonID monitoring can be switched off.

CAUTION
A switch-off of the CommonID monitoring can lead to the destruction of the user memory.
Subsequent process running time errors up to unintended movements cannot be excluded.

Variable access per DLL

Variable access via relative address

Instead of the variable name, it is possible to transfer from the 2nd access on the variable as transfer parameter address and level of the variables as well as the length of the data amount to be transferred. Besides, for security, the CommonID of the symbol file must also be transferred. These parameters are returned with the 1st access on the variable as return parameter of the rho function. This enables to save the evaluation of the symbol tables on the side of the PC in the case of repeated accesses.

The parameter level indicates if a variable is in the IRD or PKT file. Variables that are in IRD files are level 1 variables. Variables that are in PKT files are level 0 variables.

The variable access via the relative address is only allowed when the concerned IRD and PKT files are on the rho4. For accesses to IRD and PKT files that are on the PC, the access must occur always symbolically via the name of the variables.

Read access to user variable

The read access occurs via the function rVReadVar.

```
EXTERN long STDCALL rVReadVar( TVReadVar*PVReadVar);
```

The variable name is transmitted as ASCII string within TVReadVar.

The file name is also transmitted as ASCII string including the complete path name and without extension. In this string, all ‘\’-characters must be doubled.

Example

C:\\Bosch\\rho4\\Example\\Baps\\BapsVar4

Read variable values are transmitted binary. In the case of records, the BAPS record must be reproduced on the Windows page as compatible C-Struct.

The type of the transfer parameter is described in the header file rv.h.

The function rVReadVar supplies as return parameter a return code (error code) and if the return code has the value 0, the length, the element length, the relative address and the level of the desired variable and the CommonID of the symbol file.

The return parameters length, element length, relative address, level and CommonID can be used for further accesses to the same variable in later function calls.

Variable access per DLL

Write access to user variable

The write access occurs via the function rVWriteVar.

```
EXTERN long STDCALL rVWriteVar( TVWriteVar*PVWriteVar);
```

The variable name is transmitted as ASCII string within TVWriteVar.

The file name is also transmitted as ASCII string including the complete path name and without extension. In this string, all '\'-characters must be doubled.

Example

C:\\Bosch\\rho4\\Example\\Baps\\BapsVar4

The value, with which the variable is to be overwritten, is transmitted binary. In the case of records, the BAPS record must be reproduced on the Windows page as compatible C-Struct.

The type of the transfer parameter is described in the header file rv.h.

The function rVWriteVar supplies as return parameter a return code (error code) and if the return code has the value 0, the length, the element length, the relative address and the level of the desired variable and the CommonID of the symbol file.

The return parameters length, element length, relative address, level and CommonID can be used for further accesses to the same variable in later function calls.

Variable access per DLL

8.2 Administration data structure

The creation of the administration data structure (C-struct TVRdWrVar) containing the transfer parameters of the rho functions is described in the header file rv.h.

For all read and write accesses, the same C-struct is used. The variable to be transmitted is transmitted as union in this C-struct. The maximum useful data length that can be transmitted within a rho function call, is 3.75 Kbytes (3840 bytes).

On the Windows page, the BAPS data types to be transmitted must be represented through congruent C-data types.

The length of the BAPS data types allowed for read and write access is given in the following table.

BAPS data type	C-Data type	Length in bytes
BINARY	Long	4
INTEGER	Long	4
REAL	Float	4
CHAR	Unsigned char	1
POINT	Typedef float POINT [axes + belts]	4 * (axis number + belt number)
JC_POINT	Typedef float JC_POINT [axes + belts]	4 * (axis number + belt number)
WC_FRAME	Typedef float WC_FRAME [6]	24
TEXT	Typedef char TEXT [80]	80

8.3 Application example Switcher

The example of application ‘SWITCHER’ is included in the scope of delivery of the rho4. Some examples of access to BAPS variables can be found there in the module SWI_RV.C.

The examples refer to the following BAPS files:

- BAPSVAR4.QLL
- BAPSVAR4.SYM
- BAPSVAR4.PKT
- BAPSVAR4.IRD

These files are also contained in the rho4 software scope of delivery in the directory ‘C:\Bosch\rho4\Example\Baps’. The header file RV.H contains the types of the transfer parameters and other information on the assignment and the return codes of the library functions.

Variable access per DLL

Notes:

Appendix

A Appendix

A.1 Abbreviations

Abbreviation	Meaning
BAPS3	Programming language; Bewegungs- und Ablaufprogrammiersprache, Version 3; programming language
C:	Hard disk drive
CAN	Controler Area Network
DAC	Digital-analog converter
EEPROM	Electronically erasable programmable read-only memory
EGB	Elektrostatic sensitive components
ESD	Electrostatic discharge
LF	Line feed
MPP	Machine parameter program
MSD	Machine state display
PCL	Memory-programmable control
PE	Protective earth
PHG	Hand-held programming unit
POS	Actual position
PTP	Point to point
RC	Robot control
ROD	Incremental encoder
RPM	Rounds per minute
ROPS4	Robot programming system for rho4
TCP	Tool center point
WC	World coordinates

Appendix

A.2 Index

Appendix

Appendix

Notes:

Rexroth
Bosch Group

Bosch Rexroth AG
Electric Drives and Controls
P.O. Box 13 57
97803 Lohr, Germany
Bgm.-Dr.-Nebel-Str. 2
97816 Lohr, Germany
Phone +49 (0)93 52-40-50 60
Fax +49 (0)93 52-40-49 41
service.svc@boschrexroth.de
www.boschrexroth.com

1070072176

Printed in Germany
DOK-RHO*4*-DLL-BIBLIO*-PR07-EN-P