Industrial Hydraulics | **Electric Drives and Controls** | Linear Motion and Assembly Technologies | Pneumatics | Service Automation | Mobile Hydraulics

Rexroth
Bosch Group

Rexroth Rho 4.0
System description

1070072367
Edition 04

Project planning

Title Rexroth Rho 4.0
System description

Type of Documentation Project planning

Document Typecode DOK-RHO*4*-RHO4.0*SYSB-PR04-EN-P

Purpose of Documentation The present manual informs about:

- structures, functionalities and
- programming of the rho4.0

Record of Revisions

Description	Release Date	Notes
DOK-RHO*4*-RHO4.0*SYSB-PR03-EN-P	10.2003	Valid from VO07
DOK-RHO*4*-RHO4.0*SYSB-PR04-EN-P	01.2005	Valid from VO08

Validity The specified data is for product description purposes only and may not be deemed to be guaranteed unless expressly confirmed in the contract. All rights are reserved with respect to the content of this documentation and the availability of the product.

Published by 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

Manual	Contents	
Connection conditions Rho 4.0	2	System overview
	3	Installation
	4	Electrical connection
	5	Interfaces
	6	LED display
	7	Maintenance and replacement
	8	Order numbers
System description Rho 4.0	2	System overview
	3	Structure of the rho4.0
	4	PCLrho4.0
	5	CAN-Bus peripheral unit
	6	SERCOS interface
	7	Software
	8	File management
Connection conditions Rho4.1, Rho 4.1/IPC300	2	System overview
	3	Security functions
	4	Installation
	5	Electrical connection
	6	Interfaces
	7	LED display
	8	Maintenance and replacement
	9	Software
	10	Order numbers
Connection conditions Rho 4.1/BT155, Rho 4.1/BT155T, Rho 4.1/BT205	2	System overview
	3	Security functions
	4	Installation
	5	Electrical Connections
	6	Interfaces
	7	Display and Operating Controls
	8	Maintenance and Replacemant
	9	Software
	10	Order numbers
System description Rho 4.1	2	Structure of the rho4.1
	3	PCL
	4	CAN-Bus peripheral unit
	5	SERCOS interface

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

Contents

Contents

Contents

Notes:

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.

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

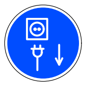
1.3 Safety markings on products

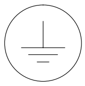

Warning of dangerous electrical voltage!

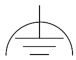

Warning of danger caused by batteries!


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

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.

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 on the structures and functionalities, as well as the programming of the rho4.0.

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 072 361
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 System overview

The rho4.0 is a robot, handling and movement control for which the whole processor capacity for the real time part is available since it can run without Windows. VxWorks is used as Task Scheduler.

The hardware core of the rho4.0 is PC-based and extended by the hardware required for a real-time control.

8 axes and 8 kinematics can be controlled in the base function scope.

The rho4.0 consists of a power pack unit, a carrier board, a PC board and a PCI plugging place, in which one of the 3 BOSCH field bus cards (CAN, INTERBUS-S and PROFIBUS-DP) can be operated.

The PC Board is equipped with a 266- or 400 MHz processor and 32-MB-SO-DIMM. A compact flash card is used as hard disk.

There are 2 types of housing: one in the form of a drive module and another one as 19”-rack.

The drive housing is located vertically besides a drive module. The 19”-housing is 1 HE (44mm) high and fitted horizontally in a 19”- control cabinet.

The 19” variant is available with extended remanence.

System overview

Notes:

3 Structure of the rho4.0

3.1 Description of the structure

The following notes refer to point 3.1.2 Block structure of the rho4.0.

Via an external PC it is possible to communicate with the rho4.0 in Windows via both a DDE server and DLL libraries. Various library functions are available.

For the connection of the library functions the OEM has four parallel channels available. The library functions are represented on Windows side as DLL.

There is one common transmission channel available for ROPS4 and DDE linking.

Four Win channels are available in BAPS for communication with the rho4.0 (Win_1 to Win_4). In addition, BAPS incorporates the capability of creating one or more BAPS clients using library functions.

Coupling to a PCLrho4.0 (PC-programmable logic control), which may run on the same computer, is provided through an internal TCP/IP connection. A data channel to the PCL is provided in the same way. The PCL is programmed using the WIN-SPS software.

The rho4.0 provides three serial channels.

The PHG2000 is the default operating device for the rho4.0. In addition to the default operating interface, it is possible to customise the interface of the PHG2000 as desired using the BDT editor. Drivers and functions are available for this purpose.

The rho4.0 incorporates a CAN interface to the digital drive amplifiers (Bosch Servodyn D, Servodyn GC).

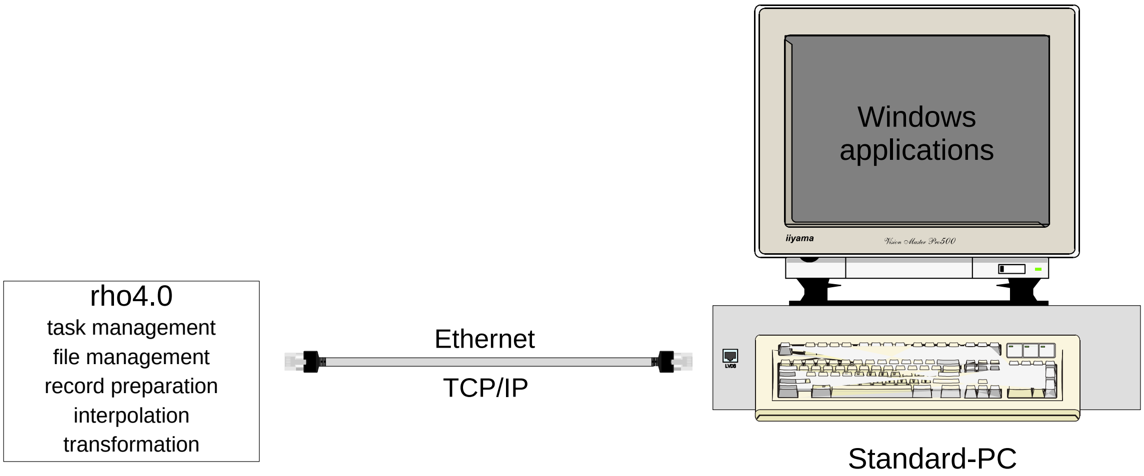
In addition, an interface for SERCOS drives is available.

A CAN interface is also used for the coupling of digital I/O. The PCLrho4.0 gives the possibility of communicating by means of PROFIBUS-DP, CAN or INTERBUS-S by using a PCI field bus card.

ROPS4 can communicate with the rho4.0 via a TCP/IP or a serially connection.
BAPS plus uses the DDE Server4.

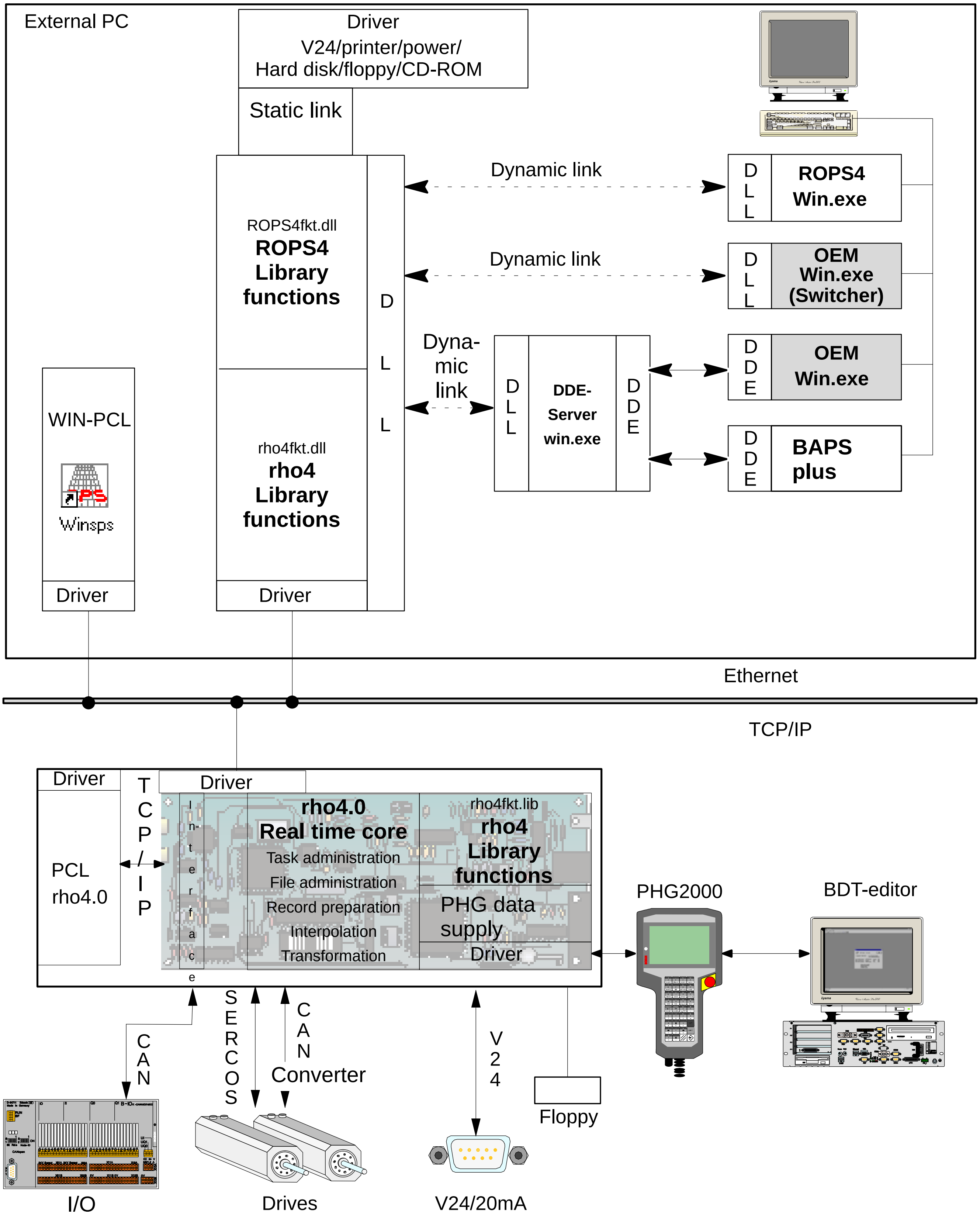
Structure of the rho4.0

3.1.1 Outline structure



Structure of the rho4.0

3.1.2 rho4.0 block diagramm



3.1.3 Operating system of the rho4.0

The operating system is saved and secured against voltage loss on a plug-in and an exchangeable compact flash disc (CF).

The change of a CF is only allowed when it is switched off by pulling the rho4.0 component. The CF is internally mounted on the component for safety reasons and can therefore not be removed from outside without intention.

In the case of change of the operating system, after the exchange of the CF, the previously saved machine parameters are to be loaded with ROPS4 and the previously saved PCLrho4.0 program with WinSPS.

3.1.4 Floppy operation

As an option an external floppy drive can be connected to the control rho4.0 as a storage medium supporting 3.5'' disks with 1.44 MB and FAT.

The disks can be read or described from the control unit as well as from Windows. On the disk, the file attribute may not be restricted, i.e. all files have the attribute A(rchiv) allocated as default.

From the control side (real-time core), the operation is made via the PHG2000.

As default the following display selection is activated:

- with the PHG mode 9.2 (list files), 9.3 (delete files), 9.4 (print files) and 9.9 (file attributes) one comes into the PHG2000 display 'Memory management'.
- with the PHG mode 9.8 (floppy management) one comes into the PHG2000 display 'Floppy management'.

Via an option byte (addr. 400062 = 1), it is possible to keep the display of the modes 9.2, 9.3, 9.4, 9.9 in the PHG3 mode.

PHG2000 display

The operating mode 'File management' gives the possibility of displaying up to 10 files of the user memory or the floppy disk at the same time on the PHG2000 display.

The file names are limited to the format 8.3. Any extension with 3 signs is allowed. Files that do not correspond to this format will be not displayed.

Directory structures of the floppy disk are not supported. All files that are in the root directory of the floppy disk are displayed. Files are copied from the control unit into the root directory of the floppy disk. Up to 224 files can be filed into the root directory. An access to subdirectories is not possible.

Structure of the rho4.0

PHG2000-Display ’Memory management’

The PHG2000 display ’Memory management’ is switched on with mode 9.2, 9.3, 9.4, 9.9.

Memory management				
GrdStell.ird	1864331	20.09.00	10:06	RWD
GrdStell.pkt	64322	18.09.00	18:12	RWD
GrdStell.qll	4313	17.06.00	19:34	RWD
Palpos01.ird	64394	10.11.99	09:54	RWD
Palpos02.ird	84385	11.11.99	11:11	RWD
Palpos05.pkt	5376	05.09.00	17:09	RWD
Pal1 .pkt	5376	05.09.00	17:09	RWD
Pal2 .pkt	6357	08.11.00	13.08	RWD
Pal12 .pkt	12356	12.10.99	16.16	RWD
Pal22 .pkt	23466	04.05.98	15.43	RWD
MB total: 4.000 occupied: 3.052 free: 0.044				
Delete rho	Print rho	Attributes rho	Copy rho –> FD	

If the PHG2000 display ’Memory management’ is active, the uppermost key row of the PHG2000 is used as function keys in the standard key assignment.

The following functions are started with it:

- ”Delete rho file“ (key code ’MOVE’)
- ”Print rho file’” (key code ’LINEAR’)
- ”Attributes rho file” (key code ’VIA’)
- ”Copy file from rho onto floppy” (key code ’TO’)

Structure of the rho4.0

PHG2000-Display ’Floppy management’

With the PHG mode 9.8 (Floppy management) the following PHG2000 display is switched on:

Floppy management			
GrdStel2.ird	2664331	20.09.00	10:06
GrdStel2.pkt	64322	18.09.00	18:12
GrdStel2.qll	4313	17.06.00	19:34
Palpos01.ird	64394	10.11.99	09:54
Palpos02.ird	84385	11.11.99	11:11
Palpos05.pkt	5376	05.09.00	17:09
Pal1 .pkt	5376	05.09.00	17:09
Pal2 .pkt	6357	08.11.00	13.08
Pal12 .pkt	12356	12.10.99	16.16
Pal22 .pkt	23466	04.05.98	15.43
MB total: 1.406 occupied: 1.354 free: 0.002			
Delete FD	Copy MP −> FD	Copy XMP −> FD	Copy FD −> rho

If the PHG2000 display ’Floppy management’ is active, the uppermost key row of the PHG2000 is used as function keys in the standard key assignment.

The following functions are started with it:

- ”Delete FD file” (key code ’MOVE’)
- ”Copy machine parameters from rho onto floppy” (key code ’LINEAR’)
- ”Copy extended machine parameters from rho onto floppy”) (key code ’VIA’)
- ”Copy files from floppy onto rho”) (key code ’TO’)

Key assignment for selection

The following key combinations correspond to the cursor movements within the file list of the display:

			Standard keyboard assignment
<up arrowl>	:	move cursor up	<Shift> <5>
<down arrow>	:	move cursor down	<Shift> < . >
< ’<’ >	:	move cursor up by a page	<Alt> <7>
< ’>’ >	:	move cursor down by a page	<Alt> <8>
<BEGIN>	:	position cursor at the beginning of the file list	<BEGIN>
<END>	:	position cursor at the end of the file list	<END>
<’[’>	:	set mark (beginning of an area)	<Alt> <WENN>
<WDH>	:	Update display (Refresh list)	<WDH>
<left arrow>	:	Quit operating mode file management	<Shift> <1>

The inversely represented file(s) is/are selected for the function key to be confirmed. Any large, coherent area over several pages can be selected. The processing of a selected area (file list) is interrupted with <left arrow>.

CAUTION
The disk may not be removed from the drive during a write access.
It can destroy (logically) the disk.

3.2 Digital I/O

On the rho4.0, (without additionally possible coupling of decentral I/O modules) 16 digital inputs and 8 digital outputs are available.

3.2.1 Digital inputs

The digital inputs of the rho4.0 are multi-function inputs. They can be used alternatively for several control functions:

Multifunction type	**can be addressed via channel number**
High-speed constant start	611 to 618
High-speed inputs in the BAPS program	801 to 816
High-speed asynchronous inputs	1801 to 1816
Processing of the inputs in the PCLrho4.0 program	−−−

High-speed constant start

For certain applications, it is necessary to trigger a movement as High-speed as possible and always with as constant idle time as possible on a condition.

By means of the “WAIT UNTIL ...” instruction − by using the input channel numbers 611 to 618 it is possible to start high-speedly and with constant delay time (with quantization of an interpolation cycle P5) on a condition.

A movement is triggered as High-speed as possible through a condition when the BAPS program contains the following instructione sequence:

```
WAIT UNTIL bedingung_erfuellt MOVE kin1 LINEAR TO P1
```

kin1:	Name of the kinematic that is moved afterward
Condition_met:	e.g. SL_INP_1=1 SL_INP_2=0

Any waiting time can be programmed before the next movement is executed:

```
.
.
1) WAIT UNTIL SL_INP_1=1 MAX_TIME=0.15

2) MOVE DEST_POS
.
.
```

After the WAIT condition is met, the movement of the set 2) with a delay of 0.15 s (150 ms) is started.

The precision resolution of the MAX_TIME is given through the interpolation cycle.

CAUTION
If the kinematic that has last moved before the WAIT-UNTIL instruction another one than the one that is to start high-speedly, the control unit must be informed via a dummy movement set which kinematic is to react high-speedly.

1) MOVE kin2 LINEAR TO P0
2) MOVE_REL kin1 LINEAR (0, 0, 0, 0); Dummy movement step
3) WAIT UNTIL condition_met
4) MOVE kin1 LINEAR TO P1
5) MOVE kin2 TO P2

If the kinematic commutation (set 2) is not programmed, the WAIT-UNTIL instruction acts only for the next movement set of the kinematic kin2, i.e. only in block 4 and block 5 would be processed without waiting.

Example: high-speed constant start

```
;;CONTROL=RHO4
;;KINEMATICS:(1=PORTAL;2=SUPPLIER)
;;PORTAL.JC_NAMES=A1,A2,A3
;;PORTAL.WC_NAMES=X1,Y1,Z1
;;SUPPLIER.JC_NAMES=BB
;;SUPPLIER.WC_NAMES=XX

PROGRAM FIXSTART

INPUT: 1=I1,                    ;Normal input
     611=SL_INP_1,              ;1st high-speed input X21.0
     612=SL_INP_2,              ;2nd high-speed input X21.1
     615=SL_INP_5,              ;5th high-speed input X21.4
     617=SL_INP_7               ;7th high-speed input X21.6

BEGIN

;;KINEMATICS=PORTAL
;;INT=LINEAR

BEG:

     MOVE SUPPLIER TO WAIT_POS
     MOVE TO START_POS
```

Structure of the rho4.0

```
     WAIT UNTIL SL_INP_1=1
     MOVE TO DEST_POS_1                    ;PORTAL (Default-Kinematic) moves
                                           ;immediately if SL_INP_1=1

     MOVE_REL SUPPLIER (0)                 ;ensures that WAIT-UNTIL instruction
                                           ;is working on SUPPLIER

     WAIT UNTIL SL_INP_2=0
     MOVE SUPPLIER TO MAGAZIN              ;SUPPLIER begins to move when SL_INP_2=0

     MOVE_REL (0,0,0)                      ;ensures that WAIT-UNTIL instruction
                                           ;is working on PORTAL

     WAIT UNTIL SL_INP_5=1
     MOVE UNTIL I1=1 TO DEST_POS_2         ;PORTAL (Default-Kinematic)
                                           ;moves immediately if SL_INP_5=1 and
                                           ;I1=0 (with I1=1 there is no movement)

     PARALLEL

        MOVE_REL SUPPLIER (0)
        WAIT UNTIL SL_INP_7=1
        MOVE SUPPLIER TO MAGAZINE_3        ;SUPPLIER and PORTAL begin to move
                                           ;immediately if SL_INP_7=1

     ALSO

        MOVE_REL (0,0,0)
        WAIT UNTIL SL_INP_7=1
        MOVE TO DEST_POS_3

     PARALLEL_END

JUMP BEG
PROGRAM_END
```

high-speed inputs in the BAPS program

Under the input channel numbers 801 to 816, the digital inputs can be processed as high-speed inputs in the BAPS program.

The high-speed inputs can be used as normal user inputs in the BAPS program, but they are acquired much more high-speedly since they are not in relation with PCLrho4.0 running times and do not depend on interpolation cycle (machine parameter P5).

Exception: the WAIT-UNTIL instruction. When there is no signal yet, the signal change is only recognized in the next interpolation cycle (P5).

Possible BAPS instructions are:

- IF input THEN
- WAIT UNTIL input
- MOVE UNTIL input
- BinaryVar=input
- SYNC Belt input

- IF STATE (’E801’) etc.

Example for an input declaration:

Input:	1	=	Inp_1	; Normal digital input
	801	=	SLInp_1	; high-speed input from X21
	1804	=	Asyn_Inp_4	; asynchronous high-speed input

High-speed asynchronous inputs

If the high-speed inputs are to be processed asynchronously − i.e. already during the set preparation −, they can be used in the BAPS program with the input channel numbers 1801 to 1816.

☞ **A maximum of 11 blocks can be processed asynchronously.**

Processing of the inputs in the PCLrho4.0 program

If the digital inputs of the rho4.0 are to be processed in the PCLrho4.0 program, this is to be adjusted in the machine parameter P36 under „ADDR.MF2 INPUT“.

ADDR.MF2 INPUT	Signification/behavior
0	Basic setting The digital inputs of the rho4.0 (X21) are copied onto the available input addresses I93.0 to I94.7 (MF2INPUT01_RCO .. MF2INPUT16_RCO).
16 to 127	Start address from which the digital inputs of the rho4.0 (X21) are copied in.
−1	Switch off copy to PCLrho4.0 Interface The digital inputs of the rho4.0 (X21) are not copied into the interface.

Under the key word P36.AdrMultifunction2Inputs, this setting can also be performed via the machine parameter converter.

Display digital inputs

The signal state of the digital inputs is displayed under PHG-Mode 7.19 ’MF2-inputs’, Byte 93 and 94.

3.2.2 Digital outputs

The digital outputs of the rho4.0 are multifunction outputs. They can be alternatively (not at the same time) used for several control functions:

Multifunction type	can be addressed via channel number
High-speed inputs in the BAPS program	801 to 816
Processing of the inputs in the PCLrho4.0 program	−−−

High-speed outputs in the BAPS program

Under the output channel numbers 801 to 808, the digital outputs can be set as high-speed outputs in the BAPS program.

The high-speed outputs can be used as normal user outputs in the BAPS program, but they are set much more high-speedly since they are not in relation with PCL running times and do not depend on interpolation cycle (machine parameter P5).

During the start of the control unit or during the input signal 'Initial position of RC', the high-speed outputs are set on ‚0'. The signal state of the high-speed outputs cannot be directly changed from the PCL, they are only addressed in the BAPS program.

Example for an output declaration:

```
OUTPUT:      1 =  Outp_1     ; Normal digital output
           801 =  SLOut_1    ; high-speed output from X11
```

In the case of the special function 1, the high-speed outputs can be transmitted as I/O number. At the moment of the switching of the special function1, the indicated value is transmitted directly to the peripheral equipment (X11).

Processing of the outputs in the PCL program

If the digital outputs of the rho4.0 are to be processed in the PCLrho4.0 program, this is to be adjusted in the machine parameter P36 under „ADDR.MF2OUTPUT".

Structure of the rho4.0

ADDR.MF2OUTPUT	Signification/behavior
0	Basic setting The available output addresses O52.0 to O52.7 are copied onto the digital outputs of the rho4.0 (X11) (MF2OUT01_RCI .. MF2OUT08_RCI).
16 to 127	Start address of the digital outputs of the rho4.0 (X11) in the PCL interface (O16.0 to O127.7).
−1	Switch off multifunction outputs of the rho4.0 (X11). The digital outputs are not used by the PCLrho4.0. This setting is required if the digital outputs of the rho4.0 are used in the BAPS programs (high-speed outputs).

Under the key word ’P36.AddrMultiFunc2Outputs’ this setting can also be performed via the machine parameter converter.

Display digital outputs

The signal state of the digital outputs is displayed under PHG−Mode 7.20 ’MF2-outputs’, Byte 52.

3.3 Operation with I/O-Gateway

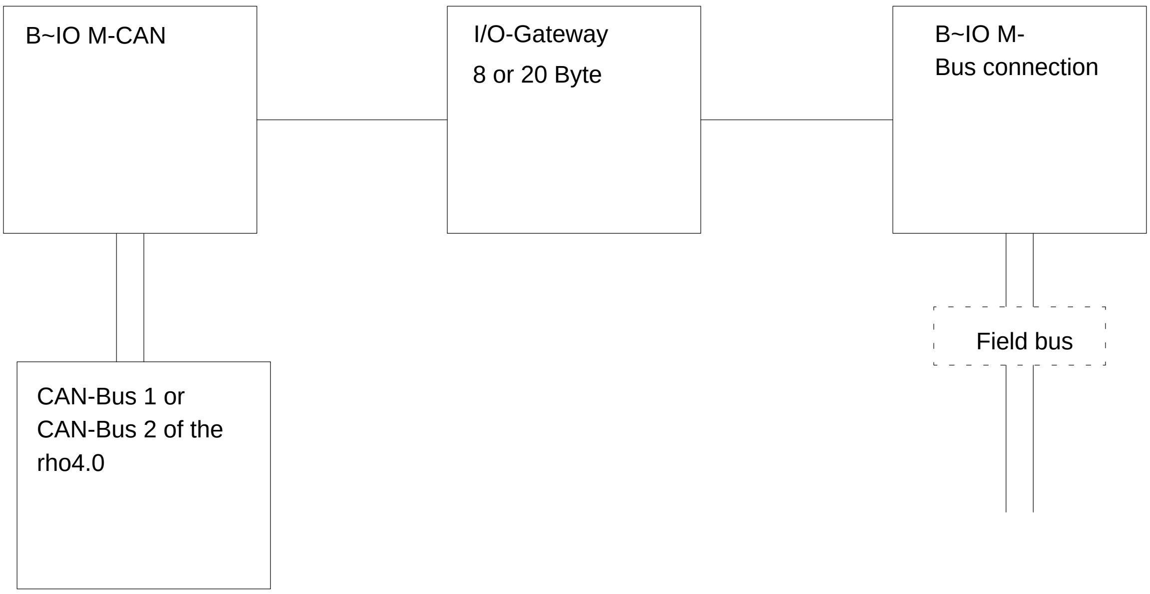
PLC controls of different busses can be connected to each other with the B~I/O gateway and the modular bus connections B~I/O M-DP or B~I/O M-IBS and B~I/O M-CAN.

A rho4.0 can also change 8 or 20 bytes inputs and outputs per B~IO M bus connection via other field busses (Profibus, Interbus S, CANopen). The selection 8 or 20 bytes can be set at the switch S1 of the I/O gateway.

☞ **The B~IO M-CAN groups can be only used from Firmware Version 1.3 for I/O-Gateway**

3.3.1 rho4.0 without PCL field bus card

The CAN bus of the rho4.0 is available for a B~IO M bus connection.



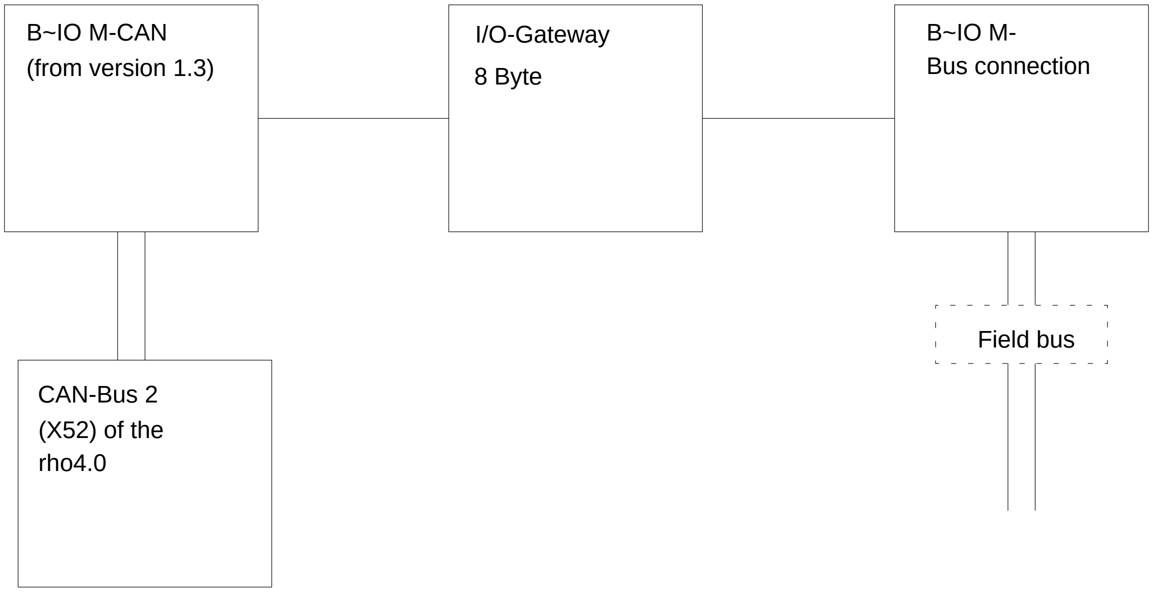
Possible combinations:

CAN-Bus of the rho4	**B~IO M-Bus connection**	**I/O-Gateway**	**B~IO M-Bus connection**	**Field bus**
Can-Bus 1 or 2	B~IO M-CAN	I/O-Gateway 8 Byte	B~IO M-DP	PROFIBUS-DP
Can-Bus 1 or 2	B~IO M-CAN	I/O-Gateway 8 Byte	B~IO M-IBS	INTERBUS-S
Can-Bus 1 or 2	B~IO M-CAN	I/O-Gateway 8 Byte	B~IO M-CAN	CANopen
Can-Bus 1 or 2	B~IO M-CAN	I/O-Gateway 20 Byte	B~IO M-DP	PROFIBUS-DP
Can-Bus 1 or 2	B~IO M-CAN	I/O-Gateway 20 Byte	B~IO M-IBS	INTERBUS-S
Can-Bus 1 or 2	B~IO M-CAN	I/O-Gateway 20 Byte	B~IO M-CAN	CANopen

3.3.2 Examples of B~IO M-CAN bus connections

Setting 8 Byte

Example of the connection of a I/O-Gateway with 8 byte setting at the CAN Bus 2 (X52) of the rho4.0 with CANrho protocol.



In the 8 byte setting at the I/O gateway, only one block inputs and one block outputs are set for the machine parameters of the rho4.0.

On the rho4.0 the following values are written in the machine parameters P30 to P32 for the connection of the CAN Bus. They can be used for all B~IO-M-bus connections with 8 Bytes.

P30 I/O Configuration
Number of the input blocks

Display PHG:

MP SET
P30 I/O-CONF. CAN
Numb.of Input-Bl.: 01

#

Structure of the rho4.0

Number of the output blocks

Display PHG:

MP SET
P30 I/O-CONF. CAN
Numb. of Out-Bl.: 01
#

Baudrate CAN-Bus 2

Display PHG:

MP SET
P30 I/O-CONF. CAN 2
Baudrate: 0
#

(Baudrate = 1 MBaud)

CANrho or CANopen CAN-Bus 2

Display PHG:

MP SET
P30 I/O-CONF. CAN 2
CANRho=0,CANopen=1 0
#

☞ **The setting CANrho must be also selected accordingly at the B~IO-M-CAN module, switch S1**

Machine parameter	**Entry Machine parameter converter**
P30 Number of the input blocks	P30.CANInpNumb=1
P30 Number of the output blocks	P30.CANOutNumb=1
P30 Baudrate CAN-Bus 2	P30.Baudrate.CANBUS2=0
P30 CANrho or CANopen CAN-Bus 2	P30.ProtocolType.CANBUS2=0

Structure of the rho4.0

P31 Address areas of the CAN inputs

IOKind (0=digital, 1=analog) of the first block

Display PHG:

```
                 MP SET
P31 ADR.CAN–I
IOKind Block 1:      0

#
```

Start address of the first block

Display PHG:

```
                 MP SET
P31 ADR.CAN–I
1stAdr.Block 1:      208

#
```

Length of the first block

Display PHG:

```
                 MP SET
P31 ADR.CAN–I
Length Block 1:      8

#
```

Identifier of the first block

Display PHG:

```
                 MP SET
P31 ADR.CAN–I
Ident. Block 1:      386

#
```

(Input identifier for Node-ID = 2, PDO1)

Structure of the rho4.0

Bus number of the first block

Display PHG:

MP SET
P31 ADR.CAN−I
Bus−no.Block 1: 2
#

Machine parameter	**Entry Machine parameter converter**
P31 Start address Block 1	P31.CANInpStAdr.Block1=208
P31 Length Block 1	P31.CANInpLeng.Block1=8
P31 Identifier Block 1	P31.CANInpIdent.Block1=386
P31 Bus number Block 1	P31.InputCANBUS.Block1=2
P31 IO type Block 1	P31.InputEAType.Block1=0

P32 Address areas of the CAN outputs

The setting of the CAN outputs occurs accordingly to the CAN inputs.

IOKind (0=digital, 1=analog) of the first block

Display PHG:

MP SET
P32 ADR.CAN−O
IOKind Block 1: 0
#

Start address of the first block

Display PHG:

MP SET
P32 ADR.CAN−O
1stAdr. Block 1: 208
#

Structure of the rho4.0

Length of the first block

Display PHG:

```
                MP SET
P32 ADR.CAN−O
Length Block  1:    8

#
```

Identifier of the first block

Display PHG:

```
                MP SET
P32 ADR.CAN−O
Ident. Block  1:    514

#
```

(Output identifier for Node-ID = 2, PDO1)

Bus number of the first block

Display PHG:

```
                MP SET
P32 ADR.CAN−O
Bus−no. Block 1:    2

#
```

Machine parameter	**Entry Machine parameter converter**
P32 start address Block 1	P32.CANOutStAdr.Block1=208
P32 length Block 1	P32.CANOutLeng.Block1=8
P32 Identifier Block 1	P32.CANOutIdent.Block1=514
P32 bus number Block 1	P32.OutputCANBUS.Block1=2
P32 EA-type Block 1	P32.OutputEAType.Block1=0

Setting S1, Bit rate

Switch 8	**Switch 7**	**Switch 6**	**Switch 5**	**Switch 4**	**Switch 3**	**Switch 2**	**Switch 1**
on	off	off	off	off	on	on	on
Setting CANrho		reserved			Baudrate	= 1 MBaud	

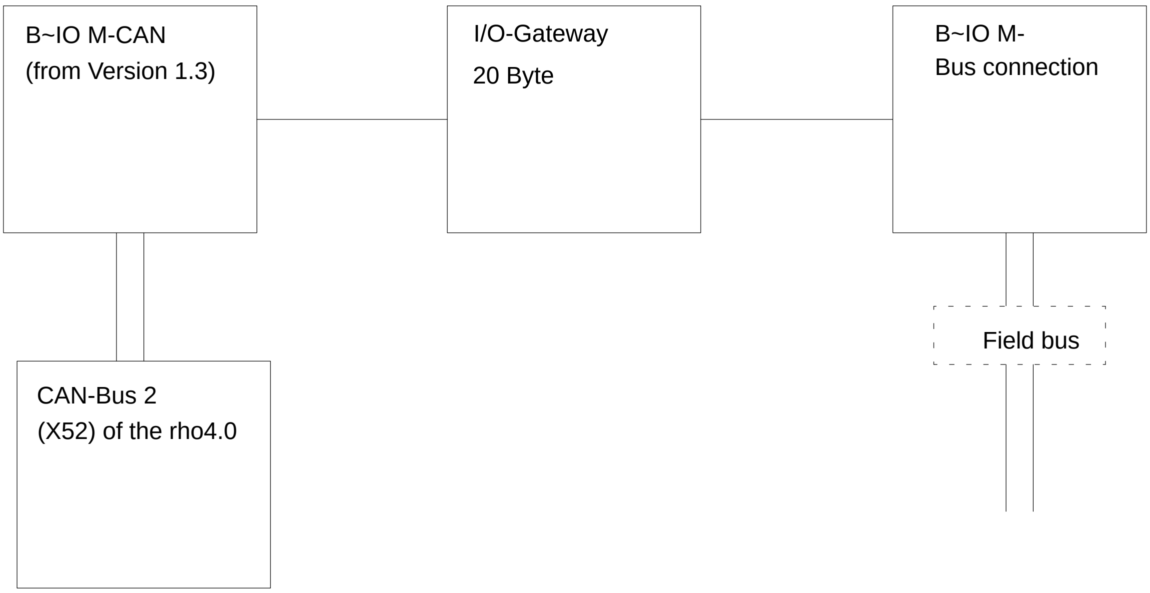
Structure of the rho4.0

Setting S2, Node-ID = 2

Switch 8	Switch 7	Switch 6	Switch 5	Switch 4	Switch 3	Switch 2	Switch 1
off	off	off	off	off	off	on	off
reserved						Node-ID	= 2

Setting 20 Byte

Example of the connection of a I/O-Gateway with 20 byte setting at the CAN Bus 2 (X52) of the rho4 with CANrho protocol.



In the 20 byte setting at the I/O gateway, only three block inputs and three block outputs are set since the maximum block length at CAN is fixed at eight bytes.

On the rho4 the following values are written in the machine parameters P30 to P32 for the connection of the CAN Bus. They can be used for all B~IO-M-bus connections with 20 Bytes.

P30 I/O Configuration

Number of the input blocks

Display PHG:

MP SET
P30 I/O-CONF. CAN
Numb. of Inp-Bl.: 03
#

Structure of the rho4.0

Number of the output blocks

Display PHG:

```
                MP SET
P30 I/O-CONF. CAN
Numb. of Out-Bl.:  03
#
```

Baudrate CAN-Bus 2

Display PHG:

```
                MP SET
P30 I/O-CONF. CAN 2
Baudrate:          0
#
```

(Baudrate = 1 MBaud)

CANrho or CANopen CAN-Bus 2

Display PHG:

```
                MP SET
P30 I/O-CONF. CAN 2
CANrho=0,CANopen=1 0
#
```

☞ **The setting CANrho must be also selected accordingly at the B~IO-M-CAN module, switch S1**

Machine parameter	**Entry Machine parameter converter**
P30 Number of the input blocks	P30.CANInpNumb=3
P30 Number of the output blocks	P30.CANOutNumb=3
P30 Baudrate CAN-Bus 2	P30.Baudrate.CANBUS2=0
P30 CANrho or CANopen CAN-Bus 2	P30.ProtocolType.CANBUS2=0

P31 Address areas of the CAN-inputs

IOKind (0=digital, 1=analog) of the first block

Display PHG:

```
                    MP SET
P31 ADR.CAN−I
IOKind Block 1:     0

#
```

Start address of the first block

Display PHG:

```
                    MP SET
P31 ADR.CAN−I
1stAdr.Block 1:     208

#
```

Length of the first block

Display PHG:

```
                    MP SET
P31 ADR.CAN−I
Length Block 1:     8

#
```

Identifier of the first block

Display PHG:

```
                    MP SET
P31 ADR.CAN−I
Ident. Block 1:     385

#
```

(Input identifier for Node-ID = 1, PDO1)

Structure of the rho4.0

Bus number of the first block

Display PHG:

	MP SET
P31 ADR.CAN−I	
Bus−no. Block 1:	**2**
#	

IOKind (0=digital, 1=analog) of the second block

Display PHG:

	MP SET
P31 ADR.CAN−I	
IOKind Block 2:	**0**
#	

Start address of the second block

Display PHG:

	MP SET
P31 ADR.CAN−I	
1stAdr.Block 2:	**216**
#	

Length of the second block

Display PHG:

	MP SET
P31 ADR.CAN−I	
Length Block 2:	**8**
#	

Structure of the rho4.0

Identifier of the second block

Display PHG:

	MP SET
P31 ADR.CAN−I	
Ident. Block 2:	**641**
#	

(Input identifier for Node-ID = 1, PDO2)

Bus number of the second block

Display PHG:

	MP SET
P31 ADR.CAN−I	
Bus−no. Block 2:	**2**
#	

IOKind (0=digital, 1=analog) of the third block

Display PHG:

	MP SET
P31 ADR.CAN−I	
IOKind Block 3:	**0**
#	

Start address of the third block

Display PHG:

	MP SET
P31 ADR.CAN−I	
1stAdr.Block 3:	**224**
#	

Structure of the rho4.0

Length of the third block

Display PHG:

```
                  MP SET
P31 ADR.CAN–I
Length Block 3:     4
#
```

☞ **Block 3 has only a length of 4 bytes, since in the 20 byte setting at the gateway already 16 bytes of blocks 1 and 2 are used**

Identifier of the third block

Display PHG:

```
                  MP SET
P31 ADR.CAN–I
Ident. Block 3:     386
#
```

(Input identifier for Node-ID = 1, PDO1 + 1)

Bus number of the third block

Display PHG:

```
                  MP SET
P31 ADR.CAN–I
Bus–no. Block 3:    2
#
```

Structure of the rho4.0

Machine parameter	Entry Machine parameter converter
P31 start address Block 1	P31.CANInpStAdr.Block1=208
P31 start address Block 2	P31.CANInpStAdr.Block2=216
P31 start address Block 3	P31.CANInpStAdr.Block3=224
P31 length Block 1	P31.CANInpLeng.Block1=8
P31 length Block 2	P31.CANInpLeng.Block2=8
P31 length Block 3	P31.CANInpLeng.Block3=4
P31 Identifier Block 1	P31.CANInpIdent.Block1=385
P31 Identifier Block 2	P31.CANInpIdent.Block2=641
P31 Identifier Block 3	P31.CANInpIdent.Block3=386
P31 bus number Block 1	P31.InputCANBUS.Block1=2
P31 bus number Block 2	P31.InputCANBUS.Block2=2
P31 bus number Block 3	P31.InputCANBUS.Block3=2
P31 EA-type Block 1	P31.InputEAType.Block1=0
P31 EA-type Block 2	P31.InputEAType.Block2=0
P31 EA-type Block 3	P31.InputEAType.Block3=0

P32 Address areas der CAN-outputs

This setting of the CAN outputs occurs according to the setting of the CAN inputs. This means that block 3 has only a length of 4 bytes, since in the 20 byte setting at the gateway already 16 bytes of blocks 1 and 2 are used.

IOKind (0=digital, 1=analog) of the first block

Length of the first block

Display PHG:

MP SET
P32 ADR.CAN−O
IOKind Block 1: 0
#

Start address of the first block

Structure of the rho4.0

Display PHG:

```
                MP SET
P32 ADR.CAN–O
1stAdr. Block 1:   208
#
```

Length of the first block

Display PHG:

```
                MP SET
P32 ADR.CAN–O
Length Block 1:      8
#
```

Identifier of the first block

Display PHG:

```
                MP SET
P32 ADR.CAN–O
Ident. Block 1:      513
#
```

(Output identifier for Node-ID = 1, PDO1)

Bus number of the first block

Display PHG:

```
                MP SET
P32 ADR.CAN–O
Bus–no. Block 1:      2
#
```

IOKind (0=digital, 1=analog) of the second block

Structure of the rho4.0

Display PHG:

	MP SET
P32 ADR.CAN−O	
IOKind Block 2:	0
#	

Start address of the second block

Display PHG:

	MP SET
P32 ADR.CAN−O	
1stAdr.Block 2:	216
#	

Length of the second block

Display PHG:

	MP SET
P32 ADR.CAN−O	
Length Block 2:	8
#	

Identifier of the second block

Display PHG:

	MP SET
P32 ADR.CAN−O	
Ident. Block 2:	769
#	

(Output identifier for Node-ID = 1, PDO2)

Bus number of the second block

Structure of the rho4.0

Display PHG:

```
                 MP SET
P32 ADR.CAN−O
Bus−no. Block 2:    2

#
```

IOKind (0=digital, 1=analog) of the third block

Display PHG:

```
                 MP SET
P32 ADR.CAN−O
IOKind Block 3:     0

#
```

Start address of the third block

Display PHG:

```
                 MP SET
P32 ADR.CAN−O
1stAdr.Block 3:     224

#
```

Length of the third block

Display PHG:

```
                 MP SET
P32 ADR.CAN−O
Length Block 3:     4

#
```

Identifier of the third block

Structure of the rho4.0

Display PHG:

	MP SET
P32 ADR.CAN−O	
Ident. Block 3:	**514**
#	

(Output identifier for Node-ID = 1, PDO1 + 1)

Bus number of the third block

Display PHG:

	MP SET
P32 ADR.CAN−O	
Bus−no. Block 3:	**2**
#	

Structure of the rho4.0

Machine parameter	Entry Machine parameter converter
P32 start address Block 1	P32.CANOutStAdr.Block1=208
P32 start address Block 2	P32.CANOutStAdr.Block2=216
P32 start address Block 3	P32.CANOutStAdr.Block3=224
P32 length Block 1	P32.CANOutLeng.Block1=8
P32 length Block 2	P32.CANOutLeng.Block2=8
P32 length Block 3	P32.CANOutLeng.Block3=4
P32 Identifier Block 1	P32.CANOutIdent.Block1=513
P32 Identifier Block 2	P32.CANOutIdent.Block2=769
P32 Identifier Block 3	P32.CANOutIdent.Block3=514
P32 bus number Block 1	P32.OutputCANBUS.Block1=2
P32 bus number Block 2	P32.OutputCANBUS.Block2=2
P32 bus number Block 3	P32.OutputCANBUS.Block3=2
P32 EA-type Block 1	P32.OutputEAType.Block1=0
P32 EA-type Block 2	P32.OutputEAType.Block2=0
P32 EA-type Block 3	P32.OutputEAType.Block3=0

Setting S1, Bitrate

Switch 8	Switch 7	Switch 6	Switch 5	Switch 4	Switch 3	Switch 2	Switch 1
on	off	off	off	off	on	on	on
Setting CANrho		reserved			Baudrate	= 1 MBaud	

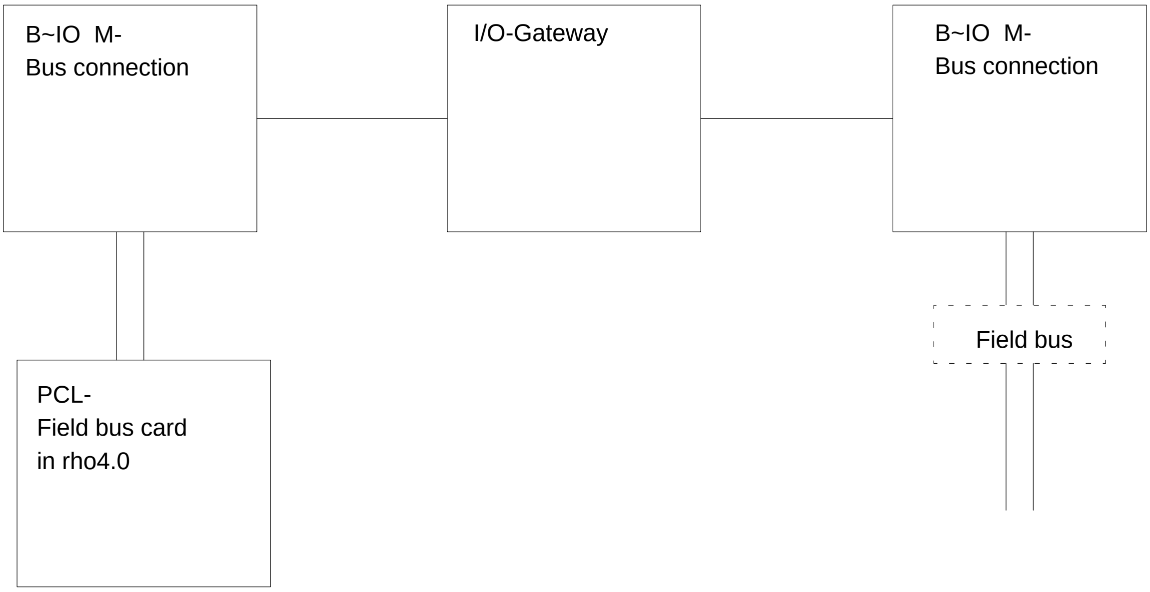
Setting S2, Node-ID = 1

Switch 8	Switch 7	Switch 6	Switch 5	Switch 4	Switch 3	Switch 2	Switch 1
off	off	off	off	off	off	off	on
reserved						Node-ID	= 1

Structure of the rho4.0

3.3.3 rho4.0 with PCL field bus card

In this variant, a rho4.0 with PCL field bus card (PCI_BM-DP, or PCI_BM-IBS, or PCI_BM-CAN) can exchange I/O information via the field busses Profibus DP or Interbus S or CANopen.



Possible combinations:

PCL- Field bus card in rho4	B~IO M-Bus connection	I/O-Gateway	B~IO M-Bus connection	Field bus
PCI_BM-DP	B~IO M-DP	I/O-Gateway	B~IOM-DP	PROFIBUS-DP
PCI_BM-DP	B~IO M-DP	I/O-Gateway	B~IO M-IBS	InterBus-S
PCI_BM-DP	B~IO M-DP	I/O-Gateway	B~IO M-CAN	CANopen
PCI_BM-IBS	B~IO M-IBS	I/O-Gateway	B~IO M-DP	PROFIBUS-DP
PCI_BM-IBS	B~IO M-IBS	I/O-Gateway	B~IO M-IBS	InterBus-S
PCI_BM-IBS	B~IO M-IBS	I/O-Gateway	B~IO M-CAN	CANopen
PCI_BM-CAN	B~IO M-CAN	I/O-Gateway	B~IO M-DP	PROFIBUS-DP
PCI_BM-CAN	B~IO M-CAN	I/O-Gateway	B~IO M-IBS	InterBus-S
PCI_BM-CAN	B~IO M-CAN	I/O-Gateway	B~IO M-CAN	CANopen

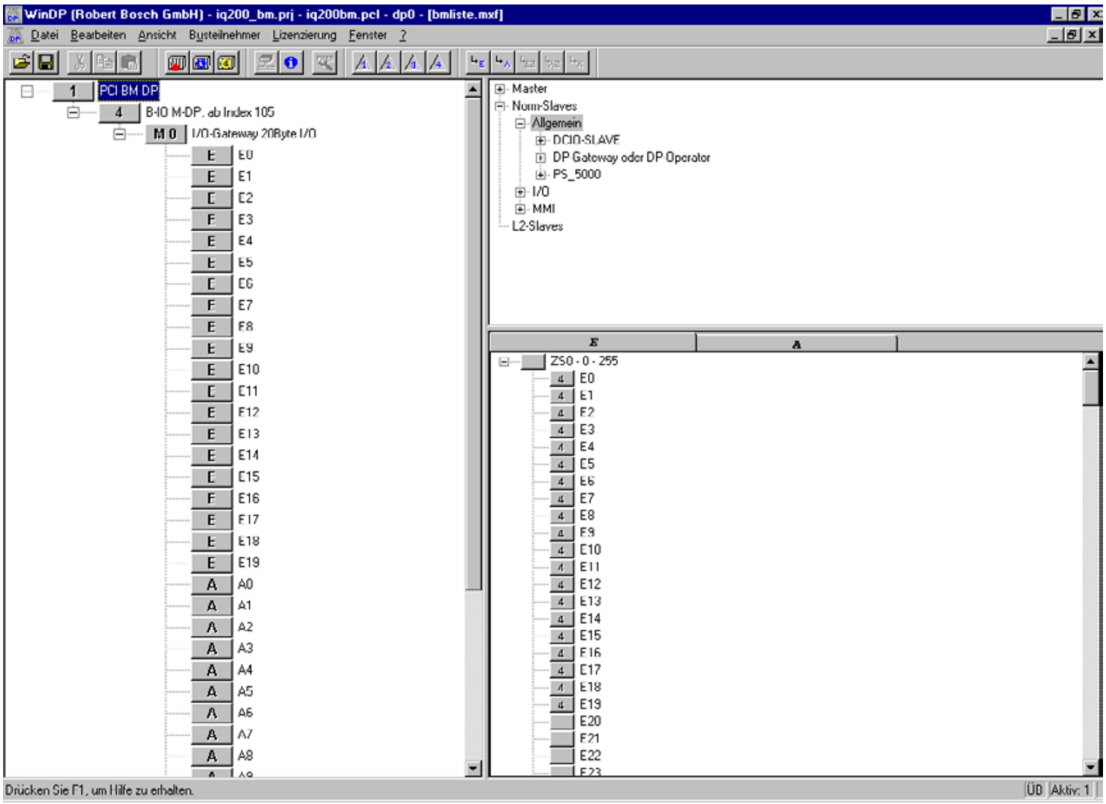
☞ **The B~IO M-CAN components can be only used from Firmware Version 1.3 for I/O-gateway**

Structure of the rho4.0

3.3.4 Example of a B~IO M-DP Profibus connection

The I/O gateway for Profibus DP is configured with the Windows tool WinDP. To be able to set there the I/O gateway, the current version of the device master file RB030133.GSD version 1.3 is required. The gateway uses here the PCL byte addresses 0..19.

The setting is performed according to the following figure.



☞ **A PCL-L licence at the Winpanel is required for the transfer of the set 20 byte**

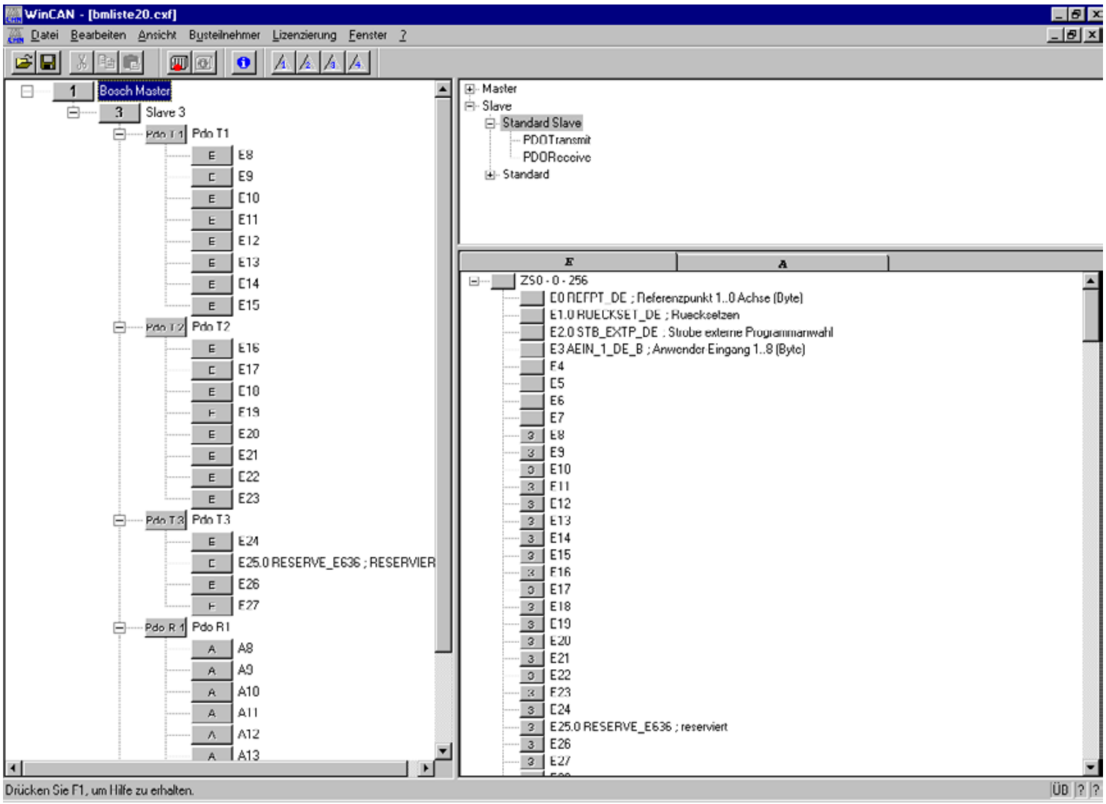
3.3.5 Example of a B~IO M-IBS bus connection

No configuration of the PCI-BM-IBS is necessary for the connection of Interbus-S since the bus master configures itself on its own. The bus master allocates addresses to the modules with increasing addresses from address 0. If a gateway is connected, the gateway uses the byte address 0 to 19.

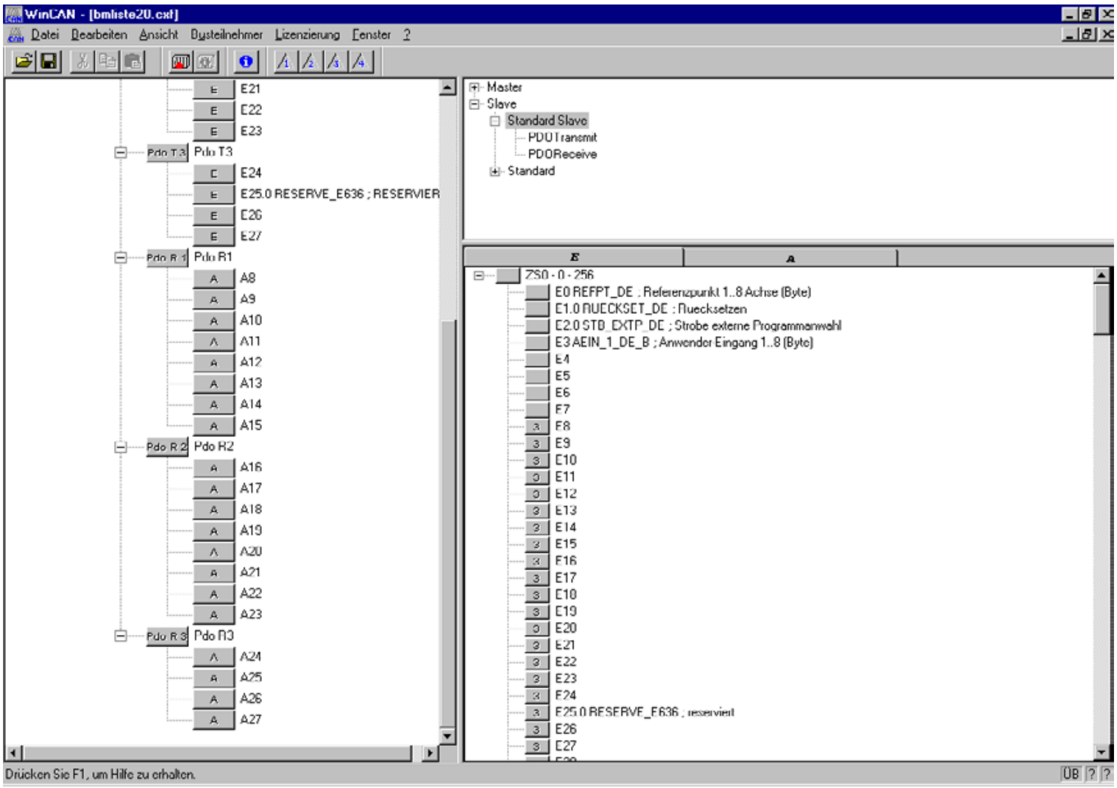
3.3.6 Example of a B~IO M-CAN bus connection

The windows tool WinCAN is used for the PCL resp. the bus master PCI_BM_CAN to configure the I/O gateway with CAN. To use the I/O gateway there for transfer, one sets for the setting 20 bytes (switch at I/O gateway) 3 input and output blocks.

The following picture shows the setting:



Structure of the rho4.0



If only 8 bytes are required for the transfer, then the switch can be set at the gateway to 8 bytes. The configuration is then considerably easier since only one input block and one output block have to be set.

3.4 Fixed IP addresses

The rho4.0 is delivered with the set IP address 192.168.4.1. Via the PHG, mode 9.1.19 the address can be changed by the user at any time.

The address of the PCLrho4.0 can be changed via mode 9.1.11. At the time of the delivery, 127.0.0.1 is the default setting.

3.5 Small linking with switches

To combine several rho4.0 in a small network, switches can be used.

3.5.1 Cabling

For the direct connection from a rho4.0 to a second rho4.0 (without network, without switch), it is possible to use e.g. the following cable:

- 2,5 m long Ethernet cable 10BaseT crossed with the order number 1070919188

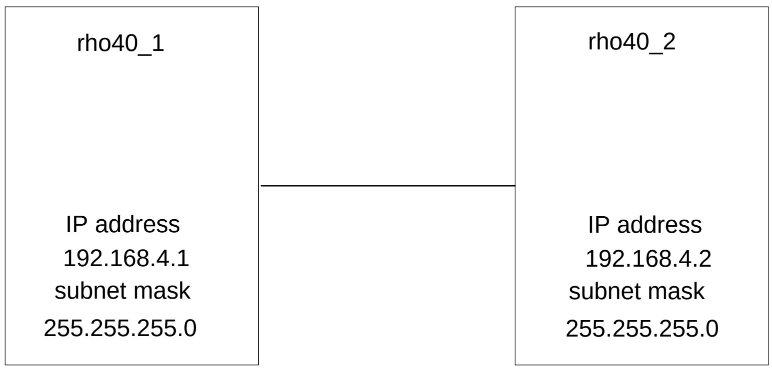
For the connection from switch to the rho4.1, it is possible to use e.g. the following cables:

- 2,5 m long Ethernet cable 10Base-T uncrossed with the order number 1070918793
- 5 m long Ethernet cable 10Base-T uncrossed with the order number 1070919258
- 10 m long Ethernet cable 10Base-T uncrossed with the order number 1070921384
- 25 m long Ethernet cable 10Base-T uncrossed with the order number 1070918796

3.5.2 Linking of two rho4.0

Connection diagram

To be able to link exactly 2 rho4.0, they must be connected with a crossed Ethernet cable 10Base-T.



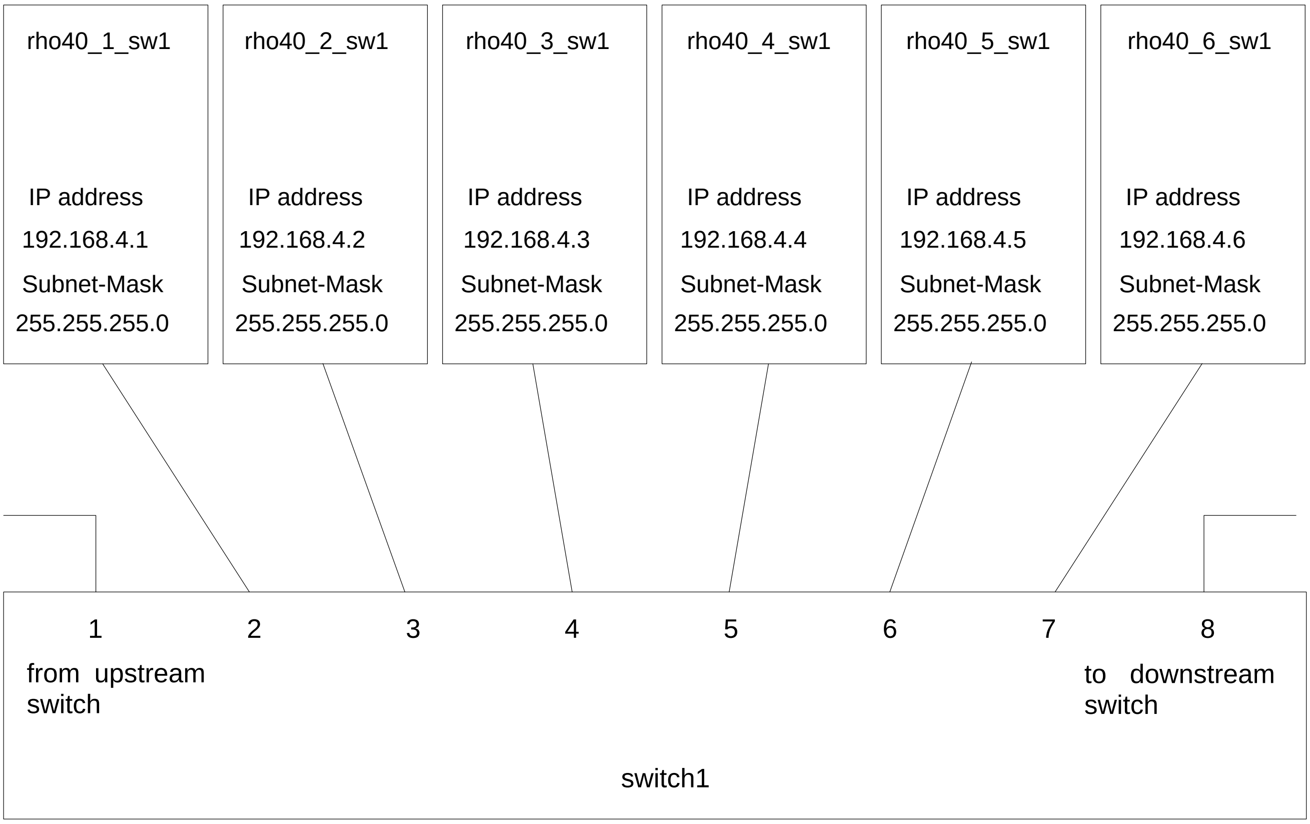
IP address and subnet mask have to be entered via PHG, mode 9.1.19.

3.5.3 Smallest linking of up to 6 rho4.1 with 8-fold switch

With a 8-fold switch, up to 8 rho4.0 can be linked. To be able to extend the smallest network to a small network, 2 switch connections are kept free.

Connection shema

For the connection of the rho4.0 to the switch, uncrossed 1:1 cables must be used.



IP address and subnet mask have to be entered via PHG, mode 9.1.19.

3.5.4 Small linking of several rho4.0 with 8-fold switch

Several rho4.0 can be linked with 8-fold switches through cascading of the switches. The maximum cascading number according to the manufacturer indication is to be taken into account.

According to the regulations according to IEEE 802.3, a maximum of 4 switches can be cascaded according to the following connection diagram.

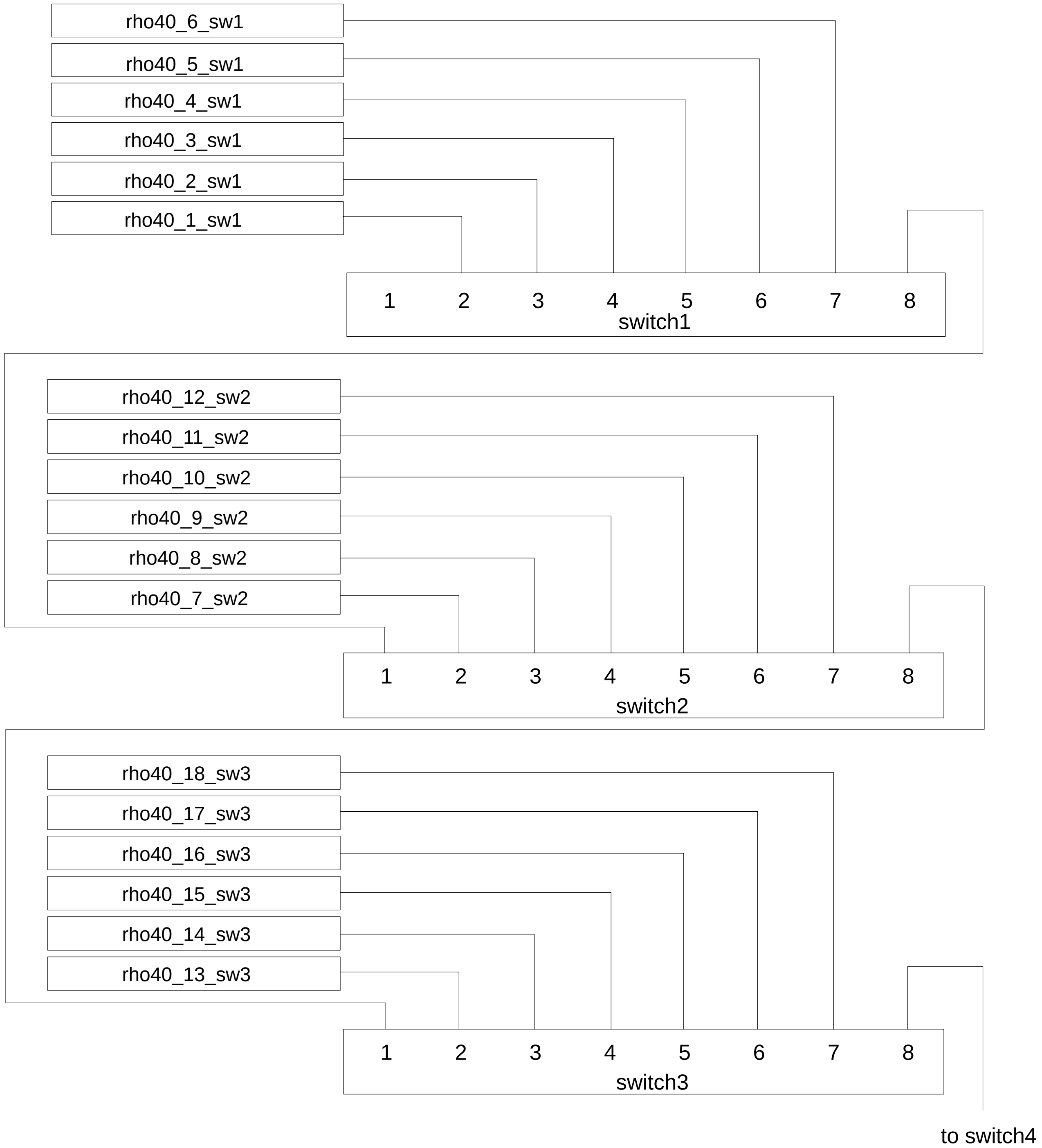
If more rho4.0 are to be linked than it is possible to connect to 4 switches maximum, a Backbone Network Topology with coaxial Ethernet cable must be used.

Connection diagram

For the connection of the rho4.0 to the switch, uncrossed 1:1 cables are to be used.

☞ **For the connection from switch to switch, a power controller for crossed or uncrossed cables must be taken into account if there is one. If the controller is set on uncrossed, uncrossed cables may be used.**

Structure of the rho4.0



IP address and subnet mask have to be entered via PHG, mode 9.1.19.

Structure of the rho4.0

3.5.5 Remote control with ROPS4 and virtual PHG

In a permanent network, the ROPS4 running on an external PC can be coupled to the rho4.0.

With the virtual PHG, the rho4.0 can be operated via TCP/IP.

Settings on the rho4.0

★ At the PHG under MODE 9.1.1 enter the coupling:

Type (0=Ser/1=Win):		Win. chan
Port number	:	6010
Interface	:	0

★ At the PHG under MODE 9.1.17 enter Virt_PHG:

Type (0=Ser/1=Win):		Win. chan
Port number	:	6100

Settings on the remote PC

★ In the file 'hosts' assign all rho4.0 that are to be remote-controlled with alias names and IP address.

★ Edit the alias name of the rho4.0 which are to be remote controlled in the file C:\Bosch\rho4\Office\VirtPHG.ini in section [VirtPHG] at the item "ServerAlias" (e.g. rho40_1).
At the item "ServerPortNo" the same port number which was edited before at the PHG under MODE 9.1.17 must be inserted here (e.g. 6100)

☞ **The file 'hosts' is to be found in the Windows directory, e.g. C:\Winnt\system32\drivers\etc**

Connecting ROPS4

★ Start ROPS4 to connect to the desired rho4.0.

★ In the setup of ROPS4-Online under [Options][Setup...] enter the rho4.0 with alias name that is to be remote-controlled:

Connection	:	TCP/IP
IP-Address/Alias name	:	rho40_1
Port number	:	6010

In this case, the rho4.0 with the alias name rho40_1 would be remote-controlled.

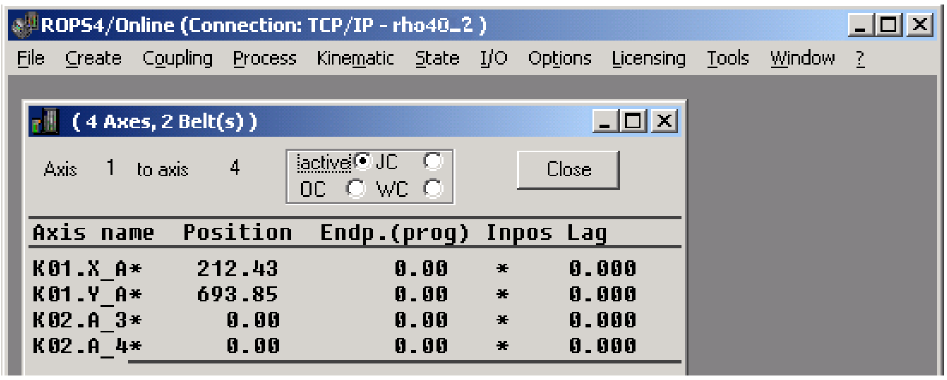
Structure of the rho4.0

If several rho4.0 are to be remote-controlled at the same time, ROPS4 can be started several times with alias names that are differently set.

When ROPS4/Online is connected with a rho4.0, the alias name and the connection type are displayed in the header. Via this display, the ROPS4 started several times can be distinguished.

Here ROPS4/Online is connected via TCP/IP with the rho4.0 with the alias name rho40_2.

Header with indication of alias name (rho40_2) and connection type (TCP/IP)



4 PCLrho4.0

The rho4.0 uses an integrated PCLrho4.0 which provides its own I/O map and access to rho4.0 I/O. The PCLrho4.0 is programmed with WinSPS (from version 3.1), a PLC programming tool.

The PCLrho4.0 has only master functionality.

There is no serial interface to the computer coupling available for the PCLrho4.0. The coupling of WinSPS occurs via the ethernet interface.

Are available as residual data in the PCLrho4.0:

- 8 kByte marker
- 32 data modules
- 16 kByte of data field

The PCLrho4.0 functionality is available in the extension stages PCLrho4.0-S, PCLrho4.0-L and PCLrho4.0-X:

- PCLrho4.0-S: 16 k statement in AWL, 16 Byte inputs and outputs
- PCLrho4.0-L: 64 k statement in AWL, 256 Byte inputs and outputs
- PCLrho4.0-X: 128 k statement in AWL, 8 Byte inputs and outputs

A detailed description is to be found in the PCL documentation (PCL and CL550, Programming manual).

PCLrho4.0 user programs are programmed with WinSPS instruction lists (IL), ladder diagram (LD) or sequential function chart (SFC). Variable names of user inputs and outputs in BAPS programs should be processed in the PLC symbol file under the same name.

The OEM can activate program modules in the PCLrho4.0 from BAPS. There are various instructions for controlling these program modules from BAPS.

PLC user tasks with fixed time pattern can be activated from BAPS user programs for quick, cyclical operation of the peripherals.

This function is implemented by programming time-controlled modules. These modules are provided on the PCLrho4.0 for the purpose of time-controlled program processing and can be called up by means of prepared program modules within a fixed time pattern.

The I/O takes place within an interpolation cycle as long as the PLC cycle time is smaller than the rho4.0 interpolation cycle. The reaction time to I/O depends on the processor used as well as the OEM configuration, number of movement systems, axes or MMI visual display.

If the rho4.0 hardware version has no PCL field bus card, the decentral I/O modules are connected to the corresponding CAN bus of the rho4.0.

PCLrho4.0

The following PCL field bus cards can be alternatively plugged into the rho4.0 hardware:

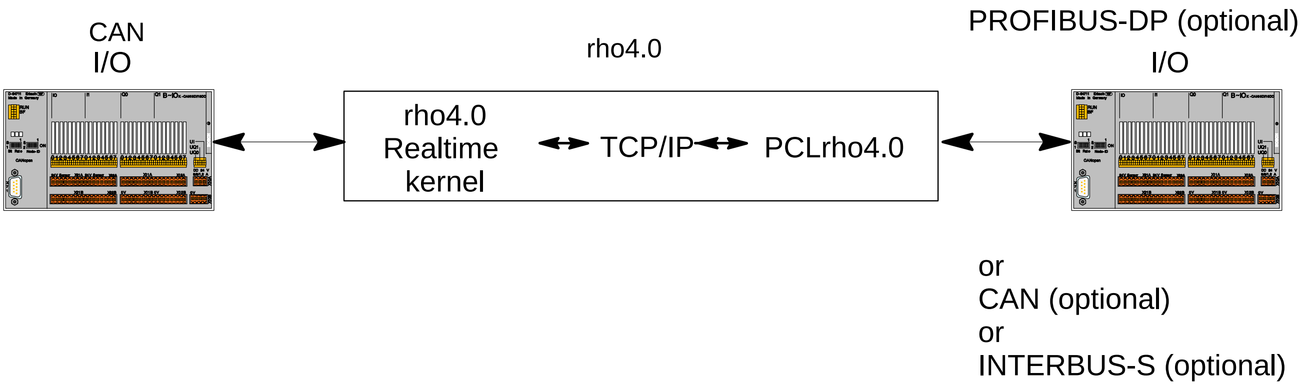
- PCI_BM_DP
- PCI_BM_CAN
- PCI_BM_IBS

Besides the PCL functionality, the corresponding field bus (PROFIBUS-DP, CAN or INTERBUS-S) is also available.

☞ **The PCLrho4.0 is programmed with WinSPS from version 3.1, since the control type PCLrho4.0 can be adjusted there. (Program files of the PCL can be adopted from a rho4.1 project as an option. For this, the check box ’Bausteine aus anderem Projekt übernehmen’ (Adopt components from another project) in the dialog ’Neue Steuerung anlegen’ (Create new control) is used.)**

rho4.0 with PCLrho4.0 Signal exchange

In the rho4.0, the rho4.0 operating system and the PLCrho4.0 are implemented on a dedicated hardware. The rho4.0 communication interface ensures the exchange of I/O and user data.



PCLrho4.0

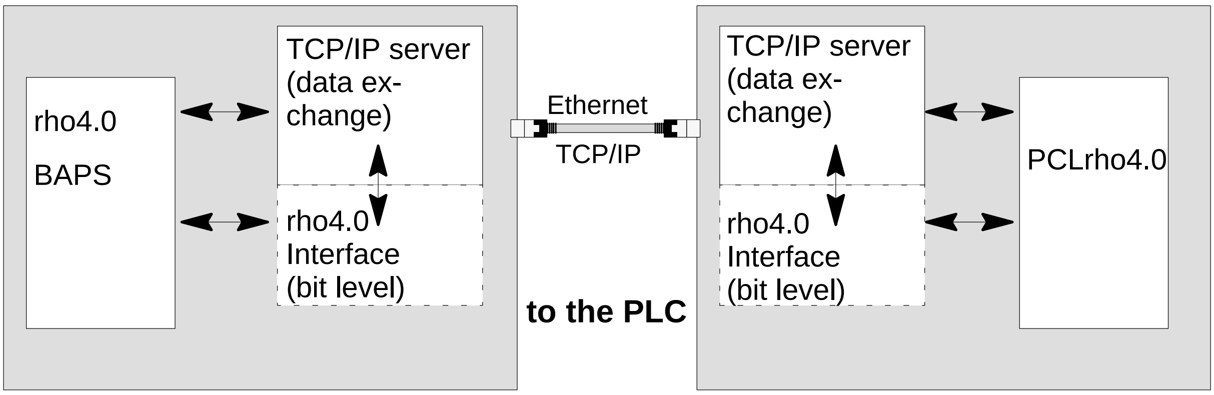
Data channel

In addition to the transmission of signalling information (bit level), data which can be accessed from BAPS can be exchanged with the PLC.

For this, there must be access to the data from BAPS.

In addition to the rho4 interface, the rho4.0 has its own data channel, which is able to transmit larger quantities of data from/to the PLC (READ/ WRITE PLC).

The TCP/IP server used with the rho4.0 is also used at the Ethernet connection to implement this data channel. The server packages data into protocol blocks and sends them to the PLC destination station. The following figure shows the data flow between the rho4 and PLC over the TCP/ IP data channel.

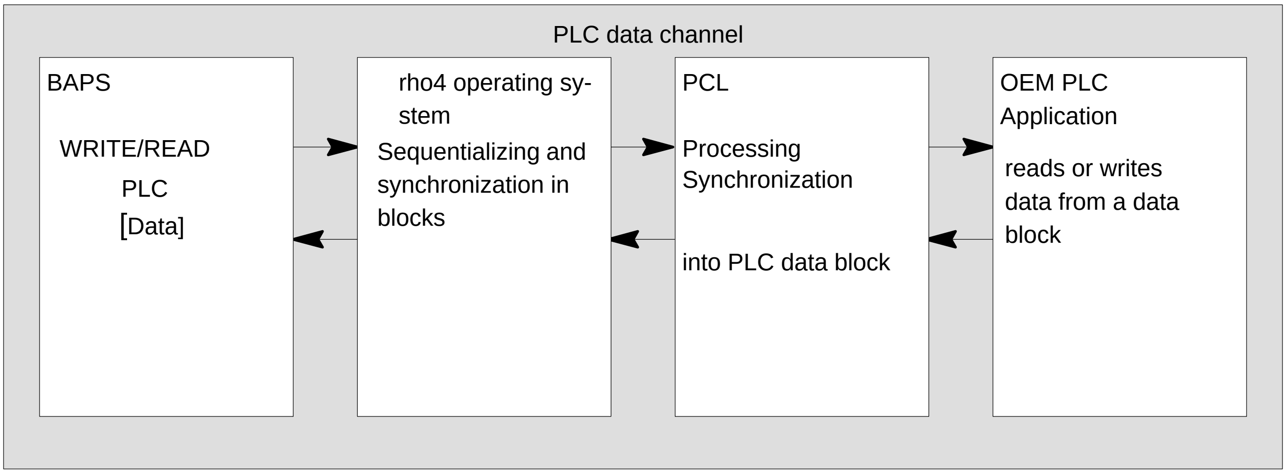


PCLrho4.0

Programming the data channel to the PLC coupling

Programming via the data channel does not depend on the type of coupling for the Ethernet/system bus. The OEM only programs the type of data to be transmitted. The data from the rho4.0 can be read out or written in on the PCLrho4.0, depending on the data block number entered in BAPS, see example.

The following figure shows the basic operation of the PLC data channel.



BAPS example program for access to PLC data via PLC data channel

```
   ; .
RECORD                                ;Record declaration in declaration part
   INTEGER : LENGTH                   ;Data length of data buffer to be transferred
   INTEGER : DM_NO                    ;Data module No. of PLC
   REAL    : REAL_VALUE               ;Components of record
   POINT   : PNTVAR
   INTEGER : INTEGERVAR
   ; .
   ; .
RECORD_END: READ_RANGE                ;Declaration of record variables
RECORD                                ;Declaration of record
   INTEGER : LENGTH                   ;Data length of data buffer to be transferred
   INTEGER : DM_NO                    ;Data module No. of PLC
   INTEGER : IVAR                     ;Components of record
```

PCLrho4.0

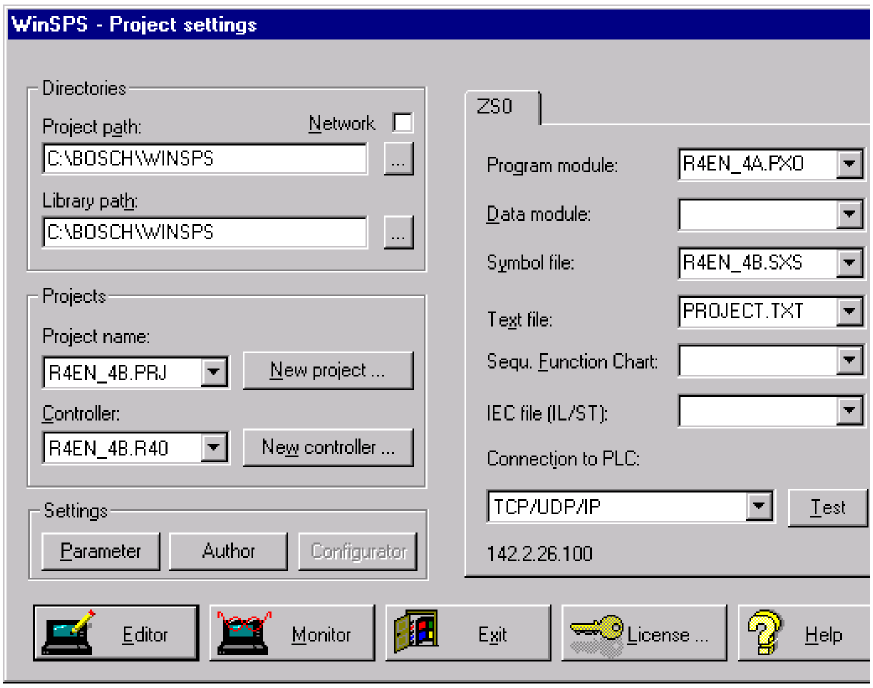
```
   REAL     : DVAR
   POINT    : PNTVAR
   ; .
RECORD_END: WRITE_RANGE            ;Declaration of record variables
REAL    : X_DVAR                   ;Auxiliary variables
INTEGER: Y_IVAR
BEGIN                              ;Length for the PLC write-range
   WRITE_RANGE.LENGTH=SIZEOF (WRITE_RANGE)
   WRITE_RANGE.DM_NO=1             ;Data module No. 1 for PLC write-range
   WRITE_RANGE.IVAR=128            ;Initialization of the components
   WRITE_RANGE.RVAR=1234.23        ;of the PLC write-range
   WRITE_RANGE.PNTVAR=STARTPOS     ;Point from PNT file
   WRITE PLC,WRITE_RANGE           ;Transfer of write-range to PLC
   READ_RANGE.LENGTH=SIZEOF(READ_RANGE)
                                   ;Length of PLC read-range
   READ_RANGE.DM_NO=2              ;Data module No. 2 for PLC read-range
   READ PLC,READ_RANGE             ;Transfer of read-range from PLC
   X_RVAR=READ_RANGE.RVAR          ;Reading out of the variables from the
   Y_IVAR=READ_RANGE.IVAR          ;PLC read-range
   Z_PNT=READ_RANGE.PNT
   ; .
PROGRAM_END
```

PCLrho4.0

4.1 PCLrho4.0 programming with WinSPS from version 3.1 (build 1406)

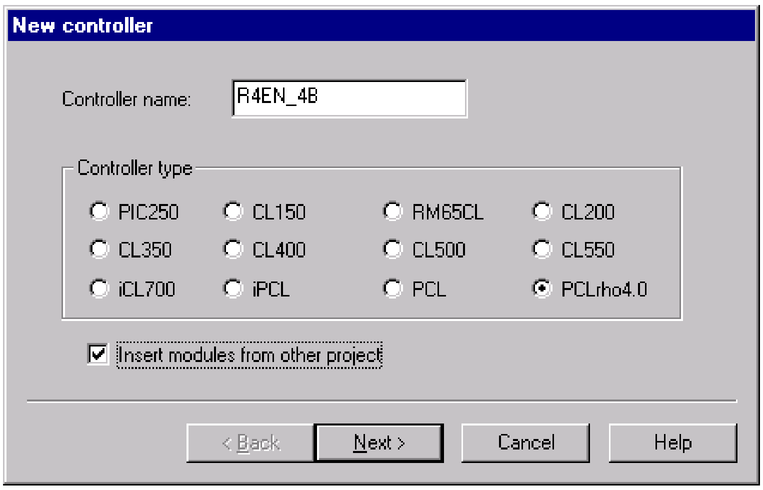
★ Select WinSPS to get into the project settings.

★ Select desired project. The rho4 makes R4EN_4B.PRJ available.



★ If no project has been yet created with the control PCLrho4.0, select with the button 'New controller' the control type 'PCLrho4.0'.

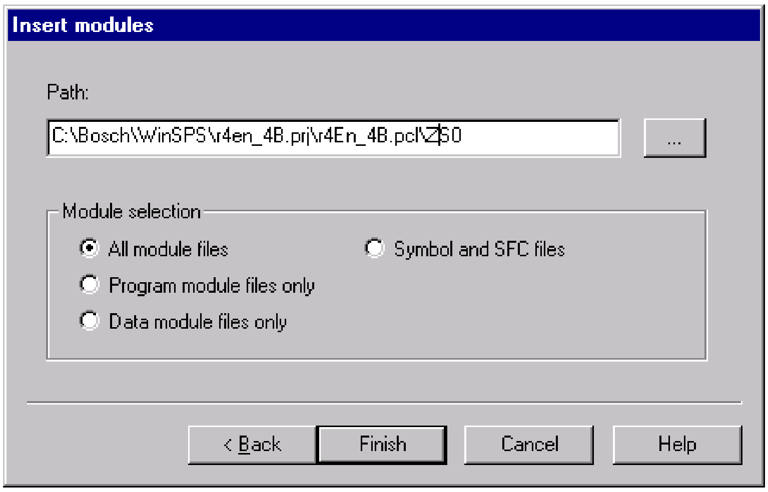
☞ **In the project R4EN_4B.PRJ a program with the identification R4EN_4B.R40 has been already created**



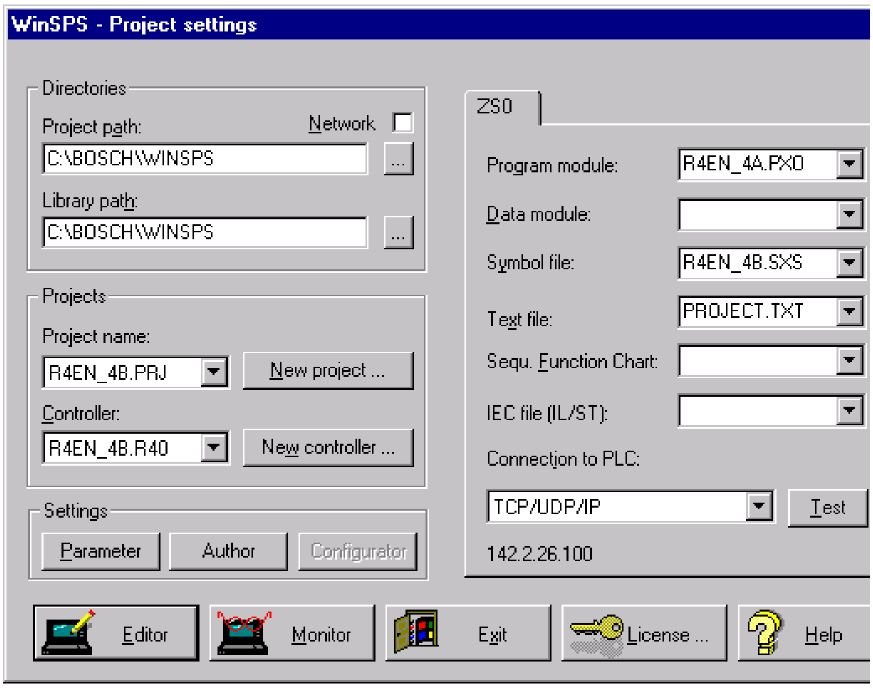
PCLrho4.0

★ Click on the checkbox next to 'Insert modules from other project' to adopt a PCL program that is already existing.

★ Via browser adjust the path of the already existing PCL program.



★ Actuate the button 'Finish' to get back into the project settings of WinSPS.



The control 'PCLrho4.0' is characterized through the identification R40 in the WinSPS project.

☞ **In the project R4EN_4B.PRJ a program with the identification R4EN_4B.R40 has been already created**

PCLrho4.0

★ With the button ’Test’ adjust the TCP/IP address of the rho4.0 to load the selected program via [Editor],[Control−>Load] into the PCLrho4.0.

☞ **The TCP/IP address can be read at the PHG2000 under mode 9.1.19**

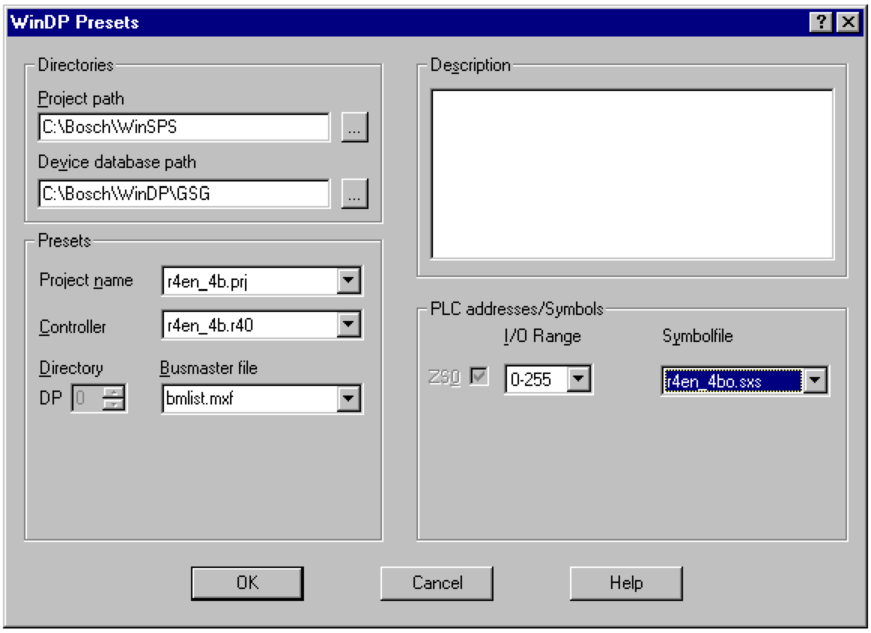
PCLrho4.0

4.2 PCLrho4.0 configuration with WinDP from version 2.10 (build 677)

To configure the PCI-BM-DP, the device master data ’Datei RB010104.GSD, Version 1.1 vom 12.03.2001’ is required.

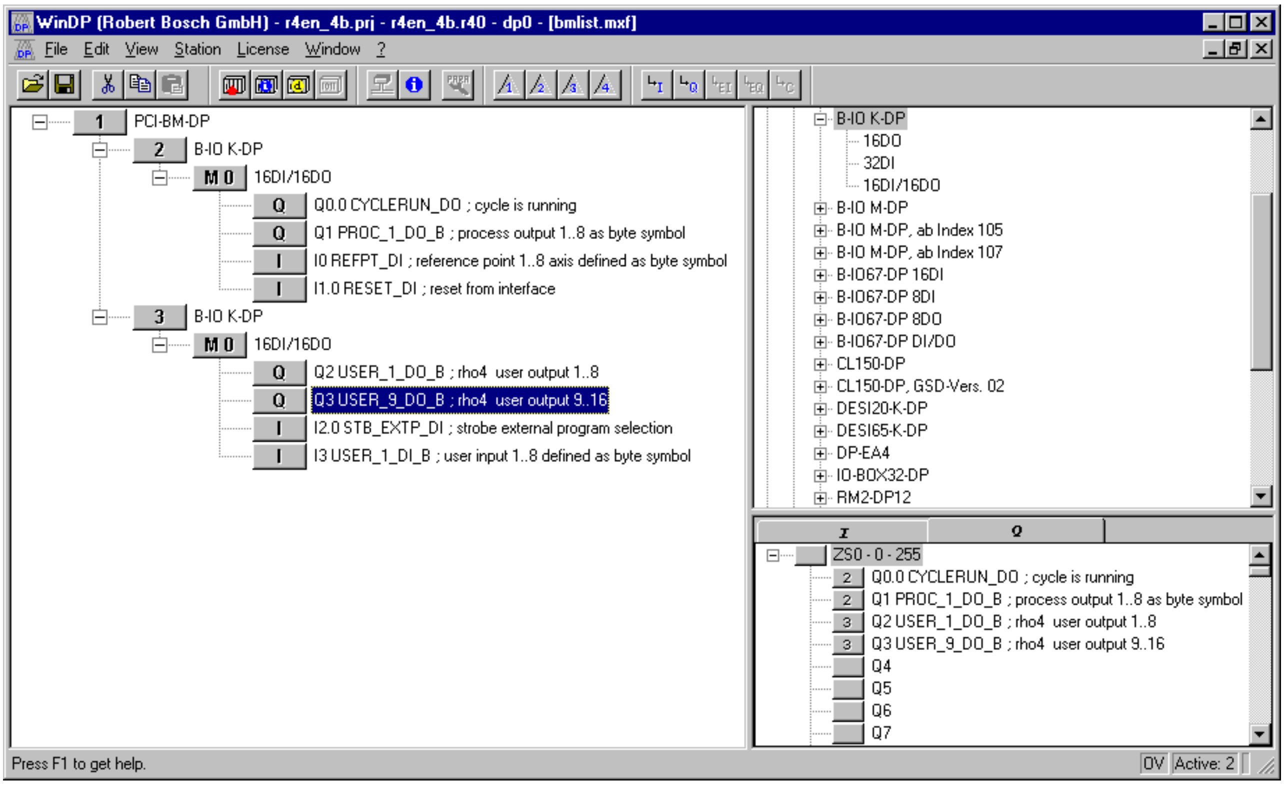
The device master data must be deposited in the directory ’Device database path’ in the pre-settings WinDP (see picture below).

As control, r4en_4b.r40 must be selected.



After setting the I/O area 0-255 and the corresponding symbol file r4en_4bo.sxs, the bus master is configured.

PCLrho4.0

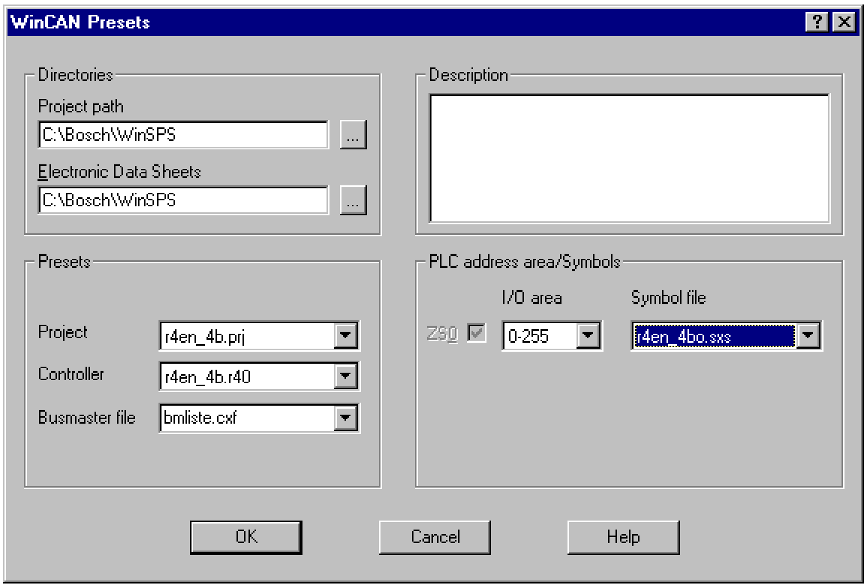


Possible configuration for two B~IO K-DP 16DI/16DO, appropriate to the PCLrho4.0 standard program R4En_4b.

PCLrho4.0

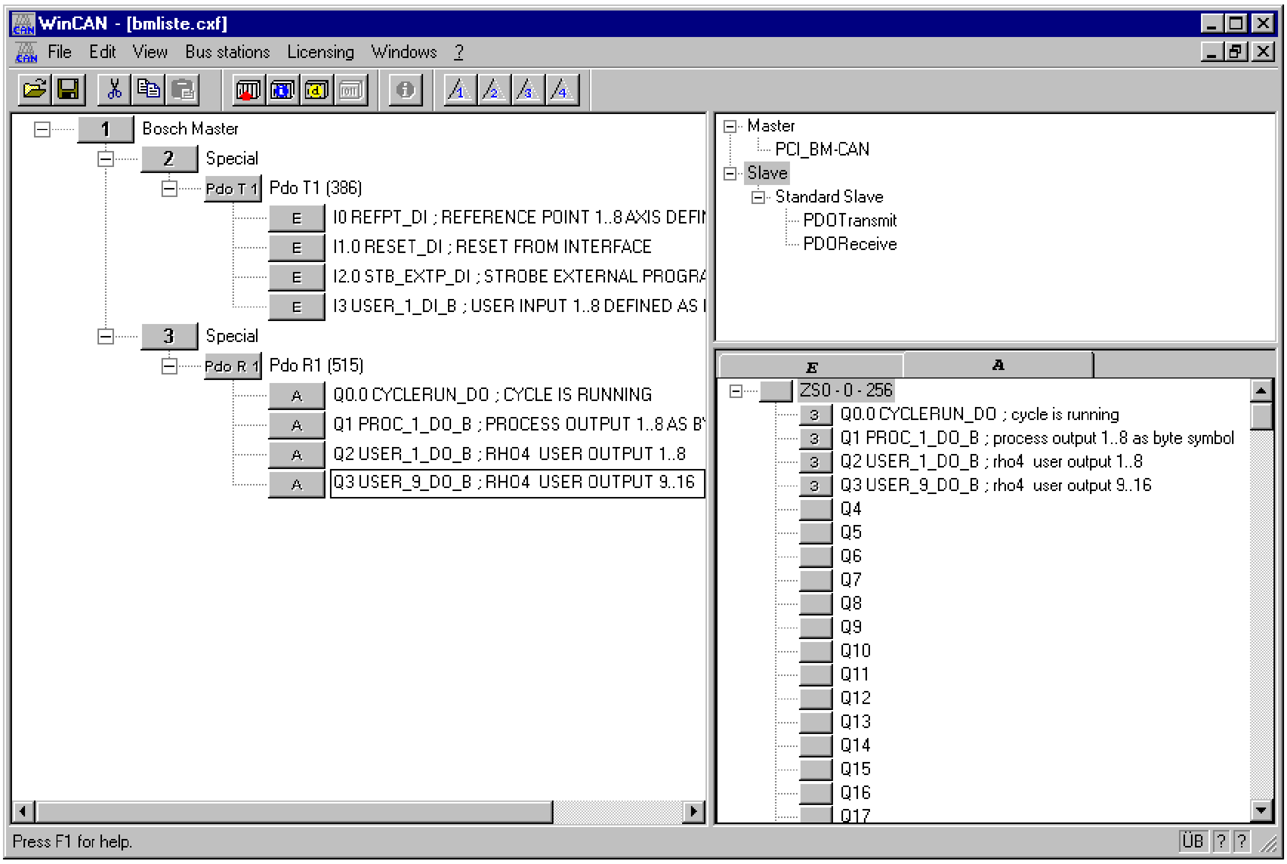
4.3 PCLrho4.0 configuration with WinCAN 1.1 (build 92)

The configuration of the PCLrho4.0 with WinCAN is required for the use of a PCI-BM-CAN component. As controller, r4en_4b.r40 must be selected.



After setting the I/O area 0-255 and the corresponding symbol file r4de_4bo.sxs, the bus master is configured.

PCLrho4.0



Configuration for 32 inputs and outputs, appropriate to the PCLrho4.0 standard program r4de_4b.

4.4 Operation of the PCLrho4.0 with INTERBUS-S

When using a PCI_BM-IBS component for the operation of the PCLrho4.0 with INTERBUS-S, there is the following possibility for the I/O configuration:

- Physical addressing of the inputs and outputs, i.e. all connected I/O components are addressed in a free increasing order.

☞ **A data exchange is limited to the inputs I0 to I511 and the outputs O0 to O511. Higher addresses cannot be used**

PCLrho4.0

Notes:

5 CAN-Bus peripheral equipment

5.1 CANopen-Interface

The CANopen interface is realized according to the guide lines of the CiA (CAN in Automation International Users and Manufacturers Group e. V.). Besides the operation of CANopen encoders, the coupling is supported by local I/O modules. The CANopen protocol cannot be operated parallel to the existing CanRho protocol at the same bus.

5.1.1 Functions

I/O modules

The I/O modules are operated according to the guide lines of the CIA Draft Standard Proposal DSP401. The data exchange of the I/O modules with the control occurs synchronously to the CANopen Sync Telegramm. Which objects from the Object Dictionary (Machine parameters of the I/O modules) are supported by the single modules, is to be read in the corresponding data sheets.

These objects are described in the same ASCII file as the drives and loaded at the startup of the control in the I/O modules (download of the I/O modules parameters).

If a I/O module from Bosch is used, Default parameters are available, which make a description of the I/O modules in the ASCII file unnecessary. To make these default parameters active, a “0” must be entered in the subparameter DssReference of the parameter 401.

SR-CAN-module

The accesses to the machine parameters and to the serial number in the SrCan module (SDO services i.e. Service Data Objects) are performed by means of the CANopen protocol. There are no additional settings required.

CANopen encoder

Detailed information, see paragraph 5.2.

5.1.2 Machine parameters

The CANopen interface is activated via machine parameter. A distinction is made between control parameters and IO parameters, i.e. there are two different, separated machine parameter files that are loaded in the control during the startup and automatically saved during the switchoff of the control. Both files are loaded by the user via the ROPS4 coupling into the control. The handling of both files does not differ.

CANopen specific control parameters

The following rho4 machine parameters concern the CANopen interface

- The transfer protocole is to be named as subparameter of the machine parameter P30. If a CAN bus is to be operated with a CANopen protocole, ’Protocol type = 1’ is to be set. If a CanRho Protocole is to be used, the subparameter ’Protocol type = 0’ is to be set.
- For each identifier of the local I/O modules, a reference to the entries in the Dss is required in the machine parameters P31 and P32. In the Bosch Dss program, all CANopen nodes are designated as ’AXIS(Index)’. The entries behind this designation apply for all CANopen nodes (e.g axes, local I/O modules, valves, ...). The index (DssRef) is introduced as subparameter of the machine parameters P31 and P32. Each index is allowed to appear only once per CAN-Bus.

CANopen specific I/O modules parameters

The CANopen specific I/O modules parameters are loaded in the startup phase of the control into the I/O modules. A ASCII file (file extension “.scs” created with the Bosch Dss or a text editor is used as basis.

For each peripheral unit (axis, encoder, I/O modules), an ASCII file must be created, in which the CANopen specific parameters are set. The number contained in the name of the ASCII file corresponds to the Dss-Ref (see also description XMP converter). The I/O-Parameter files are to be filed under the following names on the remote pc:

C:\Bosch\rho4\CANopen**Axis1.scs**

.

.

C:\Bosch\rho4\CANopen**Axis*n*.scs**

CAN-Bus peripheral equipment

Example of a scs−file for an I/O−Module:

```
PHASE = 0
PHASE = 2
AXIS = 1

    C−1400.2 = 1          ; [ ] Transmission Type   1. receive PDO
    C−1401.2 = 1          ; [ ] Transmission Type   2. receive PDO
    C−1800.2 = 1          ; [ ] Transmission Type   1. transmit PDO
    C−1801.2 = 1          ; [ ] Transmission Type   2. transmit PDO

                          ; Programming I/O module mode from
                          ; asynchronous to synchronous.

                          ; Relation between NodeId (Parameter
                 ;        ; 31,32) and DssRef => axis(x)  x = DssRef
                          ; for additional I/O modules at the bus the
                          ; contents of the file (axis(x+1)) is the same
                          ; All additional parameter for the module
                          ; can be placed here.

PHASE = 4
```

If a I/O modules from Bosch is used, Default parameters are available, which make a description of the encoder objects in the ASCII file unnecessary. To make these default parameters active, a 0 must be entered in the subparameter DssRef of the parameter 401.

The I/O modules are operated synchronously.

A data exchange occurs for each connected modules in each SYNC telegram.

5.2 CANopen-encoder

Permissible physical belt inputs are also CANopen encoders that comply with the guide lines of the CiA (Can in Automation International Users and Manufacturers Group e. V.). These measuring system inputs can also be occupied by several belts.

Encoders recommended by Bosch:

FRABA:	5812-4096-FBA1C203PG Multiturn encoder with 4096 revolutions and 4096 steps per revolution
T+R:	TR-ECE-TI-D-0035/V001 CE 65M 110-01990 Multiturn encoder with 4096 revolutions and 4096 steps per revolution

5.2.1 Functions

The encoders are driven according to the guide lines of the CIA Draft Standard Proposal DSP-406. The synchronization between control and the connected encoders is realized by means of the CANopen Sync Telegram. After sending the Sync Telegram, the encoders involved send each a position real value. Which objects from the Standardized Encoder Profile Area supporting the encoders used can be read in the corresponding data sheets.

These machine parameters are described in an ASCII file, converted, and loaded in the encoders in the startup of the control (parameter download). This ASCII file has a special format which is described under the XMP converter (extended machine parameter converter). (See also Machine parameter Software Manual, XMP converter.)

If an encoder recommended by Bosch is used, Default parameters are available, which make a description of the encoder objects in the ASCII file unnecessary. To make these default parameters active, a “0” must be entered in the subparameter DssReference of the parameter 401.

Further settings of the encoder such as baud rate, load, identifier are to be read in the technical informations of the encoder manufacturer.

CAN-Bus peripheral equipment

5.2.2 Machine parameter

The CANopen interface is activated via machine parameters. A distinction is made between control parameters and encoder parameters, i.e. there are two different, separated machine parameter files that are loaded in the control during the startup.

CANopen specific control parameters

The following rho4 machine parameters concern the CANopen interface:

- The transfer protocole is to be named as subparameter of the machine parameter P30. If a CAN bus is to be operated with a CANopen protocole, ’Protocol type = 1’ is to be set. If a CanRho Protocole is to be used, the subparameter ’Protocol type = 0’ is to be set. If CANopen encoders are used, ’Protocol type = 1’ (CANopen) is to be set.
- Settings of the machine parameter P401:
  In the case of CANopen axes or encoders, as a reference of the Dss (diagnosis & service system) the DssRef is to be taken as a further subparameter of the machine parameter P401. For each axis and each number, the corresponding number in the Dss must be indicated. For each peripheral unit (axis, encoder, I/O modules), an ASCII file must be created, in which the CANopen specific parameters are set. The number contained in the name of the ASCII file must be indicated in the rho4 parameter Dss Ref to create the relation between the rho4 and the ASCII file (see also description XMP converter).
- In the case of CANopen encoders, the subparameter of the machine parameter P401 ’drive type’ = 5 must be set.
- The identifiers for the actual values of the encoder to be read result by setting the subparameter Node-Id. The procedure is analog to the CAN axes:

  Identifier (actual value) = 384 + Node-ID

☞ **It must be ensured that there are not several identifiers on one bus. This applies to all CANopen modules involved in the bus.**

CAN-Bus peripheral equipment

Example for Machine parameter 401 :

4 axes and a CANopen encoder

A01	Servo card	1
A01	CAN-plug number	X51
A01	CAN-modules number	1
A01	CAN-axis number	1
A01	Drive type	2
A01	Encoder dist. per rotation	1.00
A01	Reference mode	0
A01	modulo value	100
A01	Measuring system factor	1000.0
A01	Command value output	1
A01	Dss-reference	1
A02	Servo card	1
A02	CAN-Plug number	X51
A02	CAN-modules number	1
A02	CAN-axis number	2
A02	drive type	2
A02	Encoder dist. per rotation	1.00
A02	Reference mode	0
A02	modulo Wert	100
A02	Measuring system factor	1000.0
A02	Command value output	2
A02	Dss-reference	2
A03	Servo card	1
A03	CAN-Plug number	X51
A03	CAN-modules number	1
A03	CAN-axis number	3
A03	drive type	2
A03	Encoder dist. per rotation	1.00
A03	Reference mode	0
A03	modulo value	100
A03	Measuring system factor	1000.0
A03	Command value output	3
A03	Dss-reference	3

CAN-Bus peripheral equipment

A04	Servo card	1
A04	CAN-Plug number	X51
A04	CAN-modules number	1
A04	CAN-axis number	4
A04	drive type	2
A04	Encoder dist. per rotation	1.00
A04	Reference mode	0
A04	modulo value	100
A04	Measuring system factor	1000.0
A04	Command value output	4
A04	Dss-reference	4
B01	Servo card	1
B01	CAN-Plug number	X51
B01	CAN-modules number	1
B01	CAN Node-ID	7
B01	drive type	5
B01	Number of rotations	4096
B01	Position offset	0.0
B01	Pulse/rotation	4096
B01	Measuring system factor	1000.0
B01	Dss-reference	0

CAN-Bus peripheral equipment

CANopen specific encoder parameters

The encoder parameters are written in ASCII files and are to be filed under the following names:

...\Axis1.scs	e.g. drive parameter axis 1 (DssRef = 1)
...\Axis(n).scs	e.g. encoder parameters (DssRef = n)

☞ **The path can be selected freely.**

Example of a scs−file for an encoder:

```
PHASE = 0
PHASE = 2
AXIS = 1

    C−1005.0 = 0x80000080  ; enable sync-Message
    C−6200.0 = 0           ; transmission period for all asynchronous
                           ; PDO’s disabled

                           ; Programminng of an encoder
                           ; to synchronous mode

                           ; Relation between NodeId (Parameter
                 ;         ; 31,32) and DssRef => axis(x)  x = DssRef

                           ; for additional encoder at the bus the con−
                           ; tents of the file (axis(x+1)) is the same
                           ; All additional parameter for the encoder
                           ; can be placed here.

PHASE = 4
```

By calling the XMP converter (in ROPS4 Online), a machine parameter binary file is created, which can then be transferred into the rho4 per coupling function. These parameters are transferred only in the initialization phase per download to the peripheral units (axes, encoders, I/O modules), they are otherwise not required in the control.

The relation between CANopen specific encoder or drive parameters and control parameters is established by the DssRef in the machine parameter P401. The adjusted DssRef indicates the corresponding ...\Axis(n).scs file.

5.3 CAN belts

Physical belt inputs can be the measuring system inputs or the Actual− or Command values of the regulated CAN axes (Servodyn-G, Servodyn-D and CANopen) as well as CANopen encoders. All supported measuring systems can be occupied by several belts. The setting occurs via the parameter 401 in analogy to the regulated axes.

5.3.1 Functions

The following measuring system inputs resp. actual− or command values of a regulated axis can be used as belt inputs in a single or even multiple path:

- Actual/Command value of a regulated Servodyn-D-axis
- Actual/Command value of a regulated CANopen-axis
- Actual/Command value of a regulated SERCOS-axis (see description Sercos interface, chapter 6)
- Measuring system input of the incremental measuring system
- Measuring system input of a CANopen encoder

5.3.2 Machine parameters

The setting of the machine parameter occurs via the parameters 401 and the group 500.

In the subparameter of P401 CAN axis number, the controlled axis indicated is the axis from which the measuring system is used for the belt logic.

If the measuring system of a CANopen encoder is used as a belt input, the CAN Node-Id instead of the axis number must be entered (see also section 5.2). In order to use the measuring system input of a CANopen encoder in a multiple path, the same CAN Node-Id must be entered.

The drive type of the belt must be adjusted as follows:

Servodyn-G-axis/belt	:	0
Servodyn-D-axis/belt	:	1
CANopen-axis/belt	:	2
CANopen-encoder	:	5

The modulo value for CANopen axes (see also CANopen description for axes) must match the value of the corresponding regulated axis.

CAN-Bus peripheral equipment

Example 1 for Machine parameter 401

4 axes and the measuring system input of the 3rd axis (CANopen axis) as belt

A01	Servo card	1
A01	CAN-Plug number	X51
A01	CAN-modules number	1
A01	CAN-axis number	1
A01	drive type	2
A01	Encoder dist. per rotation	1.00
A01	Reference mode	0
A01	modulo value	100
A01	Measuring system factor	1000.0
A01	Command value output	1
A01	Dss-reference	1
A02	Servo card	1
A02	CAN-Plug number	X51
A02	CAN-modules number	1
A02	CAN-axis number	2
A02	drive type	2
A02	Encoder dist. per rotation	1.00
A02	Reference mode	0
A02	modulo Wert	100
A02	Measuring system factor	1000.0
A02	Command value output	2
A02	Dss-reference	2

CAN-Bus peripheral equipment

A03	Servo card	1
A03	CAN-Plug number	X51
A03	CAN-modules number	1
A03	CAN-axis number	3
A03	drive type	2
A03	Encoder dist. per rotation	1.00
A03	Reference mode	0
A03	modulo value	100
A03	Measuring system factor	1000.0
A03	Command value output	3
A03	Dss-reference	3
A04	Servo card	1
A04	CAN-Plug number	X51
A04	CAN-modules number	1
A04	CAN-axis number	4
A04	drive type	2
A04	Encoder dist. per rotation	1.00
A04	Reference mode	0
A04	modulo value	100
A04	Measuring system factor	1000.0
A04	Command value output	4
A04	Dss-reference	4
B01	Servo card	1
B01	CAN-Plug number	X51
B01	CAN-modules number	1
B01	CAN-axis number	3
B01	drive type	2
B01	modulo value	100
B01	Measuring system factor	1000.0

Example 2 for Machine parameter 401

1 axis and 1 CANopen encoder, the measuring system input of which is used for 2 belts

CAN-Bus peripheral equipment

A01	Servo card	1
A01	CAN-Plug number	X51
A01	CAN-modules number	1
A01	CAN-axis number	1
A01	drive type	2
A01	Encoder dist. per rotation	1.00
A01	Reference mode	0
A01	modulo value	100
A01	Measuring system factor	1000.0
A01	Command value output	1
A01	Dss-reference	1
B01	Servo card	1
B01	CAN-Plug number	X51
B01	CAN-modules number	1
B01	CAN-Node-ID	5
B01	drive type	5
B01	Number of rotations	4096
B01	Pos-Offset	0
B01	Pulse per rotation	4096
B01	Measuring system factor	1000.0
B01	Dss-reference	0
B02	Servo card	1
B02	CAN-Plug number	X51
B02	CAN-modules number	1
B02	CAN-Node-ID	5
B02	drive type	5
B02	Number of rotations	4096
B02	Pos-Offset	0
B02	Pulse per rotation	4096
B02	Measuring system factor	1000.0
B02	Dss-reference	0

CAN-Bus peripheral equipment

5.4 CAN Interface X53 (only rho4.0L)

General

The rho4.0L, rho4.0LR, rho4.0LX and rho4.0LRX (information about the various rho40 variants you find in manual “Rexroth Rho4.0 Connectivity manual”) have an additional 3rd CAN interface. This interface however is restricted usable. The interface can only be used as belt input.

Admissible physical belt inputs are CANopen encoder, which meet the directives of the CiA (Can in Automation International Users and Manufacturers Group e. V.).

Enocers recommended by Bosch:

FRABA:	5812-4096-FBA1C203PG Multiturn encoder with 4096 revolutions and 4096 steps per revolution
T+R:	TR-ECE-TI-D-0035/V001 CE 65M 110-01990 Multiturn encoder with 4096 revolutions and 4096 steps per revolution

Parameter

P 30 (I/O-Conf. CAN)
The CAN-chipspecific parameter (baudrate, kind of protocol) of the 3rd CAN interface are prompted additionally.

P 31/P 32 (Addr. CAN-I/ Addr. CAN-O)
With P 31/P 32 it is possible to enter a “3” for the bus number

P 401 Equipment of the measuring system board
With P401 it is possible to enter “X53” for the CAN plug number, resp. a “3” for the CAN module number

4 Compatibility rho4.1 <=> rho4.0

It is not possible to activate the 3rd CAN interface by the machine parameter program at a rho4.1.
If parameter of a rho4.0 or parameter created by ROPS4, with activated 3rd CAN interface, are loaded into a rho4.1, the following message appears after the restart ”CAN-I/O-Block undef.” (Code 19328)” resp. ”incorr.comp. servo-b” (Kode 13056).

5.5 Analog I/O with CAN B~IO modules

For the processing of analog I/O signals, the coupling of analog modules is realized in the rho4. Two CAN busses are integrated which can be used for the communication of the rho4 with the analog modules.

It is possible to measure voltages or currents and process in decimal BAPS variables. For this purpose, there are two input modules variants that can be set on voltage or current measurement.

Decimal BAPS variables and further data such as axis positions, path speed and after-running can be output on analog outputs. There are two different analog output modules available for current and voltage output.

5.5.1 Bus switching

The modular field bus switching B~IO M-CAN can be driven by means of the CANopen protocol or also with the Bosch rho conformal CAN protocol. The protocol is selected by the setting of the machine parameter P30.

With a bus switching, different analog I/O modules can be combined and operated. A combination with digital I/O modules is possible. The bus switching B~IO M-CAN supports at the maximum 32 bytes inputs and outputs.

The bus switching is fixed with the I/O modules on a DIN-top hat rail. It is always on the left as the first modules. The I/O modules are fixed on the right next to the bus switching on the rail and connected with a modules connector with the corresponding neighbor modules. (See also Description of the modular bus switching B~I/O-M-CAN).

5.5.2 Analog I/O modules

Analog input modules

The analog input modules are available (see also manual ‘Input/Output modules for CL150, B~IO’):

4AI_UI
4AI_UIT

Each of these modules has 4 analog input channels with 2 bytes data each. It can be set either on voltage measurement or on current measurement. The setting applies to all 4 channels in the same way.

The measuring ranges can be adjusted via DIP switches (e.g ±10 volts). A detailed description of the setting possibilities are to be found in the documentation of the corresponding modules.

The data size of the 2 byte measuring values can be set on Straight Binary or 2-complement representation. Additionally, it is possible to choose between flush left and flush right representation in the 2 byte word in the input modules ‘4AI_UI’, since the resolution in this modules is 12 bits (including signs).

The input modules ‘4AI_UIT’ has a resolution of 14 bits (including signs) with flush left representation in the 2 byte word.

☞ **To make during the operation the handling of several different I/O modules, it is recommended to set all data sizes of all modules on flush left representation.**

The data sizes set by means of the hardware must be adjusted in the rho4 in the machine parameter P407 (‘Assignment of analog inputs’, subparameter ‘Format’ and ‘Nominal value’). See page 5−22, Assignment of the analog input channels.

Analog output modules

The analog output modules are available with 4 output channels at 2 bytes each (see also Manual ‘Input/Output modules for CL150, B~IO’):

4AO_I
4AO_U

The current and voltage output ranges can be set via DIP switches (e. g. 0 to 20 mA for current output modules and ±10 volts for voltage output modules). Detailed description of the setting possibilities, see documentation of the corresponding modules.

The data size of the 2 byte measuring values is defined for the current output modules on Straight Binary with a resolution of 16 bits.

For the voltage output modules 4AO_U it is possible to choose for a resolution of 12 bits between a representation in the 2-complement or Straight Binary and additionally between flush left and flush right representation in the 2 byte word.

☞ **To make during the operation the handling of several different I/O modules, it is recommended to set all data sizes of all modules on flush left representation.**

The data sizes set by means of the hardware must be adjusted in the rho4 in the machine parameter P405 (‘Assignment of analog inputs’, subparameter ‘Format’ and ‘Nominal value’). See page 5−23, Assignment of the analog input channels.

5.5.3 Machine parameter settings

I/O-Configuration of the CAN-Bus (P30)

In machine parameter P30, a block is to be planned for each analog B~IO-modules. A CAN block consists of 8-Byte data, which corresponds to 4 analog channels. After the entry of P30, the address areas of the I/O blocks in P31 and P32 must be adjusted.

Address areas of the CAN inputs (P31)

The image of the analog CAN inputs is deposited in the rho4 in a separate RAM area. Machine parameter P31 configurates the CAN inputs.

For each analog input block, the following subparameters are to be assigned:

- I/O-type:
  Analog modules get the code 1, digital modules the code 0.
- Start address:
  This relative address indicates the position of the data block (8 bytes) within the rho4 internal RAM area.
  The setting is to be performed as follows:

1stAdr.Block 1:	0
1stAdr.Block 2:	8
1stAdr.Block 3:	16
1stAdr.Block 4:	24
1stAdr.Block n:	(n−1) * 8

In this RAM area, only analog input blocks are taken into account. For digital input blocks, the start address relates to the rho interface.

CAN-Bus peripheral equipment

- Block length:
  The block length has for all analog modules that are available at the moment 8 bytes (4 channels).
- Identifier:
  A bus switching can supply a maximum of input blocks (32-I-byte) and 4 output blocks (32-O-Byte).
  The CAN identifier for the corresponding input block results from the node-ID set by means of the hardware of the B~IO M-CAN-modules:

Input Block	Input Byte	CAN-Identifier
1	1 to 8	384 + Node-ID
2	9 to 16	640 + Node-ID
3	17 to 24	384 + Node-ID + 1
4	25 to 32	640 + Node-ID + 1

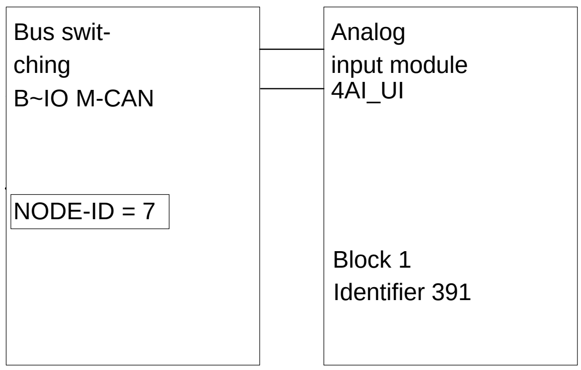
- CAN-Bus-Number:
  The CAN-Bus number indicates with which of the both CAN busses fo the rho4 the B~IO modules are operated. Only the numbers 1 and 2 are authorized. They correspond to the CAN plug number X51 and X52.

Examples: CAN input parameter

Example1:
A bus switching B~IO M-CAN is operated together with an analog input modules 4AI_UI. The node-ID of the B~IO M-CAN modules is set via the DIP switch on the value 7. The module is connected via the CAN plug X52 to the rho4.

As CAN identifier to be set, it follows:

384 + 7= 391



CAN-Bus peripheral equipment

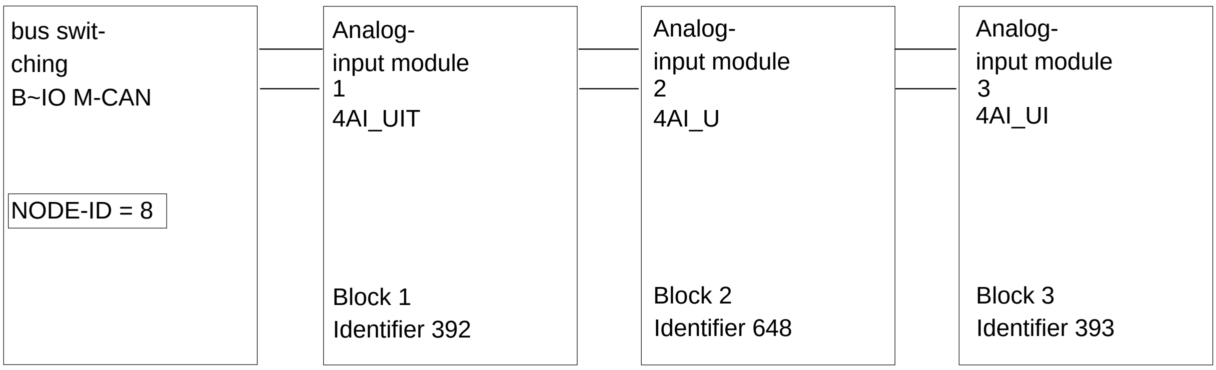
For the configuration, the following settings are to be entered under machine parameter P31:

IOKind Block 1	:	1
1stAdr.Block 1	:	0
Length Block 1	:	8
Ident. Block 1	:	391
Bus-no.Block 1	:	2

Example2:
A bus switching B~IO M-CAN is operated together with 3 analog input mo-dules. The node ID of the B~IO M-CAN module is set via the DIP switch on the value 8. The module is connected via the CAN plug X52 to the rho4.

As CAN identifiers to be set, it follows:

Block1: 384 + 8 = 392
Block2: 640 + 8 = 648
Block3: 384 + 8 + 1 = 393



For the configuration, the following settings are to be entered under machine parameter P31:

IOKind Block 1	:	1
1stAdr.Block 1	:	0
Length Block 1	:	8
Ident. Block 1	:	392
Bus-no.Block 1	:	2
IOKind Block 2	:	1
1stAdr.Block 2	:	8
Length Block 2	:	8
Ident. Block 2	:	648
Bus-no.Block 2	:	2

CAN-Bus peripheral equipment

IOKind Block 3	:	1
1stAdr Block 3	:	16
Length Block 3	:	8
Ident.Block 3	:	393
Bus-no.Block 3	:	2

Address areas of the CAN outputs (P32)

The image of the analog CAN outputs is deposited in the rho4 in a separate RAM area. Machine parameter P32 configurates the CAN outputs.

For each analog output block, the following subparameters are to be assigned:

- I/O-type:
  Analog modules get the code 1, digital modules the code 0.
- Start address:
  This relative address indicates the position of the data block (8 bytes) within the rho4 internal RAM area. For inputs and outputs there are 2 RAM areas physically separated.
  The setting is to be performed as follows:

1stAdr Block 1	:	0
1stAdr.Block 2	:	8
1stAdr Block 3	:	16
1stAdr Block 4	:	24
1stAdr Block n	:	(n−1) * 8

In this RAM area, only analog input blocks are taken into account. For digital output blocks, the start address relates to the rho interface.

- Block length:
  The block length has for all analog modules that are available at the moment 8 bytes (4 channels).
- Identifier:
  A bus switching can supply a maximum of input blocks (32-I-byte) and 4 output blocks (32-O-Byte).
  The CAN identifier for the corresponding input block results from the node-ID set by means of the hardware of the B~IO M-CAN-module:

output block	output byte	CAN-Identifier
1	1 to 8	512 + Node-ID
2	9 to 16	768 + Node-ID
3	17 to 24	512 + Node-ID + 1
4	25 to 32	768 + Node-ID + 1

CAN-Bus peripheral equipment

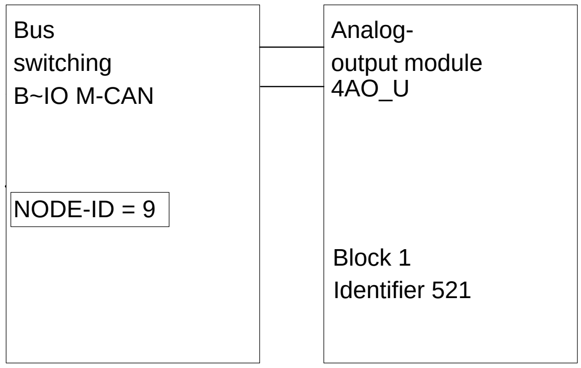
- CAN-Bus-Number:
  The CAN-Bus number indicates with which of the both CAN busses fo the rho4 the B~IO modules are operated. Only the numbers 1 and 2 are authorized. They correspond to the CAN plug number X51 and X52.

Example: CAN output parameter

Example1:
A bus switching B~IO M-CAN is operated together with an analog input module 4AO_U. The node-ID of the B~IO M-CAN module is set via the DIP switch on the value 9. The module is connected via the CAN plug X52 to the rho4.

As CAN identifier to be set, it follows:

512 + 9 = 521



For the configuration, the following settings are to be entered under machine parameter P31:

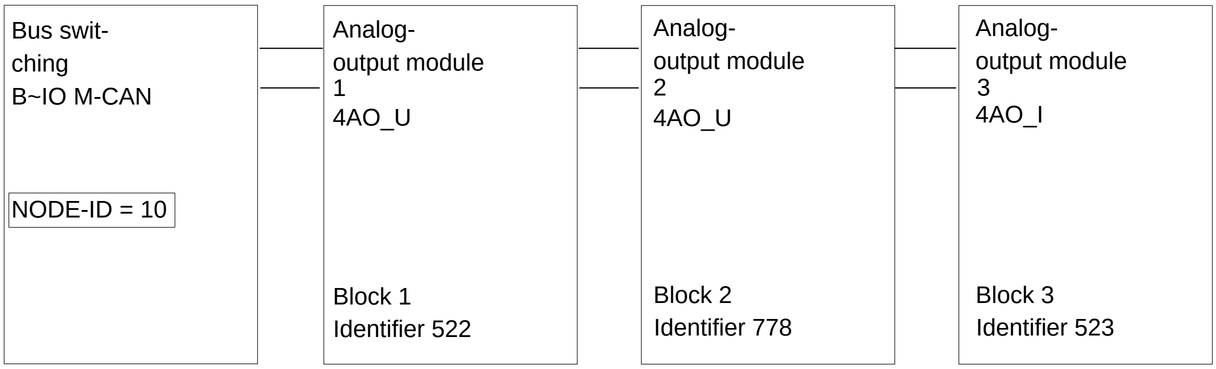
IOKind Block 1	:	1
1stAdr.Block 1	:	0
Length Block 1	:	8
Ident. Block 1	:	521
Bus-no.Block 1	:	2

CAN-Bus peripheral equipment

Example2:
A bus switching B~IO M-CAN is operated together with 3 analog input modules. The node ID of the B~IO M-CAN module is set via the DIP switch on the value 10. The module is connected via the CAN plug X52 to the rho4.

As CAN identifiers to be set, it follows:

Block1: 512 + 10 = 522
Block2: 768 + 10 = 778
Block3: 512 + 10 + 1 = 523



For the configuration, the following settings are to be entered under machine parameter P31:

IOKind Block 1	:	1
1stAdr.Block 1	:	0
Length Block 1	:	8
Ident. Block 1	:	522
Bus-no.Block 1	:	2
IOKind Block 2	:	1
1stAdr.Block 2	:	8
Length Block 2	:	8
Ident. Block 2	:	778
Bus-no.Block 2	:	2
IOKind Block 3	:	1
1stAdr.Block 3	:	16
Length Block 3	:	8
Ident. Block 3	:	523
Bus-no.Block 3	:	2

CAN-Bus peripheral equipment

5.5.4 Analog I/O parameter

The 400 group of machine parameters is used to define the number of analog I/O channels and their configuration.

Number of der analog inputs (P406)

P406 indicates the number of the analog input channels. Per input block there are 4 analog input channels to be set.

assignment of the analog input channels (P407)

In P407 the following is to be entered:

- assignment:
  BAPS channel numbers are allowed from 201 to 299. The adjusted channel number must be used in the BAPS program for the declaration of the analog input of type REAL.
  Example

```
input REAL: 201 = ANA_In_1
```

- RAM initial address:
  This relative address indicates the position of the 2-byte data word for the analog channel within the rho4 internal RAM area. For Inputs and Outputs there are 2 separate physical RAM-Areas.
  The setting is to be performed as follows:

RAM-BegAdr ANA-input 1:	0
RAM-BegAdr ANA-input 2:	2
RAM-BegAdr ANA-input 3:	4
RAM-BegAdr ANA-input 4:	6
RAM-BegAdr ANA-input n:	(n−1) * 2

- Format:
  For the data format, a numerical representation in 2's complement (positive and negative decimal numbers: Format = 0) or a straight binary representation (only positive decimal numbers: Format = 1) can be selected.
- Nominal value:
  The nominal value is a number that corresponds to the highest input value that can be represented with 2 bytes. This means the specified nominal value is obtained in the BAPS program by reading an analog input that provides the maximum possible analog input value.
  - For a nominal value of 1000 and a measuring range of ±10 V, an input value of 1000 is obtained in the BAPS program for an applied voltage of +10V, provided that left-aligned representation and Format=0 are selected.

CAN-Bus peripheral equipment

- With right-aligned representation and a resolution of 12 bits (example: 4AI_UI module), the obtained maximum value will be only 1/16 of the nominal value. This means that for an applied voltage of 10 V, an input value of 62.5 will be read in the BAPS program.

Number of der analogen Outputs (P404)

P404 gives the number of the analog output channels. Per output block, 4 analog output channels are to be set.

Assignment of the analog output channels (P405)

In P405 the following is to be entered:

- meaning:
  The first digit of the 3-digit configuration subdivides the output configuration in different catagories. The following output signals can be delivered to analog output channels:

1XX	:	Output actual position Axis XX or Belt XX (XX = 1 Number of axes + Number of belts)
2XX	:	Decimal output in a BAPS program; The specified channel number must be used to declare the analog output in the BAPS program
3XX	:	Output reference position Axis XX
400 + Kin-No	:	Output reference position Axis XX
450 + Kin-No	:	Output actual path speed Kinematic X (only for robot type= 0 , machine parameter P306)
5XX	:	Output lag Axis XX
6XX	:	Output reference speed Axis XX
7XX	:	Output actual speed Axis XX or Belt XX (XX = 1 to axis number + belt number)

- RAM start address:
  This relative address gives the position of the 2 byte data word for the analog channel within the rho4 internal RAM area. There are 2 RAM areas physically separated for inputs and outputs.
  The setting is to be performed as follows:

RAM-BegAdr ANA-input 1:	0
RAM-BegAdr ANA-input 2:	2
RAM-BegAdr ANA-input 3:	4
RAM-BegAdr ANA-input 4:	6
RAM-BegAdr ANA-input n:	(n−1) * 2

- Format:
  For the output configuration 2XX (BAPS output, the data format can be set to a numerical representation in 2's complement (positive and negative decimal numbers: Format = 0) or a straight binary representation (only positive decimal numbers: Format = 1).
  For all other configurations (1XX, 3XX, 4XX, 5XX, 6XX, 7XX), the data format 2's complement (positive and negative decimal numbers: Format = 0) is mandatory! The path speed output (4XX) yields only positive values with the data format 2's complement.

- Nominal value:
  The nominal value is the number that corresponds to the highest output value that can be represented with 2 bytes for an offset of 0.

  If the nominal value is delivered to an analog output in the BAPS program, the maximim value of the specified voltage (or current) output range is obtained, provided that a left-aligned representation has been selected.

  With right-aligned representation and a resolution of 12 bits (example: 4AO_UI module), the obtained maximum value will be only 1/16 of the nominal value.

- Offset:
  A voltage offset can be specifed for analog outouts that is added to the output value. The offset is specified as a percentage of the maximum voltage (or current).

5.5.5 Example

A bus interface with 2 analog I/O modules is connected to a rho4. A rho process requires that 4 analog outputs with an output voltage range of ±10 V be driven. In the same BAPS program, the output voltages are read via 4 analog input channels and displayed on the PHG.

The following I/O modules are used:

B~IO M-CAN	bus switching
4AO_U	Analog-output module (4 outputs)
4AI_UI	Analog input modules (4 inputs)

The 4 outputs of the 4AO_U module are connected to the inputs of the 4AI_UI module.

Hardware settings

B~IO M-CAN-bus switching:

Baud rate:	DIP-switch S1 = 1000 0111	CANrho, 1 MBaud
Node-ID :	DIP-switch S2 = 1000 0001	Node-ID = 1

CAN-Bus peripheral equipment

4AO_U analog output module:

DIP-switch S1 = 1111 0000	±10 V output voltage for channel 1 to 4

4AI_UI analog input module:

DIP-switch S1 = 0001 0000	±10 V, 2's complement, left-aligned, no averaging, no diagnostics

Machine parameter P30

I/O configuration CAN

Number of input blocks	:	1
Number of output blocks	:	1
Number of SR-CAN modules	:	0
CANopen Download	:	0
I/O configuration CAN 1		
• Baud rate	:	0
• CANrho=0, CANopen= 1	:	0
I/O configuration CAN 2		
• Baud rate	:	0
• CANrho=0, CANopen= 1	:	0

Machine parameter P31

ADR. CAN-I

IOKind Block 1	:	1
1stAdr.Block 1	:	0
Length Block 1	:	8
Ident. Block 1	:	385
Bus-no.Block 1	:	2

Machine parameter P32

ADR. CAN-O

IOKind Block 1	:	1
1stAdr.Block 1	:	0
Length Block 1	:	8
Ident. Block 1	:	513
Bus-no.Block 1	:	2

CAN-Bus peripheral equipment

Machine parameter P404

Number of analog outputs : 4

Machine parameter P405

assignment Analog outputs

Analog-output 1

- Meaning : 201
- RAM-BegAdr : 0
- Format : 0
- Nom.val. : 1000
- Volt.off. (%) : 0

Analog-output 2

- Meaning : 202
- RAM-BegAdr : 2
- Format : 0
- Nom.val. : 1000
- Volt.off. (%) : 0

Analog output 3

- Meaning : 203
- RAM-BegAdr : 4
- Format : 0
- Nom.val. : 1000
- Volt.off. (%) : 0

Analog-output 4

- Meaning : 204
- RAM-BegAdr : 6
- Format : 0
- Nom.val. : 1000
- Volt.off. (%) : 0

Machine parameter P406

Number of Analog inputs : 4

CAN-Bus peripheral equipment

Machine parameter P407

Meaning Analog inputs

Analog-input 1

- Meaning : 201
- RAM-BegAdr : 0
- Format : 0
- Nom.val. : 1000

Analog-input 2

- Meaning : 202
- RAM-BegAdr : 2
- Format : 0
- Nom.val. : 1000

Analog-input 3

- Meaning : 203
- RAM-BegAdr : 4
- Format : 0
- Nom.val. : 1000

Analog-input 4

- Meaning : 204
- RAM-BegAdr : 6
- Format : 0
- Nom.val. : 1000

CAN-Bus peripheral equipment

5.5.6 BAPS-Program

```
PROGRAM ANA_IO
;*******************************************************************************
;*Demo program CAN-Analog-I/O with B~IO
;*P405,P407: Nominal value = 1000 = 10 Volt
;*******************************************************************************
input REAL: 201 = AI1, 202 = AI2, 203 = AI3, 204 = AI4       ;analog inputs
output REAL: 201 = AO1, 202 = AO2, 203 = AO3, 204 = AO4      ;analog outputs
INTEGER: INDEX
REAL : D1, D2, D3, D4
BEGIN
      Loop:
           WRITE PHG, cls, 'D1,...,D4' WAIT 2
           INDEX = -10
           REPEAT 20 TIMES
                 AO1=100*INDEX
                 AO2 = -100
                 AO3 = -700
                 AO4 =  900
                 WAIT 1
                 D1 = AI1
                 D2 = AI2
                 D3 = AI3
                 D4 = AI4
                 WRITE PHG, cls, D1,'  ',D2,'  ',D3,'  ',D4,'  '
                 WAIT 1
                 INDEX = INDEX + 1
           REPEAT_END
      JUMP Loop
PROGRAM_END
```

5.6 SR-CAN module

To make sure that the machine parameters always fit to the kinematic (mechanics) the machine parameters concerning the mechanics are stored on an SR-CAN module. This module is firmly connected with the robot mechanics, and is maintained during the robot service life.

The communication with the SR-CAN module takes place by means of the CAN protocol. To be able to read and write the machine parameters, corresponding functions are available in BAPS and as rho4 library functions. The library functions are contained in the system group. The functions for reading the SR-CAN modules are named rSGSrCan() and rSGSerialNb(), the functions for writing in the SR-CAN module are named rSSSrCan() and rSSSerialNb(). The machine parameters and serial numbers contained in the real time core of the rho4.1 are loaded into the SR-CAN module by means of these functions. Additionally, a reading and writing operation of the machine parameters and serial numbers to the SR-CAN module with PHG2000 are offered. The PHG menu tree contains a mode protected by a password for this purpose. Mode 7.8.6.1 is assigned for reading and mode 7.8.6.2 for writing the serial number onto the SR-CAN modules. Mode 7.8.6.3 is assigned for reading and mode 7.8.6.4 for writing the machine parameters onto the SR-CAN modules.

Read/write functions

For the communication with the SR-CAN module, the before mentioned read/write functions are available. As transfer parameters, the SR-CAN module number and the kinematic (Kin1 to 16) have to be indicated in any case, since a control can address several robots at the same time.

Machine parameter and serial number are separated on the SR-CAN modules and have also to be written and read separately. During the run-up phase of the control it is possible to quickly check which machine parameters are to be activated, since only the serial number of the SR-CAN module has to be read.

When calling the write function rSSSrCan(), the complete machine parameters are loaded into the SR-CAN module. This is preferably be done while measuring the robot kinematic of a new system or when changing a defective SR-CAN module.

When calling the read function rSGSrCan(), at first also the complete machine parameters are loaded from the SR-CAN module onto an intermediate storage place in the control, yet are not active at this time. There is a filter between this intermediate storage place and the valid machine parameters that can be configured and only takes over selected machine parameters.

If all machine parameters are selected, the read function rSGSrCAN () acts as a machine parameter backup, i.e. the machine parameters are all taken over from the SR-CAN module. To activate the machine parameters, the control must be run-up again.

CAN-Bus peripheral equipment

With the write function rSSSerialNb(), the serial number, parameter P314, is loaded into the SR-CAN module. The read function rSGSerialNb() supplies as return parameter the serial number of the addressed SR-CAN module.

☞ **See also the DLL library manual.**

Configurable filter

The filter consists of a bnr file. The file name is already fixed and is called srcan.bnr. The file length is exactly 1000 bytes (character number 1 to character number 1000). A standard bnr file 'sta_srca.bnr' has been supplied on which all characters are initialized on zero.

For each machine parameter there is a character by means of which it is decided whether this parameter is taken over into the valid machine parameters after the reading from the SR-CAN module. The machine parameter and character numbers are identical. If a character has the value zero the machine parameter belonging to this character will not be taken over. If the value unequals zero, the machine parameter will be taken over.

If a file with the name srcan.bnr exists in the user memory, its contents will be used as filter for reading the machine parameters from the SR-CAN module. If this file is faulty or does not exist, no machine parameters will be taken over. This file is valid for all kinematics. The machine parameters identified for loading are always taken over for all kinematics. If all entries in the file srcan.bnr is unequal zero, a machine parameter bakkup is performed with the machine parameters located in the SR-CAN module. To activate the machine parameters, the control must be run-up again (see also “MP-Download from VO05x” in this chapter).

The file srcan.bnr is processed by means of a commercial HEX editor.

In the following example two machine parameters, P307 and P308 are taken over.

All characters in the file srcan.bnr are zero, apart from the contents of the character numbers 307 and 308.

CAN-Bus peripheral equipment

Character number	Content (Hex)
1	0
2	0
.	
307	1
308	1
.	
999	0
1000	0

Normally, only kinematic-specific machine parameters are taken over. The general system parameters remain unchanged in the real time core of the rho4. If all entries are unequal zero, a machine parameter backup is performed.

SR-CAN serial number

During the run-up of the control it is checked whether the valid machine parameters in the control correspond to those in the SR-CAN module. For a quicker verification, only the serial number in the SR-CAN module is compared with the one contained in the control. If both serial numbers are identical, the control will run up. If the serial numbers are not identical, the control will run up with the message ’serialnumber invalid’. It is then possible to generate valid machine parameters and to load them into the real time core of the rho4 (see also “MP-Download from VO05x” in this chapter).

The serial number in the SR-CAN module is not initialized at the time of shipment. The serial number consists of an Ascii string with 64 characters which is initialized after a machine parameter back-up. In this case too, the control will run up with the message ’serialnumber invalid’ (see also “MP-Download from VO05x” in this chapter).

When loading the machine parameters from the control into the SR-CAN module, the serial number remains in the control so that the comparison of the serial numbers leads until the exchange of one of the two components, to a complete run-up of the control.

The serial numbers form a part of the machine parameter (P314). For each kinematic a serial number is provided in the machine parameters of the control, since it is possible to operate up to 16 kinematics with one control. If several kinematics have been configured on one control, the handling of the serial numbers for each kinematic takes place according to the same procedure.

The operating mode of the machine parameter back-up, enabled by the key combination permission-key+Alt+Mode+0 at the PHG2000, remains unchanged.

CAN-Bus peripheral equipment

MP−Download from VO05x

From the version VO05x, the behavior of the SRCAN module regarding the machine parameter download (transfer of the machine parameters from the SRCAN module into the control) is changed.
At the startup of the control (rho4.1 and rho4.0), the serial number in the machine parameter record (P314) and the serial number saved in the SRCAN module are compared. If both serial numbers match, the behavior between the version VO05x and the previous versions (VO01x−VO04x) does not change. If the serial numbers do not match, a machine parameter download will be **automatically** started in version VO05x.
In the preceding versions (VO01x−VO04x), no machine parameter download will be started, but the error message ”serialnumber invalid” (No. 399232) is displayed. If a filter file (SRCAN.bnr) is present, the filtered machine parameters will become effective in version VO05x at the **automatically** initialized control startup.
In the preceding versions (VO01x−VO04x), the startup is not automatically initialized, the filtered machine parameters are however also adopted in the control. If there is **no** filter file in version VO05x, all machine parameters (**Backup from SRCAN module**) will become effective at the automatically initialized control startup.
In the preceding versions (VO01x−VO04x), the warning ”No MpFilter” (No. 398848) is displayed and **no** machine parameter is adopted from the SRCAN module.
For the library functions, the function rSGSrCAN() in version VO05x changes in the same way. If a filter file is present at the machine parameter download (call of rSGSrCAN ()), it will be taken into account; if there is **none**, all machine parameters will be adopted from the SRCAN module (**Backup from SRCAN module**).

Error messages from version VO05x:

If the serial number cannot be read at the startup (e.g. wrong parameter, module not connected...), the error messages ”**serialnb. n.readable**” (No. 396800) appears.
If it is detected after the machine parameter download that the machine parameters are invalid (e.g. incorrect length, Mp not yet on the SRCAN module ...), the error message ”**inval.srcan−ma−para.**” (No. 396928) appears.
During the automatically initialized machine parameter download, which can take a few seconds, the message ”machine-parameter download out of the srcan-module” is displayed at the PHG.

Machine parameters

Machine parameter P314 contains the serial number. The serial number for each kinematic is in machine parameter P314. The serial number has 64 characters and can be entered in the machine parameter program. For the movement of the cursor the greater/smaller-than-keys (< >) are used.

CAN-Bus peripheral equipment

The number of the SR-CAN modules is indicated in the machine parameter P30.

The settings for the analog I/O are described in section 5.7 Analog Inputs/Outputs of the SR-CAN module.

In machine parameter P37, the input addresses and output addresses of the read/write services are entered. These addresses are required for reading/writing the machine parameters and the serial number. In addition, the CAN-BUS number and the SR-CAN module number are entered. Address for reading, input address 1409, with offset address 1 at the module. Address for writing, output address 1537, with offset address 1 at the module.

The Offset address is set at the SR-CAN module. It must be at least 1. All input or output addresses referring to a SR-CAN module have to be adapted when changing the offset address.

The SR-CAN module is equipped with a 8-position dip switch.

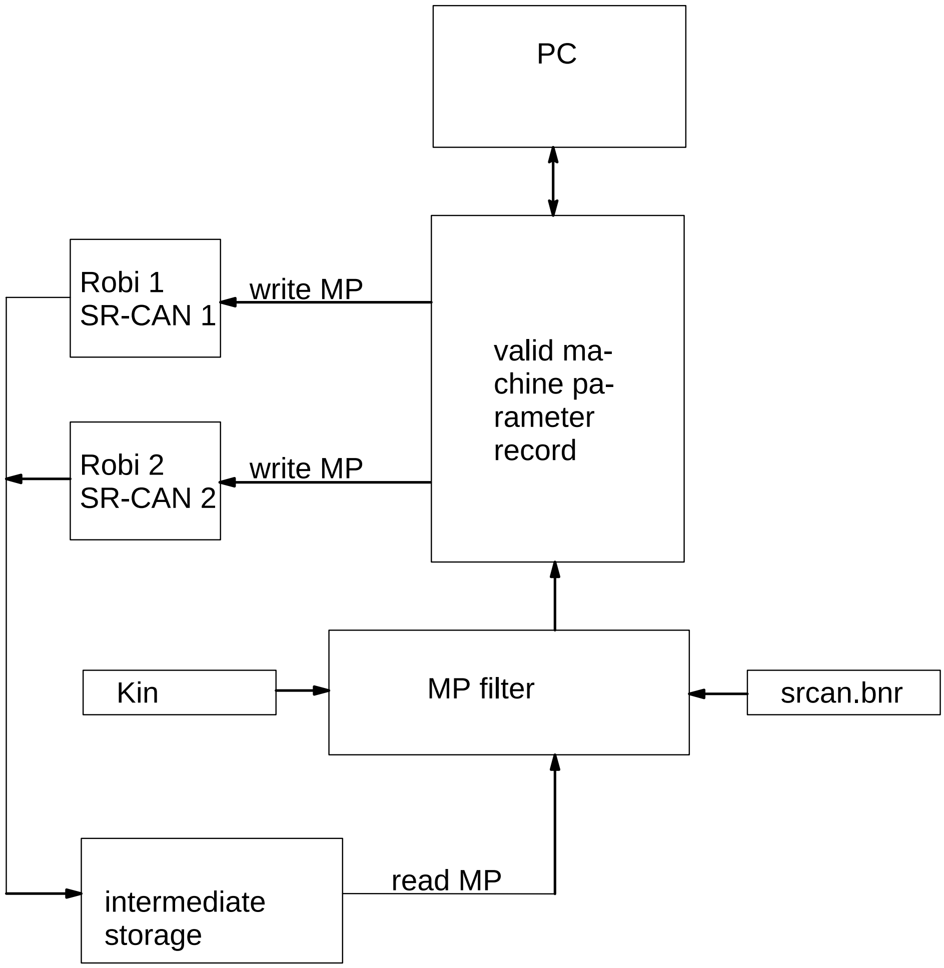
B8	B7	B6	B5	B4	B3	B2	B1

B1 to B5	Knot address
B6 to B7	Baud rate
B8	Bosch setting

Remark	**B8** **2^7**	**B7** **2^6**	**B6** **2^5**	**B5** **2^4**	**B4** **2^3**	**B3** **2^2**	**B2** **2^1**	**B1** **2^0**
reserved	x	x	x	off	off	off	off	off
Knot address 1	x	x	x	off	off	off	off	on
Knot address 2	x	x	x	off	off	off	on	off
Knot address 3	x	x	x	off	off	off	on	on
Knot address 4	x	x	x	off	off	on	off	off
Knot address 5	...							
Knot address 30	x	x	x	on	on	on	on	off
Knot address 31	x	x	x	on	on	on	on	on
Baud rate 20k	x	off	off	x	x	x	x	x
Baud rate 125k	x	off	on	x	x	x	x	x
Baud rate 500k	x	on	off	x	x	x	x	x
Baud rate 1M	x	on	on	x	x	x	x	x
CANopen	off	x	x	x	x	x	x	x
CANrho	on	x	x	x	x	x	x	x

CAN-Bus peripheral equipment

Read/write MP



5.7 Analog In-/Outputs of the SR-CAN module

The SR-CAN modules has also 4 analog inputs besides the 8 digital inputs and 8 digital outputs. The analog inputs can be read by the rho4 and processed in BAPS programs.

This description shows the possibilities of how to use the analog inputs of the SR-CAN modules and the particularities and differences from the analog inputs of the B~IO modules. It is to be considered as a complement to the of description ’Analog I/O with CAN B~IO modules’. The described facts such as the setting of machine parameters for analog inputs will form the basis.

5.7.1 CAN-bus switching

The SR-CAN module can be driven :

- CANopen protocole
- CANrho protocole

The protocole is selected by the setting of the machine parameter P30.

For the setting of the baud rate, the CAN knot address and the protocole, a 8fold DIP switch is integrated on the SR-CAN module.

☞ **The analog ínputs of the SR-CAN modules will then be transferred trouble-free even if the digital outputs of the module are transferred from the rho4. (See page 5−16, CAN-I/O parameter, Machine parameter P30..P32).**

5.7.2 Measuring ranges and data formats

The input voltage range is from −10 volts to +10 volts. The measured value read is digitally represented in a 16 bit word in the 2 complement with sign bit doubling. In this point, the SR-CAN module differs from all B~IO modules.

The doubling of the sign bit leads to the fact that the measuring value is converted into a decimal number corresponding to only the half of the real value. This particularity must be corrected by doubling the nominal value in the machine parameter P407 (assignment of the analog input channel).

5.7.3 Machine parameter settings

CAN-I/O parameters

I/O configuration of the CAN bus (P30)

In machine parameter P30, 2 input blocks and 1 output block are intended for the SR-CAN module.

A CAN block consists of up to 8 bytes data. An analog channel of the SR-CAN module consists of 2 bytes data. For the 4 analog input channels, one input block will be assigned.

The second input block is required for the digital inputs (1 byte data). The output block is required for the digital ouptut (1 byte data).

CAUTION
In order to use the analog inputs of the SR-CAN module, the digital outputs of the module must also be taken into account on the rho4 side.
Is this not the case, the transmission of the inputs will also be interrupted and the rho4 signals the status : “No transmission CAN”.

In order to use the analog inputs of the SR-CAN module, it is not necessary to set the subparameter ’Number of the SR-CAN module’. But if it is set, the parameter P37 ’Electrical type plate’ must then be adapted. (Siehe also section 5.6.)

After P30 has been entered, the address ranges of the I/O blocks are to be adapted in P31 and P32.

Address range of the CAN-Inputs (P31)

The machine parameter P31 configures the CAN ínputs. For each input block, the subparameters I/O type, initial address, block length, identifier and CAN-Bus number are to be adjusted (see also Analog I/O with CAN-B~IO modules, section 5.5.2).

Example

For the configuration of a SR-CAN module at the CAN bus 2 and adjusted node-ID=3, the following settings are to be entered under machine parameter P31:

1. Analog-input block

CAN-Bus peripheral equipment

- IOKind Block n : 1
- 1stAdr.Block n : 0
- Length Block n : 8
- Ident. Block n : 643
- Bus-no.Block n : 2

The block number n is obtained from the whole configuration of CAN modules (n=1..40).

1stAddr must be matched with other analog inputs channels. If for example a B~IO analog input module is used simultaneously with 4 analog inputs, for which the initial address is set =0, the value 8 must be set as the initial address of the SR-CAN analog inputs.

The adjustment of the initial addresses depends on which one of the two CAN busses the I/O module is driven at. It must be ensured that there is no overlap for the address assignment.

In the machine parameter P407, the 8 byte data block is assigned to the 4 input channels. See also 5−38, assignment of the analog input channels (P407).

Identifier depends on the adjusted node-ID (Offset address) of the SR-CAN module :

Identifier = 640 + Node-ID

2. Digital-input block

- IOKind Block n : 0
- 1stAdr.Block n : (212)
- Length Block n : 1
- Ident. Block n : 387
- Bus-no.Block n : 2

The block number n is obtained from the whole configuration of CAN-modules (n = 1..40).

1stAddr gives the initial address in the rho4 interface and must be matched with other digital input channels.

Identifier depends on the adjusted Node-ID (Offset address) of the SR-CAN module :

Identifier = 384 + Node-ID

CAN-Bus peripheral equipment

Address range of the CAN outputs (P32)

The machine parameter P32 configures the CAN ínputs. For each input block, the subparameters I/O type, initial address, block length, identifier and CAN-Bus number are to be adjusted (see also Analog I/O with CAN B~IO-modules, section 5.5.2).

Example

For the configuration of a SR-CAN module at the CAN bus 2 and adjusted node-ID=3, the following settings are to be entered under machine parameter P32:

Digital output block

- IOKind Block n : 0
- 1stAdr. Block n : (212)
- Length Block n : 1
- Ident. Block n : 515
- Bus-no.Block n : 2

The block number n is obtained from the whole configuration of CAN modules (n = 1..40).

1stAddr gives the initial address in the rho4 interface and must be matched with other digital input channels.

Identifier depends on the adjusted Node-ID (Offset address) of the SR-CAN module:

Identifier = 512 + Node-ID

Analog I/O parameters

In the 400 group of the machine parameters, the number of the analog I/O inputs and their assignment is set up.

Number of analog Inputs (P406)

This parameter P406 gives the number of the analog input channels. 4 analog input channels are to be adjusted per input block.

assignment of the analog input channels (P407)

In P407 is to be entered:

- meaning:
  The BAPS channel numbers from 201 to 299 are allowed. The adjusted channel number must be used in the BAPS program for declaring the analog input of the REAL type.
  Example

```
input REAL: 201 = ANA_In_1
```

CAN-Bus peripheral equipment

- RAM initial address:
  This relative address gives the position of the 2-byte data word for the analog channel within the rho4.0 internal RAM area. The first RAM initial address must be set equal to the initial address adjusted in P31. If the initial address is set on 0, it follows the following RAM initial addresses.

RAM-BegAdr ANA-input 1:	0
RAM-BegAdr ANA-input 2:	2
RAM-BegAdr ANA-input 3:	4
RAM-BegAdr ANA-input 4:	6

  If the initial address in P31 is set on 8, because for instance a B~IO module has been connected with 4 analog ínputs, the RAM initial addresses in P407 must be changed into 8,10,12,14.

- Format:
  The data format is to be set to the 2 complement (positive and negative decimal numbers: Format = 0).
- Nominal value:
  The required nominal value to be doubled, since the internal number representation of the SR-CAN module is connected to the factor 0.5. The SR-CAN module only yields the value 500 for a nominal value of 1000 in the case of the applied maximum voltage of 10 volts. In comparison with this, a B~IO module would yield the value 1000. The desired nominal value must be therefore multiplied with 2 to compensate this behaviour.

  Example
  If the decimal value read is to be 1000 for a voltage of +10 volts, the nominal value of the corresponding analog input is to be set to 2000.

5.8 Assignment of the CAN-Bus interfaces

The rho4.0 has two, resp. three separate CAN-Bus interfaces:

- Plug-No. X51
- Plug-No. X52
- Plug-No. X53 (only rho4.0 L) only usable as belt input

The interfaces X51 and X52 can be engaged in certain limits with axes, digital I/O modules and SR-CAN modules. Moreover, a coupling of serveral rho4 controls is possible via CAN-Bus.

To connect axes, digital I/O modules and SR-CAN-modules at the rho4, a certain number of O- and I-blocks is required − depending on the concerned configuration.

The required number of O- and I-blocks is calculated as follows:

Number of O-blocks =	1 (SYNC-Telegramm) + Number of axes (P302) + Number of Digital output-blocks (P30)
Number of I-blocks =	Number of axes (P302) + Number of Digital input -blocks (P30)

The number of the I- and O-blocks must be calculated separately for each CAN-Bus. When using the digital inputs and outputs on the SR-CAN-modules, they are to be taken into account for the number of the digital input and output blocks (P30).

The following tables indicate the maximum assignment of both CAN-Bus-interfaces (X51 and X52):

Screening time (P5) < 8 ms	
CAN-Bus 1 (X51)	**CAN-Bus 2 (X52)**
maximum 15 O-blocks maximum 14 I-blocks	in sum maximum 17 blocks in which maximum 15 O-blocks, maximum 14 I-blocks
or:	
in sum maximum 17 blocks in which maximum 15 O-blocks, maximum 14 I-blocks	maximum 15 O-blocks maximum 14 I-blocks

Screening time (P5) >= 8 ms	
CAN-Bus 1 (X51)	**CAN-Bus 2 (X52)**
maximum 29 O-blocks maximum 28 I-blocks	in sum maximum 17 I- bzw. O-blocks in any combination
or:	
in sum maximum 17 I- bzw. O-blocks in any combination	maximum 29 O-blocks maximum 28 I-blocks

CAN-Bus peripheral equipment

When SR-CAN-modules are used, the maximum number of I/O blocks reduces at the corresponding bus by 3 (independent on the number of the SR-CAN-modules).

The indicated values refer to a baud rate of 1 MBaud (depending on the cable length). If the baud rate is smaller, the maximum number of I/O blocks reduces by the corresponding factor (e.g factor 2 at 500 kBaud).

Cable length	max. Baud rate
up to 25 m	1 MBaud
up to 100 m	500 kBaud
up to 200 m	250 kBaud
over 200 m	125 kBaud

The screening time (P5) must be set at the minimum so that all I- and O-blocks can be transferred within the screening time. The screening time for one block is 1 MBaud CAN 110 µs. For smaller baud rates, the transfer time increases by the corresponding factor (e.g. 220 µs at 500 kBaud).

Maximum whole number of components

Whole number of axes	=	maximum 16
Number of digital output blocks + Number of SR-CAN-modules	=	maximum 40
Number of digital input blocks + Number of SR-CAN-modules	=	maximum 40

☞ **When rho4 control units are coupled via CAN-Bus, the CAN-Bus used for this purpose must not be engaged with other components**

The maximum possible number of axes with Servodyn-GC drive amplifiers is 6 per CAN-Bus. The O- and I-blocks remaning free can be engaged with any other components (also axes with Servodyn-D amplifiers).

☞ **When B~IO-modules without inputs (pure output modules) are connected, it is possible, that the red error-LED at the B~IO-modules are blinking permanent. This has no effect to the function of the B~IO-modules.**
To avoid this effect, at least one input module should be connected to each CAN-Bus.
This behaviour occurs only with CANrho protocol.

CAN-Bus peripheral equipment

5.8.1 Examples

☞ **Example 1 to 3 can be operated with screening times < 8ms. In example 4, a screening time of 8 ms at least is required.**

Example 1

- 4 axes
- 1 SR-CAN module
- 2 DESI65-K-CAN digital output modules with 1 byte each (1 O-Block/ module)
- 3 DESI65-K_CAN digital input modules with 1 byte each (1 I-Block/ module)

CAN-Bus 1 (X51)		**CAN-Bus 2 (X52)**	
4 axes		all digital I/O-modules + SR CAN-modules	
Number of O-blocks = 1 + 4 (axes)	= 5	Number of O-blocks = 1 + 2 (Digital output blocks)	= 3
Number of I-blocks = 4 (axes)	= 4	Number of I-blocks = 3 (Digital input blocks)	= 3

Example 2

- 4 axes
- 2 DESI65-K-CAN digital output-modules with 1 byte each (1 O-Block/ module)
- 3 DESI65-K_CAN digital input -modules with 1 byte each (1 I-Block/ module)
- Coupling to another rho4

CAN-Bus 1 (X51)			**CAN-Bus 2 (X52)**
4 axes + all digital I/O modules			Coupling to the second rho4 control
Number of O-blocks =	1 + 4 (axes) + 2 (Digital output blocks)	= 7	
Number of I-blocks =	4 (axes) + 3 (Digital input blocks)	= 7	

Example 3

- 5 kinematics with 4 axes each = 16 axes
- 2 B~IO-modules with 2 bytes digital outputs each (1 O-Block/module) and 2 bytes digital inputs each (1 I-Block/module)

CAN-Bus 1 (X51)		**CAN-Bus 2 (X52)**	
10 axes		6 axes + 2 B~IO-modules	
Number of O-blocks = 1 + 10 (axes)	= 11	No. of O-blocks = 1 + 6 (axes) + 2 (Digital output blocks)	= 9
Number of I-blocks = 10 (axes)	= 10	No. of I-blocks = 6 (axes) + 2 (Digital input blocks)	= 8

CAN-Bus peripheral equipment

5.9 CAN-ID assignments

The following table indicates the ID assignment of the CAN protocoles.

☞ **The grey marked IDs are reserved for the set value and actual value protocoles of the axes and must not be used for I/O IDs. If this instructions is not observed, this can give rise to overlaps between axis and digital I/O modules.**

Servodyn-G	**Servodyn-D**	**DESI65K-CAN**	**RM65M-CO BIO-K-CAN**	**CANopen I/O-modules encoder, axes**
SYNC ID = 100	SYNC ID = 100	SYNC ID = 100	SYNC ID = 100	SYNC ID = 128
Set values maximum 6 axes ID = 200 + 10 * i i = 1 to 6 210 . . 260	Set values maximum 16 axes ID = 200 + 10 * i i = 1 to 16 210 . . . 360			
		Inputs ID = 384 + i i = 1 to 99 385 . . . 483	Inputs Area 1 ID = 384 + i i = 1 to 127 385 511	Inputs Area 1 ID = 384 + i i = 1 to 127 385 511
		Outputs ID = 512 + i i = 1 to 99 513 .	Outputs Area 1 ID = 512 + i i = 1 to 127 513 .	Only I/O modules and axes: Outputs Area 1 ID = 512 + i i = 1 to 127 513 .

CAN-Bus peripheral equipment

Servodyn-G	**Servodyn-D**	**DESI65K-CAN**	**RM65M-CO** **BIO-K-CAN**	**CANopen** **I/O-modules** **encoder, axes**
		.	.	.
		.	.	.
		.	.	.

Servodyn-G	**Servodyn-D**	**DESI65K-CAN**	**RM65M-CO** **BIO-K-CAN**	**CANopen** **I/O-modules** **encoder, axes**
actual values ID = 600 + 10 * i i = 1 to 6	actual values ID = 600 + 10 * i i = 1 to 16	.	.	.
610	610	.	.	.
		611	.	.
620	620		.	.
			.	.
630	630		.	.
			639	639
640	640			
			Inputs Area 2 ID = 640 + i i = 1 to 127	Inputs Area 2 ID = 640 + i i = 1 to 127
			641	641
650	650		.	.
			.	.
660	660		.	.
	.		.	.
			.	.
	.		.	.
			.	.
	.		.	.
	760		.	.
			767	767
			Outputs Area 2 ID = 768 + i i = 1 to 127	Outputs Area 2 ID = 768 + i i = 1 to 127
			769	769
			841	.

CAN-Bus peripheral equipment

Servodyn-G	**Servodyn-D**	**DESI65K-CAN**	**RM65M-CO BIO-K-CAN**	**CANopen I/O-modules encoder, axes**
			 895	 895
				SDO area 1 ID = 1408 + i i = 1 to 127 1409 . . 1536
				SDO area 2 ID = 1536 + i i = 1 to 127 1537 . . 1663

CAN-Bus peripheral equipment

5.10 Servodyn-D-rho4 interface

5.10.1 Control data rho4 −> Servodyn-D

Interface Signal rho4	Signification in the drive
MF-output axisXOut1 (MF_OUTX_RCI)	Digital output Out1
Free	
Free	
Activation limit switch logic in drive 0 = Limit switch monitoring inactive 1 = Limit switch monitoring active (for Servodyn-G)	
0 = Manual, 1 = Automatic	Active torque limit value: 1: S-0-0092 (−> R-109 −> P-724.0) 0: S-1-0092 (−> R-110 −> P-724.1)
Drive-On axisX (DRIVE_X_RCI) No. 320 to 343, O40.0 to O42.7 (for Servodyn-G)	
Drive-On axisX (DRIVE_X_RCI) No. 320 to 343, O40.0 to O42.7	0: power final stage Out 1: power final stage In (only possible when there is no error)
Open brake axisX (BRAKE_X_RCI) No. 424 to 447, O53.0 to O55.7	0: Close brake 1: Open brake

CAN-Bus peripheral equipment

5.10.2 Status messages Servodyn-D −> rho4

Designation in the drive	**Error number**	**Effect in the rho4 Error messages, interface outputs**	**Reset*)**
Control voltage	F 35	Status message: CAN-Logic Power Code: 24320 + axis No. − 1	H
Excess voltage Undervoltage	F 30 F 99	Status message: CAN Overvoltage Code: 24448 + axis No. –1	H
Overtemperature module	F 07	Status message: CAN Controller−temp Code: 24576 + axis No. − 1	H
Overtemperature Motor	F 08	Status message: CAN Motor−temp Code: 24704 + axis No. − 1	H
Feedback Commutation error encoder error	F 11 F 99 F 70	Status message: CAN Meas. Sys fault Code: 24832 + axis No. − 1	H
Digital input In1		Interface-Signal: MF-input axisX In1 (MF_INPUTx_RCO), x = (axis No. −1 * 4) + 1)	K
CPU calculation time Timeout Temperature-trigger (ZSK2: Warning)	F 06 F 69	Output at RC-Outputs ‘I^2t limitation’ No. 232 to 255, Address O29.0 to O31.7	K
reserved for gear monitoring		Status message: CAN Current fault Code: 25216 + axis No. − 1	H
ZSK2.interface (ZSK2: Warning)		Status message: CAN communic. fault Code: 25344 + axis No. –1	H
VM-switch-off	F 98	Status message: CAN-VM−Switch-off Code: 22912 + axis No. − 1	G
Digital-input In2		Interface-Signal: MF-input axisX In2 (MF_INPUTx_RCO), x = (axis No. −1 * 4) + 2)	K
Digital-input In3		Interface-Signal: MF-input axisX In3 (MF_INPUTx_RCO), x = (axis No. −1 * 4) + 3)	K
VM-error	F 97	Status message: CAN-VM−error Code: 24192 + axis No. –1	G
Excessive deviation	F 13	Caution: Interpolator-Stop Code: 268672 + axis No. –1	K
Digital-input In4		Interface-Signal: MF-input axisX In4 (MF_INPUTx_RCO), x = (axis No. −1 * 4) + 4)	K
Collect error	All Fxx not mentioned	Status message: global CAN−Fault Code: 25472 + axis No. –1	H

CAN-Bus peripheral equipment

*) Reset condition in rho4:

H = Control start-up

G = Basic position (Interface-Signal)

K = Reset not required

6 SERCOS interface

The SERCOS interface is realized in the rho4.0 as a digital drive interface. A SERCOS ring, at which up to 16 axes can be connected, is available on the pci−rho as a hardware. The interface is realized as a position interface. It corresponds to the European norm IEC 61491.

6.1 Data exchange via SERCOS bus

6.1.1 Service channel

The required parameters are transmitted to the drives via the service channels are used in the startup phases (phases 2 and 3).

In the running time, commands, e.g. S-0-0099, Zsk1-Reset are transmitted to the drives via the service channels.

6.1.2 Cyclic data exchange

The rho4.0 supports the operating mode ‘Position interface with cyclic set-value indication’.

Setting S-0-0032:

b’0000 0000 0000 x011 for position controlling with motor encoder

b’0000 0000 0000 x100 for position controlling with external encoder

x = 0: Position controlling with haul distance

x = 1: Position controlling without haul distance (pre-controlling)

SERCOS interface

Telegramm rho4 => drive (MDT)

The cyclic telegram has the following structure:

S-0-0134	Master control word	2 Byte
S-0-0047	Position set-value	4 Byte

The master control word is operated by the rho4 as follows:

Bit 0 bis 5:		Signals for operating the service channels (see SERCOS specification)	
Bit 6/7	:	Real-time control bits:	are not used by the rho4
Bit 8/9	:	Set operating mode:	fixed on 00, corresponds to main operating mode (S-0-0032)
Bit 10	:	IPOSYNC, see SERCOS specification	
Bit 13	:	Drive break:	is set fixed on 1 by the rho4
Bit 14	:	Drive release:	rho4 interface signal ‘Drive On all axes’
			or ‘Drive On axis x’ (DRIVE_x_RCI No. 320 bis 343, O40.0 bis O42.7)
Bit 15	:	Drive on:	**up to version VO05C:** rho4 interface signal ’EMERGENCY STOP, not’ (EMERG_N_RCI, No. 128, O16.0)
			or no rho4 internal Emergency-stop condition (grave error, e.g. Servo error)
			from version VO05D: rho4 interface signal ’Open brake axes x’ BRAKE_x_RCI, No. 424..447, O53.0..55.7)

☞ **The MDT is fixed by the rho4 and cannot be modified by the user.**

SERCOS interface

Drives => rho4 (AT)

The axis telegrams are fixed on a maximum of 22 bytes per axis in the rho4. This corresponds to the maximum AT length for Bosch Rexroth Servodyn-D drives. The number of the identifiers in the configurable AT for Servodyn-D drives is limited to 5 at the maximum.

By considering the boundary conditions, the ATs are assigned by the rho4 as a standard as follows:

S-0-0135	drive status	2 Byte
S-0-0051	position actual value (motor encoder)	4 Byte
S-0-0189	hauling distance	4 Byte
S-0-0144	signal status word	2 Byte
S-0-0084	torque actual value [%]	2 Byte
S-0-0040	velocity actual value	4 Byte
	reserve	4 Byte

The drive status is a fixed part of the axis telegram and always present. The other data is preset by the rho4 during the startup via S-0-0016 (configuration list AT) in all drives.

For drives that support only short ATs, the list will be shortened correspondingly.

Example:

Maximum length of the configurable AT	= 10 Byte
S-0-0016 = (S-0-0051, S-0-0189,S-0-144)	
+ 2 bytes drive status => Total length	= 12 Byte

The configuration of the axis telegrams can be changed by the user in the drive amplifier.

To overwrite the configuration preset by the rho4, the ident number S-0-0016 must be taken over into the list of the parameters to be transferred by the rho4 (scs file).

To define the S-0-0016, the following is to be observed:

- Drive status is a fixed part of the AT
- Length of the configurable AT: 20 bytes or maximum length supported by the drive (S-0-0185), if it is smaller than 20 bytes
- Maximum number of ident numbers in the AT of the corresponding drive amplifier, e.g. 5 for Servodyn-D drives
- The first value in S-0-0016 must always be the position actual value (S-0-0051 for motor encoder or S-0-0053 for external encoder).
- Only the ident numbers that are included in the list of the configurable data set in the AT (S-0-0187) of the corresponding drive may be entered.

6.1.3 Data size of position set-values and actual values

The position set-values and actual values are transmitted in mm or degrees:

Weighting translation	:	1 LSB = 10^{-4} mm
Weighting rotation	:	1 LSB = 10^{-4}

The axes can be operated as:

- SERCOS absolute interface which enables a two-channel limit switch monitoring (in the rho4.0 and in the drive amplifier).
- Modulo interface which enablesa two-channel limit switch monitoring (in the rho4.0). The axes can be operated as endless axes, e.g. as round axis.

6.2 Machine parameter

The interface is activated via machine parameters. A distinction is made between control and drive parameters, i.e. there are two different, separate machine parameter files that are loaded into the control in the startup.

6.2.1 SERCOS specific control parameters

Machine parameter P401

With P401, the following settings declares an axis as a SERCOS axis.

A_1 Servo-B.	:	1	number of the servo board (at present always 1)
A_1 SCS Plug	:	X71	plug No. of the SERCOS interface
A_1 SCS Ring-No.	:	1	running number of the SERCOS ring (at present always 1)
A_1 SCS Axis No	:	1	SERCOS address of the corresponding drive amplifier (corresponds to the setting on the Personality module in the amplifier).
A_1 drive type	:	10	identification for Bosch Rexroth Servodyn-D-SERCOS drives
A_1 enc.dist/rot:	:	1.0	Distance of the axis per encoder revolution in mm or deg (is only required with RC-lead referencing)
A_1 Ref.-Mode	:	0	0..3 RC−lead referencing 10 drive−lead referencing

SERCOS interface

A_1 Modulo value : 0

0 Absolute interface

> 0 Modulo interface with corresponding modulo value in mm or deg

To be set for endless axes

When using the modulo interface, the following settings must be performed at the drive amplifier:

- S-0-0076 Weighting of the position data:

  Bit 7 = 1 (processing format = modulo format)

- P−0−0006 position encoder type motor

  Bit 0 = 1 (absolute value encoder)

- S-X-0103 Modulo value

  must match the rho4 module value described here.

☞**The modulo value must be set at least so high that at the most the half modulo value is runper scanning time (P5) with maximum axis velocity.**

Example:

max. axis velocity (P103) = 50 deg/s

Clock start time (P5) = 6 ms

=> path run/Clock = 50/1000 * 6 = 0,3 deg

=> min. modulo value = 0,3 deg * 2= 0,6 deg

A_1 ms.fact : 1000.00 Measuring system factor

When using Bosch Rexroth drive amplifiers always enter 1000.00.

A_1 Com.output : 1 Running numbering of the set value outputs (no double occupation admissible)

SERCOS interface

Belt encoder input

The SERCOS interface of a regulated axis can be also used as belt encoder input.

P401 is to be set as follows for the belt input:

BN1 Servo-B.	:	1	number of the servo board (at present always 1)
BN1 SCS plug-no :		X71	plug No. of the SERCOS interface
BN1 SCS ring-no.	:	1	running number of the SERCOS ring (at present always 1)
BN1 SCS axis no.	:	1	SERCOS address of the corresponding drive amplifier. (Only the address of a regulated axis already existing can be entered.)
BN1 modulo-value	:	8	Only modulo interface (entry > 0) allowed. Value must match the value of the corresponding regulated axis.
BN1 ms.fact	:	1000.00	Measuring system factor When using Bosch Rexroth drive amplifiers always enter 1000.00.

Machine parameter P38

The baud rate and the download of the drive parameters are set via P38 SERCOS-INTERFACE.

Baud rate [MBaud]:	2	Transfer rate of the interface
Par. download (0/1):	1	Perform download of drive parameters.

6.2.2 SERCOS specific drive parameters

The SERCOS specific drive parameters are called in the following extended Machine parameter (XMP). They are saved in the control in a special file. These parameters are only transferred in the initialization phase per download to the drives. They are else not required in the control.

An ASCII file created with the Bosch Rexroth DSS program is used as basis. The program DSS (Diagnosis and Service System) is used as startup help and diagnosis tool for the drive converter Servodyn-D.

After completed drive optimization, an ASCII file, which contains all drive parameters of a special axis, can be created per DSS. This must be performed for all axes.

SERCOS interface

An ASCII → Bin converter is available which generates a binary file from the ASCII files. The call of the ASCII → Bin converter occurs under ROPS4 (comp. documentation XMP converter). The created binary file is loaded into the control per ROPS4 coupling. It gets a fixed Ram area allocated in the control in analogy to the classical machine parameters.

The rho4 control is the master in the configuration-dependent drive parameters.

The following interface parameters are determined automatically by the control and transferred to the drives at every startup:

S-0-0001	NC cycle time (TNcyc)
S-0-0002	SERCOS cycle time (TScyc)
S-0-0006	Emission moment drive telegram (T1)
S-0-0008	Moment for set value-valid (T3)
S-0-0009	Initial address in the master data telegram
S-0-00010	Length Master data telegram
S-0-00015	Telegram type parameter
S-0-00016	Configuration list drive telegram (default)
S-0-00024	Configuration list master data telegram
S-0-00089	Emission moment master data telegram (T2)

If one of the listed parameters is set additionally in the DSS Ascii File, the values, except S-0-0016, are ignored during the download.

If the parameter download is activated in the machine parameter P38, it is performed automatically for all axes at the startup of the control.

☞ **The download requires much time. For a RC cycle time (P2 clock time) of 10 [msec], the loading of all essential parameters in the drive requires approx. 12 seconds per axis. The whole startup of the control will be accordingly delayed.**
Recommendation: After completed optimization, the drive parameter download is performed once for all axes. The drive parameters are saved in the Eeprom of the drive with SERCOS commando 'save memory command' (S-0-0264 for Bosch Rexroth Servodyn-D drives). The download is then switched off in the parameter P38.

SERCOS interface

6.3 Referencing

6.3.1 RC-controlled referencing

P401 Ref.-Mode = 0..3

The axes are referenced under rho4 control. Description of the different Ref modes, see software manual Control functions (No. 1070072420), chapter Reference point run at Servodyn-D with CAN interface.

The zero-point shift is set at the SERCOS drive via the parameter S-0-0177 (absolute dimension Offset 1). The axis must be defined as absolute axis => P-0-0006 (position transmitter type motor encoder) = 1. Reference point actual value and reference point offset are set in the rho4 via parameters P207 and P208. The position actual value system in the drive is not influenced, i.e. the shift into the reference system (S-0-0172) is not carried out. RC-controlled referencing is only possible with cyclically absolute and absolute measuring systems.

6.3.2 Drive-controlled referencing

P401 Ref.-Mode = 10

Referencing is started by the rho4 via command ‘Drive-controlled referencing’ (S-0-0148).
The drive performs on its own the reference run according to the set parameters (see Servodyn-D parameter manual, No. 1070066018). After the reference run is completed, the reference position determined by the drive is adopted by the rho4 as actual value.

Activation in the rho4

Parameter P402 (direction Go to reference point) must be set on <> 0. The direction of the reference run set in the rho4.0 has no effect. P403 (effectiveness reference point switch) has also no effect. The real reference run direction and effectiveness of the reference point switch is set via S-0-0147 (reference run parameter) in the drive.

The run to the reference point is started in the rho4.0 via mode 1 in the manual operation or via the command REF_PNT in the BAPS program in the automatic operation.

Special features or restrictions for drive-lead referencing:

- not appropriate for mechanically coupled axes, e.g. lift head spindle
- can only be started for a maximum of 8 SERCOS axes at the same time

SERCOS interface

  - during the attempt to start other axes, in the manual operation the warning ‘Separate referenc.! axis x’ (code-No. 268416 + axis number −1) arises.
  - in the automatic operation, the program with status message ‘Separate referenc. axis x’ (code No. 268416 + axis No. −1) is interrupted
- if the reference run is interrupted owing to an error by the drive, in the rho4 the status message ‘Ref. point error axis x’ (code No. 268288 + axis No. −1) arises.

SERCOS interface

6.4 Status messages and warnings

If an error at the SERCOS interface occurs during the startup phase, the startup of the control (without complete initialization of the SERCOS interface) will be finished. The errors that have occured can then be displayed via MODE 7.2 at the PHG or via the ROPS4/Online status.

6.4.1 Status messages at the startup of the SERCOS interface

The following status messages can arise when the SERCOS interface is initialized:

Sercos-Init-error (code = 169984)	SERCOS-ASIC cannot be initialized Check SERCOS hardware
Sercos-Ring open Phase: 0: (code = 169088)	The SERCOS ring is not closed Check SERCOS connections
Sercos error Phase1 axis: x (code = 169472 + x − 1)	The indicated axis could not be identified. x = axis number, e. g. x = 3 means 3rd axis Check axis addresses (in rho4 and drive amplifier) Check baud rate
Sercos error Phase2 axis x Y− zzz (code = 169600)	The indicated drive parameter cannot be transmitted owing to an error
Sercos-error Phase3 axis x Y− zzz (code = 169728)	x = axis number Y = SERCOS parameter type (S = Standard, P= Product specific) zzz = SERCOS ident. number
Sercos-error Phase4 axis x Y− zzz (code = 169728)	Check parameter value (heed limit values)
Sercos-error Phase2 axis x S− 127 (code = 169600)	Conversion into phase 3 cannot be performed Check via DSS list of the invalid parameters in phase 2 (S-0-0021) Additionally the invalid parameters are displayed as incorrect
Sercos-error Phase3 axis x S− 128 (code = 169728)	Conversion into phase 4 cannot be performed Check via DSS list of the invalid parameters in phase 3 (S-0-0022) Additionally the invalid parameters are displayed as incorrect

In incorrect parameter for parameter sets <> 0 (e.g. S−1−0104), the SERCOS−Ident number is displayed with 5 digits. The first digit indicates the number of the parameter set, the other four digits the parameter number.

Example:

S−1−0104 of axes 1 is incorrect

Error display: Sercos error Phase2
axis 1 S− 10104
code= 169600

6.4.2 SERCOS specific state messages for running time

Sercos-Alarm axis: x (code = 168960 + x − 1)	Drive amplifier has recognized an error (x = Axis No.) Read error code at the seven segment display of the drive amplifier or read via DSS ‘State class 1’ (S-0-0011)
separate referenc.! axis: x (code = 268416 + x − 1)	In the automatic operation the attempt was made to start drive-lead referencing for more than 8 SERCOS axes at the same time (x = axis number of the axes that cannot be started)
Ref. point error axis: x (code = 268288 + x − 1)	Drive-lead referencing has been interrupted owing to an error by the drive Read error code to the seven segment display of the drive amplifier

6.4.3 Warnings

separate referenc.! axis: x (code = 268416 + x − 1)	In the manual operation the attempt was made to start drive-lead referencing for more than 8 SERCOS axes at the same time (x = axis number of the axes that cannot be started)

SERCOS interface

6.5 Transmission of cyclic drive data to the PCLrho4.0

The data which is transmitted via the SERCOS bus in the cyclic axis telegram from the drive to the rho4.0, are copied into the I/O area of the PCL. The user can have access to drive data that can be chosen and define himself the configuration of the cyclic axis telegrams within certain limits.

6.5.1 Definition of the cyclic axis telegrams (AT)

Within the rho4.0

As input buffer for axis telegrams, a maximum of 22 bytes per axis is defined in the rho4.0. This corresponds to the maximum AT length for Bosch Rexroth Servodyn-D drives. The number of the identifiers in the configurable AT for Servodyn-D drives is limited to 5 at the maximum.

The input buffer in the rho4 is assigned as a standard as follows:

S-0-0135	2 Byte	drive status
S-0-0051	4 Byte	position actual value (motor encoder)
S-0-0189	4 Byte	hauling distance
S-0-0144	2 Byte	signal status word
S-0-0084	2 Byte	torque actual value [%]
S-0-0040	4 Byte	velocity actual value
	4 Byte	reserve

The drive status is a fixed part of the axis telegram and always present. The other data is preset by the rho4.0 during the startup via S-0-0016 (configuration list AT) in all drives.

For drives that support only short ATs, the list will be shortened correspondingly.

Example:

Maximum length of the configurable AT	= 10 bytes
S-0-0016	= (S-0-0051, S-0-0189, S-0-144)
+ 2 bytes drive status	=> Total length = 12 bytes

SERCOS interface

In the drive

The axis telegrams can be configured by the user in the drive amplifier. To overwrite the configuration preset by the rho4, the ident number S-0-0016 must be taken over into the list of the parameters to be transferred by the rho4 (scs file).

To define the S-0-0016, the following is to be observed:

- drive status is a fixed part of the AT
- Length of the configurable AT: 20 bytes or maximum length supported by the drive (S-0-0185), if it is smaller than 20 bytes
- Maximum number of ident numbers in the AT of the corresponding drive amplifier, e.g. 5 for Servodyn-D drives
- The first value in S-0-0016 must always be the position actual value (S-0-0051 for motor encoder or S-0-0053 for external encoder)
- Only ident numbers may be entered which are contained in the list of the configurable data set in the AT (S-0-0187) of the corresponding drive

6.5.2 Treating of the axis telegrams in the rho4.0

From the rho4.0, the ATs of all axes are read in each scanning cycle. Always 22 bytes per axis are reached to the PLC. If less than 22 bytes are sent by the drive, the free area is filled with zeros. The data is passed on 1:1, i.e. it is available in the PLC program accordingly to the weight set in the drive.

Example:

S-0-0051 Position actual value

Data type	:	Integer32 (4Byte)
Display format	:	Decimal
Number of digits after decimal point	:	4
Assessment factor	:	1

=> 12500 corresponds to 1.25 [mm or degree]

The first both values in the AT (drive status and position actual value) are used additionally within the rho4.0 for the error display and the position display.

SERCOS interface

6.5.3 Transmission of the axis telegrams to the PCLrho4.0

The ATs are defined in the rho4.0 with 22 bytes per axis. With a maximumconfiguration with 16 axes, it follows a transmission length of 528 bytes.

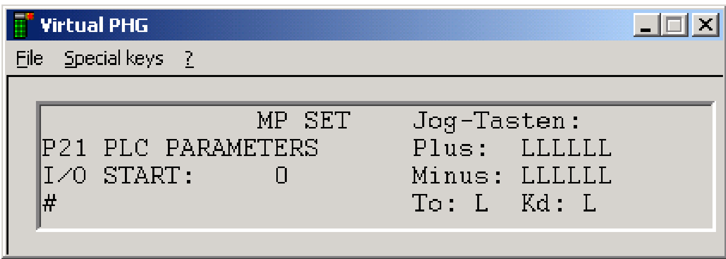
☞ **Depending on the system stress, there can be delays in the transmission, so that no clock synchronous transmission can be guaranteed. Moreover, it cannot be guaranteed that there is a transmission to the PCL at each clock (P5). If a transmission of drive data to the PCL is not yet completed and there is new drive data on the rho4.0 that is not transmitted yet, this data will be then rejected. The data consistency of the ATs to the rho4.0 I/O interface is guaranteed in this way.**

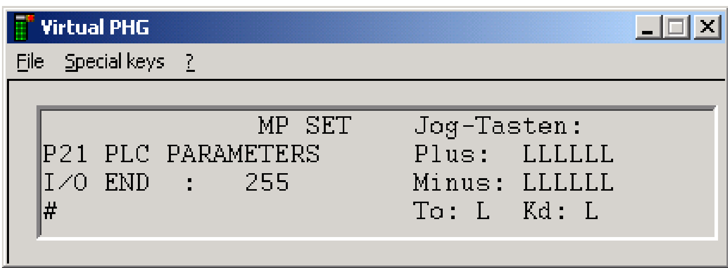
Setting P21 PLC PARAMETER

For the setting of the parameter P21, the final address of the machine parameter is increased by the number of the ATs to be transmitted.

22 Byte AT per axis * 16 axes	=	352 Byte
Reserve	=	108 Byte
rho4.0 system counter	=	4 Byte
=> Number of the bytes to be transmitted	=	464 Byte

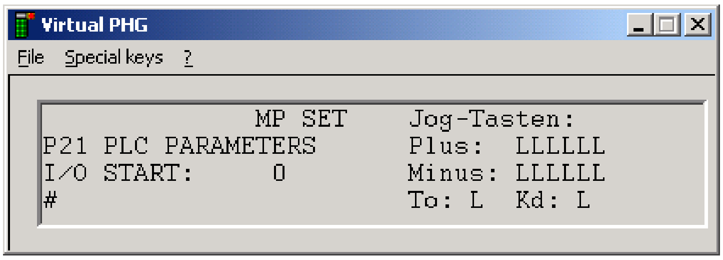
State of rho4.0 when delivered:



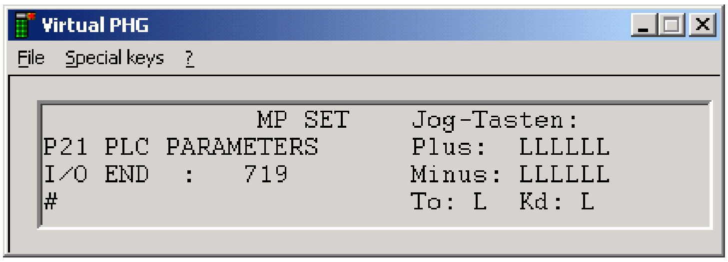


SERCOS interface

Setting with AT:
The setting of the PLC start address remains unchanged.



The setting of the PLC final address results from:
PLC final address + 464 = 719



☞ **In case of a switching with field bus (PCI_BM-DP, PCI_BM-IBS or PCI_BM-CAN), an offset of 512 bytes must be set.**

6.5.4 Transmission of the rho4.0 system counter

Additionally to the ATs, the internal rho4.0 system counter is supplied at each transmission to the PCL. The counter is increased by the clock start time set in the machine parameter P5 at each clock of the rho4.0.

The rho4.0 system counter occupies 4 bytes and has the value range −2.147.483.648 to 2.147.483.647.

☞ **The rho4.0 system counter is an allround-counter. When the upper limit is exceeded, the rho4.0 system counter becomes negative, e.g. after 2.147.483.647 follows − 2.147.483.643 at the clock start time P5 = 6ms.**

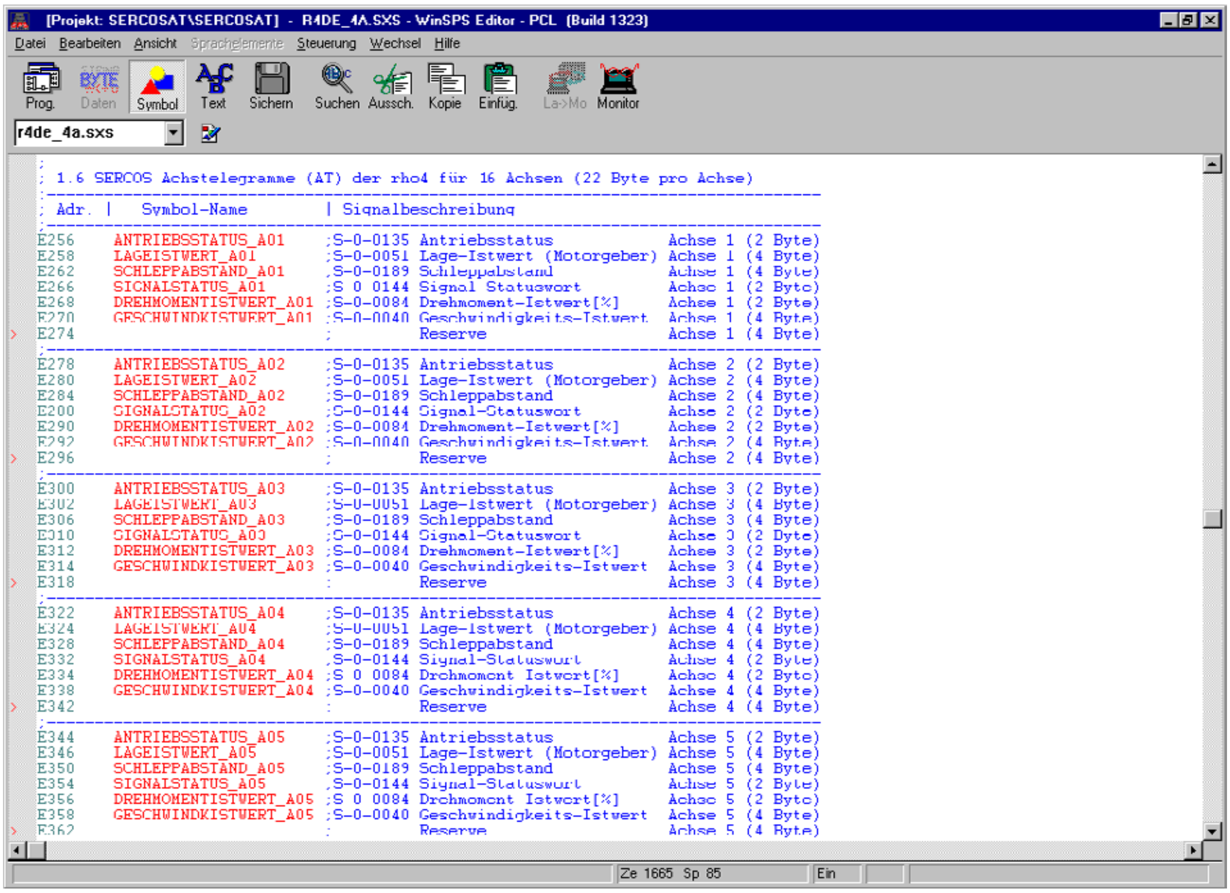
SERCOS interface

6.5.5 Representation of the ATs in the PCLrho4.0

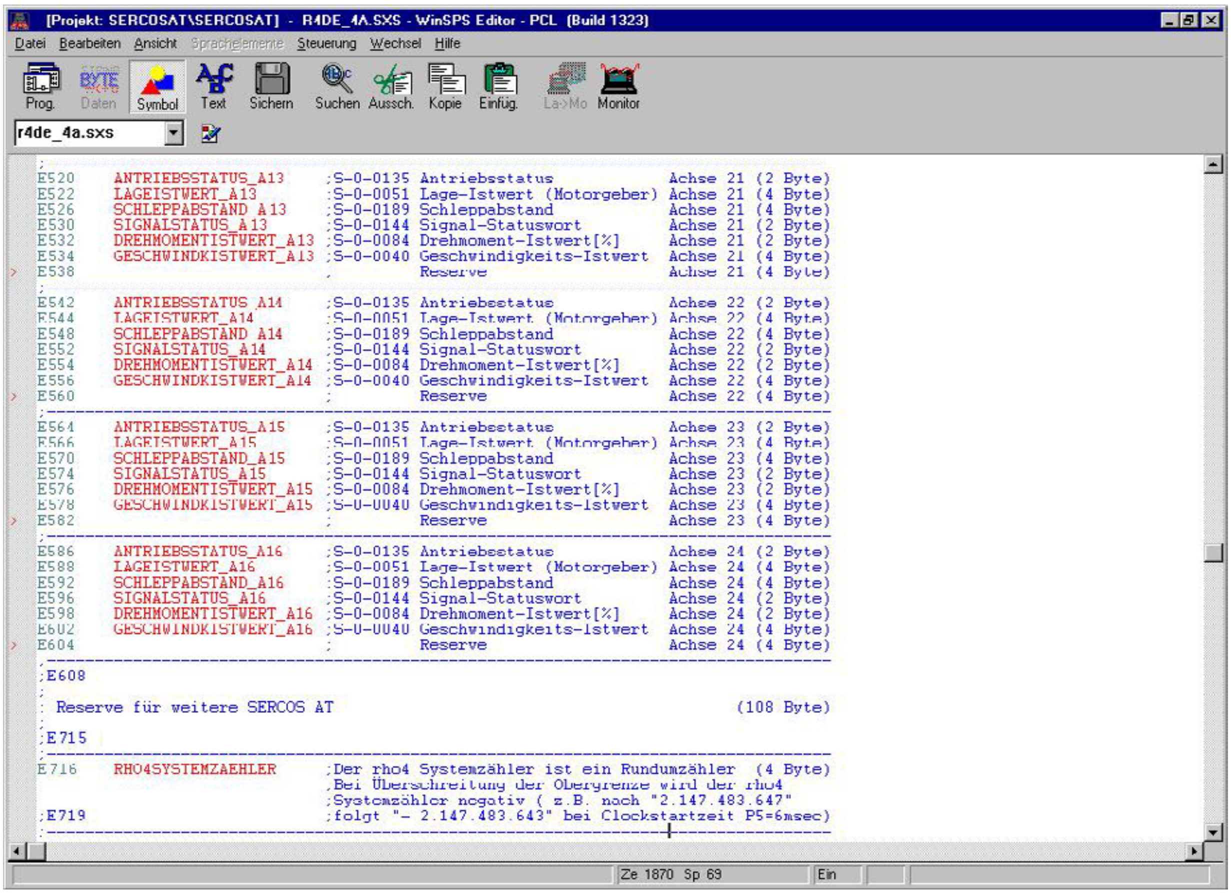
The following pictures show an example of the SERCOS axis telegrams on the PCL.

In the symbol file of the PCL program, 640 bytes in the I/O area of the PCL are reserved. The number 640 bytes results from:

22 Byte AT per axis * 24 axes	=	528 Byte
Reserve	=	108 Byte
rho4.0 system counter	=	4 Byte
=> Number of the bytes to be transmitted	=	640 Byte



SERCOS interface



☞ **The address situation of the ATs is not set for all values on 32 bit limits. The values with 4 byte characterization must be copied before the evaluation in the PCL on 32 bit limits.**

The values are to be weighted as described in section 6.5.2. This also applies to the other values of the ATs.

SERCOS interface

6.6 Belt input via external encoder at EcoDrive/IndraDrive

6.6.1 General

As an alternative to the so far available measuring system interfaces for belt encoder (incremental interface at the rho4.1, CANopen interface at rho4.1 and rho4.0), from version VO08 it is possible to connect an external incremental encoder to the drive amplifier as belt input.

6.6.2 Function

The external encoder input of a drive amplifier is usable as belt input independent of the defined controlled axis on the amplifier. To activate the external encoder, the following described parameter in the drive, resp. rho4, must be set.

6.6.3 Parameter at the EcoDrive

S−0−0076 Type of weighting for position data weighting

The type of weighting of the position data weighting determines, in which format the position data between drive and control, resp. user interface is exchanged. If the external encoder is used as belt encoder, especially for endless belt, the modulo format (Bit 7 = 1) must be adjusted.

S−0−0103 Modulo value position

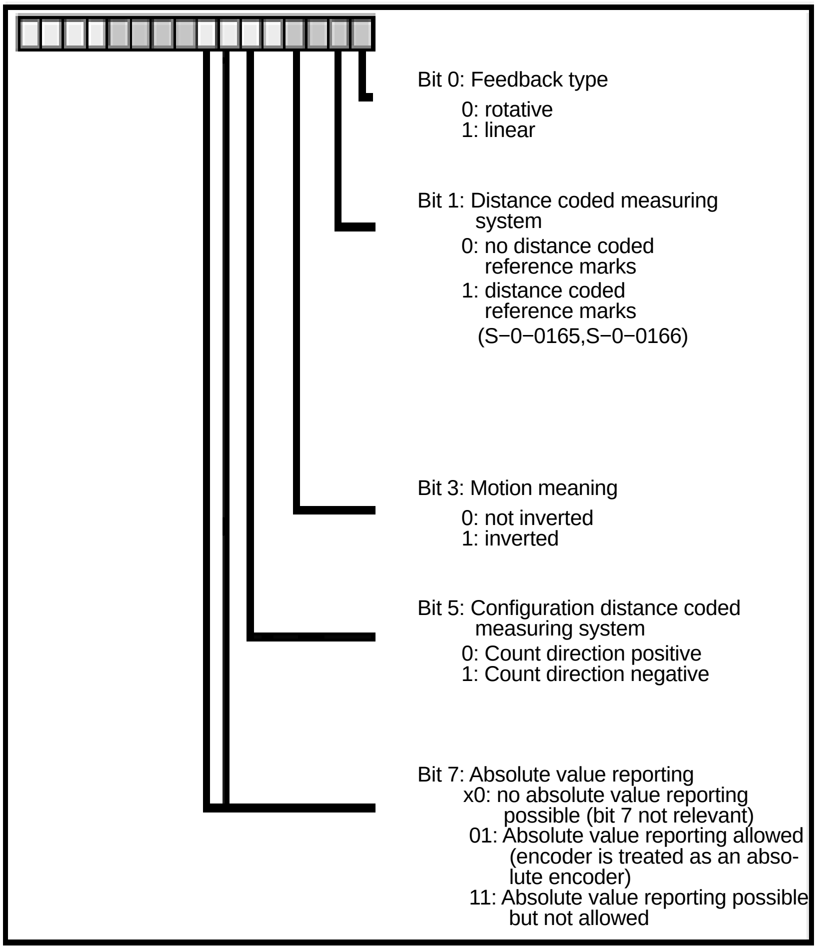
At adjusted modulo format (parameter S−0−0076, type of weighting of the position data weighting bit 7), the modulo value determines, at which numerical value the position data overflow to zero. If encoder1 is an absolute encoder, the modulo value must be adjusted greater then the traversing range of the controlled axis. The adjusted modulo value must agree with the corresponding parameter in the rho4 (P401 4. subquery ’modulo value’). The maximum value is 214748.3647 (max. traversing range).

S−0−0115 Position feedback 2 type

This parameter determines the essential properties of the external encoder.

SERCOS interface

Configuration of the parameter



S−0−0117 Feedback 2 resolution

This parameter indicates the resolution of the external encoder.

This value means:

- With rotative encoders the number of divide periods or cycles per rotation of the encoder shaft (DP/rotations)
- With linear motors the resolution in mm (mm/bar division)
- With resolver the number of pairs of poles of the resolver

The meaning of the value of S−0−0117 is determined by S−0−0115, position feedback 2 type (rotative or linear encoder).

Because for paramerization of the external encoder no own parameter for gearfactor and feedconstant are available, both values must be included in the calculation of S−0−0117.

S−0−0117 must be calculated as follows:

$$S\text{−}0\text{−}0117 = \frac{\text{enoder resolution x 360}}{\text{feedconstant}} \times \frac{\text{motor rotations n1}}{\text{output rotations n2}}$$

S−0−0391 Monitoring window feedback 2

This parameter value defines the maximal allowed deviation of the position actual value between motor encoder and external encoder (S−0−0051, position feedback value encoder 1, S−0−0053, position feedback value encoder 2).
With use of the encoder as an independent belt input, the monitoring must be disabled by setting this parameter to zero.

P−0−0075 Encoder type 2

The determination of the encoder interface where the optional encoder is connected to, takes place by this parameter. The identification number of the according interface module must set as follows:

- 2 Incremental encoder with sine signals (1V signals) by the company Heidenhain
- 5 Incremental encoder with rectangle signals by the company Heidenhain
- 8 Encoder with EnDat interface
- 9 Cogwheel with 1Vpp signals

P−0−0185 Control word of encoder 2 (optional encoder)

This parameter offers the possibility to assign a determinated functionality to the optional encoder. If the optional encoder is used as belt encoder, a “4” must be entered (optional encoder as spindle encoder).

S−0−0016 Configuration list of AT

If the belt input via drive amplifier in the rho4 is activated, the drive telegram of the corresponding axis deviating from the standard settings (see manual system description rho4.1, resp. rho4.0) is as follows:

- S−0−0135 Drive status word 2 Byte
- S−0−0051 Position feedback 1 value (motor encoder) 4 Byte
- **S−0−0053 Position feedback 2 value (ext. encoder) 4 Byte**
- S−0−0189 Following distance 4 Byte
- S−0−0144 Signal status word 2 Byte
- S−0−0084 Torque/force feedback [%] 2 Byte

SERCOS interface

If the drive telegram is reconfigured by the user, it must noticed, that the ident numbers S−0−0051 (Position feedback 1 value motor encoder) and S−0−0053 (Position feedback 2 value ext. encoder) are installed as aforementioned. The remaining parts of the drive telegram are freely usable by the user.

6.6.4 Parameter at the IndraDrive

Hardware requirements

A module for the connection of an external encoder (i.e. HFI02) must be installed in the drive amplifier.

S−0−0076 Type of weighting for position data weighting

The type of weighting of the position data weighting determines, in which format the position data between drive and control, resp. user interface is exchanged. If the external encoder is used as belt encoder, especially for endless belt , the modulo format (Bit 7 = 1) must be adjusted.

S−0−0103 Modulo value position

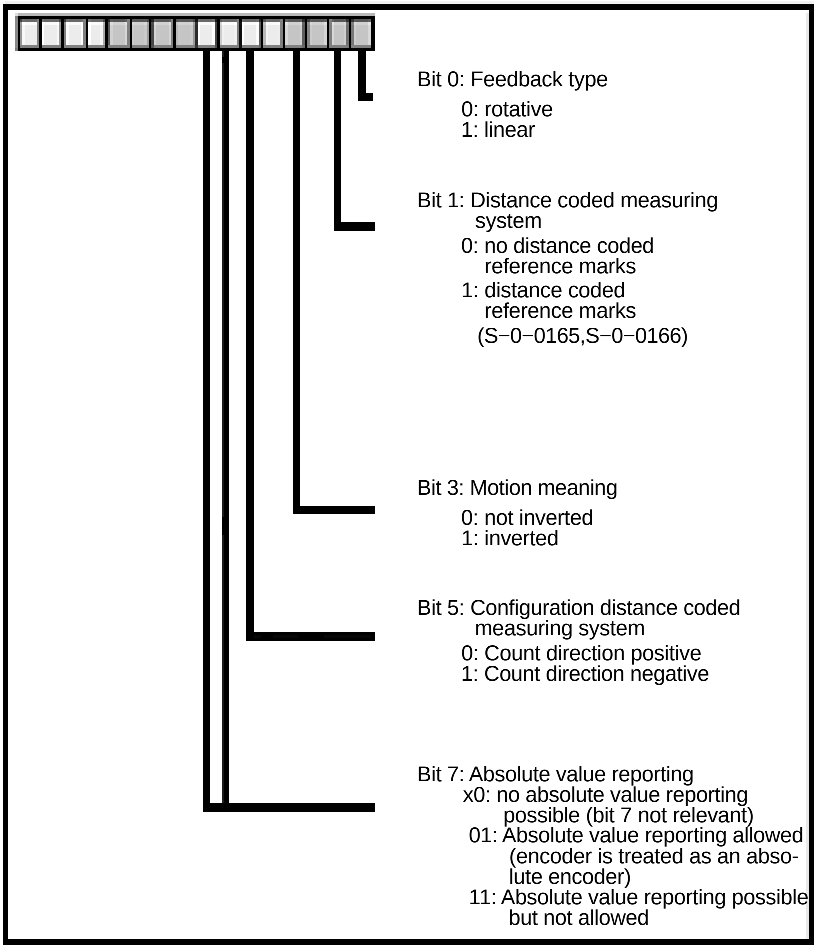
At adjusted modulo format (parameter S−0−0076, type of weighting of the position data weighting bit 7), the modulo value determines, at which numerical value the position data overflow to zero. If encoder1 is an absolute encoder, the modulo value must be adjusted greater then the traversing range of the controlled axis. The adjusted modulo value must agree with the corresponding parameter in the rho4 (P401 4. subquery ’modulo value’). The maximum value is 214748.3647 (max. traversing range).

S−0−0115 Position feedback 2 type

This parameter determines essential properties of the external encoder.

SERCOS interface

Configuration of the parameter



If the external encoder is used as belt encoder, only feeedback type “rotative” is permitted (Bit 0 = 0).

S−0−0117 Feedback 2 resolution

This parameter indicates the resolution of the external encoder. The value means with rotative encoders the number of deviding periods or cycles per rotation of the encoder shaft (DP/rotations).

Because for paramerization of the external encoder no own parameter for gearfactor and feedconstant are available, both values must be included in the calculation of S−0−0117.

S−0−0117 must be calculated as follows:

$$\text{S−0−0117} = \frac{\text{enoder resolution} \times 360}{\text{feed constant}} \times \frac{\text{motor rotations n1}}{\text{output rotations n2}}$$

S−0−0391 Monitoring window feedback 2

The value of this parameter defines the maximal admissible deviation of the position feedback value between motor encoder and external encoder S−0−0051 (Position feedback encoder 1, S−0−0053 Position feedback encoder 2).
With use of the encoder as an independent belt input, the monitoring must be disabled by setting this parameter to zero.

P−0−0075 Encoder type 2

The determination of the encoder interface where the optional encoder is connected to, takes place by this parameter. The identification number of the according interface module must set as follows:

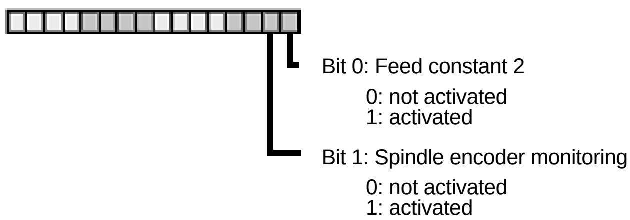
- 0 No encoder
- 1 GDS/GDM encoder from Bosch-Rexroth
- 2 Incremental encoder with sine signals (1V signals) by the company Heidenhain
- 3 Resolver with data memory
- 4 Encoder with Hiperface interface
- 5 Incremental encoder with rectangle signals by the company Heidenhain
- 6 Motor encoder of motors MSK (Motor type keyfield ”Encoder”= S2 resp. M2)
- 8 Encoder with EnDat2.1 interface by the company Heidenhain
- 10 Resolver encoder without data memory

P−0−0078 Assignment optional encoder −> option position

This parameter defines the interface which is determined for the optional (external) encoder:

- 0 none
- 1 X4 (option 1)
- 2 X8 (option 2)
- 3 X10 (option 3)

P−0−0185 Control word of encoder 2 (optional encoder)



If the external encoder is used as belt encoder, P−0−0185 must set to zero.

SERCOS interface

S−0−0016 Configuration list of AT

If the belt input via drive amplifier in the rho4 is activated, the drive telegram of the corresponding axis deviating from the standard settings (see manual system description rho4.1, resp. rho4.0) is as follows:

- S−0−0135 Drive status word 2 Byte
- S−0−0051 Position feedback 1 value (motor encoder) 4 Byte
- **S−0−0053 Position feedback 2 value (ext. encoder) 4 Byte**
- S−0−0189 Following distance 4 Byte
- S−0−0144 Signal status word 2 Byte
- S−0−0084 Torque/force feedback [%] 2 Byte

If the drive telegram is reconfigured by the user, it must be noticed, that the ident numbers S−0−0051 (Position feedback 1 value motor encoder) and S−0−0053 (Position feedback 2 value ext. encoder) are installed as aforementioned. The remaining parts of the drive telegram are freely usable by the user.

6.6.5 Machine parameter at rho4

The required settings in the rho4 must be set in **P401 Equipment of the measuring system boards**.

P401, 4. Subquery: (dependent on measuring system)

SERCOS axis number

Indication of the SERCOS-address of the corresponding drive amplifier (corresponds to the setting in the firmware module of the EcoDrive, resp. P−0−4025 with IndraDrive).
For activation of the additional external encoder as belt input, the axis address must be entered negatived.

Example: external encoder on drive with address 15
Input: −15

Modulo value

Here the adjusted value of S−0−0103 at the drive must be accepted.

P401, 5. Subquery: Measuring system factor

Always the value 1000 must be entered here.

6.6.6 Restrictions

Both measuring systems on a drive amplifier work widely independently. Merely at switch-on of the drive amplifier, resp. restart of the rho4, and with drive-controlled referencing, the position value of the external encoder (S−0−0053) is equated to the position value of the internal encoder (S−0−0051).
Therefore it must be ensured, that after a restart or referencing the belt encoder value is always set to the desired value by the belt synchronisation logic (reset belt encoder).

7 Software

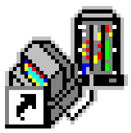
7.1 ROPS4/Online

ROPS4/Online permits the creation, testing and archiving of programs for the rho4.0 control system. The program runs under Windows 95 or Windows NT. It is not absolutely essential to use a mouse, however the use of one is recommended for the sake of improved operational comfort. ROPS4 is protected against illegal copying. A software licence must be applied for before working with it.

ROPS4/Online has the following functions:

- BAPS plus: A structure-oriented programming system, with which procedural programs can be quickly and easily created, documented and tested in a top-down design. The essential advantage of this expansion is the automatic generation of codes and the option of being able to monitor the process. The program sequence is compiled from icons in the form of a program sequence plan.
- Machine parameter converter: Using a converter, readable and editable ASCII *.amp files can, on the one hand, be created from *.bin machine parameter files. The other part of the converter converts ASCII files into machine parameter files, which can then be loaded into the control system.
- Integrated BAPS translator for translating the motion programs created using an ASCII editor.
- Program archiving (load, save, list, delete, name etc.)
- Remote control functions: Program start-up, axial position display, I/O status etc.
- Linking functions: Function for transmitting files from/to the rho4. Connection to the rho4.0 via a serial interface or TCP/IP.
- Offline/Online points editor.
- Process functions: Process selection, process start-up, process stop etc.

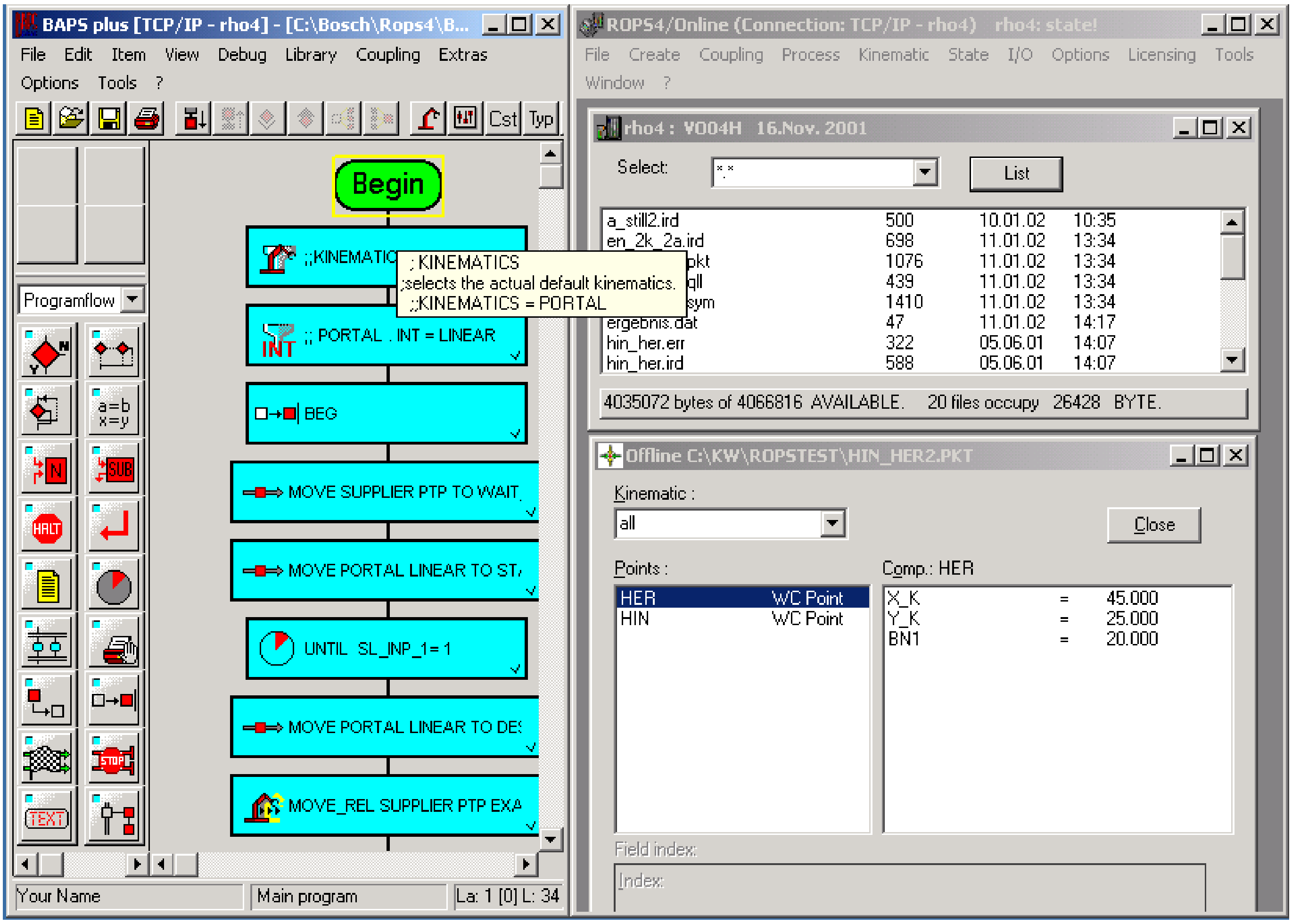
The ROPS4/Online program can be called up from the rho4 folder by selecting the appropriate icon.



Software

Screenlayout

All applications within ROPS4/Online store their size and position. A user, therefore, can always have the same screen display which he created and with which he is familiar , as shown, for example, in the following figure.



All applications which run under ROPS4/Online can be called up directly via the Start menu. The applications have in their menu bar a special menu item called Tools, under which the other programs can be accessed. This menu item can be expanded and modified by the user as required.

Software

7.2 Server/Client functionality

The ’hosts’ file

The ’hosts’ ASCII file contains the IP addresses and an associated host name. The corresponding server can be contacted under this host name. The alias names can be freely chosen and are optional.

Example of the ’hosts’ file

IP-Adresse	Host name	Alias name
192.168.4.1	rho 4	rho40
142.2.47.11	GateWay	PHG connection
142.3.00.07	PLC	PLC host computer

The file ’hosts’ is located in system directory C:\Winnt\system32\drivers\etc.

rho4.0 server

The rho4.0 has several servers which each have their own port number but which have the same IP address.

Server name	Port number	Services
Coupling	6010	ROPS link
Printer	6020	print
SER_1	6031	V24_1
SER_2	6032	V24_2
SER_3	6033	V24_3
SER_4	6034	V24_4
WIN_1	6051	Read / write in BAPS
WIN_2	6052	Read / write in BAPS
WIN_3	6053	Read / write in BAPS
WIN_4	6054	Read / write in BAPS
rho_Function_1	6091	rho4 library functions
rho_Function_2	6092	rho4 library functions
rho_Function_3	6093	rho4 library functions
rho_Function_4	6094	rho4 library functions
Virt_PHG	6100	Operation of the rho4 using the virtual PHG2000
Virt_Panel	6110	Simulation (virtual) of a PLC control panel (switches and lamps)

rho4.0 clients

The following clients are contained in the operating system of the rho4.0. The relevant servers are located in the PCL.

Client name	Port number	Services
PLC_Interface	5100	Cyclical transmission of the interface data
PLC	5110	Write PLC in BAPS

8 File management

The rho4.0 file system is used for user memory management. The visual appearance and the range of functions are similar to file management under DOS (FAT) or Windows 95 (VFAT).

Some controller-specific functions such as real-time capability and password protection.

8.1 File management function

The rho4.0 file system is block oriented, i.e. a file is distributed over a number of data blocks. The data blocks are 512 bytes in size.

Block 1 Adr: 00512	Text.dat This is a sequential file. The blocks are located at
Block 2 Adr: 01024	contiguous addresses
Block 3 Adr: 01536	**Free block**
Block 3 Adr: 02048	**Free block**
	...
	...
	...
Block n Adr: XXXXX	**Last block**
	File is sequential in memory

Block 1 Adr: 00512	Text.dat This is a **non-**sequential file. The blocks are
Block 2 Adr: 01024	**Occupied block**
Block 3 Adr: 01536	distributed in memory
Block 3 Adr: 02048	**Free block**
	...
	...
	...
Block n Adr: XXXXX	**Last block**
	File is not sequential in memory

File management

When a file is created, the file system attempts to store the file in contiguous (sequential) blocks. This presupposes that a free contiguous memory range exists that is big enough to accept the file. Where a range of this nature does not exist, the file is saved to memory in distributed form.

The memory administration occupies a part of the available memory for so-called organisation blocks in order to manage the data (data blocks) , i.e. not the whole memory storage space is available for filing net data.

Total memory = data blocks (files) + management blocks

To open files

The full name of a file must be given to open a file.

A file can be opened for reading many times (maximum 255 times). i.e. various processes can read from the same file.

However a file can only be opened for writing once.

Closing files and backup copy

When an existing file is opened for writing, the file management system first creates a copy of the original. This copy is only available to the process which opened it until it is closed down. Until then the old file remains unchanged.

After the file has been closed, the old file is removed from the directory structure and the new file is entered. If the old file still has read-access, e.g. through a process, then the file which has been removed remains and is only completely discarded when it is closed through this process.

Process handling

Processes (ird files) must be located in the memory sequentially in order to be executed.

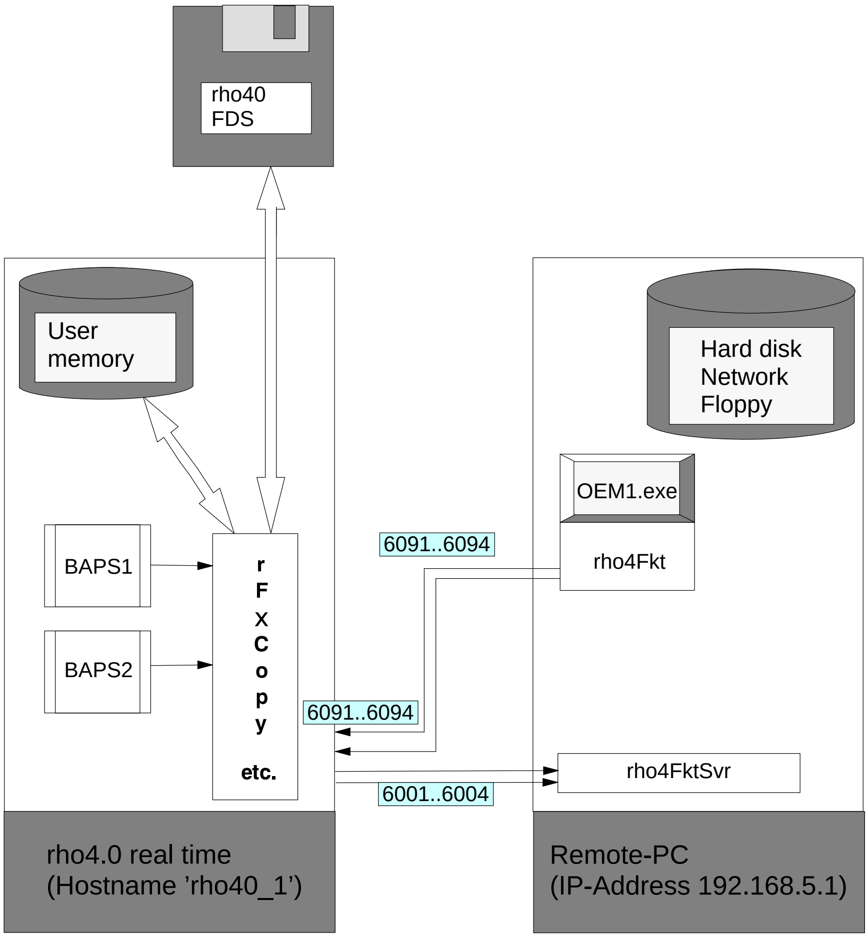
When a process is chosen the ird file is sequentialized first. If this does not succeed because no corresponding range is free, an error message is produced.

File management

8.2 Options for accessing file management

The OEM is able to have direct access into the user memory. This access is implemented through C functions which are held in the function libraries.

The rho4.0 has 2 classes for working memory access.



rho file [rF] class

The rho file class functions allow access to the rho4 file system as well as Windows file management (hard disk/network). This access is possible from a BAPS program and a Windows program.

A detailed description can be found in the manual 'DLL-library'.

Using the rho file [rF] class from Windows

The rho4Fkt.dll function library is used to link the functions under Windows. The required link types as well as detailed documentation are found in the include file rF.h.

4 servers are available in the controller (rho_Function_1 to rho_Function_4) for these services. This means: 4 Windows programs can access the file system simultaneously.

Using the rho file [rF] class from BAPS

The internal rho4 library functions, which is an internal function library, is used to link the functions under BAPS. The link types required for BAPS and detailed documentation are found in the include file rF.inc.

System channels (Win_Server_1 to Win_Server_4) are available for copy functions from and to the remote PC or from and to the floppy.

☞ **To be able to use the rhoFile functions, the program ’rho4FktSvr.exe’ and the files ’rho4FktSvr.ini’ and ’rho4fkt.dll’ are required. This program and the corresponding INI- and DLL-file are also installed during the ROPS4 installation.**
The files are then in the following directories:
C:\Bosch\rho4\winexe\rho4FktSvr.exe
C:\Bosch\origin\rho4FktSvr.ini
C:\Bosch\rho4fkt\rho4fkt.dll
Server and INI file can be filed on the remote PC in any directory (but both in the same one). The DLL must be copied into the directory ’C:\WINNT\SYSTEM32’ of the remote PC.
Before calling a rhoFile-Funktion, the program ’rho4FktSvr.exe’ must have been started on the remote PC.

The parameterization for the copy function is explained below. The parameters of the other rho file functions are analogous.

Functions in the rhoFile [rF] class

☞ **Please note: The correctness of the software developed by the OEM cannot be verified by the controller. The function library contains only outline plausibility checks.**

The function names are composed of various abbreviations. Where:
r = rho, F = File, G = Get, Mem = Memory

File management

Function name	Function no.	Comment
rFxCopy()	8015	copies a file (incl. remote PC and floppy)
rFxRemove()	8025	deletes a file (incl. remote PC and floppy)
rFxRename()	8026	renames a file (incl. remote PC and floppy)
rFReadBlk()	8030	reads a block of n bytes from a rho4 file into a buffer
rFDir()	8040	File list of the rho4 files in user memory
rFxStat()	8055	returns the status of a file (incl. remote PC and floppy)
rFChmod()	8051	modifies the access rights to a file
rFMpFloppy()	8070	copies the machine parameter file of the rho4.0 to the rho4.0−Fds (Floppy Disc)
rFxMpFloppy()	8071	copies the extended machine parameter file of the rho4.0 to the rho4.0−Fds (Floppy Disc)

rhoArchive [rA] class

☞ **Can only be called as a Windows DLL**

The ROPS4Fkt.dll function library is used to link the functions. The required link types as well as detailed documentation are found in the include file rA.h.

Only one server is available in the controller (rho-Online functions) for these services. ROPS4, the DDE Server and perhaps an OEM.exe share this server.

It is recommended that functions in the rhoFile class be used for access to the file system.
The functions in the rhoArchivierung class can also be used through serial coupling.

rhoArchive [rA] class functions

☞ **Can only be called as a Windows DLL**

These functions are used to save the contents of the working memory to the hard disk or floppy.

File management

The required C types and detailed documentation are found in the include file rA.h.

Also possible through WIN DDE. See DDE Server 4 manual.

The function names are composed of various abbreviations. With:
r = rho, A = Archive

Function name [rA]	**Function no.**	**Comment**
rAComInit()	14000	Initializes the serial port
rAComExit()	14001	Closes the serial port
rATCPCon()	14002	Establishes a connection to the IP / Port Address specified
rATCPDis()	14003	Closes the TCP connection
rAUpload()	14010	Copies a file from working memory to the hard disk
rADownload()	14020	Copies a file from the hard disk into working memory
rAList()	14030	Lists the files in the working memory
rARename()	14040	Renames a file in working memory
rADelete()	14050	Deletes a file in the working memory

8.3 File attributes

The attributes (eg read, write) of the files specify which actions (eg reading, writing) are permitted on the file concerned. Here a set attribute corresponds to the ”permission” for this action. That is to say, files with the read attribute may be read, for example. Or files without the write attribute cannot be written to.

The following attributes are available:

read	(R) Read authorization
write	(W) Write authorization
delete	(D) Delete authorization
hidden	(H) Hidden files
system	(S) System files

A file may have multiple attributes. When it is created a file has all attributes (RWD), i.e. everything is permitted.

The table below lists which actions demand which attribute:

	read	**write**	**delete**
list	X	--	--
read	X	--	--
write	--	X	--
delete	X	X	X
rename	--	X	--
copy	X	--	--
execute	X	X	--

X = attribute required
-- = no effect

The system and hidden attributes are a special case. Files with one of these attributes are not displayed by the controller, i.e. these files are hidden. Access to these files, e.g. execution, is not, however, restricted. The system and hidden attributes are password-protected.

The attributes are not copied when files are copied. This means that the copy is assigned all attributes (R, W, D).

Exception: Files created by a hidden process ’inherit’ the hidden attribute of this process.

Processes can modify the attributes of files with the function rFChmod().

File management

Process handling

The hidden attribute is ’inherited’. Hidden proceses create hidden files. Only hidden processes may modify the attributes of hidden files.

PHG2000 handling

The rho4.0 has a supervisor mode. This supervisor mode is activated and deactivated through the PHG2000. The relevant PHG2000 menu item is password-protected.
Hidden files are only displayed and may only be modified in this mode.

The supervisor mode is deactivated when the rho4.0 is started. Accordingly, hidden files will not be displayed, and the hidden attribute cannot be modified with the PHG2000.

A Appendix

A.1 Abbreviations

Abbreviation	Meaning
BAPS3	Movement and sequence programming language, version 3
C:	Drive designation, here drive C (hard disk drive)
CAN	Controller Area Network
DDE	Dynamic Data Exchange
DLL	Dynamic Link Library
EGB	Subassemblies at risk from electrostatic discharge
ESD	Electrostatic discharge Abbreviation for all terms concerning electrostatic discharges, eg ESD protection, ESD hazard
I/O	Input / Output
LCD	Liquid Crystal Display
LED	Light Emitting Diode
MMI	Man-Machine-Interface
OEM	Original Equipment Manufacturer
PCL	PC-programmable logic control
PE	Protective Earth
PHG	Hand-held programmer (Programmier Handgerät)
PLC	Programmable logic controller
ROPS4	Robot programming system for rho4
TCP/IP	Transmission Control Protocol / Internet Protocol
UPS	Uninterruptible Power Supply

Appendix

A.2 Index

Appendix

S

T

V

Appendix

Notes:

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

1070072367

Printed in Germany
DOK-RHO*4*-RHO4.0*SYSB-PR04-EN-P