Rexroth
Bosch Group

Rexroth Rho 4.1
System description

1070072185
Edition 07

Project planning

Title
Rexroth Rho 4.1
System description

Type of Documentation
Project planning

Document Typecode
DOK-RHO*4*-RHO4.1*SYSB-PR07-EN-P

Purpose of Documentation
The present manual informs about:
- the structures and functionalities
- as well as the programming of the rho4.1

Record of Revisions

Description	Release Date	Notes
DOK-RHO*4*-RHO4.1*SYSB-PR06-EN-P	10.2003	Valid from VO07
DOK-RHO*4*-RHO4.1*SYSB-PR07-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

Page

1 Safety Instructions . . . . . 1−1
1.1 Intended use . . . . . 1−1
1.2 Qualified personnel . . . . . 1−2
1.3 Safety markings on products . . . . . 1−3
1.4 Safety instructions in this manual . . . . . 1−4
1.5 Safety instructions for the described product . . . . . 1−5
1.6 Documentation, software release and trademarks . . . . . 1−7

2 Structure of the rho4.1 . . . . . 2−1
2.1 Description of the structure . . . . . 2−1
2.1.1 Outline structure . . . . . 2−2
2.1.2 rho4.1 block diagram . . . . . 2−3
2.1.3 Gateway . . . . . 2−4
2.2 Operation with I/O-Gateway . . . . . 2−6
2.2.1 rho4 without PCL field bus card . . . . . 2−6
2.2.2 Examples of B~IO M-CAN bus connections . . . . . 2−7
2.2.3 rho4.1 with PCL field bus card . . . . . 2−23
2.2.4 Example of a B~IO M-DP Profibus connection . . . . . 2−24
2.2.5 Example of a B~IO M-IBS bus connection . . . . . 2−25
2.2.6 Example of a B~IO M-CAN bus connection . . . . . 2−25
2.3 Fixed IP addresses and alias names . . . . . 2−27
2.3.1 IP-Addresses . . . . . 2−27
2.3.2 Fixed Alias names . . . . . 2−28
2.4 Small linking with switches . . . . . 2−29
2.4.1 Cabling . . . . . 2−29
2.4.2 Linking of two rho4 . . . . . 2−29
2.4.3 Smallest linking of up to 6 rho4.1 with 8-fold switch . . . . . 2−31
2.4.4 Small linking of several rho4.1 with 8-fold switch . . . . . 2−33
2.5 Telediagnostic, remote control . . . . . 2−37
2.5.1 Function . . . . . 2−37
2.5.2 Modem . . . . . 2−38
2.5.3 Permanent connecting . . . . . 2−45
2.5.4 Remote control with ROPS4 and virtual PHG . . . . . 2−54
2.5.5 Connection via modem . . . . . 2−58
2.6 MADAP Studio reporting system from version 1.2 on rho4.1 Pentium3-variants . . . . . 2−66

3 PCL . . . . . 3−1
3.1 Configuration parameter of WinPanel . . . . . 3−7
3.1.1 Start WinPanel through Winrho4 . . . . . 3−8
3.1.2 Example . . . . . 3−8

Contents

4 CAN-Bus-Peripherie . . . . . 4−1
4.1 CANopen-Interface . . . . . 4−1
4.1.1 Functions . . . . . 4−1
4.1.2 Machine parameters . . . . . 4−2
4.2 CANopen-encoder . . . . . 4−4
4.2.1 Functions . . . . . 4−4
4.2.2 Machine parameter . . . . . 4−5
4.3 CAN belts . . . . . 4−9
4.3.1 Functions . . . . . 4−9
4.3.2 Machine parameters . . . . . 4−9
4.4 Analog I/O with CAN B~IO modules . . . . . 4−13
4.4.1 Bus switching . . . . . 4−13
4.4.2 Analog I/O modules . . . . . 4−14
4.4.3 Machine parameter settings . . . . . 4−15
4.4.4 Analog I/O parameter . . . . . 4−21
4.4.5 Example . . . . . 4−23
4.4.6 BAPS program . . . . . 4−27
4.5 SR-CAN module . . . . . 4−28
4.6 Analog In-/Outputs of the SR-CAN module . . . . . 4−34
4.6.1 CAN-bus switching . . . . . 4−34
4.6.2 Measuring ranges and data formats . . . . . 4−34
4.6.3 Machine parameter settings . . . . . 4−35
4.7 Assignment of the CAN-Bus interfaces . . . . . 4−39
4.7.1 Examples . . . . . 4−41
4.8 CAN-ID assignments . . . . . 4−43
4.9 Servodyn-D-rho4 interface . . . . . 4−47
4.9.1 Control data rho4 −> Servodyn-D . . . . . 4−47
4.9.2 Status messages Servodyn-D −> rho4 . . . . . 4−48

5 SERCOS interface . . . . . 5−1
5.1 Data exchange via SERCOS bus . . . . . 5−1
5.1.1 Service channel . . . . . 5−1
5.1.2 Cyclic data exchange . . . . . 5−1
5.1.3 Data size of position set-values and actual values . . . . . 5−4
5.2 Machine parameter . . . . . 5−5
5.2.1 SERCOS specific control parameters . . . . . 5−5
5.2.2 SERCOS specific drive parameters . . . . . 5−7
5.3 Referencing . . . . . 5−9
5.3.1 RC-controlled referencing . . . . . 5−9
5.3.2 Drive-controlled referencing . . . . . 5−9
5.4 Status messages and warnings . . . . . 5−11
5.4.1 Status messages at the startup of the SERCOS interface 5−11
5.4.2 SERCOS specific state messages for running time . . . . . 5−12
5.4.3 Warnings . . . . . 5−12
5.5 Transmission of cyclic drive data to the PCL . . . . . 5−13
5.5.1 Definition of the cyclic axis telegrams (AT) . . . . . 5−13
5.5.2 Treating of the axis telegrams in the rho4 . . . . . 5−14
5.5.3 Transmission of the axis telegrams to the PCL . . . . . 5−15
5.5.4 Transmission of the rho4 system counter . . . . . 5−16
5.5.5 Representation of the ATs in the PCL . . . . . 5−17

Contents

5.6 Belt input via external encoder at EcoDrive/IndraDrive ... 5−19
5.6.1 General ... 5−19
5.6.2 Function ... 5−19
5.6.3 Parameter at the EcoDrive ... 5−19
5.6.4 Parameter at the IndraDrive ... 5−22
5.6.5 Machine parameter at rho4 ... 5−25
5.6.6 Restrictions ... 5−26

6 Software ... 6−1
6.1 Operating system variants ... 6−1
6.1.1 Installation and variants IPC300 ... 6−1
6.1.2 Installation and variants IPC 40.2 ... 6−1
6.2 Control Panels ... 6−2
6.2.1 rho4.1/IPC300 ... 6−2
6.2.2 rho4.1/IPC 40.2 ... 6−2
6.3 ROPS4/Online ... 6−3
6.4 GateWay ... 6−5
6.4.1 GateWay.ini file ... 6−7
6.4.2 Operation of the GateWay application ... 6−14
6.5 Start of Windows OEM applications ... 6−17
6.5.1 Initialization ... 6−17
6.5.2 Winrho4.ini ... 6−17
6.5.3 Example of a Winrho4.ini with the call of OEM applications ... 6−18

7 File management ... 7−1
7.1 File management function ... 7−1
7.2 Options for accessing file management ... 7−3
7.3 File attributes ... 7−6
7.4 Booting the rho4.1 ... 7−7
7.5 Support of hard disks > 1,5 GB (only rho4.1, rho4.1/IPC300) ... 7−8
7.6 Switcher ... 7−8

8 Scope of rho4.1 software ... 8−1
8.1 Hard disk folder structure ... 8−1
8.1.1 Specimen list of files ... 8−2
8.2 Software ... 8−17
8.2.1 Licences ... 8−17

A Appendix ... A−1
A.1 Abbreviations ... A−1
A.2 Index ... A−2

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.

Safety Instructions

1.2 Qualified personnel

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

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

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

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

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

Such personnel are

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

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

Safety Instructions

1.3 Safety markings on products

Warning of dangerous electrical voltage!

Warning of danger caused by batteries!

Electrostatically sensitive components!

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

Disconnect mains power before opening!

Lug for connecting PE conductor only!

Functional earthing or low-noise earth only!

Connection of shield conductor only

Safety Instructions

1.4 Safety instructions in this manual

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

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

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

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

★ This symbol is used if user activities are required.

Safety Instructions

1.5 Safety instructions for the described product

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

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

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

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

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

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

Safety Instructions

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

CAUTION
use only spare parts approved by Rexroth!

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

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

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

Safety Instructions

1.6 Documentation, software release and trademarks

Documentation

The present manual provides information on the structures and functionalities, as well as the programming of the rho4.1

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 072360	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 Structure of the rho4.1

2.1 Description of the structure

The following notes refer to point 2.1.2 Block structure of the rho4.1.

The exchange of data between the real-time kernel of the rho4.1 and the Windows user level is implemented via a Shared Memory TCP/IP connection. The individual groups of connections are described in the following text.

It is possible to communicate with the rho4.1 in Windows via both a DDE server and DLL libraries. Various library functions are available.

The OEM has four parallel channels for linking the library functions. The library functions are implemented as DLLs on the Windows side.

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

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

Coupling to a 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. Data communication is handled through Windows and a network card if the PC-programmable logic control is part of an external computer. The PCL is programmed using the WIN-SPS software. The connection between the PCL and the rho4.1 real-time kernel is also a shared memory TCP/IP connection.

Four serial channels are provided for the rho4.1 real-time kernel, one of which is occupied by the PHG2000.

The PHG2000 is the default operating device for the rho4.1. 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. Optional a SERCOS interface is available.

The rho4.1 real-time kernel incorporates a CAN interface to the digital drive amplifiers (Bosch Servodyn D, Servodyn GC).

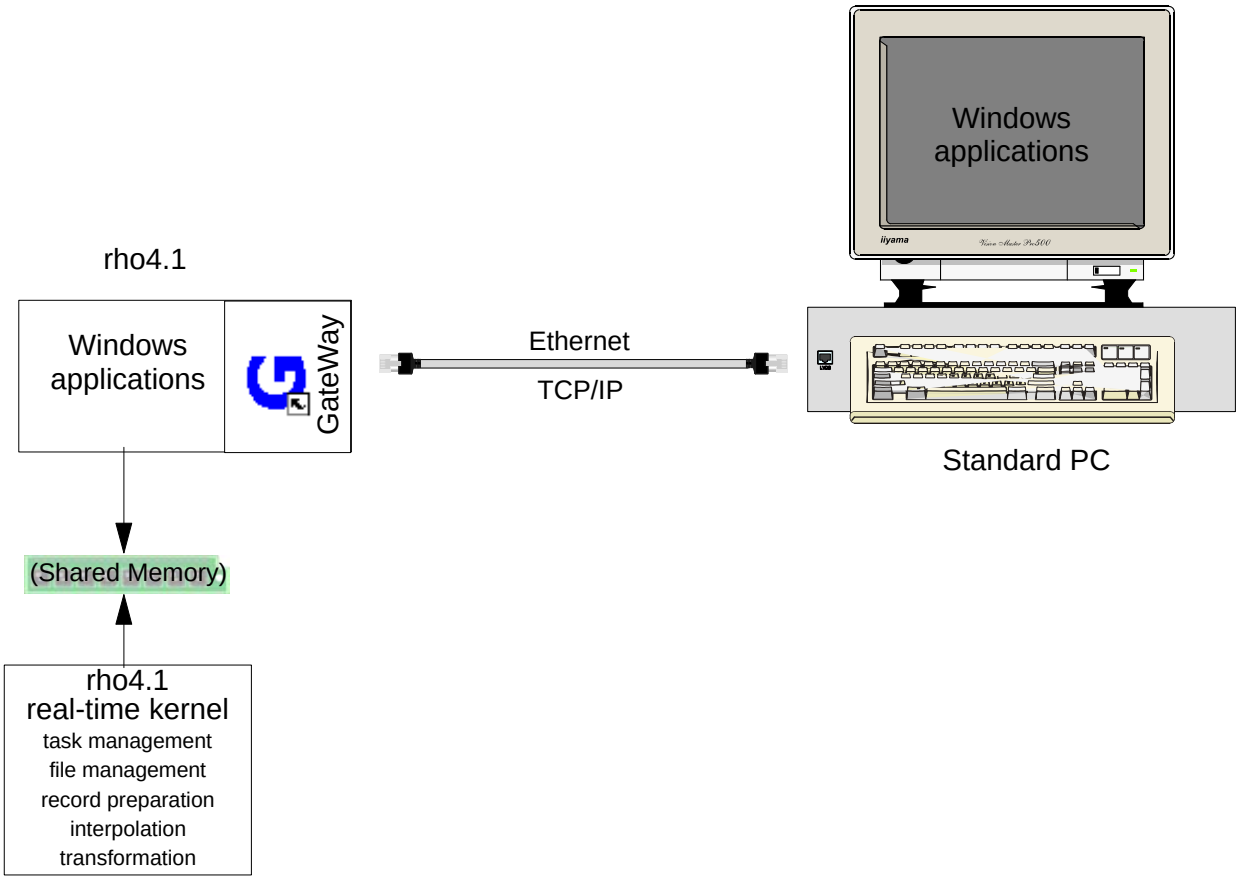
Structure of the rho4.1

A CAN interface is also used to tap digital I/Os from the real-time kernel. The PCL offers the capability of communicating by means of Profibus-DP if additional hardware is used.

ROPS4 can communicate with the real-time kernel both via a dynamic link library and a shared memory TCP/IP connection and serially. BAPS plus uses DDE Server4.

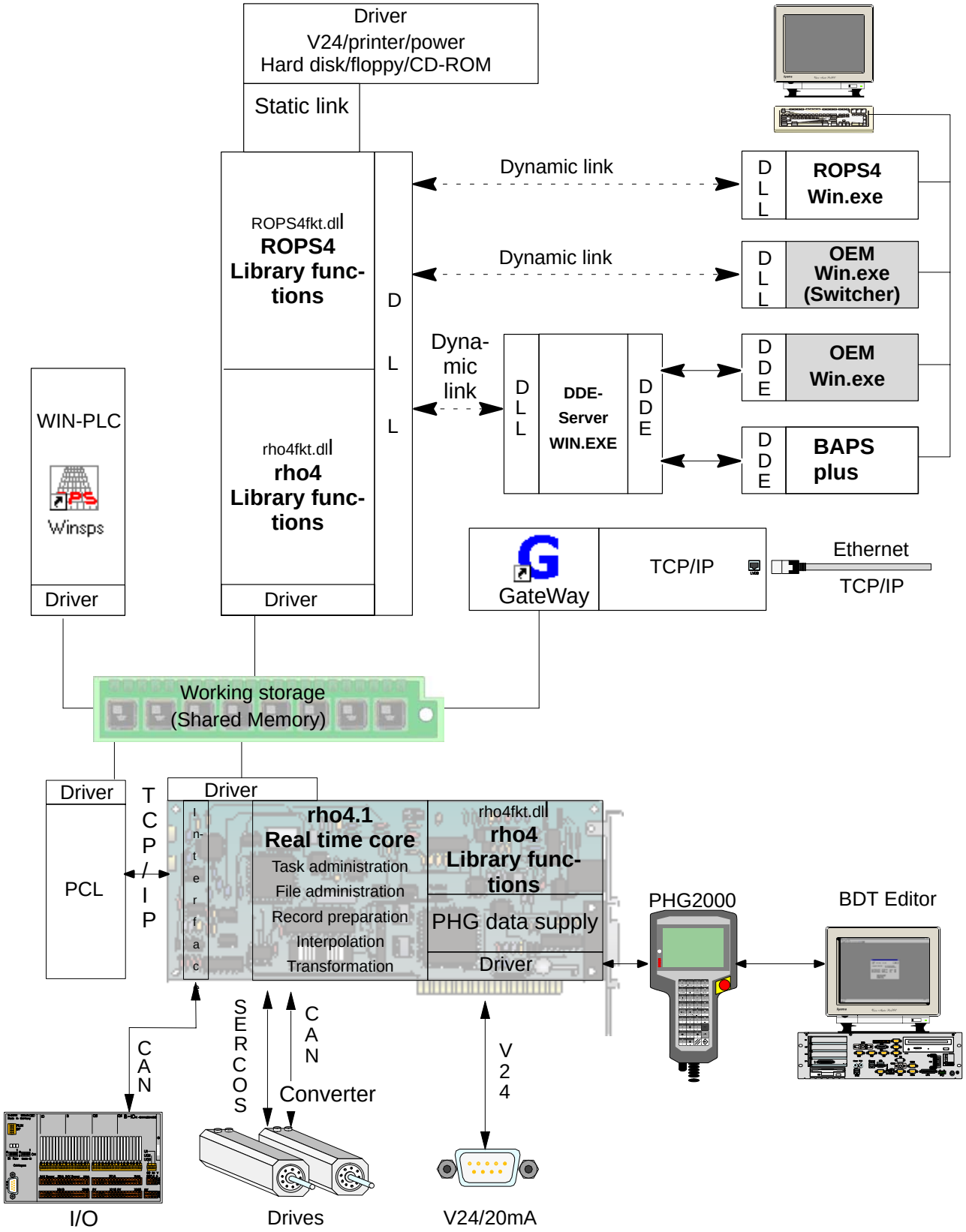
A gateway is supplied in the rho4.1's scope of supply. The gateway allows several TCP/IP connections to be established simultaneously, eg through a network card in a local network or to the WWW (World Wide Web).

2.1.1 Outline structure



Structure of the rho4.1

2.1.2 rho4.1 block diagram



Structure of the rho4.1

2.1.3 Gateway

The GateWay program is a Windows application for Windows 95 or Windows NT. It is used for TCP/IP communication between the rho4.1 and all external clients and servers.

☞ **See also DLL library manual.**

External applications are those that are not installed on the same hardware as the rho4.1.

A client is an application that makes use of services provided by a server.

Example 1

Typical services are the determination of the actual position of individual axes or reading errors that have occurred in the rho4.1.

The client requests the axis position from the rho4.1 server. The server determines the axis position and sends this to the client. For the purposes of this example, the server is running on the rho4.1 and the client is running on an external PC. These are the rho server 2 and the Windows PC 2 respectively in the following diagram.

The situation of client and server may also be reversed.

Example 2

A vision system determines image data and supplies this data as server services.

An ird program running in the control system functions as the client and request this data from the server. For the purposes of this example, the Windows PC 3 is the server and the rho client 1 is the client in the following diagram.

Use of GateWay

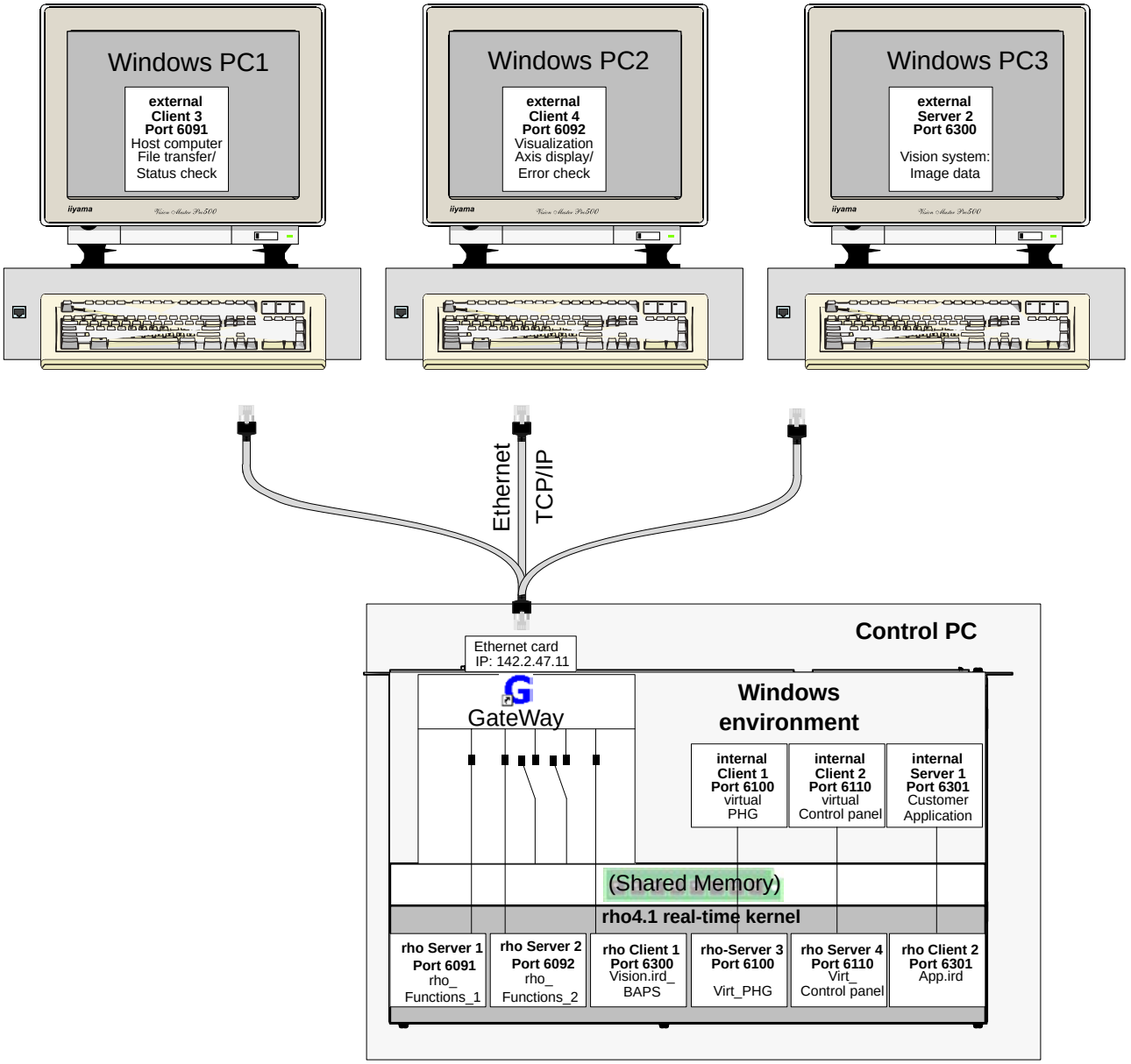
It is not possible to access the hardware components of one part from those of another part. The rho4.1 real-time kernel cannot access the Ethernet card directly. Thus the control unit has no physical access to the network. All connections are made through a shared-memory driver and the GateWay.

GateWay’s task is to establish the connection to an external client or server and to transfer data over this connection. These are Clients 3 and 4 and the external Server 2 in the following diagram.

A client or server running on the same hardware as the rho4.1, in other words an internal client, does not use the GateWay. These are Clients 1 and 2 and the internal Server 1 in the following diagram.

Structure of the rho4.1

GateWay's functioning



The rho4.1 hardware should be viewed as two parts.

One part consists of the PC's hardware components running the Windows 95 operating system, in which the rho4.1 is installed. The hardware components consist of

- the monitor
- the keyboard
- the hard disk
- the Ethernet card

The second part is the real-time kernel of the rho4.1. This consists of

- a part of the working memory (RAM)
- the rho4.1 card.

Structure of the rho4.1

2.2 Operation with I/O-Gateway

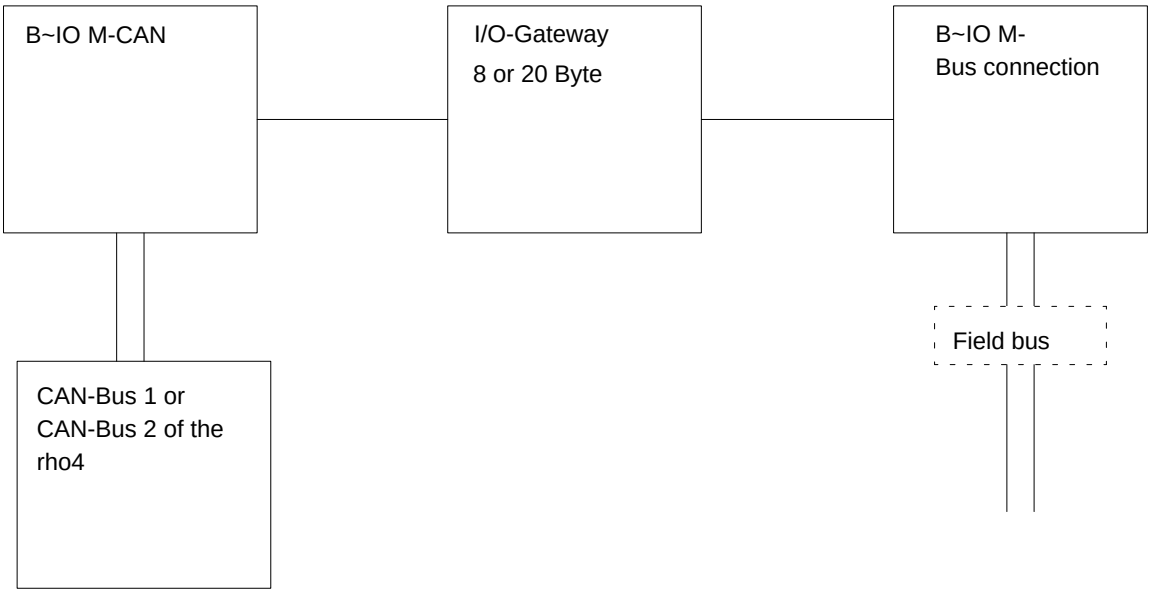
PLC controls of different busses can be connected to each other with the B~I/O gateway and the modular bus connections B~I/O M-DP or B~I/O M-IBS and B~I/O M-CAN.

A rho4 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.**

2.2.1 rho4 without PCL field bus card

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



Possible combinations:

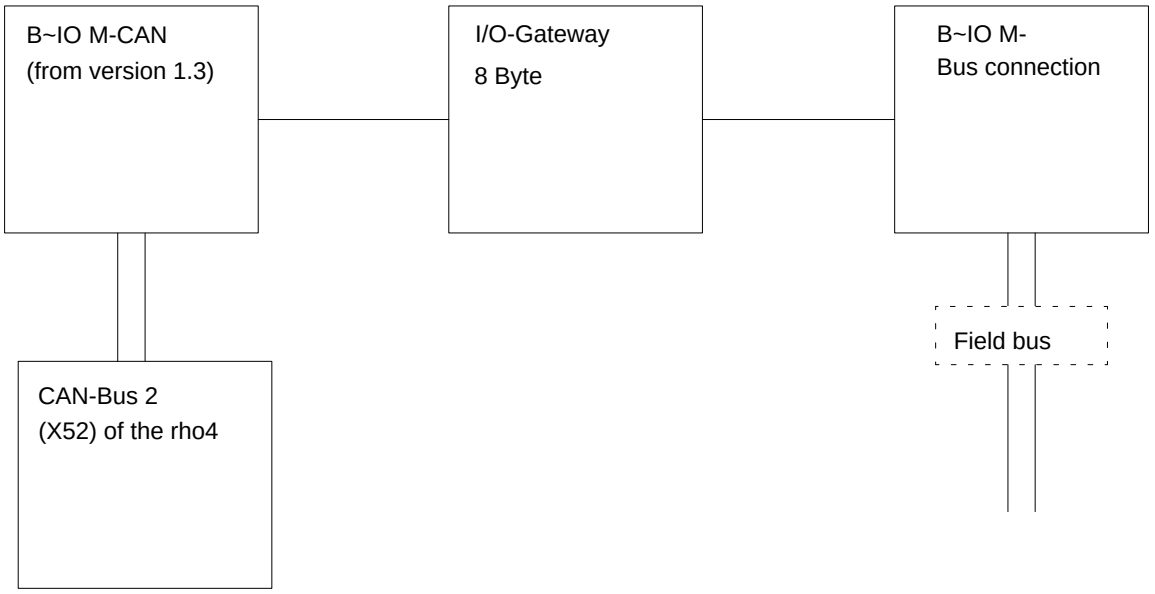
CAN-Bus of the rho4	**B~IO M-Bus connection**	**I/O-Gateway**	**B~IO M-Bus connection**	**Field bus**
Can-Bus 1 or 2	B~IO M-CAN	I/O-Gateway 8 Byte	B~IO M-DP	PROFIBUS-DP
Can-Bus 1 or 2	B~IO M-CAN	I/O-Gateway 8 Byte	B~IO M-IBS	InterBus-S
Can-Bus 1 or 2	B~IO M-CAN	I/O-Gateway 8 Byte	B~IO M-CAN	CANopen
Can-Bus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

Structure of the rho4.1

2.2.2 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 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.

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 8 Bytes.

Structure of the rho4.1

P30 I/O Configuration

Number of the input blocks

Display PHG:

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

#
```

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.**

Structure of the rho4.1

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

P31 Address areas of the CAN inputs

EA-type (0=digital, 1=analog) of the first block

Display PHG:

```
                    MP SET
P31 ADR.CAN-I
IOKind Block1:      0
#
```

Start address of the first block

Display PHG:

```
                    MP SET
P31 ADR.CAN-I
1st Adr.Block1:     208
#
```

Length of the first block

Display PHG:

```
                    MP SET
P31 ADR.CAN-I
Length Block1:      8
#
```

Identifier of the first block

Structure of the rho4.1

Display PHG:

MP SET
P31 ADR.CAN-I
Ident. Block1: 386
#

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

Bus number of the first block

Display PHG:

MP SET
P31 ADR.CAN-I
Bus-No. Block1: 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.CANInpdent.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.

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

Display PHG:

MP SET
P32 ADR.CAN-O
IOKind Block1: 0
#

Start address of the first block

Structure of the rho4.1

Display PHG:

```
                    MP SET
P32 ADR.CAN-O
1st Adr. Block1:    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:      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
#
```

Structure of the rho4.1

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

Setting S1, Bit rate

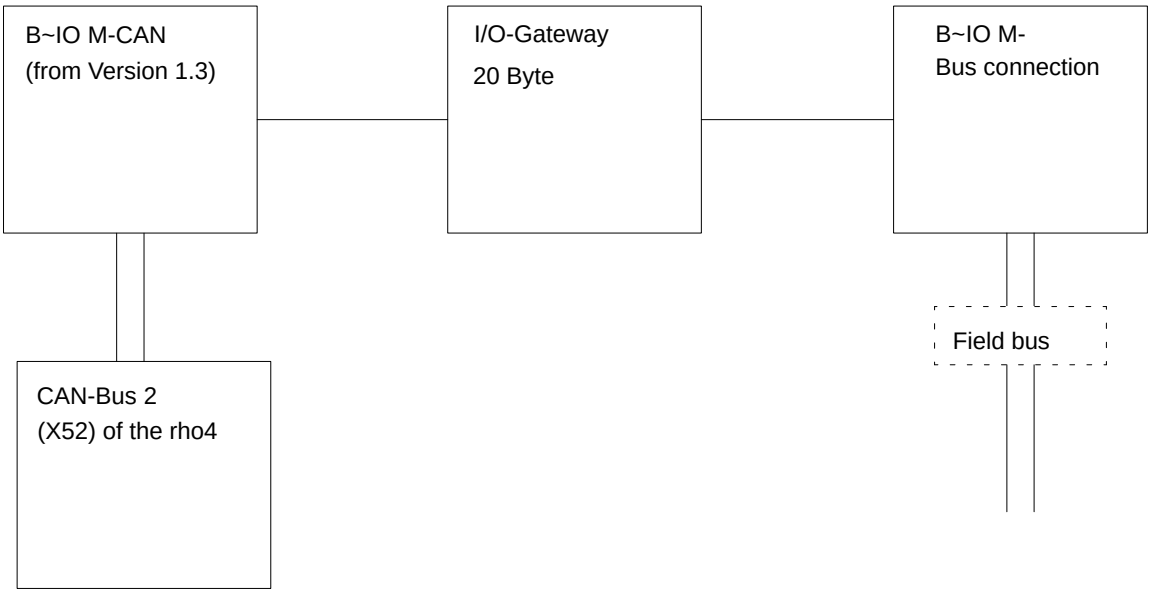
Switch 8	**Switch 7**	**Switch 6**	**Switch 5**	**Switch 4**	**Switch 3**	**Switch 2**	**Switch 1**
on	off	off	off	off	on	on	on
Setting CANrho	reserved				Baudrate	= 1 MBaud	

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
reserviert						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.

Structure of the rho4.1

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
#
```

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
#
```

Structure of the rho4.1

☞ **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

EA-type (0=digital, 1=analog) of the first block

Display PHG:

```
                    MP SET
P31 ADR.CAN-I
IOKind Block1:      0
#
```

Start address of the first block

Display PHG:

```
                    MP SET
P31 ADR.CAN-I
1st Adr.Block1:     208
#
```

Length of the first block

Display PHG:

```
                    MP SET
P31 ADR.CAN-I
Length Block1:      8
#
```

Identifier of the first block

Structure of the rho4.1

Display PHG:

```
                    MP SET
P31 ADR.CAN-I
Ident. Block1:      385
#
```

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

Bus number of the first block

Display PHG:

```
                    MP SET
P31 ADR.CAN-I
Bus-No. Block1:     2
#
```

EA-type (0=digital, 1=analog) of the second block

Display PHG:

```
                    MP SET
P31 ADR.CAN-I
IOKind Block2:      0
#
```

Start address of the second block

Display PHG:

```
                    MP SET
P31 ADR.CAN-I
1st Adr.Block2:     216
#
```

Length of the second block

Structure of the rho4.1

Display PHG:

```
                    MP SET
P31 ADR.CAN-I
Length Block2:      8
#
```

Identifier of the second block

Display PHG:

```
                    MP SET
P31 ADR.CAN-I
Ident. Block2:      641
#
```

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

Bus number of the second block

Display PHG:

```
                    MP SET
P31 ADR.CAN-I
Bus-No. Block2:     2
#
```

EA-type (0=digital, 1=analog) of the third block

Display PHG:

```
                    MP SET
P31 ADR.CAN-I
IOKind Block3:      0
#
```

Start address of the third block

Structure of the rho4.1

Display PHG:

```
                    MP SET
P31 ADR.CAN-I
1st Adr.Block3:     224
#
```

Length of the third block

Display PHG:

```
                    MP SET
P31 ADR.CAN-I
Length Block3:      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. Block3:      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. Block3:     2
#
```

Structure of the rho4.1

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.CANInpdent.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.

EA-type (0=digital, 1=analog) of the first block

Display PHG:

```
                    MP SET
P32 ADR.CAN-O
IOKind Block1:      0
#
```

Start address of the first block

Display PHG:

```
                    MP SET
P32 ADR.CAN-O
1st Adr. Block1:    208
#
```

Structure of the rho4.1

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-Nr. Block 1:     2
#
```

EA-type (0=digital, 1=analog) of the second block

Display PHG:

```
                    MP SET
P32 ADR.CAN-O
EA-type Block2:      0
#
```

Start address of the second block

Structure of the rho4.1

Display PHG:

```
                    MP SET
P32 ADR.CAN-O
1st Adr.Block2:     216
#
```

Length of the second block

Display PHG:

```
                    MP SET
P32 ADR.CAN-O
Length Block2:      8
#
```

Identifier of the second block

Display PHG:

```
                    MP SET
P32 ADR.CAN-O
Ident. Block2:      769
#
```

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

Bus number of the second block

Display PHG:

```
                    MP SET
P32 ADR.CAN-O
Bus-Nr. Block2:     2
#
```

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

Structure of the rho4.1

Display PHG:

```
                    MP SET
P32 ADR.CAN-O
IOKind Block3:      0
#
```

Start address of the third block

Display PHG:

```
                    MP SET
P32 ADR.CAN-O
1st Adr.Block3:     224
#
```

Length of the third block

Display PHG:

```
                    MP SET
P32 ADR.CAN-O
Length Block3:      4
#
```

Identifier of the third block

Display PHG:

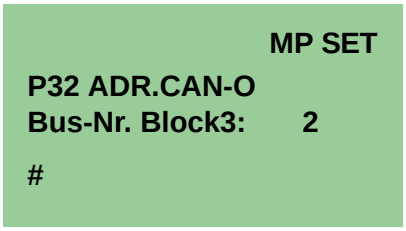
```
                    MP SET
P32 ADR.CAN-O
Ident. Block3:      514
#
```

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

Bus number of the third block

Structure of the rho4.1

Display PHG:



Machine parameter	**Entry Machine parameter converter**
P32 start address Block 1	P32.CANOutStAdr.Block1=208
P32 start address Block 2	P32.CANOutputStAdr.Block2=216
P32 start address Block 3	P32.CANOutputStAdr.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	

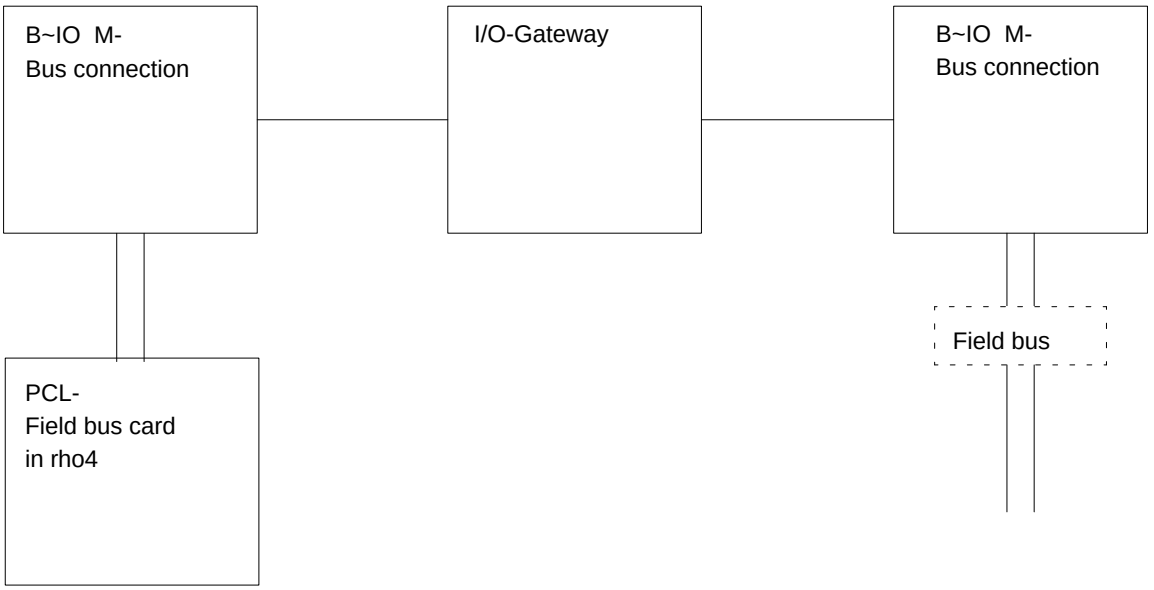
Setting S2, Node-ID = 1

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 rho4.1

2.2.3 rho4.1 with PCL field bus card

In this variant, a rho4 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

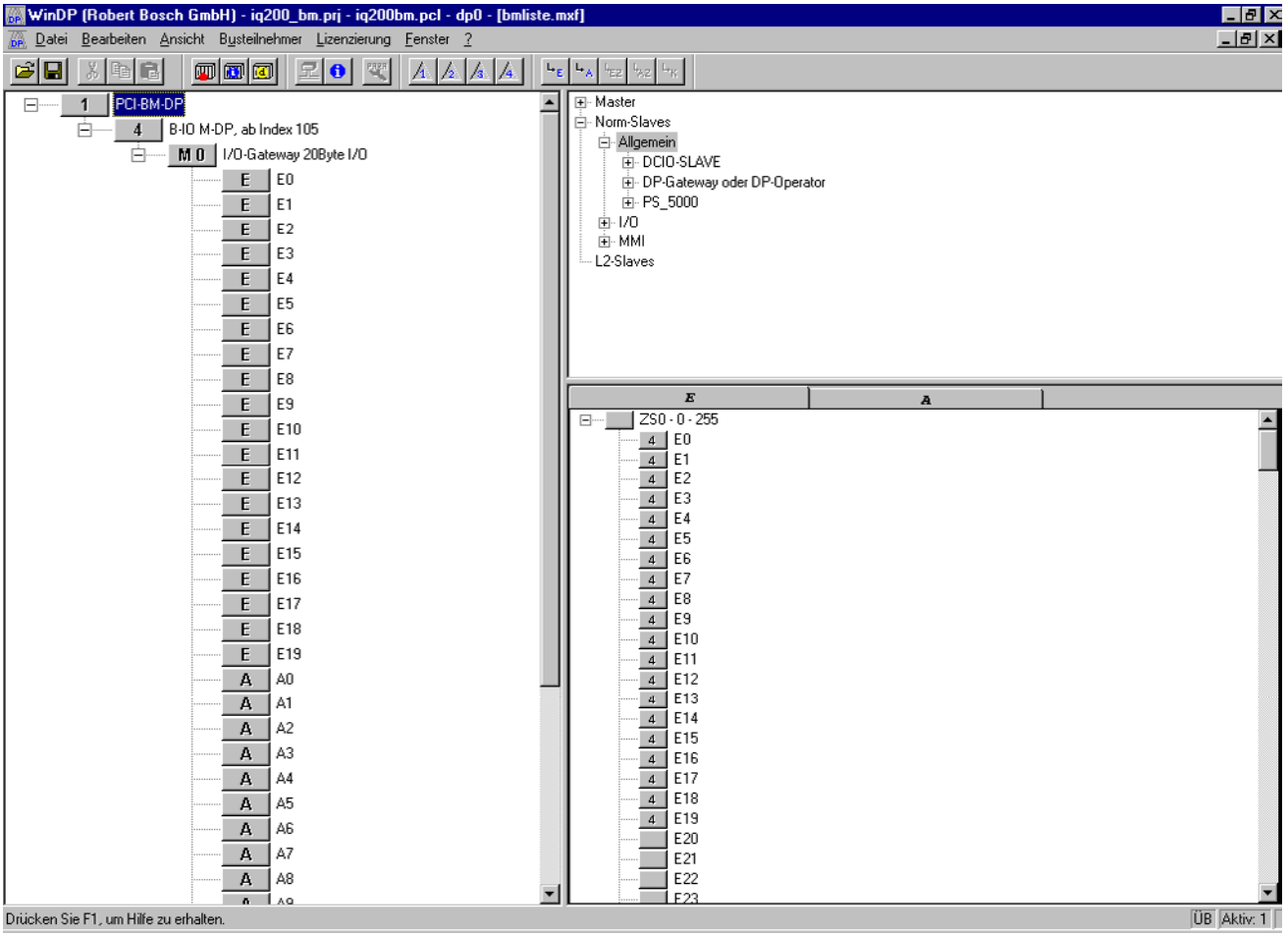
☞ **The B~IO M-CAN components can be only used from Firmware Version 1.3 for I/O-gateway.**

Structure of the rho4.1

2.2.4 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.**

Structure of the rho4.1

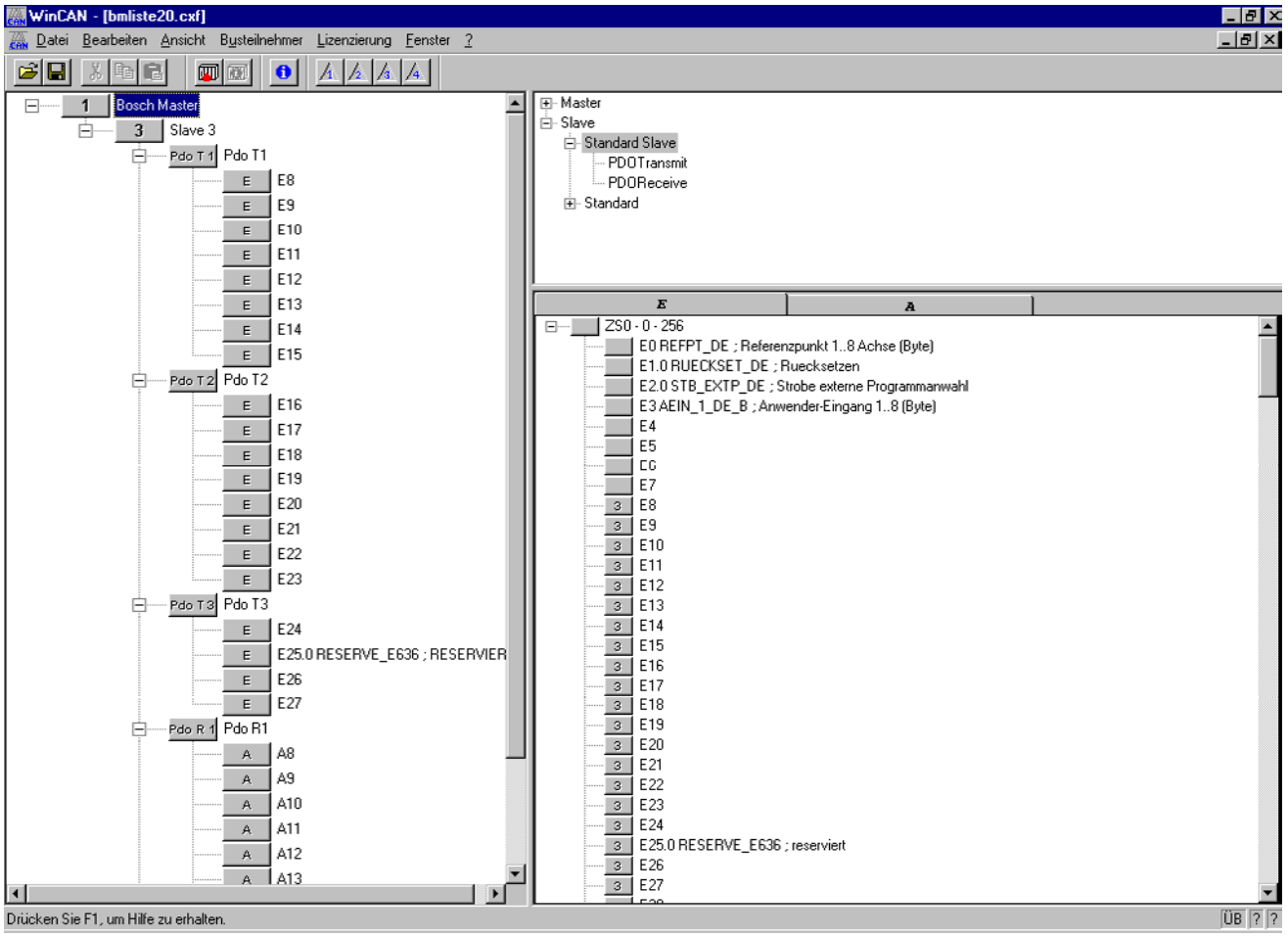
2.2.5 Example of a B~IO M-IBS bus connection

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.

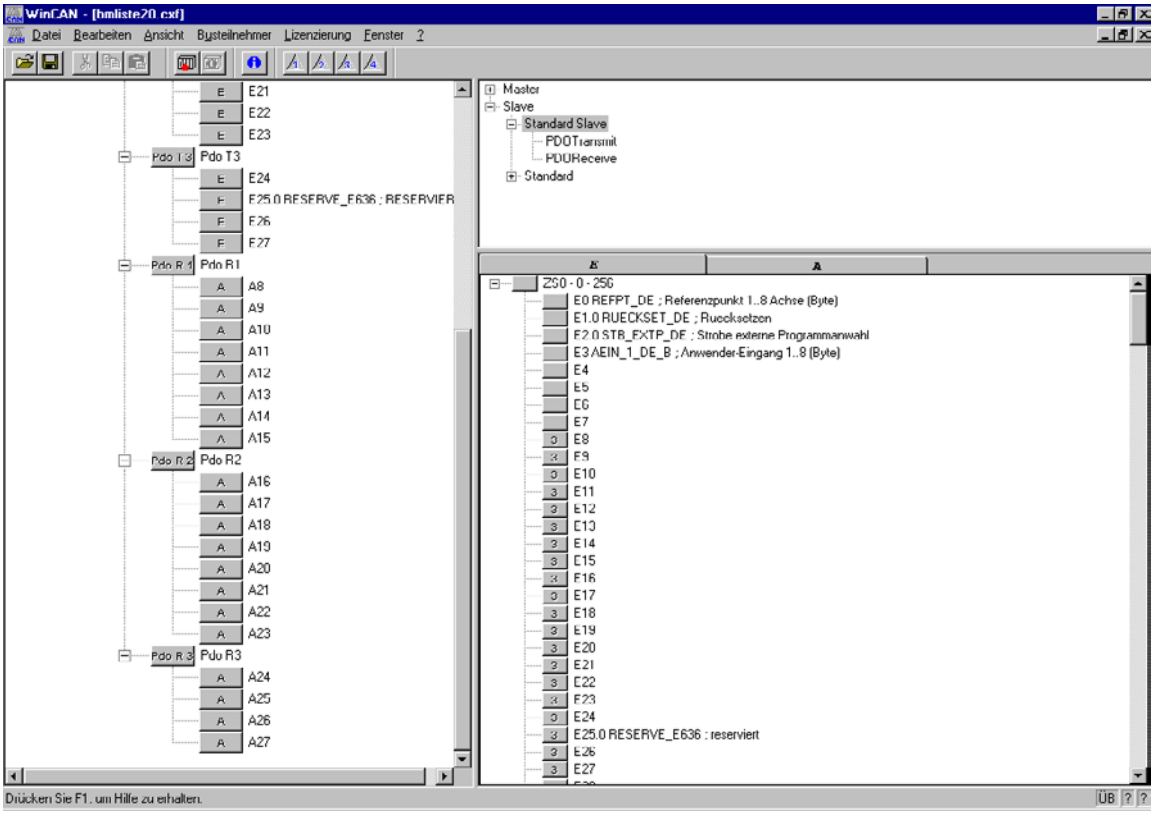
2.2.6 Example of a B~IO M-CAN bus connection

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 rho4.1



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.

Structure of the rho4.1

2.3 Fixed IP addresses and alias names

2.3.1 IP-Addresses

☞ **The IP-addresses may only be changed in conflict case after consulting Bosch.**

The TCP/IP-address of the physical network card is preset on “192.168.4.1”. If the rho4.1 is to be integrated into an existing network, a new address which is to be defined by the administrator must be entered under [Control Panel] [Network] [Protocols] [TCP/IP].

192.0.1.1	Name	TCP/IP −> Bosch Win/VVxWorks Shared memory network
	Setting under Windows NT	The entry in [Start][Settings][Control Panel][Network][Protocols][TCP/IP −> Bosch Win/VVxWorks Shared memory Network] [Properties][IP Address] is 192.0.1.1
	Address	Windows-Address from rho4
	Explanation	This IP address is required when the rho4 is client, e.g in the channels Client PLC and PLC_Interface, or e.g. for a call of rTClientCon in BAPS
192.0.1.2	Name	rho4
	Setting under Windows	In the file ‘hosts’ there is 192.0.1.2 rho4
	Address	Address fo the rho4 from Windows
	Explanation	Under this IP address, the TCP/IP functions of the rho4 are addressed, e.g all rho4-library functions. It is also required for the communication from Winrho4 to rho4, e.g. loading of the machine parameters and the user memory for the start of the rho4.1 The port number is fixed which function is addressed
127.0.0.1	Name	PCL
	Setting under Windows	−−−
	Address	Address of the PCL from the rho4
	Explanation	Under this IP address the PCL is addressed when rho4.1 and PCL run jointly on the rho4.1 hardware
127.0.0.1	Name	General Windows application
	Setting under Windows	In the file ‘hosts’ there is 127.0.0.1 localhost
	Address	Address of the general Windows applications under each other when they jointly run on the rho4.1 hardware
	Explanation	‘localhost’ can always be used for communication between general Windows applications when they jointly run on the rho4.1 hardware

Structure of the rho4.1

2.3.2 Fixed Alias names

The alias name ‘rho4’ is reserved and may only be used for the IP address of the rho4.1 from Windows. In the file ‘hosts’, the Alias name for ’rho4’ may in no case be modified or removed. The file ’hosts’ must contain the entry ’192.0.1.2 rho4’.

The Alias name ‘rho4’ may not be used with any other meaning.

Structure of the rho4.1

2.4 Small linking with switches

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

2.4.1 Cabling

For the direct connection from a rho4.1 to a second rho4.1 (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

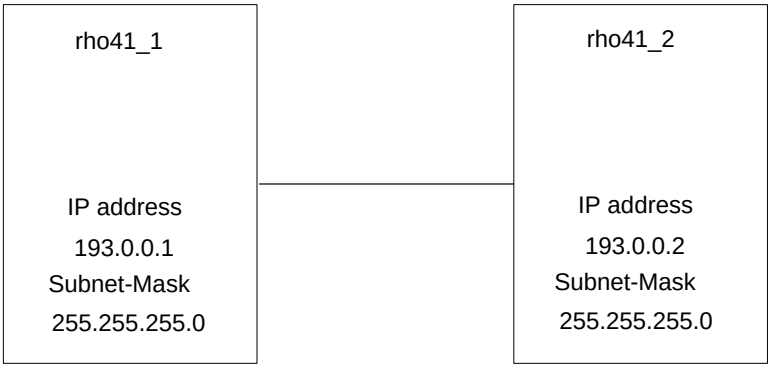
For the connection from switch to the rho4.1, it is possible to use e.g. the following cables:

- 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

2.4.2 Linking of two rho4

Connection diagram

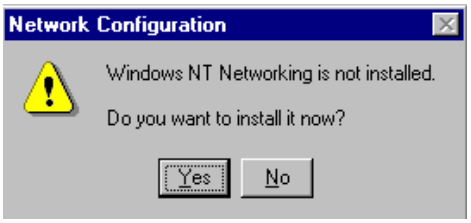
To be able to link exactly 2 rho4.1, they must be connected with a crossed Ethernet cable 10Base-T.



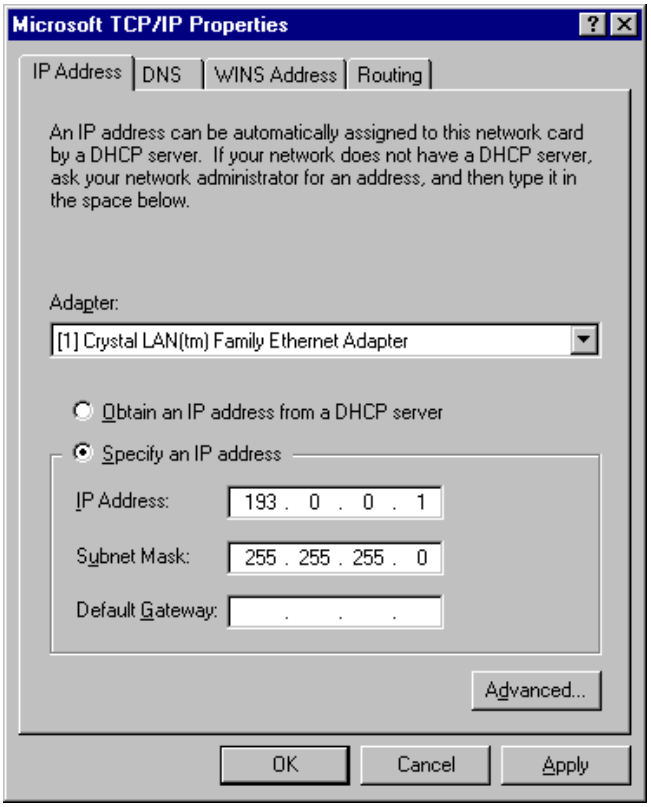
Structure of the rho4.1

IP address and subnet masks are to be entered under [Start] [Settings] [Control Panel] [Network] [Protocols][TCP/IP −> Crystal LAN(tm) Family Ethernet Adapter] [Properties] [IP Address] [Specify an IP address].

Only with Windows NT:
After selecting the item “Network” in the Control Panel, the following window will be displayed:



☞ **Do not enter “Yes”. Close this window by pressing “No”.**



Structure of the rho4.1

File ‘hosts’

The file ‘hosts’ which can be loaded in both rho4.1 is as follows:

127.0.0.1	localhost
192.0.1.2	rho4
193.0.0.1	rho41_1
193.0.0.2	rho41_2

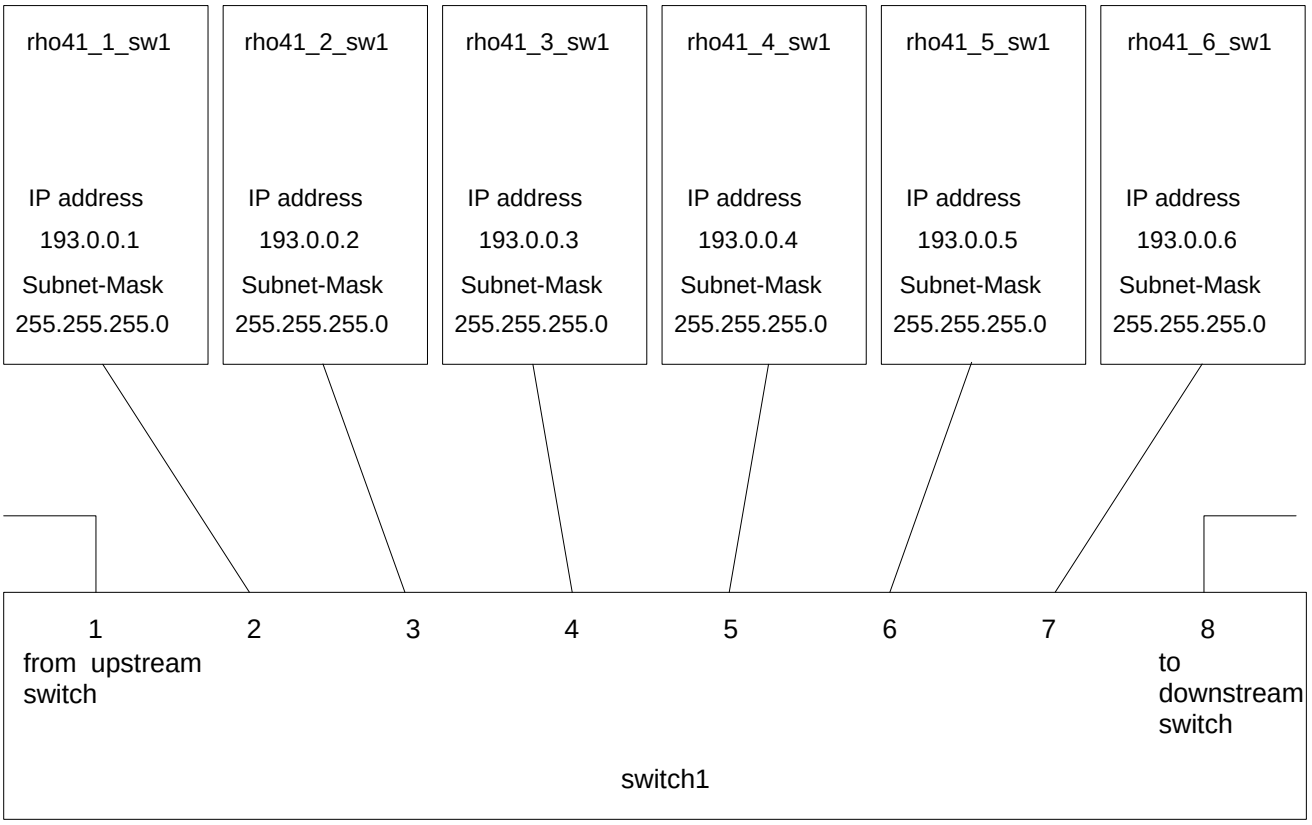
2.4.3 Smallest linking of up to 6 rho4.1 with 8-fold switch

With a 8-fold switch, up to 8 rho4.1 can be linked. To be able to extend the smallest network to a small network, 2 switches connections are kept free.

8-, 16- or 24-fold switches are commercially available.

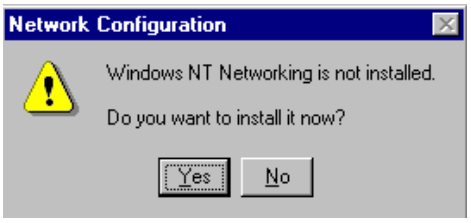
Connection shema

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

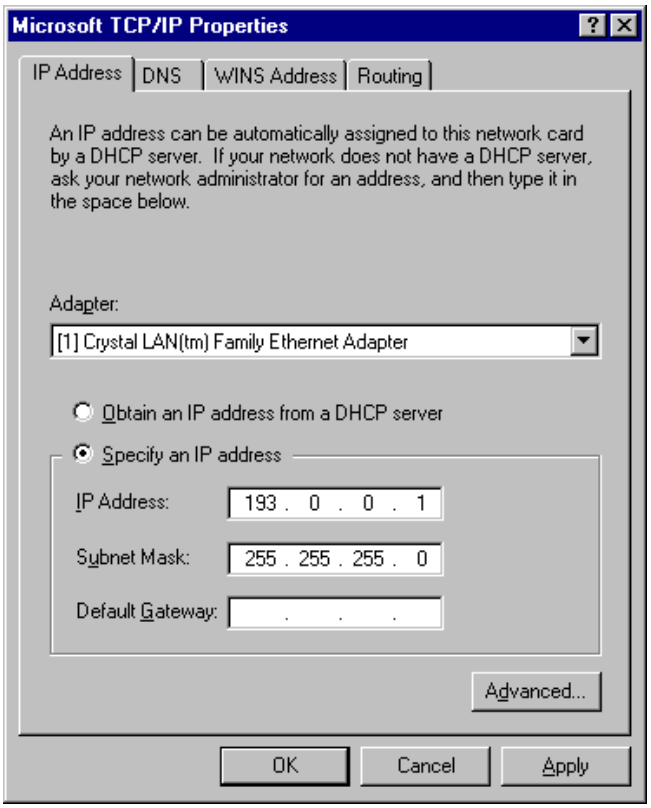


The IP address and the Subnet mask are to be entered under [Start] [Settings] [Control Panel] [Network] [Protocols] [TCP/IP −> Crystal LAN(tm) Family Ethernet Adapter] [Properties] [IP Address] [Specify an IP address].

Only with Windows NT:
After selecting the item “Network” in the Control Panel, the following window will be displayed:



☞ **Do not enter “Yes”. Close this window by pressing “No”.**



Structure of the rho4.1

file 'hosts'

The file 'hosts' that can be loaded in all rho4.1 in this smallest network, is as follows:

127.0.0.1	localhost
192.0.1.2	rho4
193.0.0.1	rho41_1_sw1
193.0.0.2	rho41_2_sw1
193.0.0.3	rho41_3_sw1
193.0.0.4	rho41_4_sw1
193.0.0.5	rho41_5_sw1
193.0.0.6	rho41_6_sw1

2.4.4 Small linking of several rho4.1 with 8-fold switch

Several rho4.1 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 rho4.1 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.

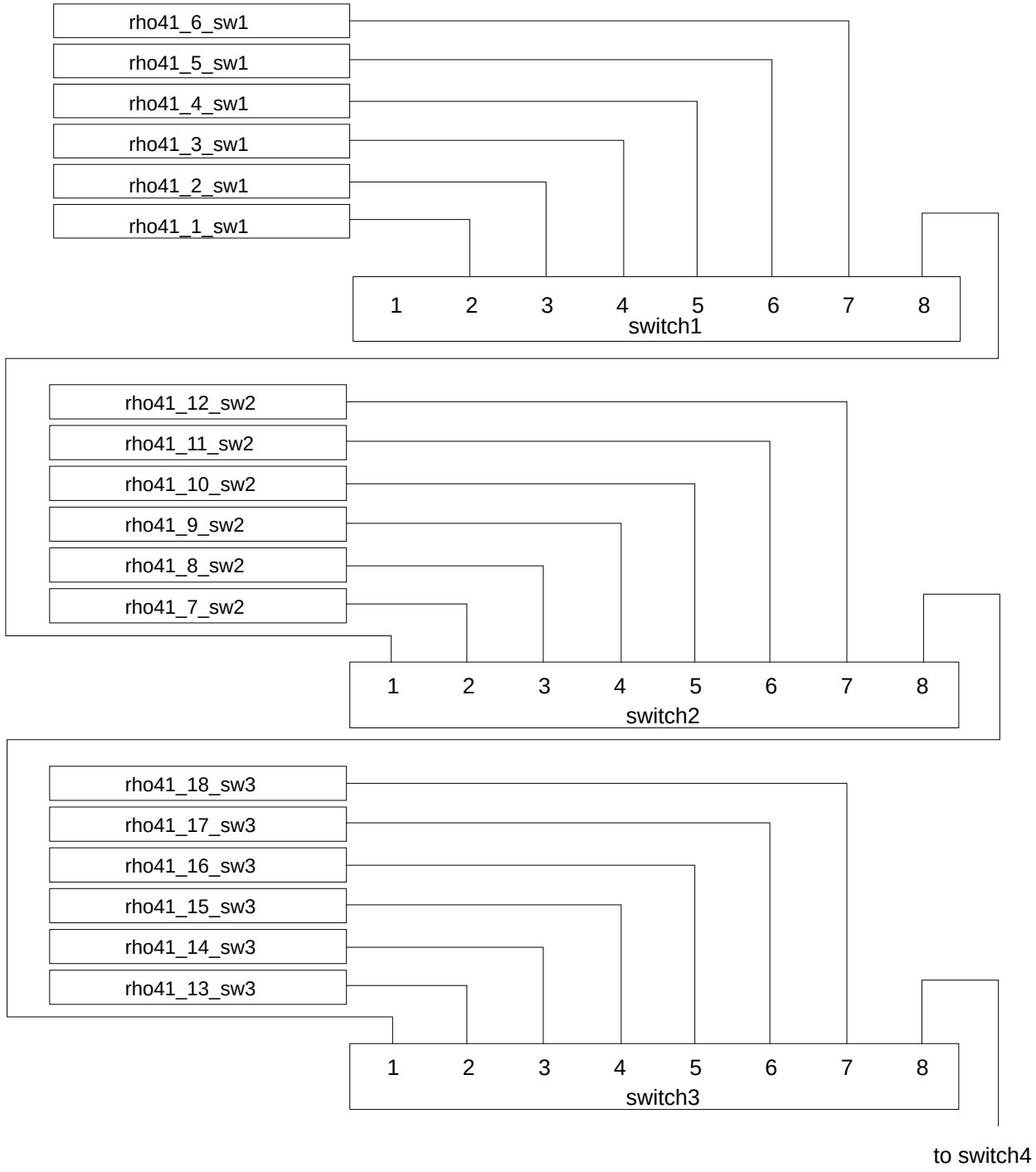
Connection diagram

☞ **This description assumes a rho4.1 under Win95.**

For the connection of the rho4.1 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.**

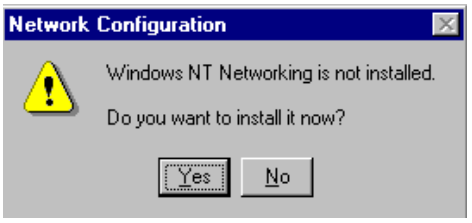
Structure of the rho4.1



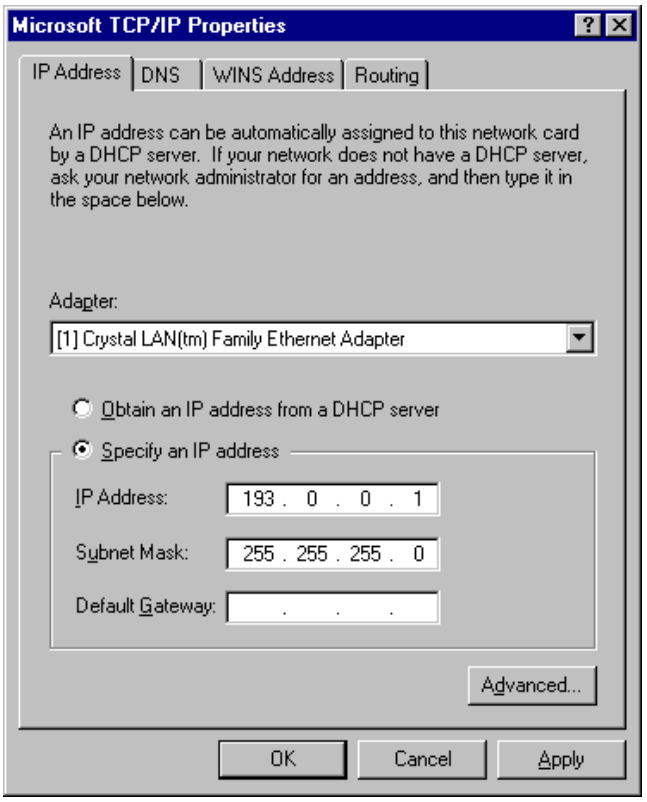
Structure of the rho4.1

IP address und Subnet-Mask are to be entered under [Start] [Settings]Control Panel] [Network] [Protocols] [TCP/IP −> Crystal LAN(tm) Family Ethernet Adapter] [Properties] [IP Address] [Specify an IP address].

Only with Windows NT:
After selecting the item “Network” in the Control Panel, the following window will be displayed:



☞ **Do not enter “Yes”. Close this window by pressing “No”.**



Structure of the rho4.1

File ’hosts’

The file ‘hosts’ that can be loaded in all rho4.1 in this small network is as follows:

127.0.0.1	localhost
192.0.1.2	rho4
193.0.0.1	rho41_1_sw1
193.0.0.2	rho41_2_sw1
193.0.0.3	rho41_3_sw1
193.0.0.4	rho41_4_sw1
193.0.0.5	rho41_5_sw1
193.0.0.6	rho41_6_sw1
193.0.0.7	rho41_7_sw2
193.0.0.8	rho41_8_sw2
193.0.0.9	rho41_9_sw2
193.0.0.10	rho41_10_sw2
193.0.0.11	rho41_11_sw2
193.0.0.12	rho41_12_sw2
193.0.0.13	rho41_13_sw3
193.0.0.14	rho41_14_sw3
193.0.0.15	rho41_15_sw3
193.0.0.16	rho41_16_sw3
193.0.0.17	rho41_17_sw3
193.0.0.18	rho41_18_sw3

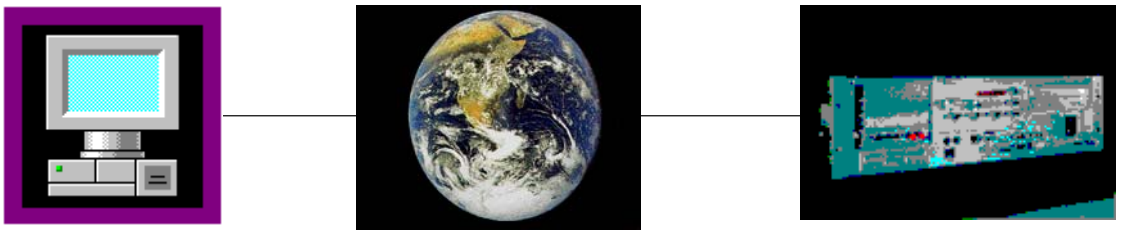
Structure of the rho4.1

2.5 Telediagnostic, remote control

2.5.1 Function

☞ **The description only applies to a rho4.1 from SW-Version VO02B. From Version VO04H the ReachOut software is not longer preinstalled on the rho4.1. The customer can obtain the software or a licence by the Bosch Rexroth AG.**

The rho4.1 has in its basic scope an Ethernet connection, via which it can be linked worldwide, enabling telediagnostic and remote control.
For the remote control, ReachOut of the Stac company is available. ReachOut is already installed from SW-Version VO02B to VO03G on the rho4.1, it must be installed moreover on the local PC used for remote control.



Local PC

Remote control and telediagnostic

TCP/IP

Permanent linking or connection via a modem

rho4.1

The innovative movement control of Bosch

DANGER
Machines movements without supervision and operated through remote control can lead to person injuries and material damages.

The user must observe the relevant safety regulations and the valid rules for prevention of accidents, so that no danger should arise. For the correct use, only the user is responsible. Observe the following procedure.

Procedure

★ Keep telephone contact with a trained service technician, who is at the machine, during the whole session.

Structure of the rho4.1

★ Comment important operation steps and have them confirmed so that you can be informed of the state of the machine at every time also via the telephone.

★ Carry out all operations only after previous agreement.

★ Carry out all operating steps slowly and through-out. Remember that the remote control runs in general via a DFÜ network and therefore can only take place with a slight delay at the machine.

ReachOut

ReachOut can communicate via the network as well as via a modem. The description is based on the ReachOut Release 8.30.

★ For the installation of ReachOut, close all running applications on the local PC.

★ Install ReachOut including WinSock 2.0.

★ New start of PC.

2.5.2 Modem

The installation of a modem is described on the basis of the example of the AcerModem 56000bps, Model No. AME-TG00. The installation indications of the modem manufacturer must be observed.

Installation on the rho4.1

★ Shut-down and switch off rho4.1. Wait until the uninterruptible power supply (UPS) has switched off.

★ Connect modem to a serial interface, e.g COM1. Establish all further connections such as connection from modem to telephone socket and network connection, according to indications of the modem manufacturer.

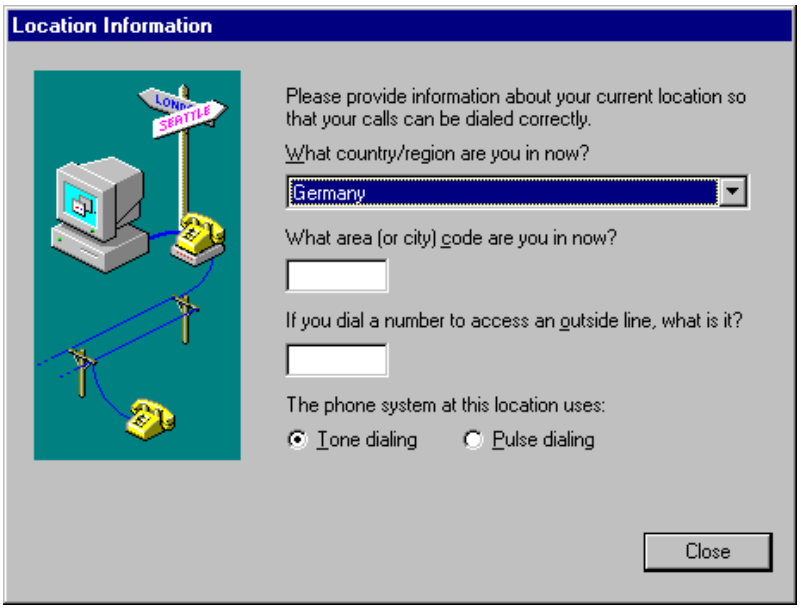
★ Switch on modem.

★ Switch on rho4.1. During the start, Windows recognizes automatically the modem newly connected and displays for a few seconds a dialog field showing the new hardware and the designation of the modem. Windows begins automatically with the installation.

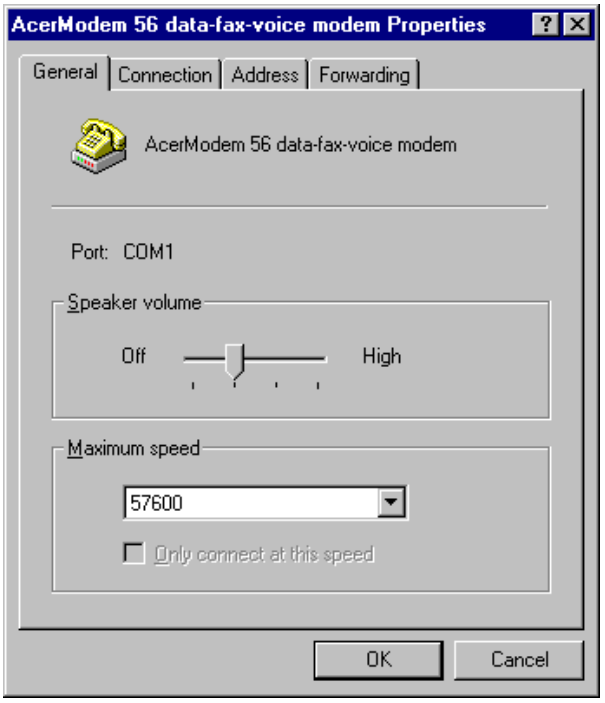
★ On the request of Windows, insert the driver disk or the driver CD-ROM of the modem and confirm with OK. Windows copies the drivers.

Structure of the rho4.1

After the installation is completed, the following or analog settings are to be performed under [Start][Settings]Control Panel[Modems]:

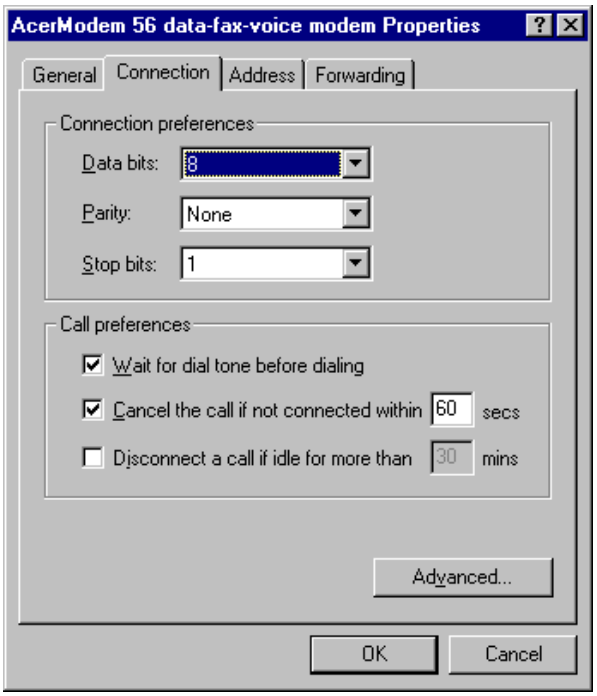


Structure of the rho4.1

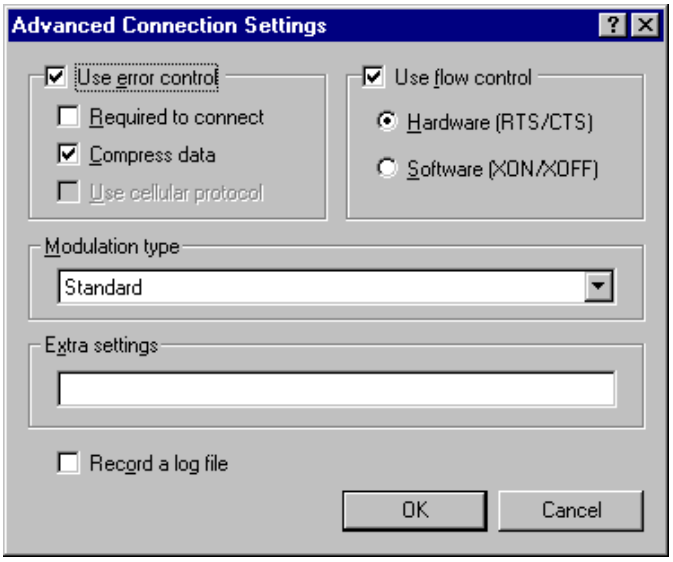


The maximum speed is adjusted on 57600 bauds since the real time part of the rho4.1 interrupts transfers from Windows/BIOS to UART.

The modem adjusted here allows only a maximum baud rate of 57 kBauds. This means no speed loss in the transfer. The connection to the COM1 ist arbitary. Another free COM interface can also be chosen (COM2 or COM3).



Structure of the rho4.1



Structure of the rho4.1

Installation on local PC

The process describes the installation of a modem on the local PC. Local PC means the PC, from which the remote control or telediagnostic is carried out.

★ Shut down and switch off local PC.

★ Connect modem to a serial interface, e.g COM1. Establish all further connections such as connection from modem to telephone socket and network connection, according to indications of the modem manufacturer.

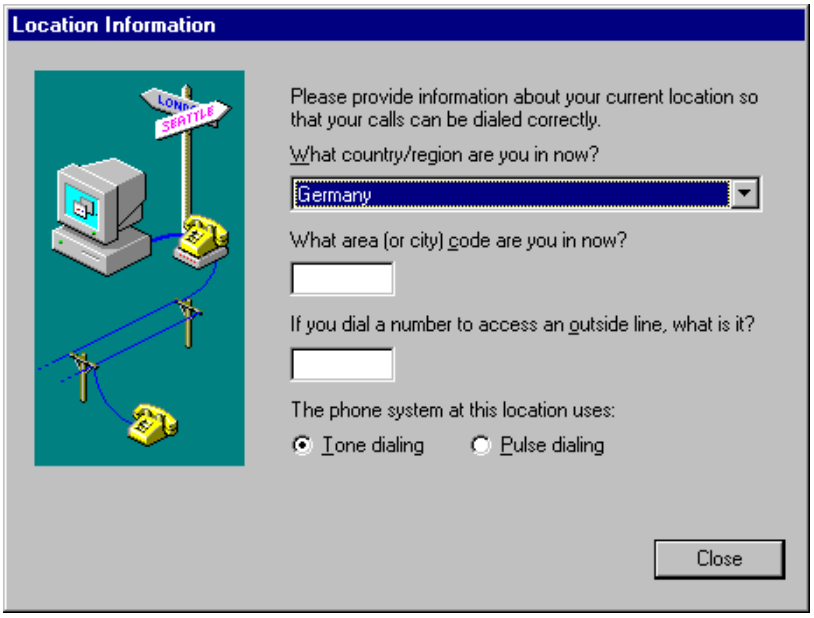
★ Switch on modem.

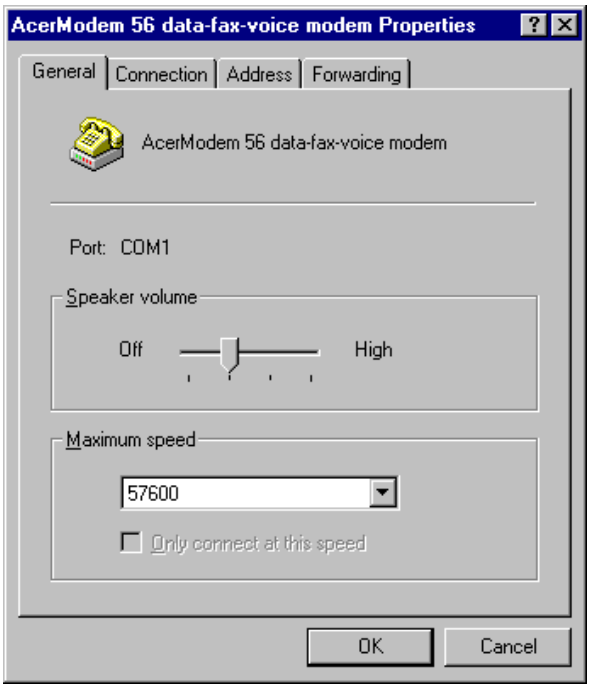
★ Switch on local PC. During the start, Windows recognizes automatically the modem newly connected and displays for a few seconds a dialog field showing the new hardware and the designation of the modem. Windows begins automatically with the installation.

★ On the request of Windows, insert the driver disk or the driver CD-ROM of the modem and confirm with OK. Windows copies the drivers.

After the installation is completed, the following or analog settings are to be performed under [Start][Settings]Control Panel[Modems]:

Structure of the rho4.1

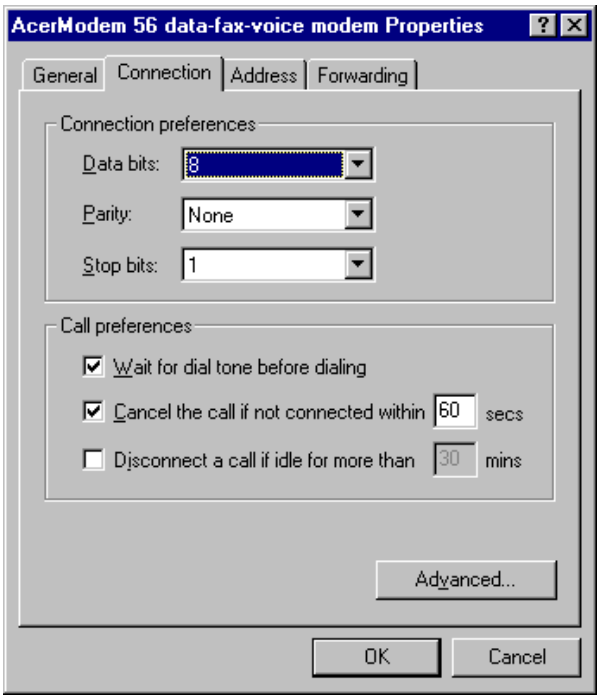




The maximum speed is adjusted on 57600 bauds since in this case the local PC allows no higher baud rate from the UART power.

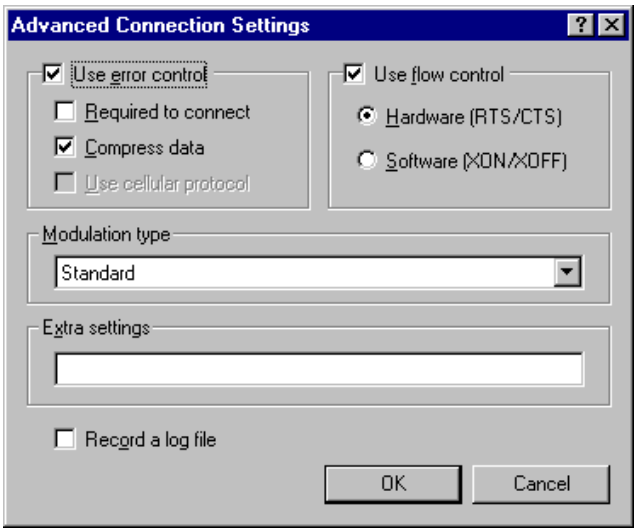
Structure of the rho4.1

The modem adjusted here allows only a maximum baud rate of 57 kBauds. This means no speed loss in the transfer. The connection to the COM1 ist arbitary. Another serial interface can also be chosen.



Under the call settings, it is to be taken into account:

- If it is to be dialed from an internal telephone network, the tick must be deactivated before ’Wait for dial tone before dialing’.
- If it is to be dialed from a direct telephone, the tick must be activated before ‘Wait for dial tone before dialing’.



Structure of the rho4.1

2.5.3 Permanent connecting

If a network installed permanently is available, each rho4.1 in the network can be diagnosed and used from a local PC (telediagnostic service) in the network.

A remote control and telediagnostic service is possible with:

- ROPS4
- virtual PHG and Gateway
- ReachOut

Remote control with ReachOut

ReachOut is a software that simulates the keyboard and the screen content of a remote PC (rho4.1). The screen content of the remote PC is represented in a window of the local telediagnostic PC.

The ReachOut Explorer allows to transfer files between the local PC and the rho4.1 and vice-versa.

During the installation, an administrator name and a password are requested. If a connection is to be established later to this rho4.1, the administrator name and password are to be entered on the local PC for the connecting.

Settings on the rho4.1

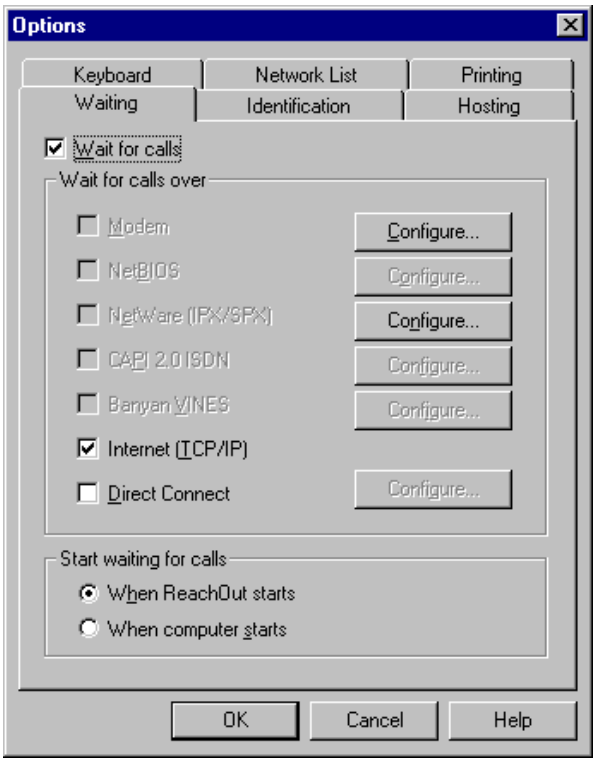
For the communication via TCP/IP, an IP address must be defined on the rho4.1 under [Start] [Settings] [Control Panel] [Network] [Protocols] [TCP/IP −> Crystal LAN(tm) Family Ethernet Adapter] [Properties] [IP Address] [Specify an IP address].

ReachOut must be installed and started on the rho4.1. A call in the auto start should be fixed so that even after a boot of the rho4.1 a connection can be made.

Structure of the rho4.1

For the communication via TCP/IP, the following setting must be made in the installed ReachOut under [Configure] [Options...][Waiting]:

☞ **This is the basic setting if for the installation of ReachOut on the rho4.1 no modem is defined.**



The setting means that after the start of ReachOut, the establishment of a connection via a TCP/IP network is waited for, enabling the establishment of a connection between ReachOut on a local PC and a rho4.1.

Structure of the rho4.1

Settings on the local PC

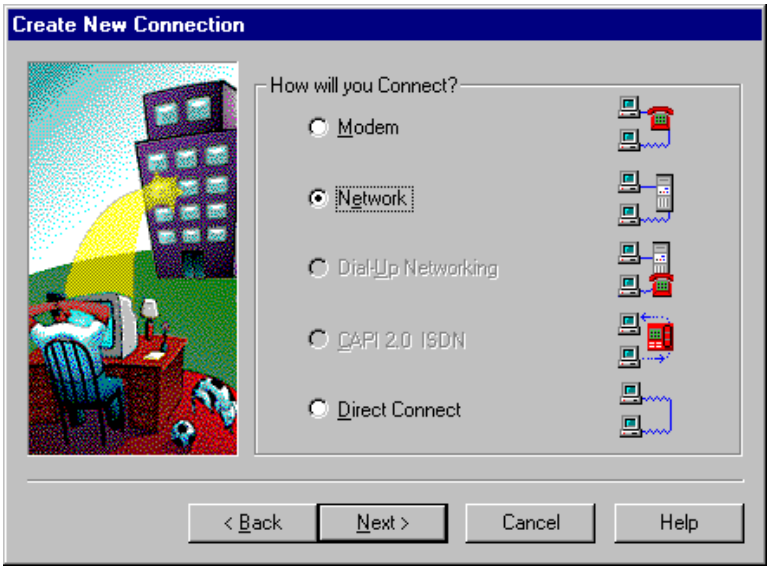
For the communication via TCP/IP, an IP address must be defined on the rho4.1 under [Start] [Settings] [Control Panel] [Network] [Protocols] [TCP/IP –> Crystal LAN(tm) Family Ethernet Adapter] [Properties] [IP Address] [Specify an IP address].

On the local PC (telediagnostic PC), ReachOut must be installed and started and a connection declared.

★ Call menu item [Connection] [New...] to declare a connection.

★ Enter connection name, e.g rho41_1.

★ Enter connection type, here ‘Network’.

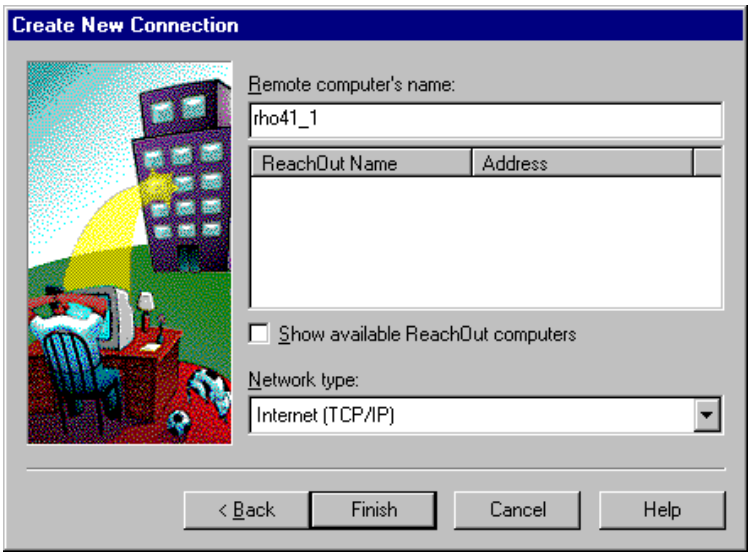


Structure of the rho4.1

★ Enter name of the rho4.1 (Remote computers name) to which later this connection is to be established, e.g rho41_1.

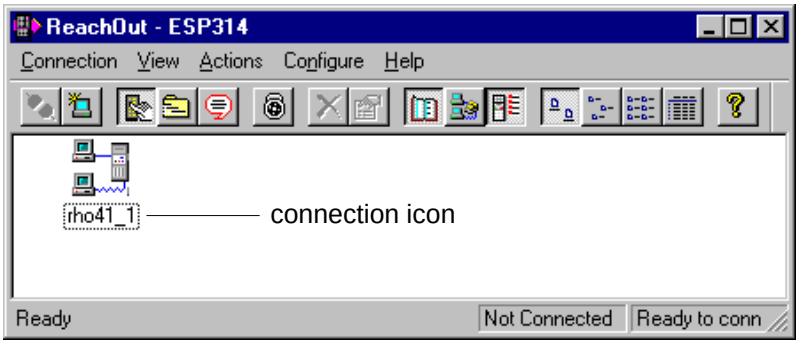
★ Enter communication type (Network type), here: Internet (TCP/IP).

★ In the file ‘hosts’ of the local PC (telediagnostic PC), enter the IP-address of rho41_1, e.g. 193.0.0.1 rho41_1.



★ Actuate button ‘Finish’.

A connection icon is made in which the settings are available with which it is possible to connect to the rho4.1.



Structure of the rho4.1

Connect to ReachOut

The following points are executed on the local PC. The condition is that ReachOut is started both on the local PC and on the rho4.1.

There are 3 different kinds of use for the coupling, it is possible to activate the 3 kinds at the same time:

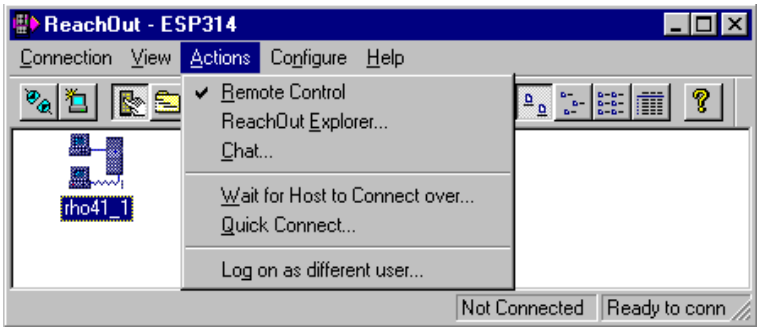
- Connect ReachOut to remote control
- Connect ReachOut to transfer file from and to the rho 4.1
- Connect ReachOut to the remote control and simultaneous file transfer

Connection to remote control

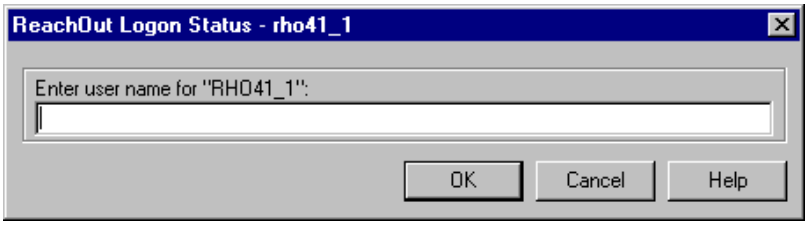
★ Activate connection icon through single mouse click.

★ Activate under menu item [Actions] the tick at Remote Control.

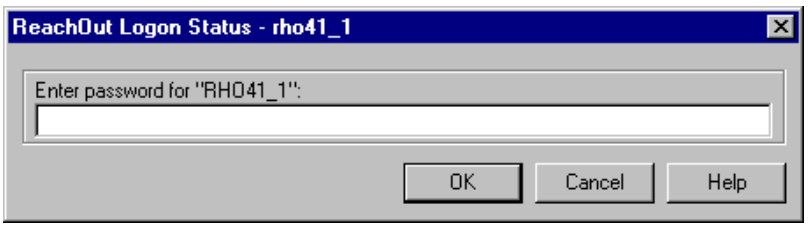
★ Double click with the mouse on the connection icon.



★ Enter user name which was entered during the installation of ReachOut on the rho41_1. The user name defined at the installation on the rho4.1 is ‘user’.

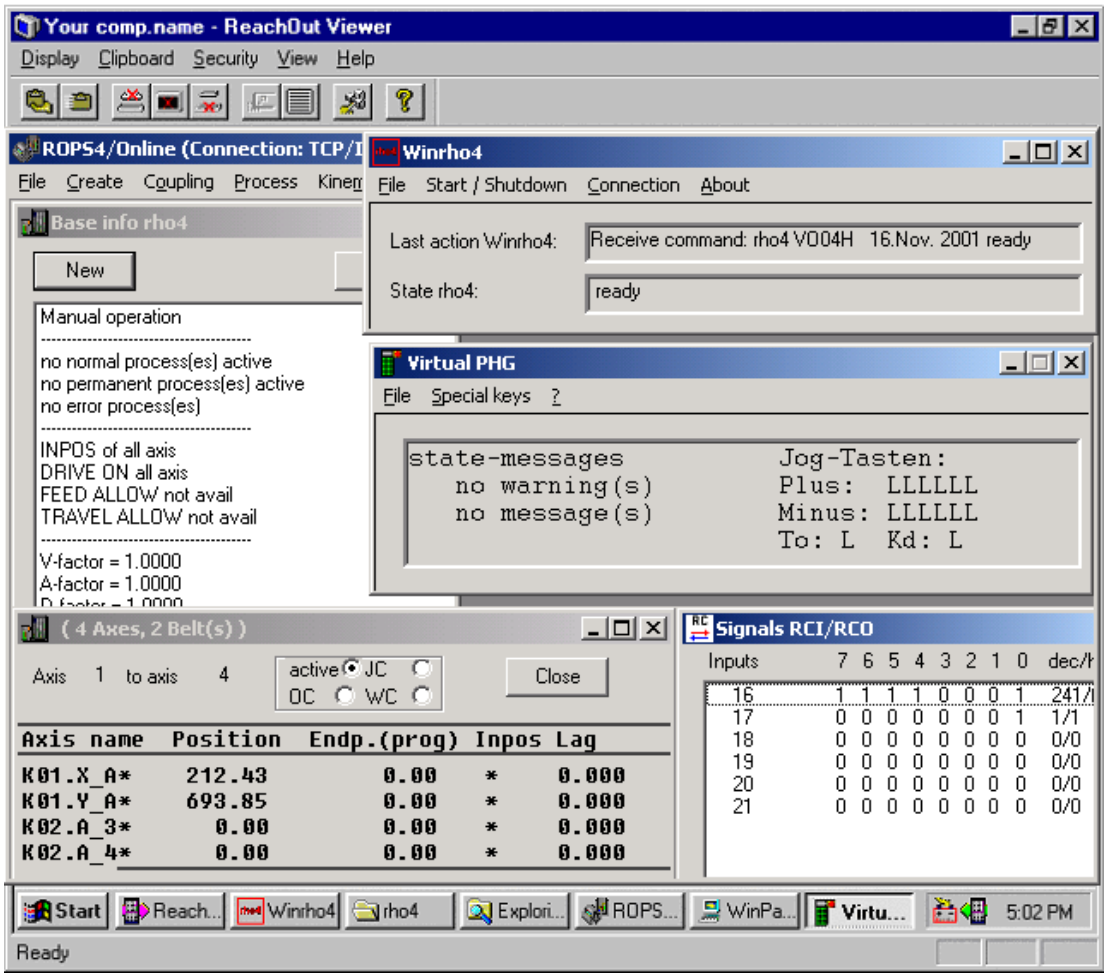


★ Enter password which was entere during the installation of ReachOut on the rho41_1. The password defined at the installation on the rho4.1 is ‘password’.



Structure of the rho4.1

ReachOut on the rho41_1 answers. The rho41_1 can be remote controlled.



The following can be used e.g.:

- Windows
- ROPS4
- virtual PHG

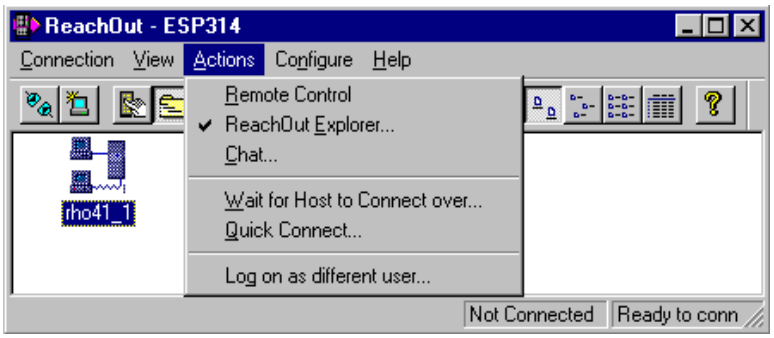
Structure of the rho4.1

Connection to transfer files from and to the rho4.1

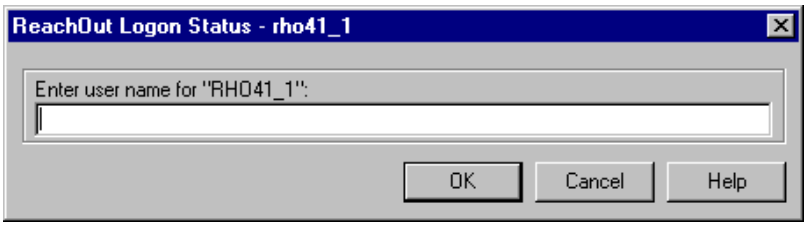
★ Activate connection icon through single mouse click.

★ Activate under menu item [Actions] the tick at ReachOut Explorer.

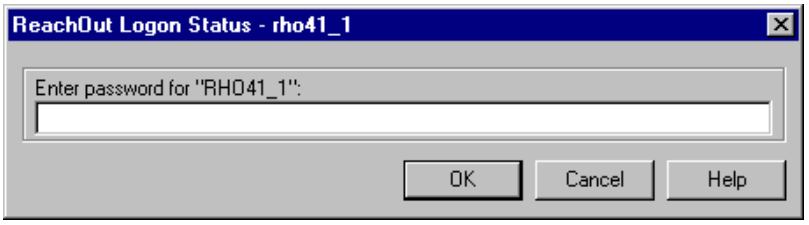
★ Double click with the mouse on the connection icon.



★ Enter user name which was entered during the installation of ReachOut on the rho41_1. The user name defined at the installation on the rho4.1 is ‘user’.



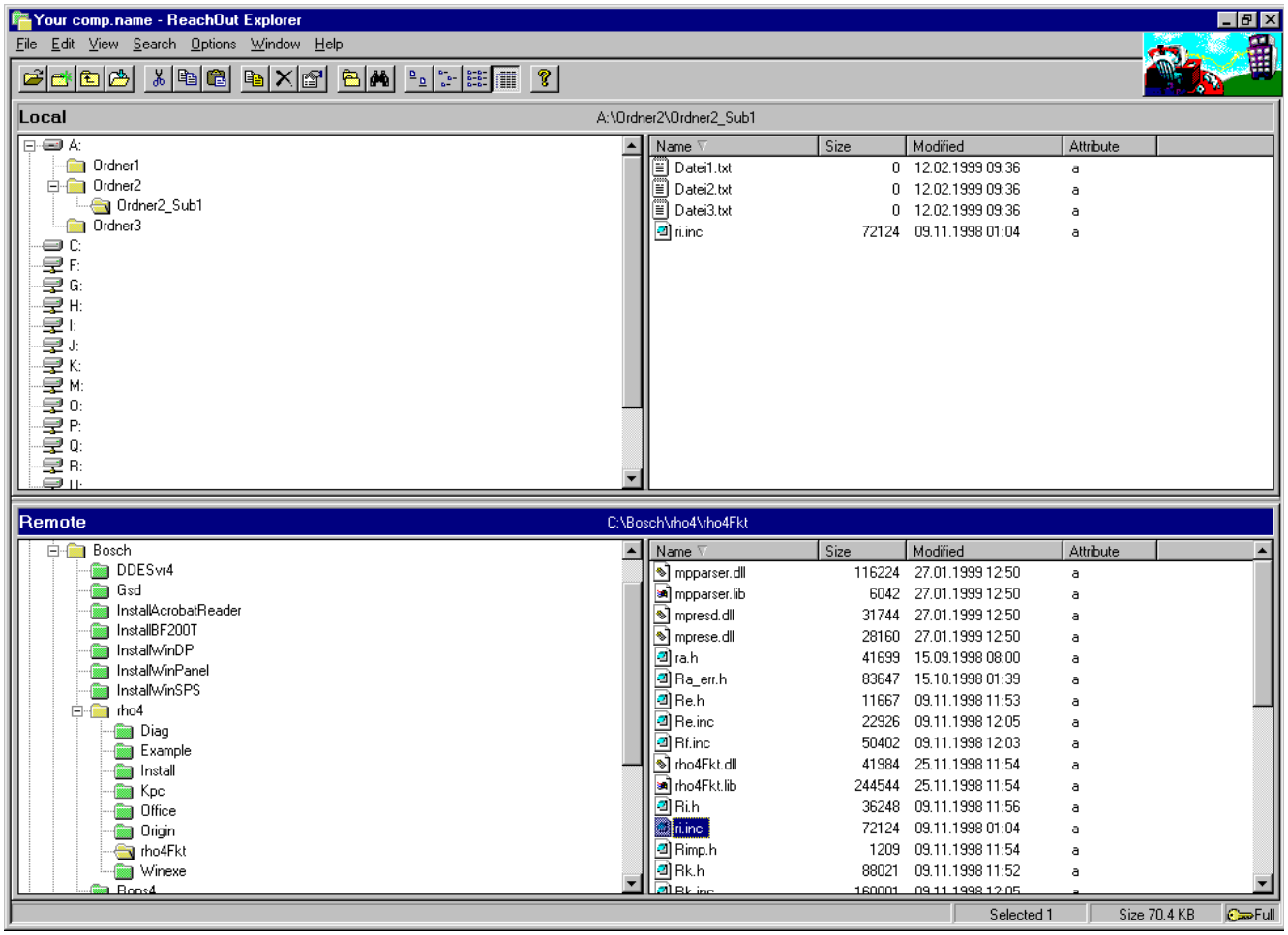
★ Enter password which was entered during the installation of ReachOut on the rho41_1. The password defined at the installation on the rho4.1 is ‘password’.



Structure of the rho4.1

ReachOut represents in the upper display part the files of the local PC and in the lower one the files of the rho41_1.

The ReachOut Explorer is used for file transfer like the Windows Explorer, e.g. a file can be copied in the standard way (<Ctrl C>) on the remote PC and inserted on the local PC (<Ctrl V>).



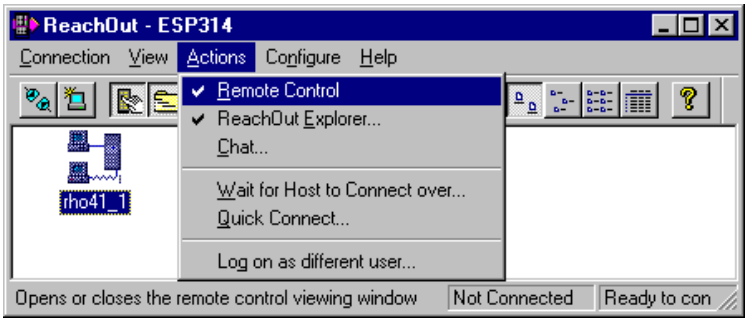
Connection to remote control and simultaneous file transfer

★ Activate connection icon through single mouse click.

★ Activate under menu item [Actions] the ticks at Remote Control and ReachOut Explorer.

★ Double click with the mouse on the connection icon.

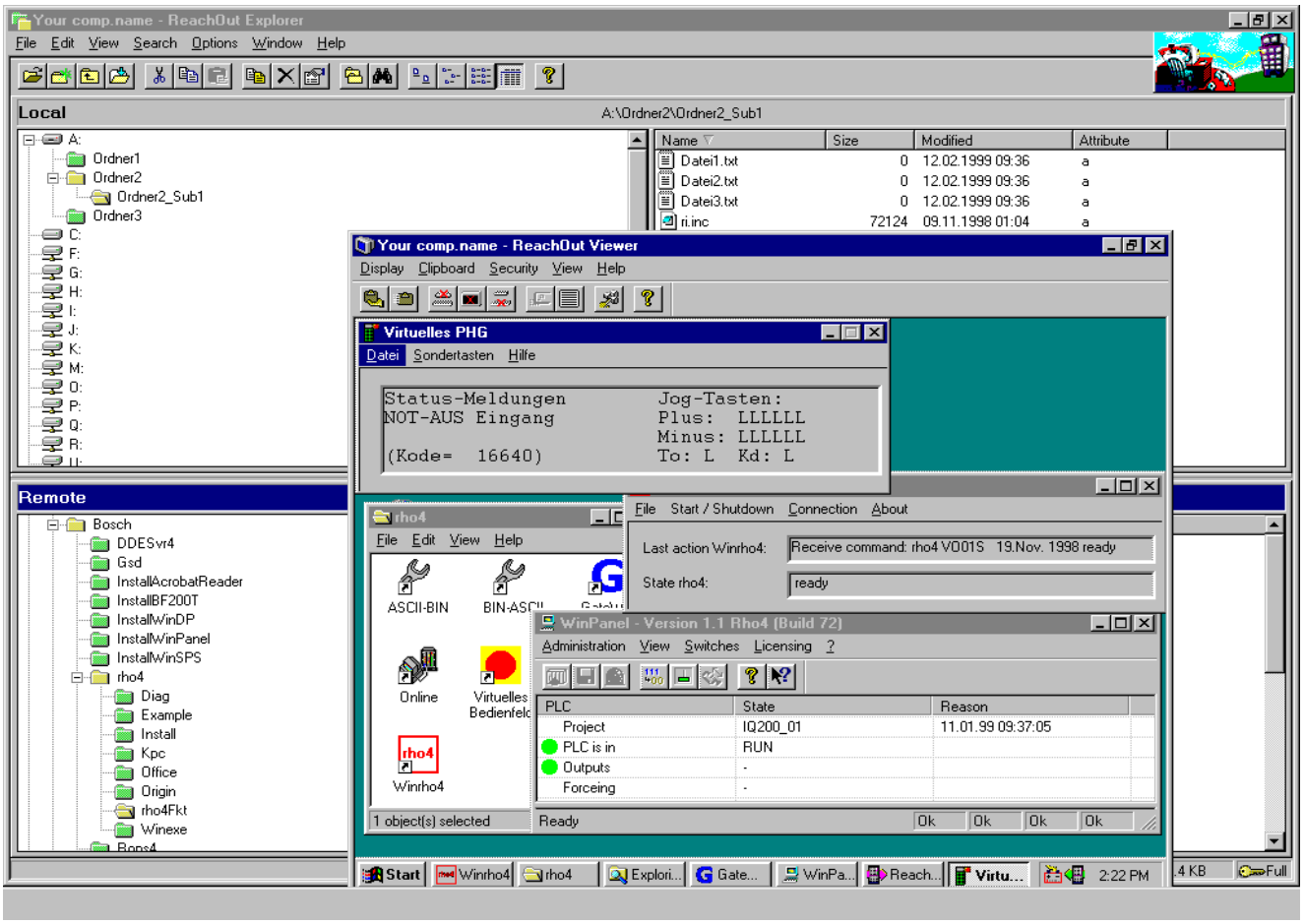
Structure of the rho4.1



A remote control as well as a file transfer with the same connection declaration is possible.

After the establishment of a connection, the following is opened:

- a window with the desktop of the rho4.1 for remote control.
- a window with ReachOut Explorer for the file transfer.

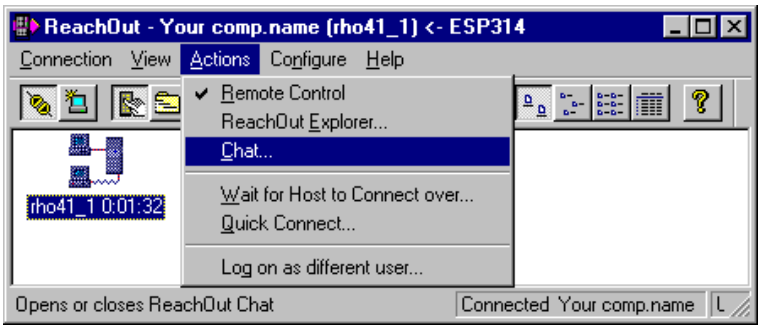


Connect ReachOut and chat

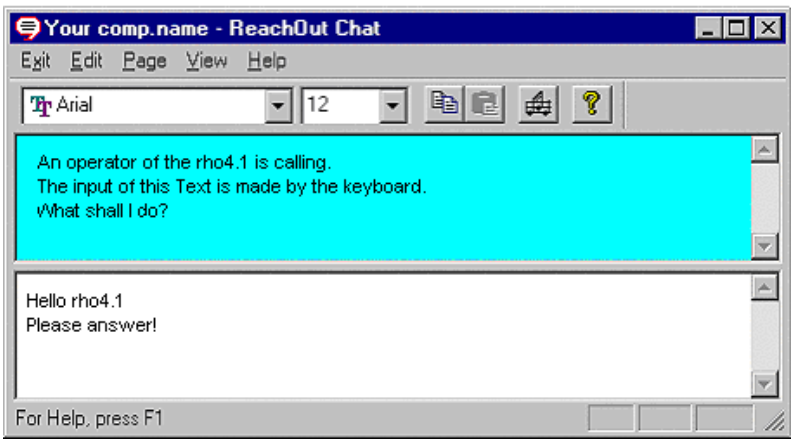
If a coupling to a rho4.1 is existing, it is possible to open in addition to Remote Control and ReachOut Explorer a chat window.

The chat window is opened both on the rho4.1 and on the local PC and is used for the mutual information exchange. Entries on the keyboard of the rho4.1 appear on the local PC and vice-versa.

★ In the case of an existing connection under the menu item [Actions] click on [Chat...].



The chat window is opened and the communication is then possible.



2.5.4 Remote control with ROPS4 and virtual PHG

If an operation or a modification in the Windows part of the rho4.1 is not necessary, it is possible to link the ROPS4 running on the local PC via the gateway to the rho4.1.

With the virtual PHG running on the local PC, it is possible to operate the rho4.1 via the gateway. (See also rho4 manual ROPS4, chapter ‘TCP/IP-settings’).

Structure of the rho4.1

Settings on the rho4.1

★ At PHG under MODE 9.1.1 coupling enter:

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

★ At PHG under MODE 9.1.17 Virt_PHG enter:

Typ (0=Ser/1=Win):		Win. channel
Port number	:	6100

★ In the file ‘GateWay.ini’ enter:

ConnectionNo	=	2
1	=	coupling
2	=	Virt_PHG
[Coupling]	=	
ServerAlias	=	rho4
ServerPortNo	=	6010
GateWayPortNo	=	6010
Msglen	=	256
[Virt_PHG]	=	
ServerAlias	=	rho4
ServerPortNo	=	6100
GateWayPortNo	=	6100
Msglen	=	156

☞ **The file ‘GateWay.ini’ is to be found in the directory C:\Bosch\rho4\Winexe.**

If further channels should be addressed via the gateway besides the coupling channel and the channel for the virtual PHG, ‘ConnectionNo’ is to be adjusted on the desired channel number.

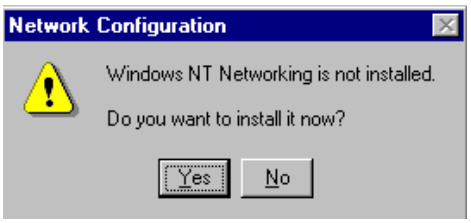
★ Start the GateWay via the file ‘GateWay.exe’, so that ROPS4 and the virtual PHG keep a connection.

☞ **The file ‘GateWay.exe’ is to be found in the directory C:\Bosch\rho4\Winexe.**

Structure of the rho4.1

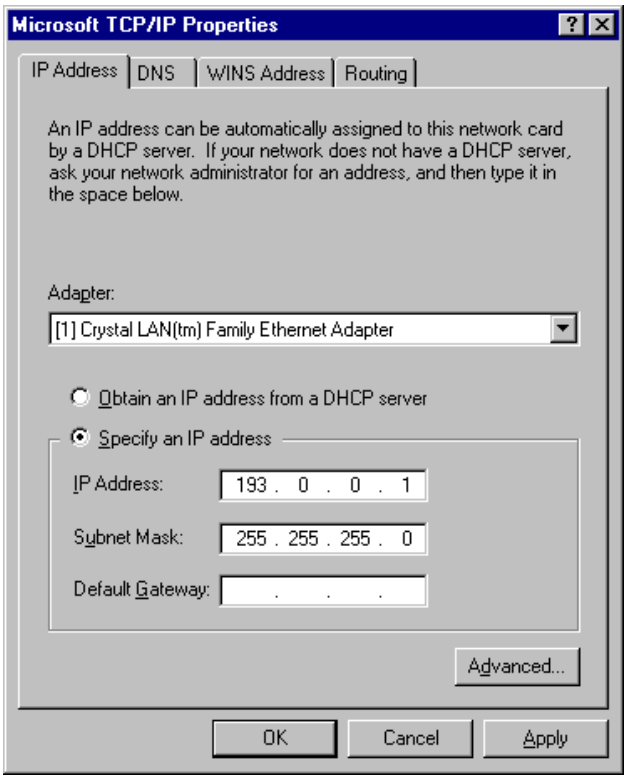
★ Enter the IP address and the Subnet mask under [Start] [Settings] [Control Panel] [Network] [Protocols] [TCP/IP −> Crystal LAN(tm) Family Ethernet Adapter] [Properties] [IP Address] [Specify an IP address].

Only with Windows NT:
After selecting the item “Network” in the Control Panel, the following window will be displayed:



☞ **Do not enter “Yes”. Close this window by pressing “No”.**

Example for an IP address:



Settings on the local PC

★ In the file ‘hosts’ define all rho4.1 that should be remote controlled with alias name and IP address.

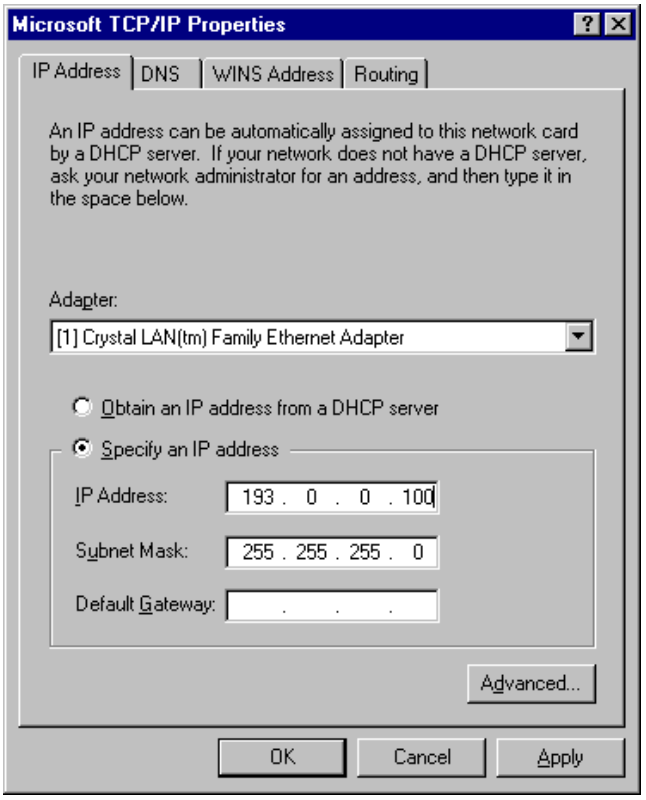
Structure of the rho4.1

193.0.0.1	rho41_1
193.0.0.2	rho41_2
193.0.0.3	rho41_3
193.0.0.4	rho41_4
193.0.0.4	rho41_5
.	.
.	.
.	.

☞ **The file ‘hosts’ is to be found in the Windows directory, e.g C:\Windows or C:\WINNT\System32\drivers\etc.**

★ Enter IP address and the Subnet mask under [Start] [Settings] [Control Panel] [Network] [Protocols] [TCP/IP −> Crystal LAN(tm) Family Ethernet Adapter] [Properties] [IP Address] [Specify an IP address]:

IP address	:	193.0.0.100
Subnet mask	:	255.255.255.0



Link ROPS4

★ Start ROPS4 to connect to the desired rho4

Structure of the rho4.1

★ In the setup of ROPS4-Online under [Optionen][Setup...] enter the rho4.1 with alias name that is to be remote controlled:

Connection	:	TCP/IP
IP-Adress/Aliasname	:	rho41_2
Port number	:	6010

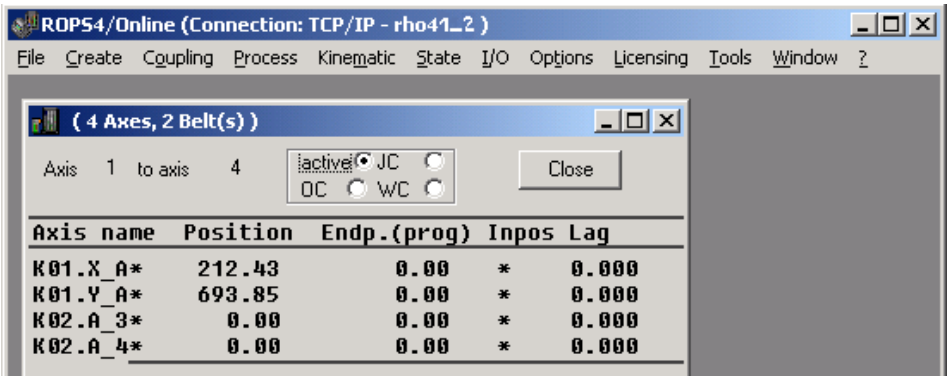
The rho4.1 with the alias name rho41_2 would be in this case remote controlled.

If several rho4.1 should be remote-controlled at the same time, ROPS4 can be started several times with alias names adjusted differently.

If ROPS4/Online is connected with a rho4.1, the alias name and the connection type are displayed in the header. Via this display it is possible to distinguish the ROPS4 started several times.

Here the ROPS4/Online is connected via TCP/IP with the rho4.1 with the alias name rho41_2.

Header with alias name (rho41_2) and connection type (TCP/IP)



2.5.5 Connection via modem

If a network installed permanently is available, each rho4.1 equipped with a modem can be diagnosed and operated from a PC equipped with a modem.

A remote control and a telediagnostic is for instance possible with ReachOut. For the duration of a remote control and telediagnostic, the normal telephone network is used. The arising costs are the same as for a normal telephone conversation.

Remote control with ReachOut

ReachOut is a software that simulates the keyboard and the screen content of a remote PC (rho4.1). The screen content of the remote PC is represented in a window of the local telediagnostic PC.

Structure of the rho4.1

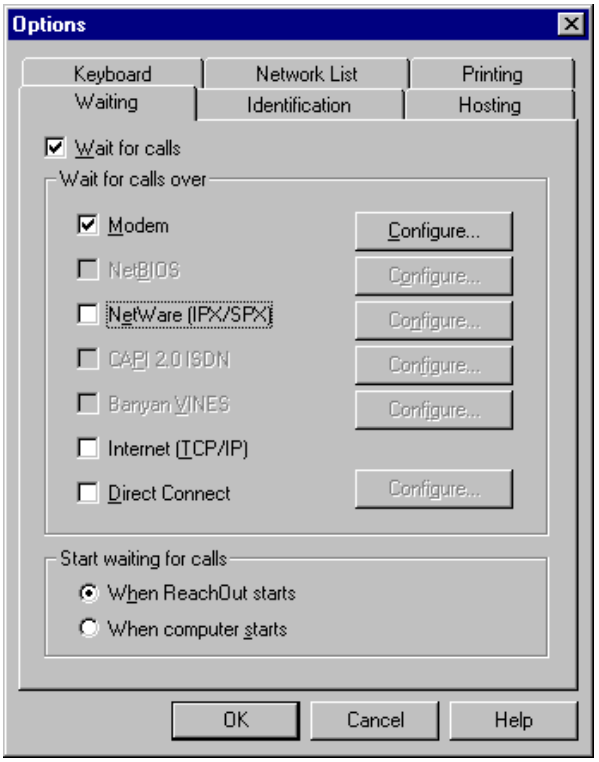
The ReachOut Explorer allows to transfer files between the local PC and the rho4.1 and vice-versa.

During the installation, an administrator name and a password are requested. If a connection is to be established later to this rho4.1, the administrator name and password are to be entered on the local PC for the connecting.

Settings on the rho4.1

★ Start ReachOut on the rho4.1. A call in the auto start should be fixed so that even after a boot of the rho4.1 a connection can be made.

★ In ReachOut under [Configure][Options...][Waiting] perform the following setting:

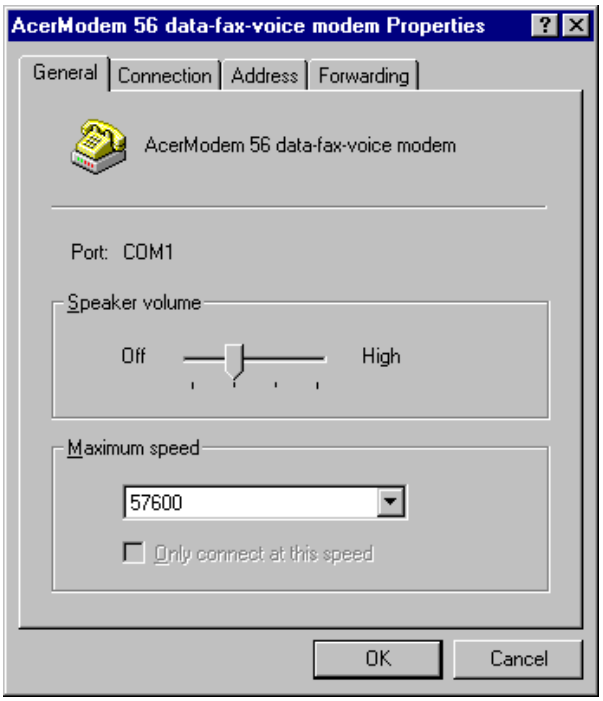


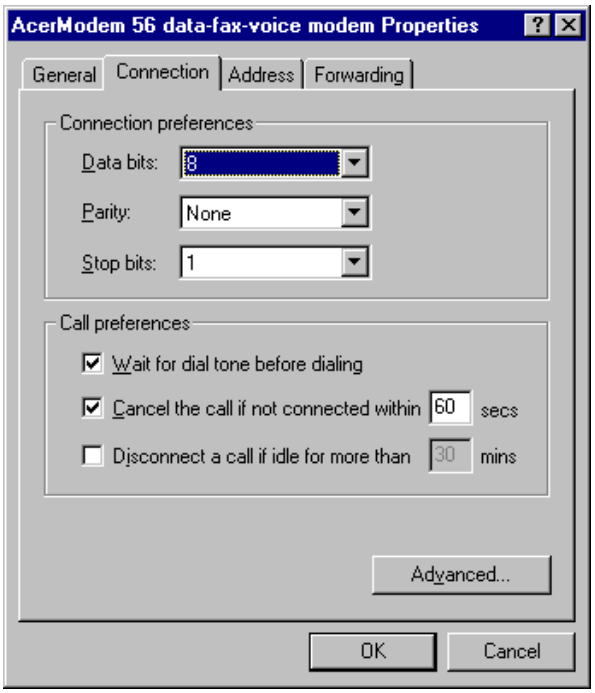
The setting means that after the start of ReachOut the establishment of a connection via a modem is waited for, enabling ReachOut to establish a connection to rho4.1 via a local PC.

Structure of the rho4.1

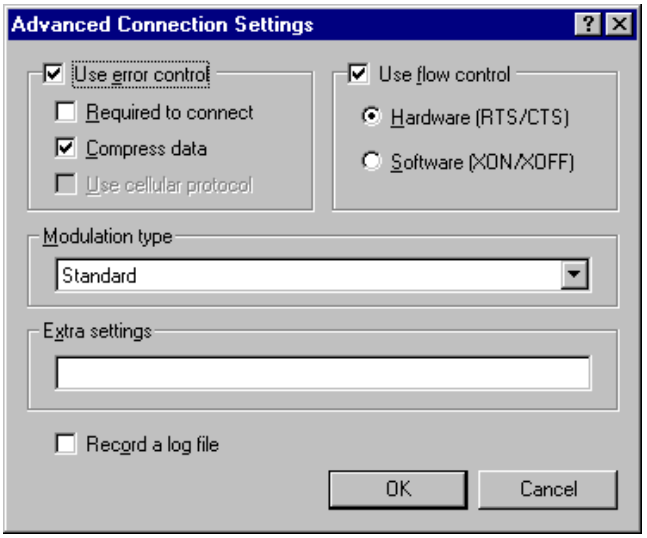
Under [Configure][Options...][Waiting][Configure...] perform the following or analog settings:

☞ **Make sure that the settings below are the same as during the installation of the modem on the rho4.1, see page 2−38 ff.**





Structure of the rho4.1



Settings on local PC

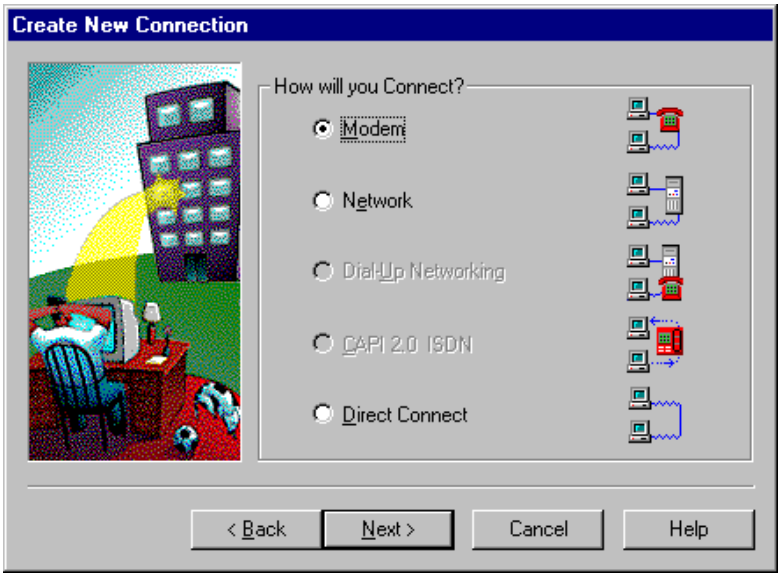
On the local PC (telediagnostic PC), ReachOut must be installed and started and a connection declared.

★ Call menu item [Connection] [New...] to declare a connection.

★ Enter connection name, e.g rho41_1.

★ Enter connection type, here ‘Modem’.

Structure of the rho4.1



★ Enter name of the rho4.1 (Remote computers name) to which later this connection is to be established, e.g rho41_1.

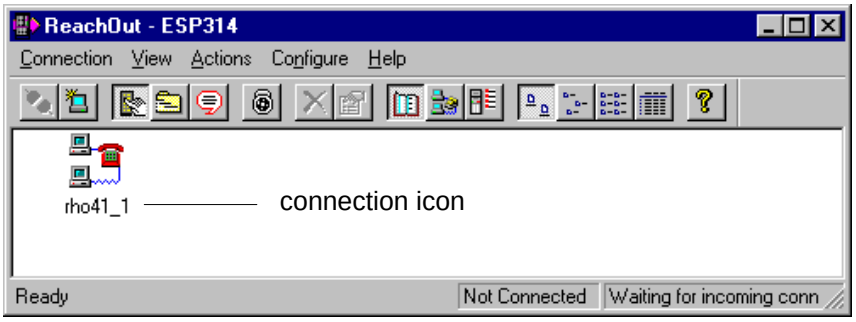
The parts of the telephone number mean:

01	dialling code (only for operation at a telephone network)
0030	International country code (0030 for Greece)
1	National area code (here Athen)
47110815	Telephone number of the modem connection (user number)

★ Actuate button ‘Finish’.

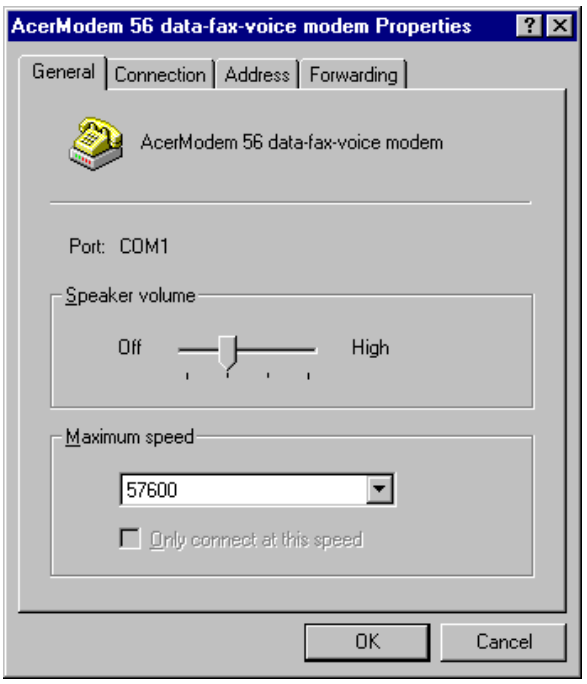
A connection icon is made in which the settings are available with which it is possible to connect to the rho4.1.

Structure of the rho4.1

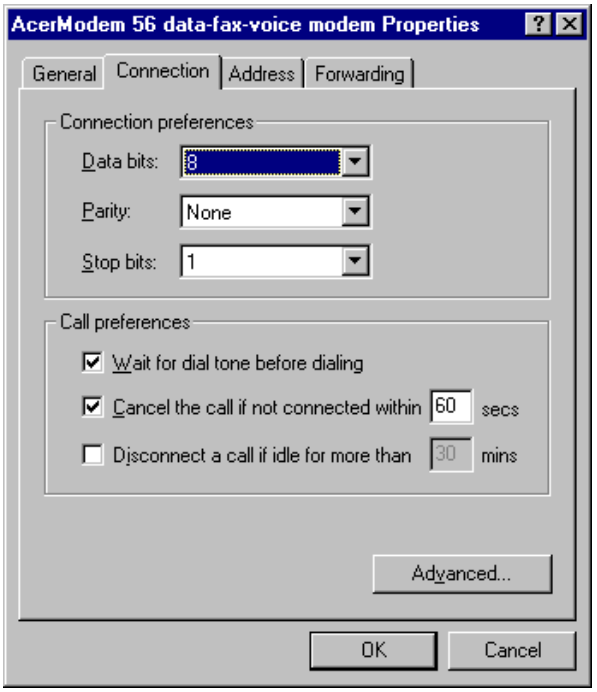


Under [Configure][Options...][Waiting][Configure...] perform the following or analog settings:

☞ **Make sure that the settings below are the same as during the installation of the modem on the rho4.1, see page 2−38 ff.**

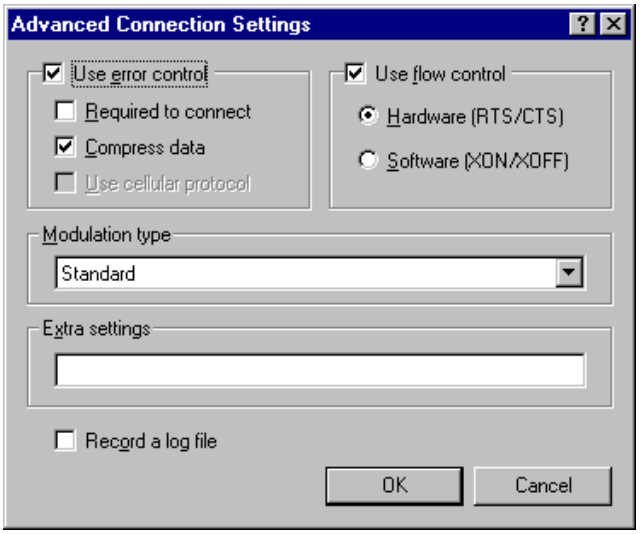


Structure of the rho4.1



Under the call settings, it is to be taken into account:

- If it is to be dialed from an internal telephone network, the tick must be deactivated before 'Wait for dial tone before dialing'.
- If it is to be dialed from a direct telephone, the tick must be activated before 'Wait for dial tone before dialing'.



Structure of the rho4.1

Connect to ReachOut

The following points are executed on the local PC. The condition is that ReachOut is started both on the local PC and on the rho4.1.

There are 3 different kinds of use for the coupling, it is possible to activate the 3 kinds at the same time:

- Connect ReachOut to remote control
- Connect ReachOut to transfer file from and to the rho 4.1
- Connect ReachOut to the remote control and simultaneous file transfer

For further procedure, see page 2−49 ff.

Structure of the rho4.1

2.6 MADAP Studio reporting system from version 1.2 on rho4.1 Pentium3-variants

General

MADAP Studio is a windows-based visualisation tool which run on all rho4.1 Pentium3 variants from software version VO06B. The rho4.1 control supplies library functions to display all system messages of the control and the drive by the reporting system MADAP Studio (from version 1.2). As well, additional detailed informations are available.
Both components form a complete system, which run on the same PC.

Installation instructions to MADAP Studio

The installation is made by a Windows-Setup-Program.
The following settings should be used :

1. Monitor screen resolution VGA 1024 x 768
2. Delete current version, save customer specific data
3. Would you like to install Auto-Startup ? >>> ”Yes”
4. TCP/IP-communication with protocol TCP/IP
5. TCP/IP driver selection:
   PCL local, 1x Rho-System, ext. Reporting system, PLC-IO, Sequential function chart range 1-64, Panel-No. ”MMI 1”, no Touch
6. RHO driver selection:
   Rho/IP-Address 192.0.1.2, Type: Rho4.1, Selection:Channel 4
7. Select ”Yes, I want to restart my computer now.”

By the supposition that ”Winrho4.exe” is in the Startup folder of Windows, point 3 causes, that the MADAP Studio visualisation gets absolute priority when the rho4.1 is switched on and a rho-communication can be build up. This means, that the MADAP Studio application starts, the MADAP Studio communication to the rho4.1 is build up and the MADAP Studio user interface is displayed on the screen. All MADAP Studio functions inclusive the extended reporting system are active.

Functionality

MADAP Studio offers a user interface for plc-controlled machines and installations, to

· control,
· display conditions,
· make available statistics about processes
and
· diagnostic signals and interruptions.

Parallel to the MADAP Studio, several Windows applications may be active.

Structure of the rho4.1

Extended reporting system

Unvisible to the user, MADAP Studio has cyclic accesses to actual rho4.1 system messages. The contents of existing messages are read and stored (inclusive date and time) in the display list. Due to the display width, detail- and help informations must asked for separate (see XML file access).
MADAP Studio differs between state messages (red), warnings (yellow) and notes (green). Because the rho4.1 informs between new and reset messages, MADAP Studio takes over automatically the reset, resp. the deletion within the display list.
MADAP Studio supports at runtime max. two rho4.1 controls at the same time.

XML file access

For each stand next rho4.1 system message, a so called longtext exist, which gives detailed information about the possible cause and makes available more legends.
These add on texts depend on language and device. The file format of these textfiles is XML. They are in C:\Bosch\rho4\rho4fkt\Meldungen\Texte. The user must copy these files to the MADAP Studio subdirectory "...\Texte". The files may be individual expanded by the user.

Structure of the rho4.1

Notes:

PCL

3 PCL

The rho4.1 uses an integrated PCL which provides its own I/O map and access to rho4 I/O. The PCL is programmed with WinSPS, a PLC programming tool.

PCL user programs are programmed with WinSPS statement lists, KOP contact plan or AS process language. 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 PCL 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 PCL for the purpose of time-controlled program processing and can be called up by means of prepared program modules within a fixed time pattern.

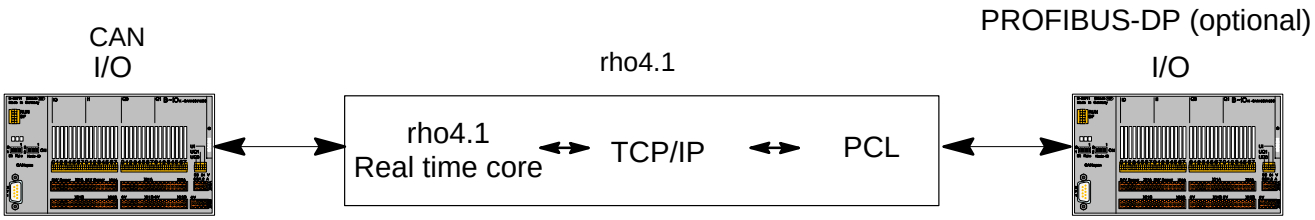
The I/O takes place within an interpolation cycle as long as the PLC cycle time is smaller than the rho4 interpolation cycle. The reaction time to I/O depends on the processor used as well as the OEM configuration, number of movement systems, axes or MMI visual display.

The concept supports two PLC coupling variants of the rho4. A communication interface is set up which is the same for both variants.

PCL

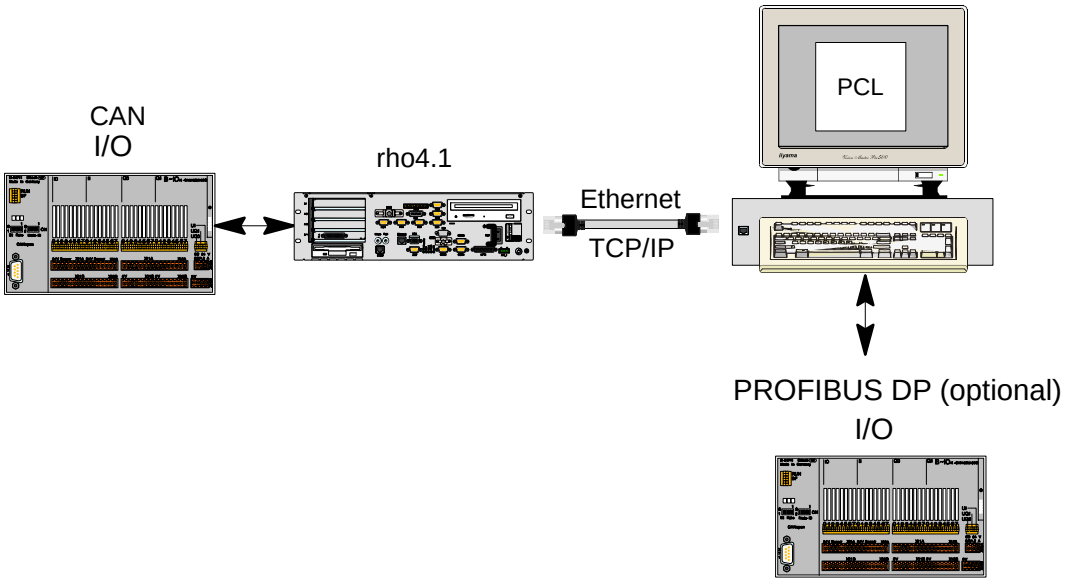
rho4.1 with PCL on one PC

With this variant, the rho4.1 operating system and the PCL are implemented in the hardware of one PC. The rho4.1 communication interface sees to the exchange of I/O and user data.



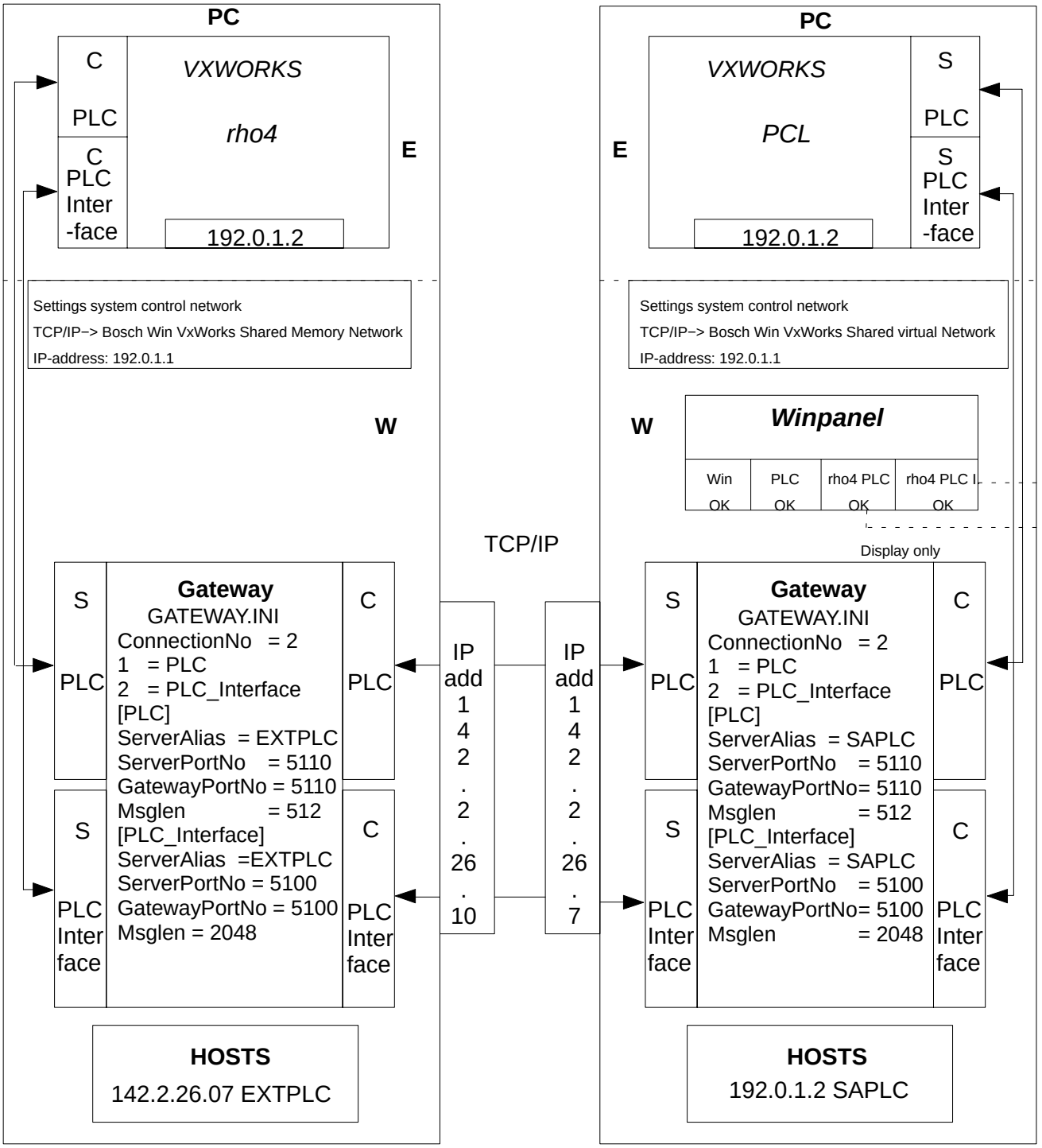
rho4.1 and PCL on separate PCs

With this variant, the PCL operates on a separate PC. Transmission to the rho4.1 is implemented using TCP/IP. This variant mirrors the rho4.1 interface on the PCL PC. The user, therefore, also has access to his data from the rho4.1 interface on the PCL PC.



PCL

rho4.1 only



Machine parameters:

P20 I/O assem. config.
41
P21 PLC parameter
.
Time control: 5000

PHG:

Mode 9.1.11 READ/WRITE PLC
IA: 192.0.1.1
Port number: 5110
Mode 9.1.12 PLC_Interface
IA: 192.0.1.1
Port number: 5100

Meaning:

C = Client
S = Server
E = Real time core
W = Windows

Winrho4.ini

;;Mode C:The Winpanel
;;runs on an external PC
;;To select this mode,please delete the semicolon in the
;;following two lines
[WinPanel]
Exe =

PCL

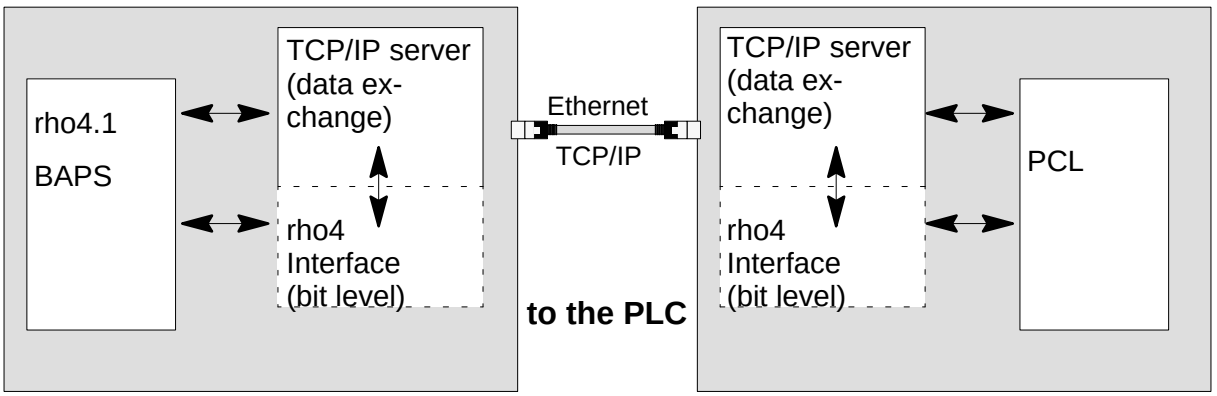
Data channel

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 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 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. Here, no distinction is made as to whether the PLC software runs on the same hardware or whether it runs externally. The following figure shows the data flow between the rho4 and PLC over the TCP/IP data channel.

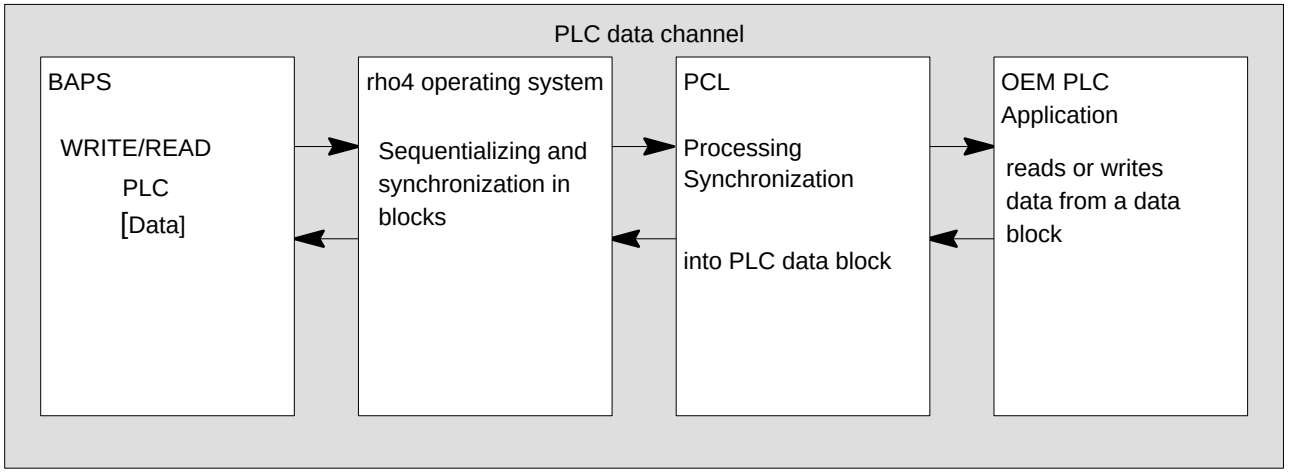


PCL

Programming the data channel to the PLC coupling

Programming via the data channel does not depend on the type of coupling for the Ethernet/Profibus/system bus. The OEM only programs the type of data to be transmitted. The data from the rho4 can be read out or written in on the PCL, 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
```

PCL

```
   REAL    : RVAR

   POINT   : PNTVAR

   ; .

RECORD_END: WRITE_RANGE           ;Declaration of record variables

REAL   : X_RVAR                   ;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
```

PCL

3.1 Configuration parameter of WinPanel

The WinPanel is started by Winrho4.exe.

The following configuration parameters can be entered at the start:

/OPC	Activation of the OPC server of the PLC (License required)
/L	Activated of the display of the menu item licensing
/M	Internal system messages of the WinPanel are not output
/K1	WinSPS runs on the rho4.1 hardware
/K2	Activation of the UDP protocol (also required for BOSCHCOM) WinSPS runs via UDP protocol on an external PC
/K16COM1-19200	WinSPS runs on an external PC with serial protocol on the COM1 interface of the rho4.1 with 19200 Bauds
/K16COM2-19200	WinSPS runs on an external PC with serial protocol one the COM2 interface of the rho4.1 with 19200 baud
/K16COM1-19200	Combination of /K1 und /K2 and /K16COM1-19200
/K16COM2-19200	Combination of /K1 und /K2 and /K16COM2-19200
/R	Activation of the computer interface
/R10	Wait 10 ms before a new connection to the computer interface
/Y4	During the installation of a PCL field bus card with InterBus-S, the option ‘fast InterBus-S’ is activated
/h	Deactivation of the Hardlock (Dongle), only a Crypkey license is searched for

PCL

3.1.1 Start WinPanel through Winrho4

The WinPanel is started by Winrho4.exe.

Winrho4.exe is in the directory C:\Bosch\rho4\Winexe. Before the start of the WinPanel, Winrho4.exe checks if in the directory C:\Bosch\rho4\Winexe the file Datei Winrho4.ini is present.

If Winrho4.ini is not present, the WinPanel is started with the following default settings:

/K19COM1-19200/M/OPC/R10/L

The meaning of the switches can be found in the initialization file “Winrho4.ini”.

In the directory C:\Bosch\rho4\Origin, the file Winrho4.ini is as a mere ’Comment file’. If the WinPanel should be started with other switches (as default), the file Winrho4.ini must be copied into the directory C:\Bosch\rho4\Winexe\. At the corresponding places in Winrho4.ini, the comment signs are to be removed.

3.1.2 Example

During a start of the WinPanel with the configuration ‘/K19COM1-19200/M/OPC/R10/L/h’, the following is set:

/K16COM1-19200	WinSPS can ● run on the rho4.1 ● run on an external PC and coupled via Ethernet with UDP protocol ● run on an external PC and can be coupled via the serial interface with 19200 Bauds
/M	Internal system messages WinPanel are not output
/OPC	The OPC server of the PCL is activated
/R10	RSS computer interface is activated A new connection is established after a waiting time of 10 ms
/L	The menu item Licensing of the WinPanel is displayed
/h	The WinPanel does not notice any plugged Hardlock (Dongle) and recognizes a Crypkey license if present

4 CAN-Bus-Peripherie

4.1 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.

4.1.1 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-modules

The accesses to the machine parameters and to the serial number in the SrCan modules (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 4.2.

CAN-Bus-Peripherie

4.1.2 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 Protocol 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 DssRef (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**

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

CAN-Bus-Peripherie

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 to the parameter 401.

The I/O-modules are operated synchronously.

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

CAN-Bus-Peripherie

4.2 CANopen-encoder

As a physical belt input, an incremental measuring system is available at the rho4.1. This measuring system CAN be occupied by several belts.

Allowed physical belt inputs are also CANopen encoders that observe the guidelines 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

4.2.1 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 objects (machine parameters of the encoders) 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).

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-Peripherie

4.2.2 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-Peripherie

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 value	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-Peripherie

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

CANopen specific encoder parameters

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

c:\Bosch\rho4\CANopen\Axis1.scs	e.g. drive parameter axis 1 (DssRef = 1)
c:\Bosch\rho4\CANopen\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.

CAN-Bus-Peripherie

4.3 CAN belts

As a physical belt input, an incremental measuring system is available at the rho4.1. This measuring system CAN be occupied by several belts.

Physical belt inputs CAN be the measuring system inputs 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 in analogy to the regulated axes.

4.3.1 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 5)
- Measuring system input of the incremental measuring system
- Measuring system input of a CANopen encoder

4.3.2 Machine parameters

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

The regulated axis from which the measuring system is used for the belt logic is indicated in the subparameter of P401 CAN Axis No.

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 4.2). 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-Peripherie

Example 1 for Machine parameter 401

4axes 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 value	100
A02	Measuring system factor	1000.0
A02	Command value output	2
A02	Dss-reference	2

CAN-Bus-Peripherie

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-Peripherie

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-Peripherie

4.4 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 rho 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.

4.4.1 Bus switching

The modulear field bus switching B~IO M-CAN CAN be driven by means of the CANopen protocole or also with the Bosch rho conformal CAN protocole. The protocole 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).

☞ **CAUTION:**
If analog and digital I/O modules are operated with a bus connection, it must be imperatively ensured that after the bus connection the analog at first and then the digital modules are engaged. If this instruction is not heeded, this can lead to inconsistent data.

CAN-Bus-Peripherie

4.4.2 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 4−21, 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.

CAN-Bus-Peripherie

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 P407 (‘Assignment of analog inputs’, subparameter ‘Format’ and ‘Nominal value’). See page 4−22, Assignment of the analog input channels.

4.4.3 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-Peripherie

- 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

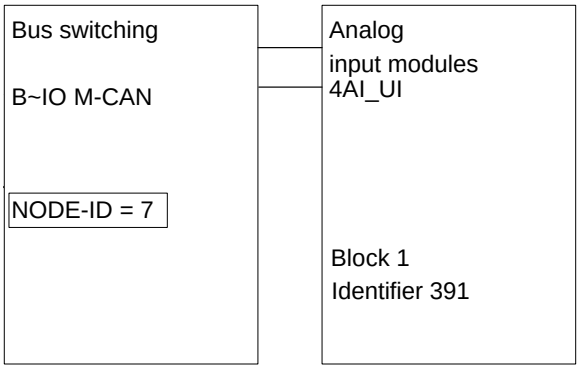
- CAN-Bus-Number:
  The CAN-Bus number indicates with which of the both CAN busses fo the rho4 the B~IO modules are operated. Only the numbers 1 and 2 are authorized. They correspond to the CAN plug number X51 and X52.

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 modules is connected via the CAN plug X52 to the rho4.

As CAN identifier to be set, it follows:

384 + 7= 391



CAN-Bus-Peripherie

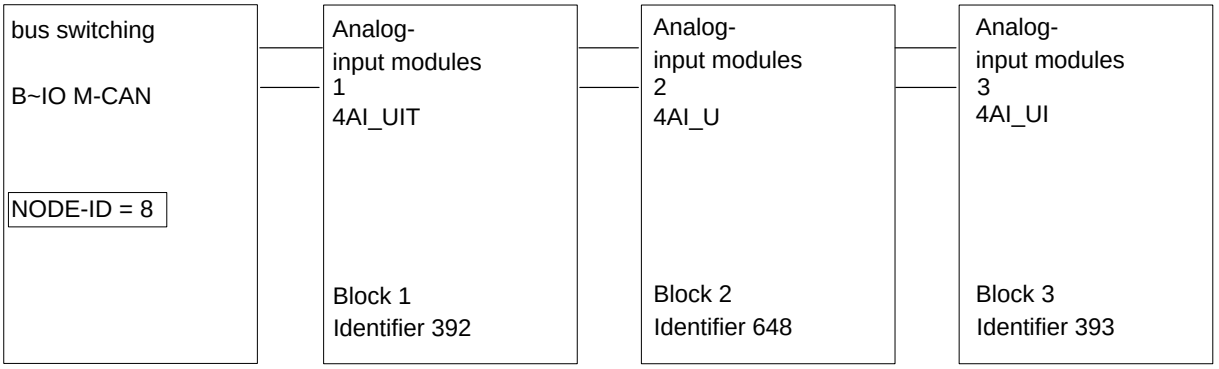
For the configuration, the following settings are to be entered under machine parameter P31:

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 modules. The node ID of the B~IO M-CAN modules is set via the DIP switch on the value 8. The modules 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

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:

- IO-Kind:
  Analog modules get the code 1, digital modules the code 0.
- 1st 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-modules:

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-Peripherie

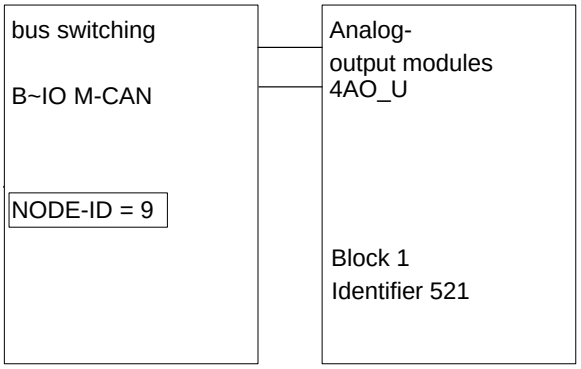
- CAN-Bus-Number:
  The CAN-Bus number indicates with which of the both CAN busses fo the rho4 the B~IO modules are operated. Only the numbers 1 and 2 are authorized. They correspond to the CAN plug number X51 and X52.

Example: CAN output parameter

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

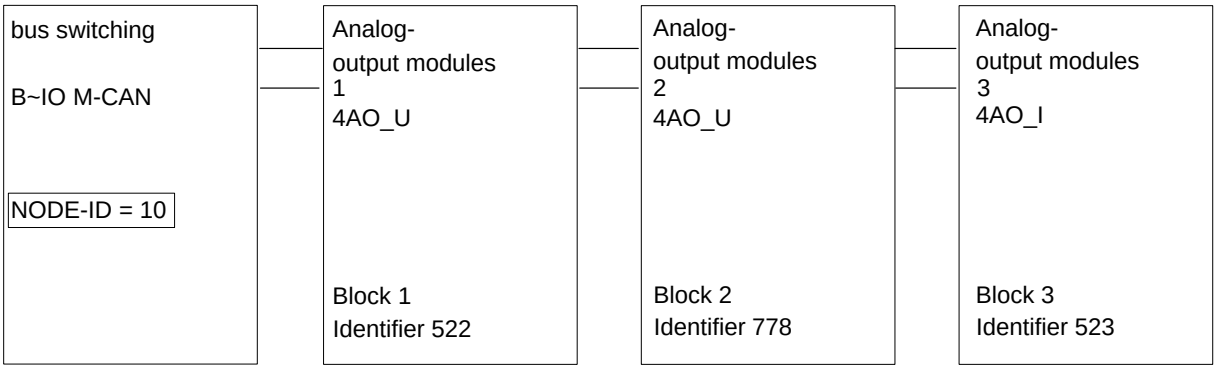
IOKind Block 1	:	1
1stAdr Block 1	:	0
Length Block 1	:	8
Ident Block 1	:	521
Bus-No. Block 1	:	2

CAN-Bus-Peripherie

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 modules is set via the DIP switch on the value 10. The modules 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-Peripherie

4.4.4 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.

Assignement of the analog input channels (P407)

In P407 the following is to be entered:

- Meaning:
  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 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-Peripherie

- 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-Nr	:	Output reference path speed Kinematic X
450 + Kin-Nr	:	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

CAN-Bus-Peripherie

- 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).

4.4.5 Example

A bus interface with 2 analog I/O modules is connected to a rho4.1. 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:

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

CAN-Bus-Peripherie

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

Numb. of Inp.-Bl.	:	1
Numb. of Out-Bl.	:	1
nmb. SRCAN modules	:	0
CANopen Downl.	:	0
I/O configuration CAN 1		
• Baudrate	:	0
• CANrho=0, CANopen= 1	:	0
I/O configuration CAN 2		
• Baudrate	:	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-Peripherie

Machine parameter P404

Number of analog outputs : 4

Machine parameter P405

Assignement 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-Peripherie

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-Peripherie

4.4.6 BAPS program

```
PROGRAM ANA_IO
;*****************************************************************************
;*Demo program CAN-Analog-I/O with B~IO
;*P405,P407: Nominal value = 1000 = 10 Volt
;*****************************************************************************
INPUT REAL: 201 = AE1, 202 = AE2, 203 = AE3, 204 = AE4      ;analog inputs
OUTPUT REAL: 201 = AA1, 202 = AA2, 203 = AA3, 204 = AA4     ;analog outputs
INTEGER: INDEX
REAL : D1, D2, D3, D4
BEGIN
      Loop:
            WRITE PHG, cls, 'D1,...,D4' WAIT 2
            INDEX = -10
            REPEAT 20 TIMES
                  AA1=100*INDEX
                  AA2 = -100
                  AA3 = -700
                  AA4 =  900
                  WAIT 1
                  D1 = AE1
                  D2 = AE2
                  D3 = AE3
                  D4 = AE4
                  WRITE PHG, cls, D1,'  ',D2,'  ',D3,'  ',D4,'  '
                  WAIT 1
                  INDEX = INDEX + 1
            REPEAT_END
      JUMP Loop
PROGRAM_END
```

CAN-Bus-Peripherie

4.5 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 rSGSRCOn() and rSGSerialNb(), the functions for writing in the SR-CAN module are named rSSSRCOn() 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.

CAN-Bus-Peripherie

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.

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 legnth 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. The file is located in the directory C:\BOSCH\rho4\ORIGIN.

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-Peripherie

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 rho4.1. If all entries are unequal zero, a machine parameter bakkup 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 'serial number 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-Peripherie

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.

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

CAN-Bus-Peripherie

The settings for the analog I/O are described in section 4.6 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.

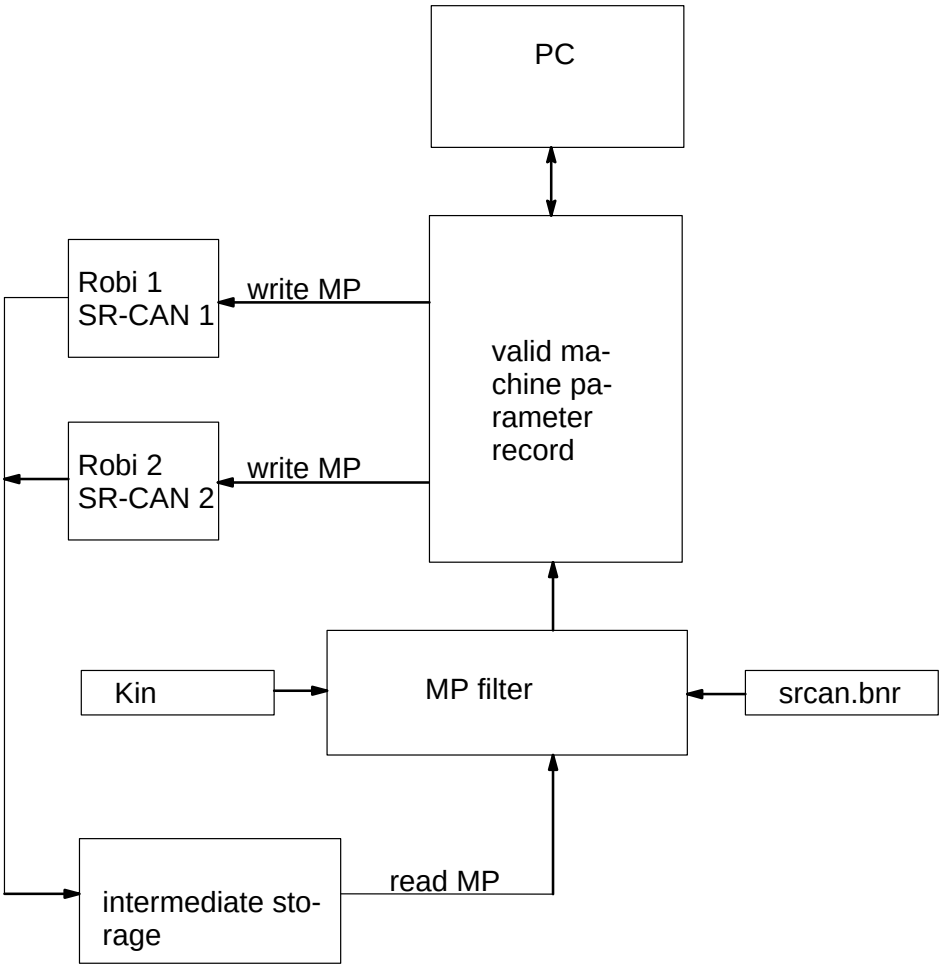
B8	B7	B6	B5	B4	B3	B2	B1

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
Baudrate 20k	x	off	off	x	x	x	x	x
Baudrate 125k	x	off	on	x	x	x	x	x
Baudrate 500k	x	on	off	x	x	x	x	x
Baudrate 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-Peripherie

Read/write MP



CAN-Bus-Peripherie

4.6 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 (description of the hardware, see chapter 4.7). 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.

4.6.1 CAN-bus switching

The SR-CAN module can be driven by:

- CANopen protocol
- CANrho protocol

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

For the setting of the baud rate, the CAN knot address and the protocol, a 8-fold 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 4−15, CAN-I/O parameter, machine parameter P30, P32).**

4.6.2 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 (assignement of the analog input channel).

CAN-Bus-Peripherie

4.6.3 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 ouputs of the module on the rho4 side must also be taken into account.
If this is not the case, the transfer of the inputs will be also interrupted and the rho4 signals the state: 'no transfer 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 4.5)

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 4.4.2).

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

- 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 assignement.

In the machine parameter P407, the 8 byte data block is assigned to the 4 input channels. See also 4−37, Assignement of the analog input channels (P407).

The 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.

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

Identifier = 384 + Node-ID

CAN-Bus-Peripherie

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 4.4.2).

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.

The 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 inputs and their assignement 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.

Assignement 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
```

- RAM initial address:
  This relative address gives the position of the 2-byte data word for the analog channel within the rho4.1 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.

CAN-Bus-Peripherie

4.7 Assignment of the CAN-Bus interfaces

The rho4 has two separate CAN-Bus interfaces:

- Plug-No. X51
- Plug-No. X52

These interfaces 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-Peripherie

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. Baudrate
up to 25 m	1 MBaud
up to 100 m	500 kBaud
up to 200 m	250 kBaud
up to 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 24
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-Peripherie

4.7.1 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 A-Block/module)
- 3 DESI65-K-CAN digital input -modules with 1 byte each (1 E-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 = 20 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-Peripherie

CAN-Bus 1 (X51)		**CAN-Bus 2 (X52)**	
14 axes		6 axes + 2 B~IO-modules	
Number of O-blocks = 1 + 14 (axes)	= 15	Number of O-blocks = 1 + 6 (axes) + 2 (Digital output blocks)	= 9
Number of I-blocks = 14 (axes)	= 14	Number of I-blocks = 6 (axes) + 2 (Digital input blocks)	= 8

Example 4

- 6 kinematics with 4 axes each = 24 axes
- 1 RM65M-CAN module with 16 bytes digital output each (2 O-blocks/ module) and 16 bytes digital input each (2 I-blocks/module)

CAN-Bus 1 (X51)		**CAN-Bus 2 (X52)**	
24 axes		RM65M-CAN-module	
Number of O-blocks = 1 + 24 (axes)	= 25	Number of O-blocks = 1 + 2 (Digital output blocks)	= 3
Number of I-blocks = 24 (axes)	= 24	Number of I-blocks = 2 (Digital input blocks)	= 2

CAN-Bus-Peripherie

4.8 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 E/A-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 24 axes ID = 200 + 10 * i i = 1 to 24 210 . . . 370			
		Inputs ID = 384 + i i = 1 to 99 385	Inputs Area 1 ID = 384 + i i = 1 to 127 385	Inputs Area 1 ID = 384 + i i = 1 to 127 385
	390	.	.	.
		.	.	.
	400	.	.	.
		.	.	.
	410	.	.	.
		.	.	.
	420	.	.	.
		.	.	.
	430	.	.	.
		.	.	.
	440	.	.	.
		.	.	.

CAN-Bus-Peripherie

Servodyn-G	**Servodyn-D**	**DESI65K-CAN**	**RM65M-CO BIO-K-CAN**	**CANopen E/A-modules encoder, axes**
		.	.	.
		.	.	.
		.	.	.
		483	.	.
			.	.
			511	511
		Outputs ID = 512 + i i = 1 to 99	Outputs Area 1 ID = 512 + i i = 1 to 127	Only I/O modules and axes: Outputs Area 1 ID = 512 + i i = 1 to 127
		513	513	513
		.	.	.
		.	.	.
		.	.	.
		.	.	.
		.	.	.
		.	.	.
Actual values ID = 600 + 10 * i i = 1 to 6	Actual values ID = 600 + 10 * i i = 1 to 24	.	.	.
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		.	.
	.		.	.

CAN-Bus-Peripherie

Servodyn-G	Servodyn-D	DESI65K-CAN	RM65M-CO BIO-K-CAN	CANopen E/A-modules encoder, axes
			.	.
	.		.	.
			.	.
	.		.	.
	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
	770		.	.
	.		.	.
	.		.	.
	.		.	.
	.		.	.
	.		.	.
	.		.	.
	830		.	.
	.		.	.
	840		840	.
			841	.
			.	.
			.	.
			.	.
			.	.
			895	895
				SDO Area 1 ID = 1408 + i i = 1 to 127
				1409
				.
				.
				1536

CAN-Bus-Peripherie

Servodyn-G	**Servodyn-D**	**DESI65K-CAN**	**RM65M-CO BIO-K-CAN**	**CANopen E/A-modules encoder, axes**
				SDO Area 2 ID = 1536 + i i = 1 to 127 1537 . . 1663

CAN-Bus-Peripherie

4.9 Servodyn-D-rho4 interface

4.9.1 Control data rho4 −> Servodyn-D

Interface Signal rho4	Signification in the drive
MF-output axisX Out1 (MF_Out_x_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 01_RCI..BRAKE24_RCI) No. 424 to 447, O53.0 to O55.7	0: Close brake 1: Open brake

CAN-Bus-Peripherie

4.9.2 Status messages Servodyn-D −> rho4

Designation in the drive	**Error number**	**Effect in the rho4 Error messages, interface outputs**	**Reset*)**
Control voltage	F 35	Error message: CAN-Logic Power Code: 24320 + axis No. − 1	H
Excess voltage Undervoltage	F 30 F 99	Error message: CAN Overvoltage Code: 24448 + axis No. –1	H
Overtemperature module	F 07	Error message: CAN Controller-Temp. Code: 24576 + axis No. − 1	H
Overtemperature Motor	F 08	Error message: CAN Motor-Temp. Code: 24704 + axis No. − 1	H
Feedback Commutation error encoder error	F 11 F 99 F 70	Error 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		Error message: CAN current fault Code: 25216 + axis No. − 1	H
ZSK2.interface (ZSK2: Warning)		Error message: CAN communic. fault Code: 25344 + axis No. –1	H
VM-switch-off	F 98	Error message: CAN-VM-Switchoff 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	Error message: CAN-VM-error Code: 24192 + axis No. –1	G
Excessive deviation	F 13	Warnung: 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	Error message: global CAN-fault Code: 25472 + axis No. –1	H

CAN-Bus-Peripherie

*) Reset condition in rho4:

H = Control start-up

G = Basic position (Interface-Signal)

K = Reset not required

CAN-Bus-Peripherie

Notes:

SERCOS interface

5 SERCOS interface

The SERCOS interface is realized in the rho4 as a digital drive interface. A SERCOS ring, at which up to 24 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.

5.1 Data exchange via SERCOS bus

5.1.1 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.

5.1.2 Cyclic data exchange

The rho4 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’ ,x=1..24 (DRIVE_x_RCI Nr.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, Nr. 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-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	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 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

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).

SERCOS interface

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

SERCOS interface

5.2 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.

5.2.1 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 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 analysis
When using BOSCH 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-K.	:	1	number of the servo board (at present always 1)
BN1 SCS Plug	:	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 analysis When using BOSCH 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.

5.2.2 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 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
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 Servodyn D drives). The download is then switched off in the parameter P38.

SERCOS interface

5.3 Referencing

5.3.1 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-lead referencing is only possible with cyclically absolute and absolute measuring systems.

5.3.2 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 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 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-controlled 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

- 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

5.4 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.

5.4.1 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-Loop open Phase: 0: (code = 169088)	The SERCOS ring is not closed Check SERCOS connection
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) Sercos-error Phase3 axis x Y− zzz (code = 169728) Sercos-error Phase4 axis x Y− zzz (code = 169728)	The indicated drive parameter cannot be transmitted owing to an error x = axis number Y = SERCOS parameter type (S = Standard, P= Product specific) zzz = SERCOS ident. number 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

SERCOS interface

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

5.4.2 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-controlled 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-controlled referencing has been interrupted owing to an error by the drive Read error code to the seven segment display of the drive amplifier

5.4.3 Warnings

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

SERCOS interface

5.5 Transmission of cyclic drive data to the PCL

From version VO04X, the data which is transmitted via the SERCOS bus in the cyclic axis telegram from the drive to the rho4, 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.

5.5.1 Definition of the cyclic axis telegrams (AT)

Within the rho4

As input buffer for axis telegrams, a maximum of 22 bytes per axis is defined in the rho4. This corresponds to the maximum AT length for BOSCH-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 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

5.5.2 Treating of the axis telegrams in the rho4

From the rho4, 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
Numb. 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 for the error display and the position display.

SERCOS interface

5.5.3 Transmission of the axis telegrams to the PCL

The ATs are defined in the rho4 with 22 bytes per axis. With a maximum configuration with 24 axes, it follows a transmission length of 528 bytes.

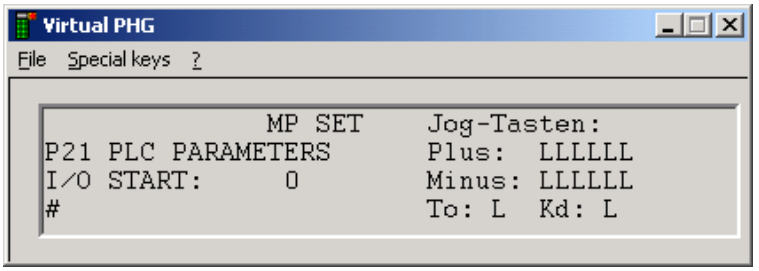
☞ **Depending on the system stress, there can be delays in the transmission, so that no clock synchronous transmission can be guaranteed. Moreover, it cannot be guaranteed that there is a transmission to the PCL at each clock (P5). If a transmission of drive data to the PCL is not yet completed and there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

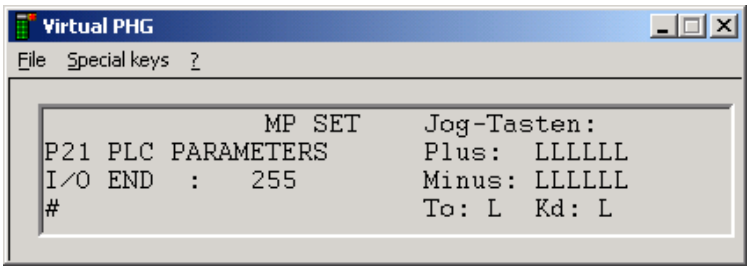
Setting P21 PLC PARAMETER

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 * 24 axes	=	528 Byte
Reserve	=	108 Byte
rho4 system counter	=	4 Byte
=> Number of the bytes to be transmitted	=	640 Byte

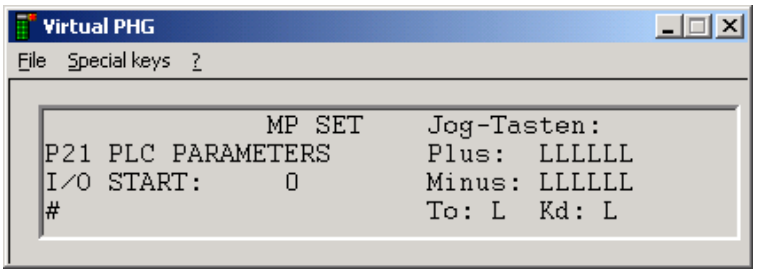
State of rho4.1 when delivered:



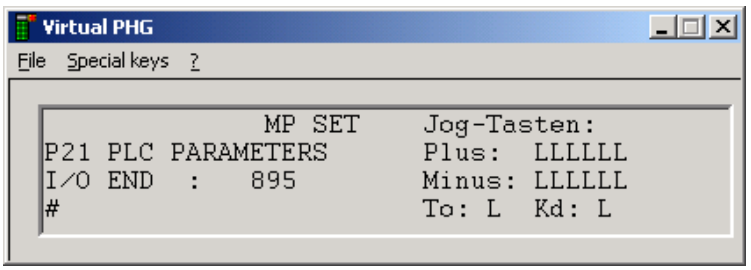


SERCOS interface

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



The setting of the PLC final address results from:
PLC final address + 640 = 895



☞ **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.**

5.5.4 Transmission of the rho4 system counter

Additionally to the ATs, the internal rho4 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.

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

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

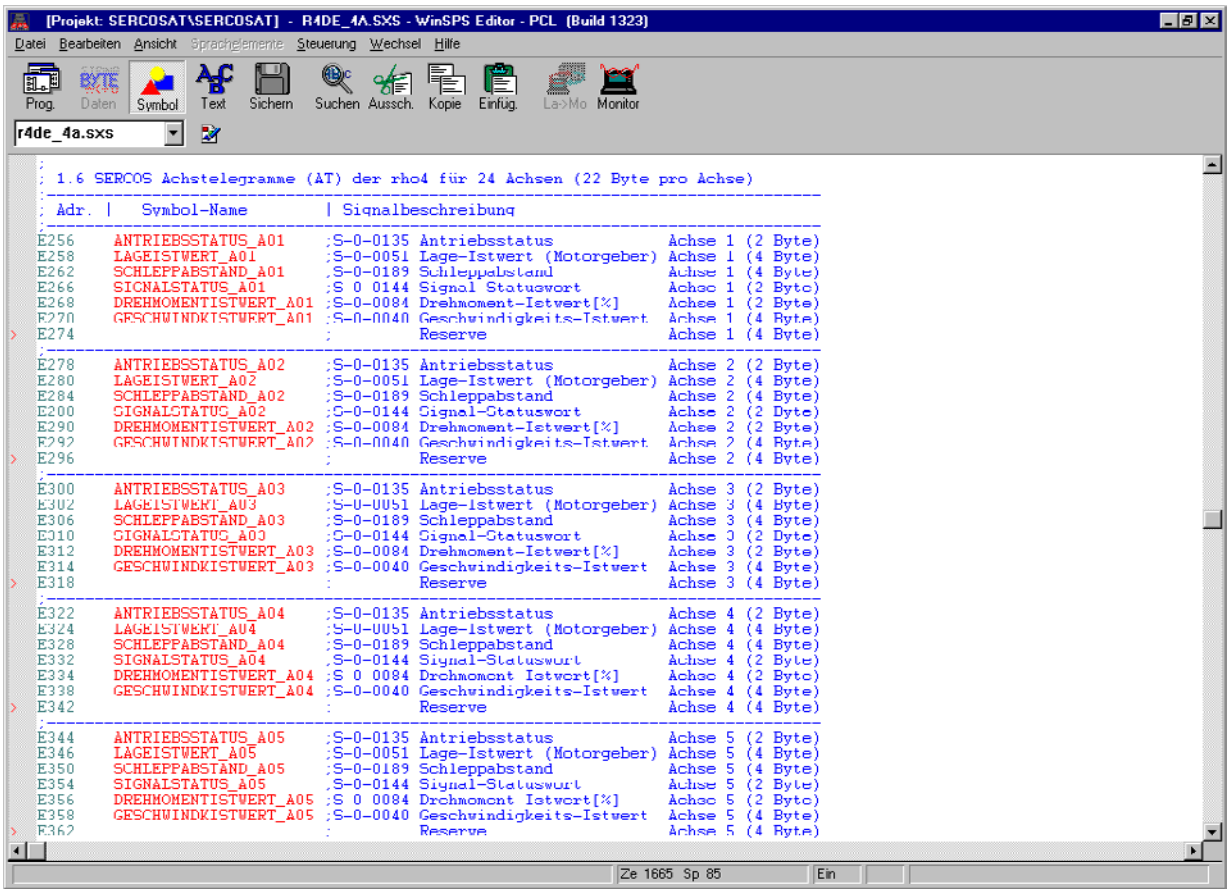
SERCOS interface

5.5.5 Representation of the ATs in the PCL

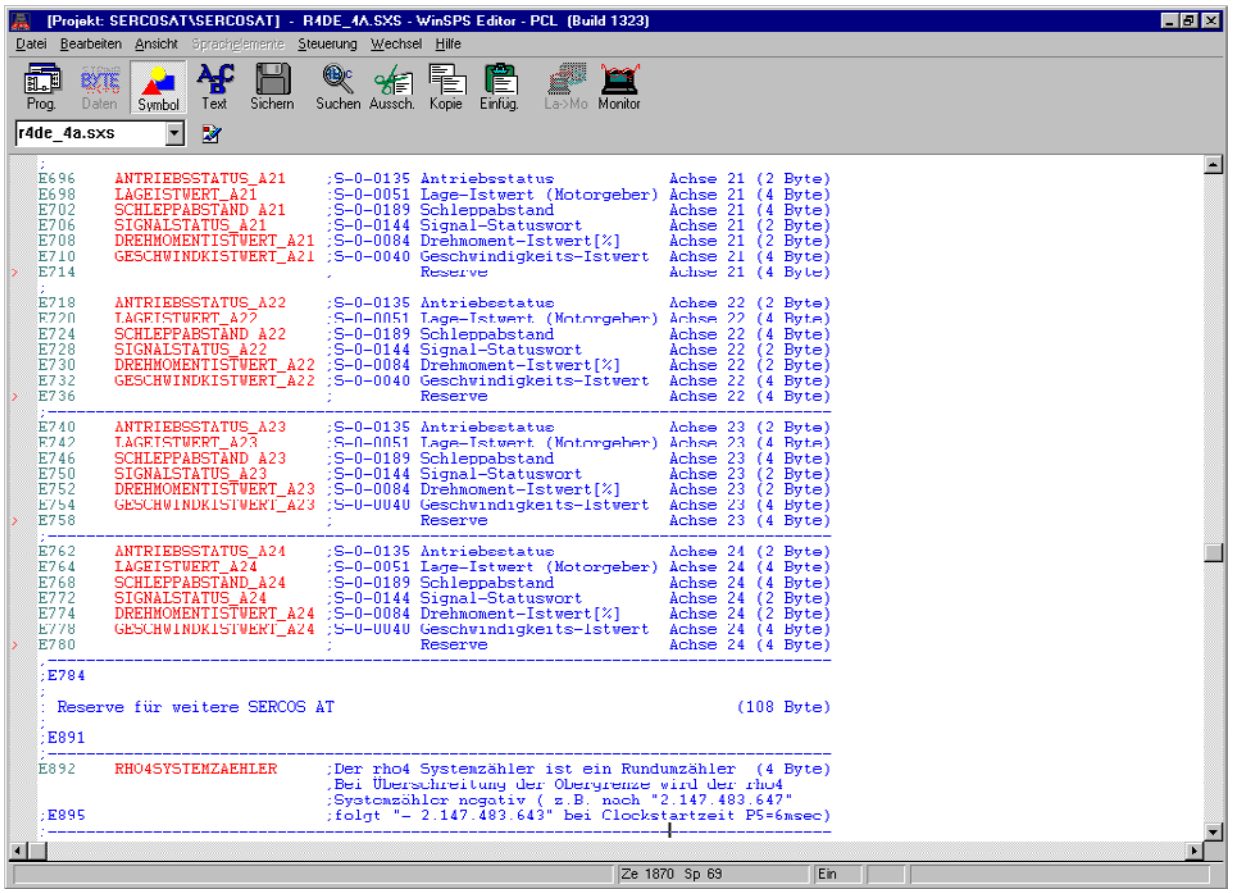
The following pictures show an example of the SERCOS axis telegrams on the PCL.

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 system counter	=	4 Byte
=> Number of the bytes to be transmitted	=	640 Byte



SERCOS interface



The symbol file of the rho4 PCL program is prepared for the assignement of the AT values.

☞ **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 5.5.2. This also applies to the other values of the ATs.

SERCOS interface

5.6 Belt input via external encoder at EcoDrive/IndraDrive

5.6.1 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.

5.6.2 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.

5.6.3 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

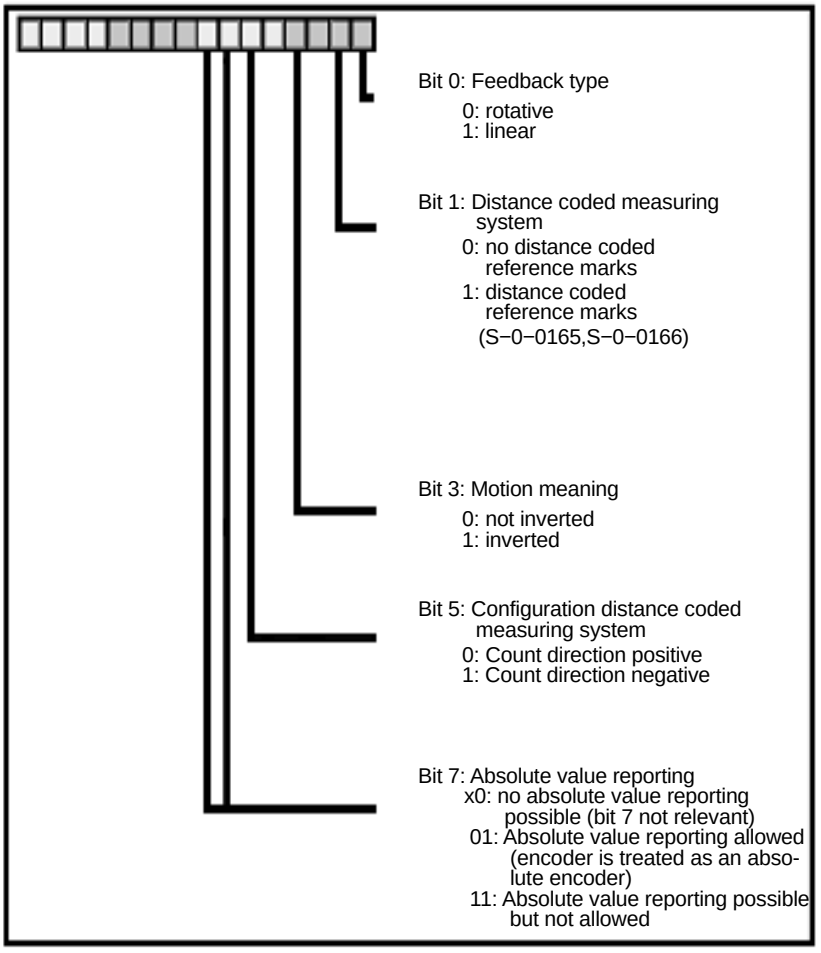
At adjusted modulo format (parameter S−0−0076, type of weighting of the position data weighting bit 7), the modulo value determines, at which numerical value the position data overflow to zero. If encoder1 is an absolute encoder, the modulo value must be adjusted greater then the traversing range of 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.

SERCOS interface

S−0−0117 must be calculated as follows:

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

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.

5.6.4 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

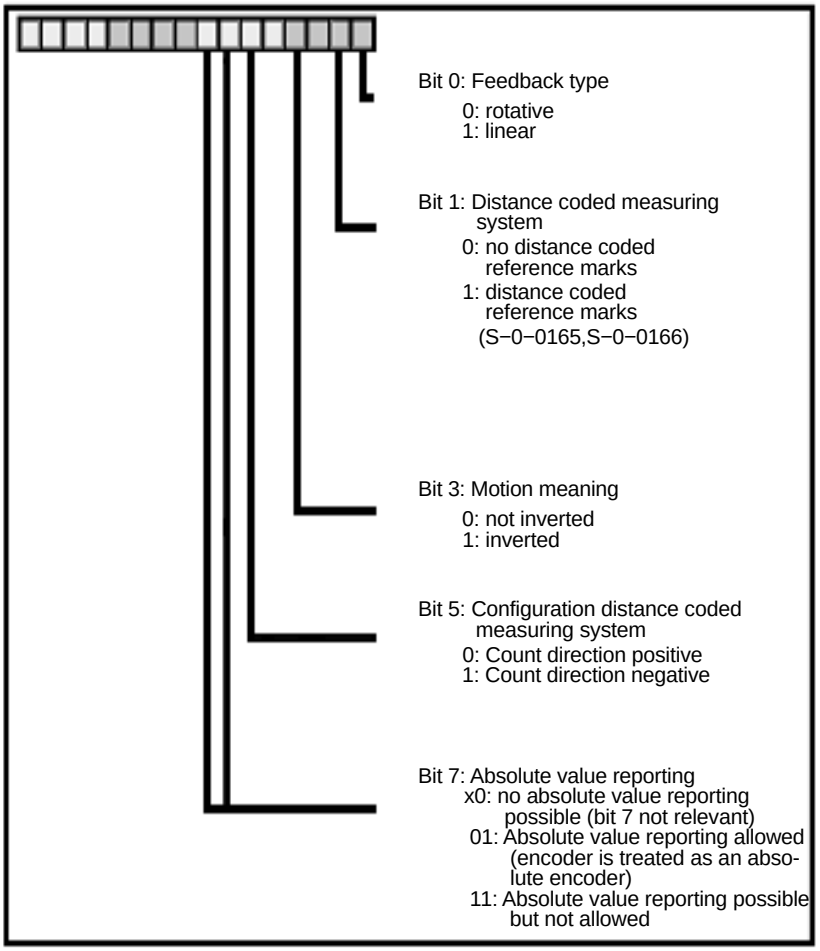
At adjusted modulo format (parameter S−0−0076, type of weighting of the position data weighting bit 7), the modulo value determines, at which numerical value the position data overflow to zero. If encoder1 is an absolute encoder, the modulo value must be adjusted greater then the traversing range of 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:

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

SERCOS interface

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:

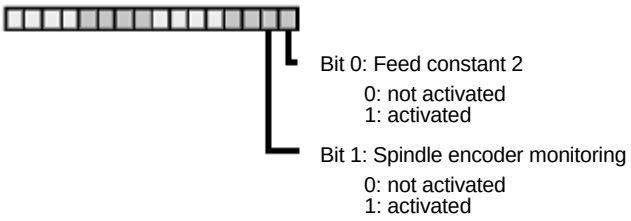
- 0 No encoder
- 1 GDS/GDM encoder from Bosch-Rexroth
- 2 Incremental encoder with sine signals (1V signals) by the company Heidenhain
- 3 Resolver with data memory
- 4 Encoder with Hiperface interface
- 5 Incremental encoder with rectangle signals by the company Heidenhain
- 6 Motor encoder of motors 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.

5.6.5 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.

SERCOS interface

5.6.6 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).

Software

6 Software

6.1 Operating system variants

In the rho4.1 variants the following basic software is installed:

- rho4.1/IPC300 Pentium III 400 MHz Windows NT4.0
- rho4.1/IPC40.2 Pentium III 933 MHz Windows XP

6.1.1 Installation and variants IPC300

The IPC300 software variants of the rho4.1 will be delivered as an image. Equal to the existing procedure, the image is copied by help of a bootdisc to the harddrive of the PC.
Available is Windows NT 4.0 in german or english. Always the actual version of the rho4-software is included.
The image CD's are only running on a P3 variant, i.e. they cannot be installed on a 266 MHz version. The disc for restoration the delivery state is suitable for Pentium III and Pentium I variants.

Software

Hardware variant	WNT De P3	WNT En P3	WNT De	WNT En	DISK INST VOXXX
rho4.1/IPC300 266 MHz	––––––	––––––	√	√	√
rho4.1/IPC300 P3 400 MHz	√	√	–––	–––––	√

☞ **The rho4.1/IPC300 with 266 MHz as well as the control panel variants rho4.1/BT155, rho4.1/BT155T and rho4.1/BT205 are no longer available.**

6.1.2 Installation and variants IPC 40.2

For Windows XP variants there is no method to restore the delivery state. Provided is a CD for basic installation of Windows XP and the update CD of the rho4 software. Due to frequently changes of the Microsoft Windows XP operating system, the provided CD normally is not equal to the delivery state.

Software

Software

	Windows XP Professional De	Windows XP Professional En
rho4.1/IPC 40.2 933 MHz	√	√

6.2 Control Panels

6.2.1 rho4.1/IPC300

rho4.1BT155/rho4.1BT205

The following devices are available for the rho4.1/IPC300 P3:

− BF315T P3
− BF315T P3 TA
− BF312T P3

The BF315T P3 TA ist built equal to the existing BF315T TA for Pentium I. Only the picture transmission was adapted to work with the P3 (400 MHz). It works not with the Pentium I (266 MHz).
For the rho4.1/IPC300 with Pentium I (266 MHz) only the BF315T TA is available.
The variants with prepared keys, resp. without prepared keys are not available for Pentium I (266 MHz) versions.

BF315T TA

BF315T **P3** TA

BF315T **P3** OTA

BF315T **P3**

6.2.2 rho4.1/IPC 40.2

The following control panels are available for the rho4.1/IPC 40.2:

Software

- VDP 16.2 12” Display
- VDP 40.2 15” Display
- VDP 60.1 12” Display

The control panels VDP 16.2 and VDP 40.2 are available with a keypad or a touchscreen. The VDP 60.1 is only available with keypad.

6.3 ROPS4/Online

ROPS4/Online permits the creation, testing and archiving of programs for the rho4 control system. The program runs under Windows 95, Windows NT or Windows XP. 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 via a serial interface or TCP/IP.
- Offline/Online points editor.
- Process functions: Process selection, process start-up, process stop etc.

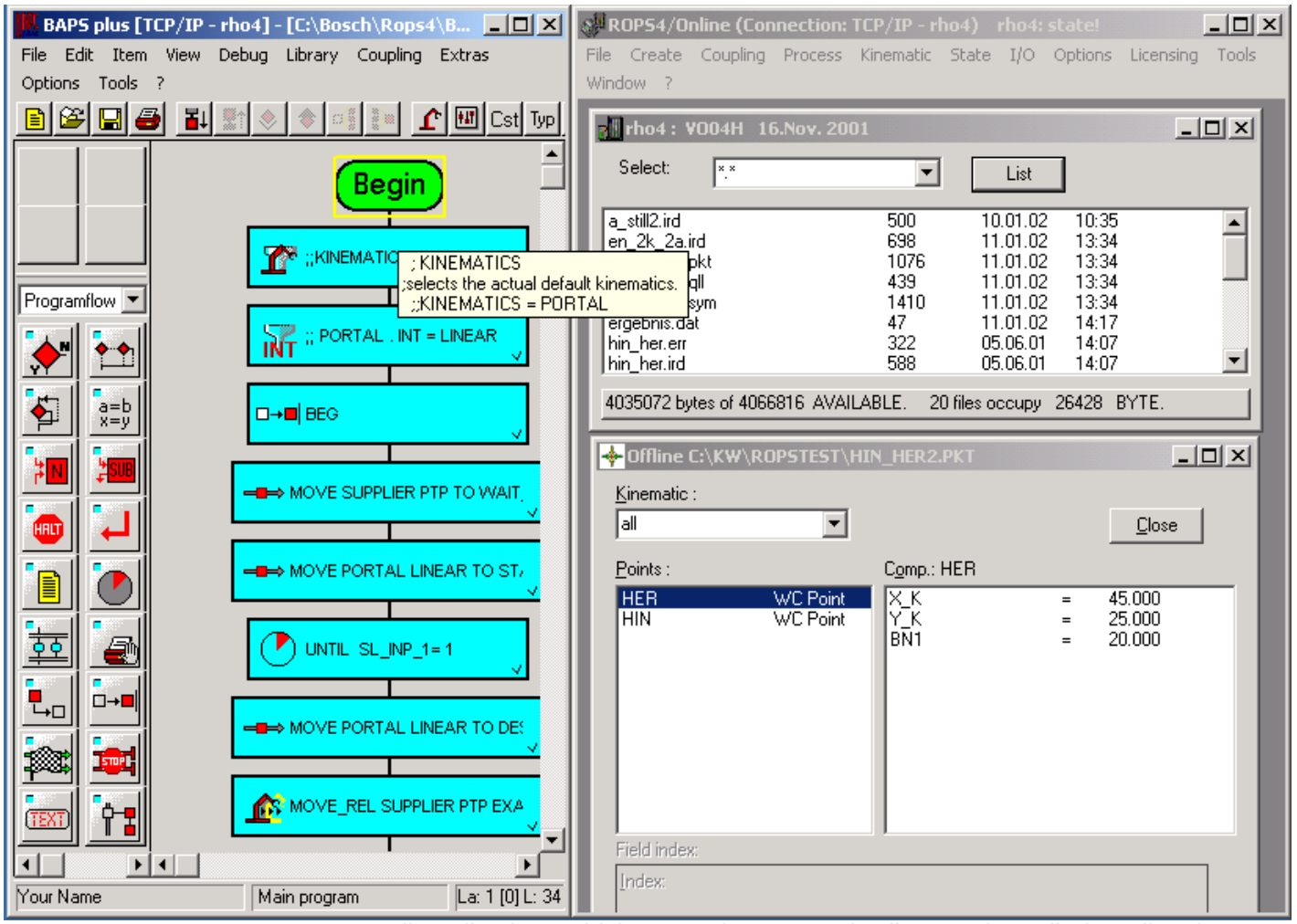
The ROPS4/Online program can be called up from the rho4 folder

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

6.4 GateWay

The GateWay program is a Windows application for Windows 95, Windows NT or Windows XP. It is used for TCP/IP communication between the rho4.1 and all external clients and servers.

IP address and port number

A TCP/IP address consists of two components, the IP address of the computer (host) and a port number. The IP address is the globally unique address of the PC. In postal terms, this would correspond to the country, city, street and house number. The port number is the number of a server or client on this PC. In postal terms, this would correspond to the name of a person living at the above-mentioned address. The port numbers 6000 to 6200 and 5000 to 5200 are reserved for Bosch Rexroth.

The rho4.1 has its own IP address (192.0.1.2). This is different to the IP address of the PC (142.2.47.11) on which it runs. The IP address of the rho4.1 is contained in the file ’hosts’ (see below).

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 address	Host name	Alias names
192.0.1.2	rho 4	rho_control
142.2.47.11	Gateway	PHG connection
142.3.0.7	PLC	PLC host computer

The file ’hosts’ is located in the Windows system directory.

rho Server

The rho4.1 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

Software

Server name	**Port number**	**Services**
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)

rho4.1 clients

The following clients are contained in the operating system of the rho4. 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

Establishing a connection

The GateWay will be started up automatically after the control system has been booted.

During initialization, the GateWay establishes a TCP/IP connection to the corresponding server for all connections set up in the GateWay.ini file. At the same time, a GateWay server branch is started up for each of these connections and is available for connecting a client. Since the GateWay runs under the Windows partition of the PC, its IP address is the IP address of the physical Ethernet card.

An external client must, therefore, set the IP address of the Windows PC and the port number of the corresponding server branch of the GateWay.

Software

An internal client must set the IP address of the rho4.1 and the port number of the rho4.1 server.
Details of the settings can be found under point 6.4.1.

An rho4.1 server which is reserved by a GateWay can no longer be contacted by another client. Each server can serve only one client.

6.4.1 GateWay.ini file

The GateWay application has one ini file, all initialization data is stored in the ASCII file (name: “GateWay.ini”). The file must be located in the same directory as the exe file (GateWay.exe). The entries of an ini file comprise a section name ([GateWay]) and the relevant codes (connection no, 1, server alias etc.). The semicolon (;) marks a comment, the text after the semicolon is not interpreted by the GateWay application.

Changes to GateWay.ini are only implemented when GateWay.exe is restarted.

Explanation of GateWay.ini

[Section name]/ Keyword		**Entry**	
[GateWay]			This section name must not be changed, it gives the initialization root.
ConnectionNo	=	n	Specifies the number of connections. The number ’n’ is variable and can be adjusted by the user. A section must be defined for each ’n’ connection.
1	=	Name	’Name’ is the section name for a connection. The ’Name’ can be freely allocated. The GateWay application uses the ’Name’ for error and status messages.
2	=	Name 2	...
3	=	Name 3	...
...	=	...	...
N	=	Name n	...
[Name]			Section name of a connection eg the connection under the keyword ’1’ in the section [GateWay].
ServerAlias	=	Host name	Symbolic host name of the server from the ’hosts’ file.
ServerPortNo	=	Number	Port number of the connection between the server and GateWay. This number must correspond with the relevant entry in the machine parameters.
GateWayPortNo	=	Number	Port number of the connection between GateWay and the client. This number must correspond with the client’s request.
Msglen	=	Number	Number is the maximum length of a record for this connection.

Software

Example of a GateWay.ini with 3 connections

```
[GateWay]                       ; fixed section name
ConnectionNo    = 3             ; 3 connections
1               = WIN_1         ; the 1st connection is for WIN_1
2               = SER_1         ; the 2nd connection is for SER_1
3               = rho_Function_1 ; the 3rd connection is for rho_Function_1

[WIN_1]                         ; section name of the 1st connection
ServerAlias     = rho4          ; symbolic name of the rho4 from the hosts file
ServerPortNo    = 6051          ; port number of the rho4 for the WIN_1
GateWayPortNo   = 6051          ; port number of the GateWay for the WIN_1
Msglen          = 256           ; message length of the TCP connection

[SER_1]                         ; section name of the 2nd connection
ServerAlias     = rho4          ; symbolic name of the rho4 from the hosts file
ServerPortNo    = 6031          ; port number of the rho4 for the SER_1
GateWayPortNo   = 6031          ; port number of the GateWay for the SER_1
Msglen          = 256           ; message length of the TCP connection

[rho_Function_1]                ; section name of the 3rd connection
ServerAlias     = rho4          ; symbolic name of the rho4 from the hosts file
ServerPortNo    = 6091          ; port number of the rho4 for the library functions
GateWayPortNo   = 6091          ; port number of the GateWay for the library functions
Msglen          = 256           ; message length of the TCP connection
```

Software

Example of a GateWay.ini with 4 connections

```
[GateWay]                       ; fixed section name
ConnectionNo        = 4         ; 4 connections
1                   = Virt_PHG  ; the 1st connection is for the virtual PHG
2                   = Virt_Panel ; the 2nd connection is for the virtual control panel
3                   = WIN_1     ; the 3rd connection is for WIN_1
4                   = SER_1     ; the 4th connection is for SER_1

[Virt_PHG]                      ; section name of the 1st connection
ServerAlias         = rho4      ; symbolic name of the rho4.1 from the hosts file1
ServerPortNo        = 6100      ; port number of the rho4.1 for the virtual PHG
GateWayPortNo       = 6100      ; port number of the GateWay for the virtual PHG
Msglen              = 156       ; this entry should only be changed following consultation
                                ; with Bosch Rexroth

[Virt_Panel]                    ; section name of the 2nd connection
ServerAlias         = rho4      ; symbolic name of the rho4 from the hosts file
ServerPortNo        = 6110      ; port number of the rho4 for the virtual control panel
GateWayPortNo       = 6110      ; port number of the GateWay for the virtual control panel
Msglen              = 6         ; this entry should only be changed following consultation
                                , with Bosch Rexroth

[WIN_1]                         ; section name of the 3rd connection
ServerAlias         = rho4      ; symbolic name of the rho4 from the hosts file
ServerPortNo        = 6051      ; port number of the rho4 for the WIN_1
GateWayPortNo       = 6051      ; port number of the GateWay for the WIN_1
Msglen              = 256       ; message length of the TCP connection

[SER_1]                         ; section name of the 3rd connection
ServerAlias         = rho4      ; symbolic name of the rho4 from the hosts file
ServerPortNo        = 6031      ; port number of the rho4 for the SER_1
GateWayPortNo       = 6031      ; port number of the GateWay for the SER_1
Msglen              = 256       ; message length of the TCP connection
```

Software

The GateWay.ini in the ’as supplied’ setting

Shown below is a the GateWay.ini in its ’as supplied’ state. This GateWay.ini has the same pre-initialisation as the machine parameters of the rho4. The section names correspond with the names in the PHG2000 menu (mode 9.1). The ’port numbers’ and ’message length’ settings are identical.

The host names are the same as those in the ’hosts’ file supplied.

The entries can be changed if this is required by other applications.

The number of connections (ConnectionNo) is 0, i. e. the GateWay is inactive.

The user must modify the ini file to suit his requirements.

Sections not used do not have any effect. They are not interpreted by the GateWay application. In this file this includes all sections except for the [GateWay] section.

```
[GateWay]                       ; fixed section name
ConnectionNo        = 0         ; no connection
1                   = Coupling  ; connection for the ROPS link

2                   = Printer   ; connection for printing
3                   = SER_1     ; connection for V24_1
4                   = SER_2     ; connection for V24_2
5                   = SER_3     ; connection for V24_3
6                   = SER_4     ; connection for V24_4
7                   = WIN_1     ; connection for read/write in BAPS
8                   = WIN_2     ; connection for read/write in BAPS
9                   = WIN_3     ; connection for read/write in BAPS
10                  = WIN_4     ; connection for read/write in BAPS
11                  = PLC       ; connection for write PLC in BAPS
12                  = PLC_Interface ; connection for cyclic transmission of the interface data
13                  = rho_Function_1 ; connection for rho4 library functions
14                  = rho_Function_2 ; connection for rho4 library functions
15                  = rho_Function_3 ; connection for rho4 library functions
16                  = rho_Function_4 ; connection for rho4 library functions
17                  = Virt_PHG  ; connection for the virtual PHG
18                  = Virt_Panel ; connection for the virtual control panel
```

Software

```
[Coupling]                      ; section name of the 1st connection
ServerAlias         = rho4      ; symbolic name of the rho4 from the hosts file
ServerPortNo        = 6010      ; port number of the rho4 for the ROPS link
GateWayPortNo       = 6010      ; port number of the GateWay for the ROPS link
Msglen              = 256       ; this entry should only be changed following consultation
                                ; with Bosch Rexroth

[Printer]                       ; section name of the 2nd connection
ServerAlias         = rho4      ; symbolic name of the rho4 from the hosts file
ServerPortNo        = 6020      ; port number of the rho4 for printing
GateWayPortNo       = 6020      ; port number of the GateWay for printing
Msglen              = 256       ; message length of the TCP connection

[SER_1]                         ; section name of the 3rd connection
ServerAlias         = rho4      ; symbolic name of the rho4 from the hosts file
ServerPortNo        = 6031      ; port number of the rho4 for V24_1
GateWayPortNo       = 6031      ; port number of the GateWay for V24_1
Msglen              = 256       ; message length of the TCP connection

[SER_2]                         ; section name of the 4th connection
ServerAlias         = rho4      ; symbolic name of the rho4 from the hosts file
ServerPortNo        = 6032      ; port number of the rho4 for V24_2
GateWayPortNo       = 6032      ; port number of the GateWay for V24_2
Msglen              = 256       ; message length of the TCP connection

[SER_3]                         ; section name of the 5th connection
ServerAlias         = rho4      ; symbolic name of the rho4 from the hosts file
ServerPortNo        = 6033      ; port number of the rho4 for V24_3
GateWayPortNo       = 6033      ; port number of the GateWay for V24_3
Msglen              = 256       ; message length of the TCP connection

[SER_4]                         ; section name of the 6th connection
ServerAlias         = rho4      ; symbolic name of the rho4 from the hosts file
ServerPortNo        = 6034      ; port number of the rho4 for V24_4
GateWayPortNo       = 6034      ; port number of the GateWay for V24_4
Msglen              = 256       ; message length of the TCP connection
```

Software

```
[WIN_1]                 ; section name of the 7th connection
ServerAlias     = rho4  ; symbolic name of the rho4 from the hosts file
ServerPortNo    = 6051  ; port number of the rho4 for read/write in BAPS
GateWayPortNo   = 6051  ; port number of the GateWay for read/write in BAPS
Msglen          = 256   ; message length of the TCP connection

[WIN_2]                 ; section name of the 8th connection
ServerAlias     = rho4  ; symbolic name of the rho4 from the hosts file
ServerPortNo    = 6052  ; port number of the rho4 for read/write in BAPS
GateWayPortNo   = 6052  ; port number of the GateWay for read/write in BAPS
Msglen          = 256   ; message length of the TCP connection

[WIN_3]                 ; section name of the 9th connection
ServerAlias     = rho4  ; symbolic name of the rho4 from the hosts file
ServerPortNo    = 6053  ; port number of the rho4 for read/write in BAPS
GateWayPortNo   = 6053  ; port number of the GateWay for read/write in BAPS
Msglen          = 256   ; message length of the TCP connection

[WIN_4]                 ; section name of the 10th connection
ServerAlias     = rho4  ; symbolic name of the rho4 from the hosts file
ServerPortNo    = 6054  ; port number of the rho4 for read/write in BAPS
GateWayPortNo   = 6054  ; port number of the GateWay for read/write in BAPS
Msglen          = 256   ; message length of the TCP connection

[PLC]                   ; section name of the 11th connection
ServerAlias     = PLC   ; symbolic name of the PLC from the hosts file
ServerPortNo    = 5110  ; port number of the rho4 for write PLC in BAPS
GateWayPortNo   = 5110  ; port number of the GateWay for write PLC in BAPS
Msglen          = 512   ; this entry should only be changed following consultation
                        ; with Bosch Rexroth

[SPS_Interface]         ; section name of the 12th connection
ServerAlias     = PLC   ; symbolic name of the PLC from the hosts file
ServerPortNo    = 5100  ; port number of the rho4 for the cyclical transmission of the
                        ; interface data
GateWayPortNo   = 5100  ; port number of the GateWay for the cyclical transmission
                        ; of the interface data
```

Software

```
Msglen          = 2048      ; this entry should only be changed following consultation
                            ; with Bosch Rexroth

[rho_Function_1]            ; section name of the 13th connection
ServerAlias     = rho4      ; symbolic name of the rho4 from the hosts file
ServerPortNo    = 6091      ; port number of the rho4 for the library functions
GateWayPortNo   = 6091      ; port number of the GateWay for the library functions
Msglen          = 256       ; message length of the TCP connection

[rho_Function_2]            ; section name of the 14th connection
ServerAlias     = rho4      ; symbolic name of the rho4 from the hosts file
ServerPortNo    = 6092      ; port number of the rho4 for the library functions
GateWayPortNo   = 6092      ; port number of the GateWay for the library functions
Msglen          = 256       ; message length of the TCP connection

[rho_Function_3]            ; section name of the 15th connection
ServerAlias     = rho4      ; symbolic name of the rho4 from the hosts file
ServerPortNo    = 6093      ; port number of the rho4 for the library functions
GateWayPortNo   = 6093      ; port number of the GateWay for the library functions
Msglen          = 256       ; message length of the TCP connection

[rho_Function_4]            ; section name of the 16th connection
ServerAlias     = rho4      ; symbolic name of the rho4 from the hosts file
ServerPortNo    = 6094      ; port number of the rho4 for the library functions
GateWayPortNo   = 6094      ; port number of the GateWay for the library functions
Msglen          = 256       ; message length of the TCP connection

[Virt_PHG]                  ; section name of the 17th connection
ServerAlias     = rho4      ; symbolic name of the rho4 from the hosts file
ServerPortNo    = 6100      ; port number of the rho4 for the virtual PHG
GateWayPortNo   = 6100      ; port number of the GateWay for the virtual PHG
Msglen          = 156       ; this entry should only be changed following consultation
                            ; with Bosch Rexroth
```

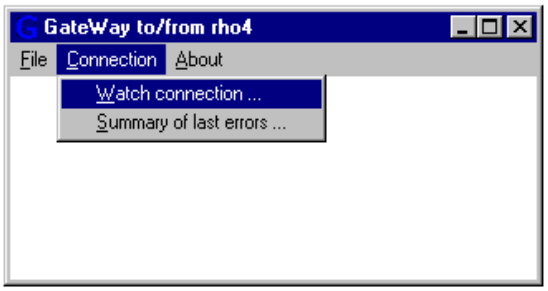
Software

```
[Virt_Panel]                    ; section name of the 18th connection
ServerAlias       = rho4        ; symbolic name of the rho4 from the hosts file
ServerPortNo      = 6110        ; port number of the rho4 for the virtual control panel
GateWayPortNo     = 6110        ; port number of the GateWay for the virtual control panel
Msglen            = 6           ; this entry should only be changed following consultation
                                ; with Bosch Rexroth
```

6.4.2 Operation of the GateWay application

The GateWay application only has a few control elements as it remains in the background during normal operation and does not need to be operated. The application can also run without a monitor and control elements.

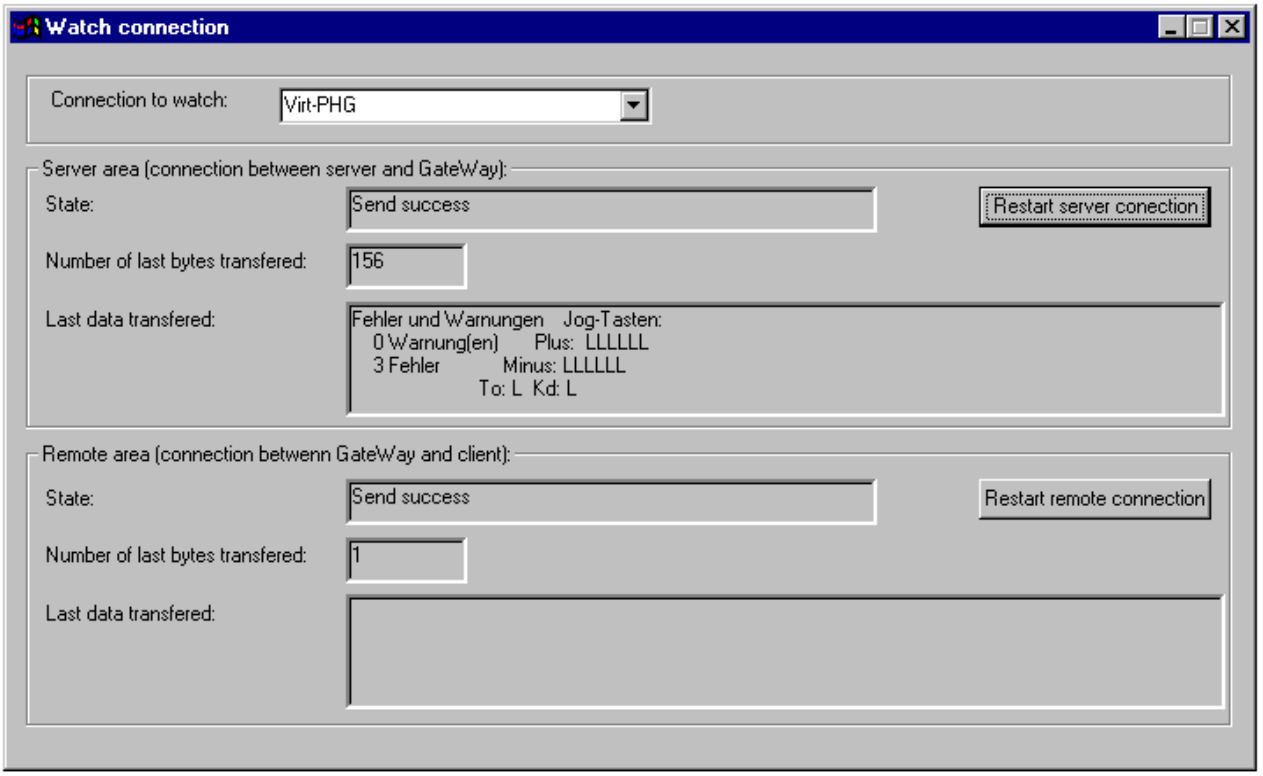
The application can be brought to the foreground and operated from there for start-up purposes or to aid troubleshooting.



Menu	**Submenu**	**Action**
File	Exit	Quit application
Connection	Watch connection	Open 'Watch connection' dialog box
	Summary of last errors	Open 'Last error messages' dialog box
About	About	Display version number, operating system and host name

Software

'Watch connection' dialog box



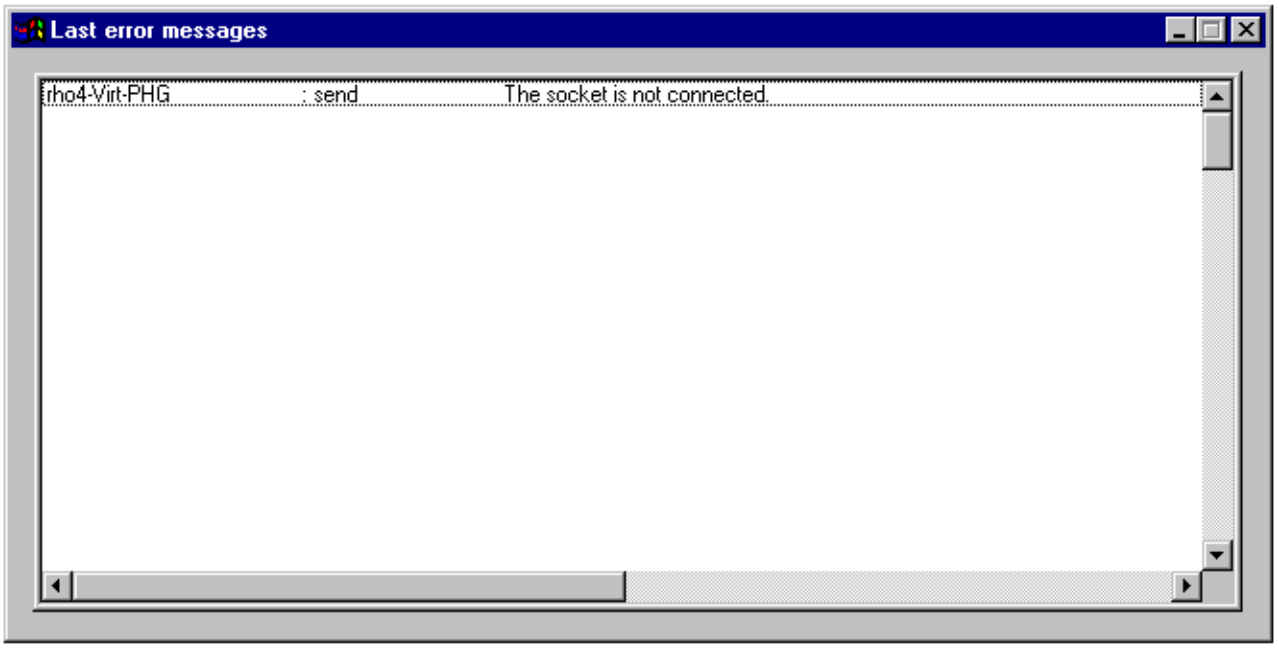
The dialog contains three areas

Area	Description
Connection to watch	The connection to be monitored and controlled can be selected here. The names in the selection box are the section names from the GateWay.ini file (see above)
Server area	The link between GateWay and server is displayed / controlled here
Remote area	The link between GateWay and client is displayed / controlled here

Option	Description
State	Current status of link
Number of last bytes transferred	The amount of data last transferred
Last data transferred	The data which was last transferred
Restart server connection	The GateWay - server connection is re-established
Restart remote connection	The GateWay - Client server branch is re-started

Software

”Last error messages” window



The dialogue displays all errors. In the form:

Host name - section name: Error text function

The example given above shows how the Virt_PHG connection has produced the error message ’The socket is not connected’ whilst sending. The ’Watch connection’ dialog box shows whether the error message relates to the server area or to the remote area.

The ’Last error messages’ window is opened automatically in the event of an error. Once the error has been eliminated, the relevant connection can be re-started using the ’Watch connection’ dialogue window. The error messages remain in the ’Last error messages’ window. They are not deleted.

Software

6.5 Start of Windows OEM applications

To give the possibility to the OEM to start automatically own applications after the start of the rho4.1, ”Winrho4” offers the function ”Call of Windows OEM applications”.This extension stage is available from the rho4.1 software package VO03D.

This function allows to avoid that the applications with the rho4 itself may be started at the same time by the autostart of the PC. The reason is that the applications communicating immediately after their call with the rho4 would produce a timeout error because the start stage of the rho4.1 is generally longer than the initialization phase of a Windows Exefile.

CAUTION
Winrho4.exe must not be closed, since not all functions of the rho4.1 are else still guaranteed (e.g. shutdown).

6.5.1 Initialization

After the rho4 has signalled that the start was correct, the OEM applications are started by the Winrho4.exe application. The order results from the entries in the Winrho4.ini.

Possible errors are numerically displayed in the window ”Watch connection”, but only if the window has been opened manually before. The allocation of error number to error text is located in the description ”Platform SDK: Win32 API / Win32 Error Codes”.

6.5.2 Winrho4.ini

The OEM applications are written in the Winrho4.ini. The Winrho4.ini must be near the Exefile Winrho4.exe in the directory c:\bosch\rho4\Winexe. An example Winrho4.ini is to be found in c:\Bosch\rho4\origin.

Syntax		Description
[Autostart]		/*The key word is fixed and may not be modified/*
NumberOfExeCalls	=	/*Number of applications that are to be started /* /*Possible entries: 0 .. 10 /*
1	=	/*Section name of the first application /*
2	=	/*Section name of the second application, etc. /*
/*for each application, the following section must be created/*		
[Application]	=	/*Section name of the application, see above /*

Software

Syntax		**Description**
Exefilename	=	/*Name and path of the Exefiles. If no path is given,/* /*Windows searches for the Exefile in the order:/* /*1. the current directory /* /*2. the Windows system directory /* /*3. the Windows directory /* /*4. the directories that are given in the /* /* Environment-Variable PATH /*
Parameter	=	/*possible parameters for the Exefile; /* /*may remain empty /*
Everyrho4Start	=	/*determines if the application is to be called at every/* /*start / restart of the rho4.1 (Everyrho4Start = 1) or/* /*only at the first start of the rho4.1 since the start /* /*of Winrho4.exe (Everyrho4Start = 0) /*

6.5.3 Example of a Winrho4.ini with the call of OEM applications

```
;;-------------------------------------------------------------------------

;;Two applications are started, section WinExe1 and WinExe2

[Autostart]

NumberOfExeCalls = 2

1                = WinExe1

2                = WinExe2

;;-------------------------------------------------------------------------

;;Section WinExe1:

;;      'Exefilename'         drive, directory and Exefilename of the application

;;      'Parameter'           no parameters

;;      'Everyrho4Start'      to be called only after the first start of the rho4.1

;;

;;The following example calls the Windows application

;;"c:\Bosch\rho4\Example\switcher\Switcher.exe".

;;Switcher.exe contains no parameters.

;;This application is only called once.

[WinExe1]

Exefilename      = c:\Bosch\rho4\Example\Switcher\Switcher.exe
```

Software

```
Parameter        =

Everyrho4Start   = 0

;;-------------------------------------------------------------------------------

;;-------------------------------------------------------------------------------

;;Section WinExe2:

;;     'Exefilename'         Exefilename of the application

;;     'Parameter'           the call parameters of the application.

;;     'Everyrho4Start'      to be called always

;;

;;The following example calls the Windows application

;;'Notepad.exe'

;;Notepad.exe contains as parameter "c:\Bosch\rho4\Example\baps\Test.dat"

;;Notepad.exe is called at each start/restart of the rho4.1.

[WinExe2]

Exefilename      = Notepad.exe

Parameter        = c:\Bosch\rho4\Example\baps\Test.dat

Everyrho4Start   = 1

;;-------------------------------------------------------------------------------
```

Software

Notes:

7 File management

The rho4 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, password protection and special backup strategies have been added.

7.1 File management function

The rho4 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 it.

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.

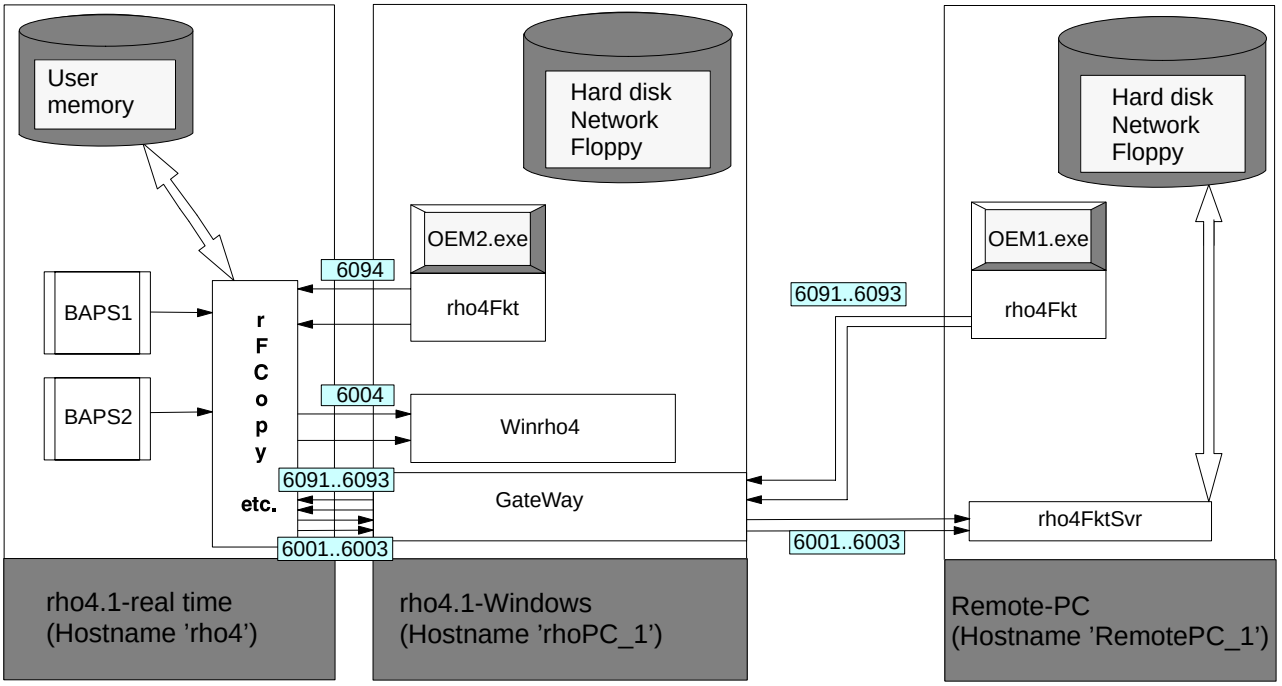
When a process is chosen the ird file is sequentialized first. If this does not succeed because no corresponding range is free, an error message is produced.

File management

7.2 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 rho4.1 has 2 classes for working memory access.



rhoFile [rF] class

The rhoFile 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 rhoFile [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.

See also the example program Switcher.exe.

File management

Using the rhoFile [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.

Four system channels (Win_Server_1 to Win_Server_4) are available for the copy functions to and from the hard disk. This means: 4 BAPS processes can access the hard disk simultaneously.

See also the example programs in BAPS.

The configuration 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

Function name	Function no.	Comment
rFxCopy()	8010	copies a file
rFxRemove()	8021	deletes a file
rFxRename()	8022	renames a file
rFReadBlk()	8030	reads a block of n bytes from a rho4 fileinto a buffer
rFDir()	8040	File list of the rho4 files in user memory
rFChmod()	8051	modifies the access rights to a file
rFxStat()	8055	returns the status of a file
rFMem()	8080	supplies the actual state of the user memory (Size, number of files etc.)

rhoArchives [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.

File management

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.

rhoArchives [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.

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. Where:
r = rho, A = Archives

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

File management

7.3 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

The system and hidden attributes have identical behaviour, but are protected by different passwords.

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 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 is started. Accordingly, hidden files will not be displayed, and the hidden attribute cannot be modified with the PHG2000.

7.4 Booting the rho4.1

When the rho4.1 is booted, a system-specific swap file is read from the hard disk. This gives the working memory and the machine parameters the same status as before shut down. The precondition for this is that the rho4.1 underwent a controlled shut down.

Shutting down the rho4.1

Shut-down is triggered from the PHG2000. The function can also be operated via a remote control input on the interface if the system is not equipped with a PHG2000.

All the contents of the working memory and the machine parameters are swapped out to the hard disk when the controller is shut down. The swap file is managed in the rho4.1 and cannot be read by DOS or Windows tools.

UPS

The rho4.1 is fitted with an uninterruptible power supply (see manual 'Rexroth Rho 4.1, Rho 4.1/IPC300 Connectivity Manual', resp. 'Rexroth Rho 4.1/IPC 40.2 Connectivity Manual'). This triggers a shut-down if there is a loss of power. This ensures that data consistency is maintained, even in the event of a mains power outage.

File management

7.5 Support of hard disks > 1,5 GB (only rho4.1, rho4.1/IPC300)

From rho4-SW version VO03D, hard disks > 1,5 GB are used in the rho4.1. This leads to the following restrictions:

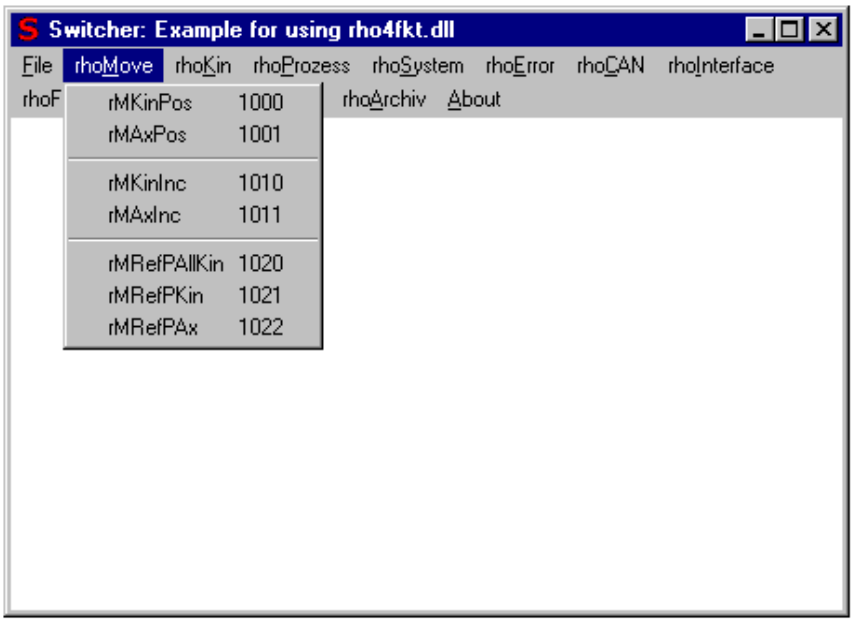
- The downgrade from version 3 to version 2 or 1 is not possible.
  Reason: after the partitioning, the CD-ROM drive gets the letter E:. The images of the preceding versions expect however the CD-ROM drive at D:.
  Remedy: edit the Autoexec.bat “DISK INST VOXXX”. The drive letter D: must be replaced by O:.
- The restore of a Windows95 version via a NT version is not possible.
  Reason: if a NT version is booted with W95-DOS, there cannot be accessed to the hard disk.
  Remedy: Restore the Windows95 delivery state twice

Delivered hard disks and formats

Disk size	LW C:\FAT32	LW D:\FAT32	CD-ROM
1,5 GB	1,51 GB	not existing	Drive D:
2,1 GB	1,51 GB	0,6 GB	Drive E:
3,0 GB	1,51 GB	1,5 GB	Drive E:
XY GB	1,51 GB	XY−1,51 GB	Drive E:

7.6 Switcher

The example Switcher.exe shows how the rho4 library functions are linked.



File management

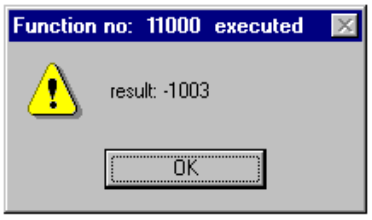
All source and header files, and the project for MS Visual C/C++ Version 4.1, are included in the scope of supply.

☞ **See also DLL library manual**

A comprehensive description of the functions and their parameters is included in the appropriate header files.

A function is executed immediately it is selected through the menu. No messages or prompts will be issued.

The Switcher.exe application only shows the return code for the function concerned.



In the above example, this means that function 11000 returned Result −1003.
11000 is the rTClientCon() function. It is described in file rT.h. The return codes are also described here. The associated source file is swi_rt.c

Creating the example Switcher.exe

This project was created using MS-Visual C/C++ Version 4.1.

The following files belong to the Switcher:

- switcher.c: WinMain()
- switcher.rc: Resources
- switcher.ico: Icons
- resource.h: Resource includes
- swi_rm.c: 1000 class function calls
- swi_rk.c: 2000 class function calls
- swi_rp.c: 3000 class function calls
- swi_rs.c: 4000 class function calls
- swi_re.c: 5000 class function calls
- swi_rc.c: 6000 class function calls
- wi_ri.c : 7000 class function calls
- swi_rf.c: 8000 class function calls
- swi_rmp.c: 9000 class function calls
- swi_rt.c: 11000 class function calls

File management

- swi_ra.c: 14000 class function calls
- swi_rmxg.c: 19000 class function calls
- swi_rmxs.h: 20000 class function calls
- r_imp.h: Import statement for C and C++ programs
- rmain.h: Main types for all functions
- rm.h: Types and descriptions for the 1000 class
- rk.h: Types and descriptions for the 2000 class
- rp.h: Types and descriptions for the 3000 class
- rs.h: Types and descriptions for the 4000 class
- re.h: Types and descriptions for the 5000 class
- rc.h: Types and descriptions for the 6000 class
- ri.h : Types and descriptions for the 7000 class
- rf.h: Types and descriptions for the 8000 class
- rmp.h: Types and descriptions for the 9000 class
- rmps.h: Types and descriptions for the 10000 class
- rt.h: Types and descriptions for the 11000 class
- ra.h: Types and descriptions for the 14000 class
- rMxG.h: Types and descriptions for the 19000 class
- rMxS.h: Types and descriptions for the 20000 class

Additional DLL

rho4Fkt.dll: DLL for rho4 library functions

The DLL is stored in the \Windows\System folder. A copy is stored in the \bosch\rho4\rho4Fkt folder.

Compiler switches

- Warning level = 3
- Optimizations = Disable (Debug)
- Debug Info = Program Database
- Preprocessor definitions = WIN32, _DEBUG, _WINDOWS
- Processor = Blend *
- Calling convention = _cdecl *
- Use run-time library = debug Single-threaded
- Struct member alignment = 4

Total of all compiler switches

/nologo /Zp4 /MLd /W3 /Gm /GX /Zi /Od /D ”WIN32” /D ”_DEBUG” /D ”_WINDOWS” /FR”Debug/” /Fp”Debug/switcher.pch” /YX /Fo”Debug/” /Fd”Debug/” /c

File management

Linker switches

- Object / library modules = kernel32.lib user32.lib rho4fkt.lib
- Generate Debug Info = Microsoft Format

Total of all linker switches

kernel32.lib user32.lib \bosch\rho4\rho4fkt\rho4fkt.lib /nologo /subsystem:windows /incremental:no /pdb:”Debug/switcher.pdb” /debug /machine:IX86 /out:”Debug/switcher.exe”

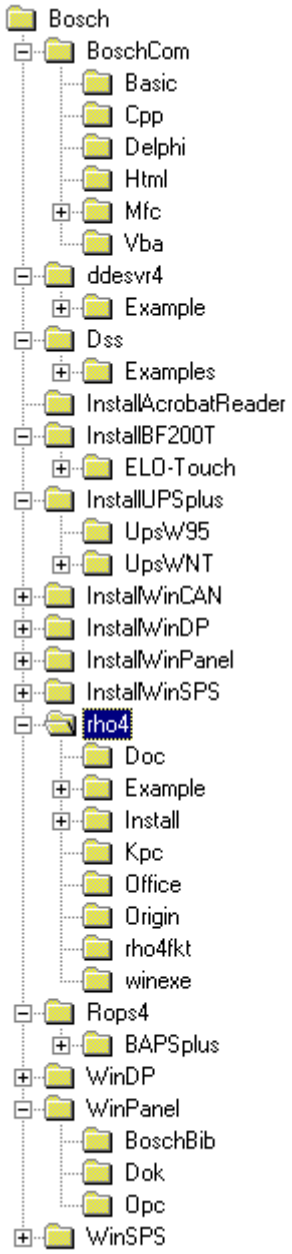
File management

Notes:

8 Scope of rho4.1 software

8.1 Hard disk folder structure

☞ **By new developments and enhancements (hard- and software) the represented folder structure and its contents may vary. That applies to programs and whose files which are decribed in this chapter.**



Principal structure of the hard disk

Scope of rho4.1 software

8.1.1 Specimen list of files

Where it is desirable for the rho4.1 user to be able to read, modify or call the files, these files are listed individually or as examples. All other files belong to the rho4.1 software package and must not be read, modified or called by the user.

\Bosch\BoschCom folder

This is the root folder of the software interface BoschCom enabling the open communication with ethernet-capable control units (e.g. PLC). In the Word document ‘BoschCom_Ext.doc’, the supplied example programs are described.

Subfolder	Description	Comments
Basic	Program files for an example with Microsoft Visual Basic	
Cpp	Program files for an example with Microsoft Visual C++	
Delphi	Program files for an example with Delphi	
Html	Contains example program in HTML format	
Mfc\TestCom	Contains files of the Microsoft Foundation Classes	
Vba	Contains example program for Microsoft Excel	

Bosch\DDESvr4 folder

This is the root folder for the rho4 DDE server. The document ’rho4-DDServer4Verxxx.doc’ stored in C:\Bosch\DDESvr4 provides an exhaustive description of the server and the files of relevance to the user. The subfolder ’Example’ contains example programs for the communication of the DDE server with Windows programs.

\Bosch\Dss\Examples\rho4 folder

In this folder there are the example configuration files for SERCOS and CANopen.

\Bosch\InstallAcrobatReader folder

In this folder there are the installation files (German and English) of the Acrobat Reader, version 5.05. The Acrobat Reader is required to open the rho4 manuals (PDF files) located in the folder Bosch\rho4\doc.

Scope of rho4.1 software

\Bosch\InstallBF200T\ELO-Touch folder (only rho4.1, rho4.1/IPC300)

In the subfolders MMWin95 or MMWinNT there are the installation files for the Touch Screen of the control panels BF200T and BF212T. Windows 95 and Windows NT 4.0 are supported.

C:\Support\Touch folder (only rho4.1/IPC 40.2)

In the subfolders DMC or Elo there are the installation files for the Touch Screen of the control panels VDP 16.2 and VDP 40.2. Windows XP is supported.

\Bosch\InstallUPSPlus folder (only rho4.1, rho4.1/IPC300)

In the subfolders UpsW95 and UpsWNT there are the installation files for the UPS. Windows 95 and Windows NT 4.0. are supported. An installation of this software is not required in general since the UPS software is automatically installed during an update or a new installation of the rho4 software.

C:\Support\UPS folder (only rho4.1/IPC 40.2)

In this subfolder there are the installation files for the UPS. Windows XP is supported. An installation of this software is not required in general since the UPS software is automatically installed during an update or a new installation of the rho4 software.

\Bosch\InstallWinCAN folder

In this folder there are the installation disks of the configuring software for the setting of CAN-I/O modules etc. at the PCI_BM-CAN (CAN field bus) of the PCL-SoftSPS.

\Bosch\InstallWinDP folder

In this folder there are the installation disks of the configuring software for the setting of PROFI-Bus-DP I/O modules etc. at the PCI_BM-DP (PROFI-Bus) of the PCL-SoftSPS.

\Bosch\InstallWinPanel folder

In this folder there are the installation disks of WinPanel. An installation of this software is not required in general since the WinPanel is automatically installed during an update or a new installation of the rho4 software.

\Bosch\InstallWinSPS folder

In this folder there are the installation disks of the programming tool WINSPS required for the programming of the PCL-SoftSPS.

Scope of rho4.1 software

\Bosch\rho4 folder

This is the root folder of the robot and handling control rho4. It contains subfolders and a Readme file on the rho4 version, e.g. 'Rho4 new or different in VO03G Ver101.doc'.

Subfolders:

Subfolder	Description	Comments
Doc	Rho4 manuals in PDF format	To open these files, the Acrobate Reader must be installed.
Example	Examples: a) BAPS programs b) use of rho4 library functions c) PHG2000-Projects	To load the PHG projects, the BDT editor software must be installed
Install	Installation files	
	eWeb-Project files	Folder includes all files of the Jbuilder project.
KPC	System folder for the rho4.1	These files must not be edited with Windows or DOS tools.
Office	Applications in the Office version	Operation of the der rho4.1 using a virtual control panel and virtual PHG
Origin	bnr, ini, dll and link files in the original state in which they were supplied	This allows the files to be restored to the original state in which they were supplied if they are inadvertently lost. This only applies to files contained within this folder.
rho4Fkt	Files for rho4.1 library functions	
Winexe	Windows.exe files	

Files:

Filename	Description	Comments
Rho4-New or different in VO03D Ver101.doc	Readme file containing new features and changes in the current rho4 version in over the previous version	Text file which can be opened in Notepad, for example.

Scope of rho4.1 software

\Bosch\rho4\Example folder

This folder contains only subfolders in which sample BAPS programs and examples of links to the rho4 library functions and PHG2000 projects may be found.

Subfolder	Description	Comments
BAPS	Sample BAPS program	Collection of a variety of different BAPS programs
Einrich	C++ sample for the use of rho4 library functions. Specifically, rhoTCP, rhoKin and rhoMove	Includes all source files. Developed using MS Developer Studio. The C:\Bosch\rho4\ExampleEinrich\Help subfolder contains a Help file in which the C++ program description for this sample application can be found.
LSWIN_x	C++-example for READ/WRITE WIN channel	Including all source files Development under the MS-Developer Studio In the subfolder C:\Bosch\rho4\Example\LSWIN_x\Help there is a help file with the C++ program description for this sample application In the subfolder BAPS or BAPSplus there are the BAPS programs for this application
Phg2000	Example projects, file list and process info for the PHG2000	To load the projects into the PHG2000, the BDT editor software must be installed
switcher	C sample for the use of rho4 library functions	Includes all source files. Developed using MS Developer Studio. This contains the source code for the correct way of calling each rho4 library function.
updownld	C++ sample for the use of rho4 library functions. Specifically, rhoTCP, rhoArchive	Includes all source files. Developed using MS Developer Studio. The C:\Bosch\rho4\Example\updownld\Help subfolder contains a Help file in which the C++ program description for this sample application can be found.
VbSwitcher	Visual Basic sample for the use of rho4 library functions	Includes all source files. Developed using Visual Basic. This contains the source code for the correct way of calling each rho4 library function.

Scope of rho4.1 software

\Bosch\rho4\Example\BAPS folder

This folder contains sample BAPS programs. No claim of completeness is made. Sample programs are not provided for all BAPS keywords.

☞ **No liability shall be accepted for correct programming, function etc**

Should you need to know the correct programming syntax for a particular BAPS keyword, you can search all *.qll files in this subfolder for the relevant keyword using, for example, Windows Explorer.

Filename	Description	Comments
Rf02010.qll	Sample BAPS program for the 2010, rKGInfo rho4 library function	Text file
Rf03012.qll	Sample BAPS program for the 3012, rPGExtData rho4 library function	Text file
Rfyyyyy.qll	Sample BAPS program for the yyyyy, rxyz rho4 library function	Text file
iiiijjjj.qll	iiiijjjj.qll are general sample BAPS programs of the correct syntax for a great variety of BAPS keywords	Text files

\Bosch\rho4\Example\Einrich folder

This folder contains a sample C++ program for the use of rho4 library functions. Specifically, the rho4 library functions of the rhoTCP, rhoKin and rhoMove groups are used here. All source files are included.

This sample application has been developed using MS Developer Studio version 4.1.

The C:\Bosch\rho4\Example\Einrich\Help subfolder contains a Help file in which the C++ program description for this sample application can be found.

☞ **No liability shall be accepted for correct programming, function etc**

Scope of rho4.1 software

Filename	Description	Comments
*.cpp	C++ source files	See 'C:\Bosch\rho4\Example\Einrich\Help\ EINRICHTEN.HLP' for a description
*.h	Header files	See 'C:\Bosch\rho4\Example\Einrich\Help\ EINRICHTEN.HLP' for a description
Einrichten.exe	Executable exe file of this sample application	MS Developer Studio release
*.aps, *.clw, *.mdp etc.	Other files for this application	Some files are generated automatically when a project is created in MS Developer.

\Bosch\rho4\Example\Switcher folder

Contains a sample of how to link each rho4 library function into a C program.

☞ **No liability shall be accepted for correct programming, function etc**

Scope of rho4.1 software

Filename	Description	Comments
switcher.exe	Application for the use of library functions.	
switcher.c	WinMain()	
switcher.rc	Resources	
switcher.ico	Icons	
swi_rm.c	1000 class function calls, rhoMove, motion functions in manual mode	
swi_rk.c	2000 class function calls, rhoKin, movement system information such as axis position etc.	
swi_rp.c	3000 class function calls, rhoProcess, process information and functions such as process status etc.	
swi_rs.c	4000 class function calls, rhoSystem, system information and functions	
swi_re.c	5000 class function calls, rhoError, status messages and warnings	
swi_ri.c	7000 class function calls, rhoInterface, reading and setting I/Os in the interface	
swi_rf.c	8000 class function calls, rhoFile, file functions such as: rFCopy, copy file	
swi_rmpg.c	9000 class function calls, rhoMPGet, read rights on the machine parameters	
swi_rmps.c	10000 class function calls, rhoMPSet, write access to the machine parameters	
swi_rt.c	11000 class function calls, rhoTCP, establish and drop a TCP/IP connection	
swi_rmxg.c	19000 class function calls, rhoMxGet, reading access to the machine parameters	
swi_rmxs.c	20000 class function calls, rhoMxSet, writing access to the machine parameters	
swi_ra.c	rhoArchive class function calls. File functions over the coupling interface	
*.h, *.mak, *.mdp etc.	Other files in the application	

Scope of rho4.1 software

\Bosch\rho4\Example\Updownld folder

This folder contains a sample C++ program for the use of rho4 library functions. Specifically, the rho4 library functions of the rhoTCP and rho-Archive groups are used. All source files are included.

This sample application has been developed using MS Developer Studio version 4.1.

☞ **No liability shall be accepted for correct programming, function etc**

The C:\Bosch\rho4\Example\updownld\Help subfolder contains a Help file in which the C++ program description for this sample application can be found.

Filename	Description	Comments
*.cpp	C++ source files	See 'C:\Bosch\rho4\Example\Updownld\Help\ Updownld.hlp' for a description
*.h	Header files	See 'C:\Bosch\rho4\Example\Updownld\Help\ Updownld.hlp' for a description
Updownld.exe	Executable exe file of this sample application	MS Developer Studio release
*.aps, *.clw, *.mdp etc.	Other files in this application	Some files are generated automatically when a project is created in MS Developer.

\Bosch\rho4\Install\VxW95Drv or VxWinDrv or VxWXPDrv folder

Installation tools, for Bosch only. No user access.

\Bosch\rho4\Kpc folder

rho4 binary files, for Bosch only. No user access.

\Bosch\rho4\Office folder

This is the root folder for the rho4 Office version applications.

The rho4 can be operated via a virtual control panel simulating the I/O, and a virtual PHG simulating the real PHG, almost fully as if one were using the real I/O and the real PHG. Exceptions to this are, for example, teaching and defining in PHG2000 mode.

Scope of rho4.1 software

Filename	Description	Comments
VirtInterface.exe	Executable exe file for the virtual control panel	Virtual control panel. This application assumes that the default PLC pass-through program is running in the PLC and that there are no real remote I/O modules applied in the rho4. See C:\WinSPS\Rho4_xx.prj\ SoftSPS.pcl\zs0. xx means the characterization of the rho4-SW version, e.g. 3D (VO03D)
VirtInterface.Ini	ini file for VirtInterface.exe	
VIRTINTERFACE.HLP	Helpfile for the virtual control panel	
VirtPHG.exe	Executable exe file for the virtual PHG	Virtual PHG This application assumes that there is no real PHG connected to the rho4.
VirtPHG.Ini	ini file for VirtPHG.exe	
Virtphg.hlp	Helpfile for the virtual PHG	

\Bosch\rho4\Origin folder

This folder contains original rho4 files in the ’as-supplied’ state.

This allows the files to be restored to the ’as-supplied’ state if they are inadvertently lost. This only applies to files contained within this folder.

A copy of almost all these files incorporating any settings changed by the user can be found in other subfolders.

Scope of rho4.1 software

Filename	Description	Comments
*.pif	Shortcuts which can be found in the Rho4 folder on the Desktop.	
*.dll	DLL files which are copied to the Windows\System folder during installation.	
*.lib	Library files for the *.dll files mentioned above	
*.ini	Initialization files in the original state in which the rho4 was supplied	
*.bin	Bosch rho4 system file	This file must not be edited with Windows or DOS tools. ☞**Use only in consultation with and under the direction of Bosch**
hosts	File for the assignment of IP addresses and logical names. The file is copied to the Window folder during the initial installation.	The file can be expanded by the user in the Windows folder. However, the entry relating to ’rho4’ must remain unaltered.
Sta_srca.bnr	Filter file for the SR-CAN module	For details, see chapter 4 section 4.5 SRCAN module
Sta_tast.bnr	Standard key assignment table for the PHG2000	For details, see rho4 manual ‘PHG2000 Software manual’, chapter ‘Variable assignment of the PHG keys’
*.bix *.exe *.ocx	Bosch rho4 system files Files for the registration of the Bosch software, e.g. ROPS4 Files for the registration of the Bosch software, e.g. ROPS4	This file must not be edited with Windows or DOS tools. ☞**Use only in consultation with and under the direction of Bosch**

Scope of rho4.1 software

\Bosch\rho4\rho4Fkt folder

This folder contains the Header, Library, .DLL and inc files for the use of rho4 library functions.

Filename	Description	Comments
rho4Fkt.dll	DLL for rho4 library functions	This DLL contains all functions except the rhoArchive group and their TCP functions for rhoArchive.
rho4Fkt.lib	Import library for dll function of rho4 library functions	
rops4fkt.dll	DLL for ROPS4 library functions	This DLL contains the functions of the rhoArchive group and their TCP functions for rhoArchive.
rops4fkt.lib	Import library for the dll functions of ROPS4 library functions	
mpparser.dll	DLL for machine parameter converter functions	
mpparser.lib	Import library for the dll functions of machine parameter converter functions	
Ra.h	14000 class functions, rhoArchive	
Ra_err.h	Constants declaration file for the status and warning messages of class 14000 functions, rhoArchive	
Re.h	5000 class functions, rhoError	
Ri.h	7000 class functions, rhoInterface	
Rimp.h	dll import statement	Must always be included.
Rk.h	2000 class functions, rhoKinematic	Includefile for Windows application.
Rm.h	1000 class functions, rhoMove	Includefile for Windows application.
Rmain.h	Main agreements for all functions	Includefile for Windows application. Must always be included.
Rmpg.h	9000 class functions, rhoMpGet	Includefile for Windows application.
Rmps.h	10000 class functions, rhoMpSet	Includefile for Windows application.
Rmxs.h	20000 class functions, rhoMxSet	Includefile for Windows application.
Rp.h	3000 class functions, rhoProcess	Includefile for Windows application.
Rs.h	4000 class functions, rhoSystem	Includefile for Windows application.
Rt.h	11000 class functions, rhoTCP	Includefile for Windows application.
*_En.h	English version of the corresponding Header file	Includefile for Windows application.
Re.inc	5000 class functions, rhoError	Includefile for BAPS program
Rf.inc	8000 class functions, rhoFile	Includefile for BAPS program
Ri.inc	7000 class functions, rhoInterface	Includefile for BAPS program
Rk.inc	2000 class functions, rhoKinematic	Includefile for BAPS program

Scope of rho4.1 software

Filename	Description	Comments
Rmain.inc	Main agreements for all functions	Includefile for BAPS program , must always be included
Rmpg.inc	9000 class functions, rhoMpGet	Includefile for BAPS program
Rmxg.inc	19000 class functions, rhoMxGet	Includefile for BAPS program
Rmps.inc	10000 class functions, rhoMpSet	Includefile for BAPS program
Rmxs.inc	20000 class functions, rhoMxSet	Includefile for BAPS program
Rp.inc	3000 class functions, rhoProcess	Includefile for BAPS program
Rs.inc	4000 class functions, rhoSystem	Includefile for BAPS program
Rt.inc	11000 class functions, rhoTCP	Includefile for BAPS program
*_En.inc	English version of the corresponding includefile	Includefile for BAPS program

\Bosch\rho4\Winexe folder

This folder contains the executable exe files and their initialization files.

Filename	Description	Comments
Winrho4.exe	Windows part of the rho4 Connection between Windows and the real-time part of the rho4	This is automatically invoked if Autostart is activated when the system is switched on.
Winrho4.ini	Initialization file for Winrho4.exe	
GateWay.exe	TCP/IP connection between the Windows part of the rho4 to other PCs	Required, for example, if ROPS4 is running on an external laptop and a connection is to be established over the TCP/IP Ethernet to the rho4
GateWay.ini	Initialization file for GateWay.exe	In this file the number, type, IP addresses and port numbers of the channels to rho4 are fixed.
Crt121.exe	Executable driver file for the control panels BF200T and BF212T	For details, see comment in the file C:\Autoexec.bat
rho4FktSvr.exe	Server for rho4 functions	See manual 'DLL-library', chapter Filesystem rhoFile [rF] 8000
rho4FktSvr.ini	Ini-file for the rho4Fct server	See manual 'DLL-library', chapter Filesystem rhoFile [rF] 8000

Scope of rho4.1 software

\Bosch\Rops4 folder

This is the root folder of ROPS4.

Filename	Description	Comments
Editor.ini	Initialization file for the editor	
mp_bin_to_ascii.exe	Executable file of the machine parameter converter BIN −> ASCII	Creates from a binary machine parameter file (.bin) of rho4 a readable file in ASCII format
mp_ascii_to_bin.exe	Executable file of the machine parameter converter ASCII −> BIN	Creates from a readable machine parameter file in the ASCII format a loadable binary machine parameter file for the rho4
ROPS4-New or changed in X.y.txt	Short description of the new functions and modifications in the ROPS4 version X.y	
Online.exe	Executable file for ROPS4	
Rops4win.ini	Initialization file for ROPS4	

The folder \Bosch\Rops4 also contains the subfolder for the graphical programming and testing system BAPS plus.

Subfolder	Description	Comments
BAPSplus	Root folder of the graphical programming and testing system BAPS plus.	

\Bosch\Rops4\BAPSplus folder

This is the root folder of BAPS plus, the graphical programming and testing system of the rho4.

Subfolder	Description	Comments
BAPSplus	Root folder of the graphical programming and testing system BAPS plus.	

It contains subfolders with the following purposes:

Scope of rho4.1 software

Subfolder	Description	Comments
Bibl	Libraries	Definition of the BAPS instructions. The user must store user-defined libraries, including any new user-defined commands, in this subfolder
Code	*.ird-, *.qll-, *.sym-, *.pkt- and *.err files	Files created by the BAPS translator.
Dat	BAPS plus files	These files have the extension opd
Vorlagen	Sample templates	Program templates. May be used as a basis under 'File, New'.

\Bosch\WinPanel folder

This is the master folder of the WinPanel. It contains files of the PCL and the WinPanel.

\Bosch\WinSPS folder

This is the master folder of the WinSPS.

\Bosch\WinSPS\r4En_3D.prj\r4En_3D.PCL\Zs0 folder

This folder contains the files for the PCL standard pass-through program of the rho4. The ending '3D' refers to the rho4 version (here VO03D), from which these programs are available.

Filename	Description	Comments
Cande_1f.pxo	Program module for ● reading out an analog input of the electronic type plate ● writing on two BAPS-INTEGER-inputs	A BAPS program suitable for this program module is to be found under: C:\Bosch\rho4\Example\BAPS\Lesecan.qll
Datde_1f.pxd	Data module for data required from BAPS	WRITE/READ PLC
Exjde_1f.pxo	Program module for external Jog mode (Jog)	The module is activated by removing the commentary character ';' in the program module Om1_1d.pxo
Read_2a.pxd	Data module for READ PLC	A BAPS program suitable for this program module is to be found under: C:\Bosch\rho4\Example\BAPS\Spslese.qll
Read_2a.pxo	Program module for READ PLC	A BAPS program suitable for this program module is to be found under: C:\Bosch\rho4\Example\BAPS\Spslese.qll
Om1_3d.pxo	Organization module 1, PCL main program	
Om2_1f.pxo	Organization module 2, initialization module of the PCL	

Scope of rho4.1 software

Filename	**Description**	**Comments**
Om18_1f.pxo . . Om25_1f.pxo	Organization module 18..25, time-controlled processing for rho4 time task 1..8	
Pm001_1f.pxo . . Pm024_1f.pxo	Program module 1..24 for BAPS-instruction:START/STOP PLC_PROCESS (program number)	A BAPS program suitable for this program module is to be found under: C:\Bosch\rho4\Example\BAPS\Strtstop.qllPLC_PROCESS (program number) runs as cyclic task
Pm025_1f.pxo	Program module 25 for BAPS-instruction: PLC_PROCESS (program number)	A BAPS program suitable for this program module is to be found under: C:\Bosch\rho4\Example\BAPS\Spsproz.qllPLC_PROCESS (program number) runs only once
r4ene_3d.pxo	rho4 standard pass-through program	The ending ’3d’ of the filename stands just as in all program organization data modules etc. for the version ’3d’ (VO03D).
r4en_3d.sxs	rho4-symbol file It contains all used program modules and signal declarations	
r4en_3do.sxs	rho4-symbol file It contains all program modules just as ’r4en_3d.sxs’ with the modification that the signal declarations are determined for a profibus DP application.	The signals are shifted for 512 bytes.They start with I512 and O512
Wr_2a.pxd	Data module for required data for the BAPS-instructionWRITE PLC ...	A BAPS program suitable for this program module is to be found under: C:\Bosch\rho4\Example\BAPS\Spswrit2.qll
Sysde_1f.pxo	Program module for the transfer of the PCL system statuses onto the rho4	
Time1_1f.pxo . . Time8_1f.pxo	Program module 1..8 for PCL time task 1..8	
Timev11f.pxo . . Timev81f.pxo	Program modules 11,21,31...81 for the time value transfer for time task 1..8	A BAPS program suitable for this program module is to be found under: C:\Bosch\rho4\Example\BAPS\Spszeita.qll

Scope of rho4.1 software

8.2 Software

Software included in the basic package of the rho4.1:

- Software PLC, PC-PCL configuration level, without PCI bus card
- Windows 95, Windows NT 4.0, Windows XP
- ROPS4/Online (installed)
- DDE Server 4 (installed)
- WinPanel (installed and licensed)
- Winrho4 (installed and licensed)
- UPS-Software (installed, no licence required)
- WinSPS (installation files)
- WinDP (installation files)
- Sample programs
- Documentation
- Applications
  - virtual PHG
  - virtual Interface
  - Switcher
  - rho4 fct server

8.2.1 Licences

Licences for the use of the software on additional PC may be obtained for:

- ROPS4/Online
- DDE-Server4
- WinSPS
- WinDP
- BoschCom

Scope of rho4.1 software

Notes:

Appendix

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

A
Alias names, 2−27
Analog I/O, 4−13, 4−14
  SR−CAN modules, 4−32
Analog I/O Parameter (P404−P407), 4−21
Assignment of the CAN−Bus interfaces, 4−39
Axis telegram, 5−3

B
B~IO M−CAN, 2−23
B~IO M−CAN bus connection, 2−25
B~IO M−CAN Bus switching, 4−13
B~IO M−CAN modules, 4−13
B~IO M−DP, 2−23
B~IO M−DP Profibus connection, 2−24
B~IO M−IBS, 2−23
Belt input via external encoder at EcoDrive/Indra-Drive, 5−19
Betriebssystem Varianten, 6−1
Booting the rho4, 7−7

C
CAN belts, 4−9
CAN−ID assignments, 4−43
CANopen, 2−8, 2−23
CANopen encoder, 4−4
CANopen protocole, 4−34
CANopen−Interface, 4−1
CANrho, 2−8
CANrho protocole, 4−34
Chat, 2−53
Control Panels, 6−2
Cyclic axis telegrams, 5−13
Cyclic drive data, 5−13

D
Data channel, 3−4
DESI65K−CAN, 4−43
Documentation, 1−7
Drive−lead referencing, 5−9

E
EA−type, 2−18
EMC Directive, 1−1
EMERGENCY−STOP devices, 1−5
ESD
  Electrostatic discharge, 1−6
  grounding, 1−6
  workplace, 1−6
ESD−sensitive components, 1−6
Example of a B~IO M−CAN bus connection, 2−25
Example of a B~IO M−IBS bus connection, 2−25

F
File ’hosts’, 2−36, 6−5
File attributes, 7−6
File management, 7−1
Floppy disk drive, 1−7

G
GateWay, 6−5
GateWay.ini, 2−55
GateWay.ini file, 6−7
Grounding bracelet, 1−6

H
Hard disk drive, 1−7
Hub, 2−29

I
I/O modules, 4−1
I/O−Gateway, 2−6, 2−23
Interbus S, 2−23
IP address, 6−5
IP addresses, 2−27

L
Licences, 8−17
Low−Voltage Directive, 1−1

M
Machine parameter backup, 4−30
Machine parameter settings for analog I/O, 4−15
MADAP Studio reporting system, 2−66
Master control word, 5−2
Modem, 2−38
Modules sensitive to electrostatic discharge. *See* ESD−sensitive components
MP filter for SR−CAN module, 4−29

P
PCI_BM−CAN, 2−23
PCI_BM−DP, 2−23
PCI_BM−IBS, 2−23
PCL, 3−1
PCL field bus card, 2−23
PHG2000 handling, 7−7
PLC data channel, 3−4
Process handling, 7−7
Profibus DP, 2−23

Q
Qualified personnel, 1−2

Appendix

R
RC−lead referencing, 5−9
ReachOut, 2−38
READ/WRITE PLC, 3−4
Release, 1−8
Remote control, 2−37
rho4 file system, 7−1
RM65M, 4−43
ROPS4/Online, 6−3
rSGSerialNb, 4−28
rSGSRCan, 4−28
rSSSRCan, 4−28

S
Safety instructions, 1−4
Safety markings, 1−3
Scope of rho4.1 software, 8−1
Screening time (P5), 4−39
Screenlayout, 6−4
SERCOS interface, 5−1
SERCOS machine parameter, 5−5
Servodyn−D−rho4−Interface, 4−47
Size of the hard disk, 7−8
Small linking, 2−29
Software basic package, 8−17
Spare parts, 1−6
SR−CAN Machine parameter P314, 4−31
SR−CAN module, 4−28
  read, 4−28
  write, 4−28
SR−CAN MP Download, 4−31
SR−CAN serial number, 4−30
SR−CAN−modules, 4−1
Standard operation, 1−1
Status messages Servodyn−D −> rho4, 4−48
Support of hard disks > 1,5 GB at the rho4.1, 7−8
Switcher, 7−8

T
Telediagnostic, 2−37
Test activities, 1−5
To start WinPanel, 3−7
Trademarks, 1−8

U
UPS, 7−7

V
Virtual PHG, 2−54

W
Windows OEM applications, 6−17
Winrho4.ini, 6−17

Appendix

Notes:

Rexroth
Bosch Group

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

Printed in Germany
DOK-RHO*4-RHO4.1*SYSB-PR07-EN-P