Rexroth Inline Terminal for Serial Data Transmission

R911170482
Edition 02

R-IB IL RS 485/422 PRO(-2MBD)-PAC

1 serial input and output channel
RS 485 or RS 422

09/2006

1 Description

The terminals are designed for use within an Inline station. It is used to operate standard I/O devices with serial interfaces on a bus system.

Features

- Serial input and output channel (RS-485 or RS-422)
- Various protocols supported
- Transmission speed adjustable up to 38400 baud
- Number of data bits, stop bits, and parity can be set
- 4-kbyte receive buffer and 1-kbyte transmit buffer
- Parameterization and data exchange via the bus using process data

This data sheet is only valid in association with the application description for the Rexroth Inline system (see "Documentation" on page 3).

Make sure you always use the latest documentation. It can be downloaded at www.boschrexroth.com.

Table of Contents

2 Ordering Data

Terminal Blocks

Description	Type	MNR	Pcs./Pkt.
Inline terminal for serial data transmission, complete with accessories (connectors and labeling fields); Transmission speed 500 kbps	R-IB IL RS 485/422-PRO-PAC	R911170442	1
Inline terminal for serial data transmission, complete with accessories (connectors and labeling fields); Transmission speed 2 Mbps	R-IB IL RS 485/422-PRO-2MBD-PAC-PAC	R911170443	1

Documentation

Description	Type	MNR	Pcs./Pkt.
"Automation Terminals of the Rexroth-Inline Product Range" application description	DOK-CONTRL-IL-SYSINS***-AW..-EN-P	R911317021	1
„Configuring and Installing the Rexroth-Inline Product Range for INTERBUS“ application description	DOK-CONTRL-IL-SYSPRO***-AW..-EN-P	R911317023	1

For further ordering data (accessories), please refer to our product catalog at www.boschrexroth.com.

3 Technical Data

General Data	
Housing dimensions (width x height x depth)	24,4 mm x 120 mm x 71,5 mm
Weight	135 g (including connector)
Mode	Process data mode with 12 byte
Connection method for sensors	2 and 3-wire technology
Permissible temperature (operation)	-25°C to +55°C
Permissible temperature (storage/transport)	-25°C to +85°C
Permissible humidity (operation/storage/transport)	10 % to 95 % according to DIN EN 61131-2
Permissible air pressure (operation/storage/transport)	70 kPa to 106 kPa (up to 3000 m above sea level)
Degree of protection	IP 20 according to IEC 60529
Class of protection	Class 3 according to VDE 0106, IEC 60536
Connection data of connector	
Connection method	spring-cage terminals
Conductor cross-section	0.2 mm^2 to 1.5 mm^2 (solid or stranded), AWG 24 -16

Interfaces	
Bus	
Local bus	Through data routing
Serial Interfaces	
Type	RS485 half duplex or RS422 full duplex cannot be operated simultaneously Electrical data according to EIA (RS) 485, EIA (RS) 422, CCITT V.11
Line termination resistance	120 Ω, typical
Permissible input differential voltage	±5,7 V, maximum

Owing to the 120 W termination resistance the input differential voltage is limited to ±5.7 V.

Hysteresis	50 mV, typical
Input sensitivity	-0.2 V, minimum; +0.2 V, maximum
Output differential voltage (with 100 Ω load)	±2.0 V minimum
Output differential voltage (with 54 Ω load)	±1.5 V minimum
Short-circuit output current	±80 mA, typical

Transmission Speed	
R-IB IL RS 485/422-PRO-PAC	500 kbps
R-IB IL RS 485/422-PRO-2MBD-PAC	2 Mbps

Power Consumption	**500 kbps**	**2 Mbps**
Communications power U_L	7.5 V DC	7.5 V DC
Current consumption at U_L	170 mA, typical; 260 mA, maximum*	185 mA, typical; 260 mA, maximum*
Total power consumption	Approximately 1.275 W, typical, 1.950 W, maximum*	Approximately 1.388 W, typical, 1.950 W, maximum*

* All serial interface connections short circuited.

CAUTION

A continuous short-circuit can damage the device.

This terminal takes no current from the U_M and U_S potential jumpers.

Supply of the Module Electronics Through the Bus Coupler	
Connection method	Potential routing

Power Dissipation	
Power dissipation in the module	P_{EL} = 1.4 W
Power dissipation of the housing P_{HOU}	1.4 W, maximum (within the permissible operating temperature)

Limitation of Simultaneity, Derating	
$T_A \leq 50°C$	No derating
$T_A > 50°C$	I_Q = 4 A I_Q: Total diagonal routing current I_M/I_S/GND

Safety Equipment

None

Electrical Isolation/Isolation of the Voltage Areas

Electrical isolation of the logic level from the serial interface is ensured by the DC/DC converter.

Common Potentials

The serial interface control and data lines have galvanically the same potential. FE is a separate potential area.

Separate Potentials in the System Consisting of Bus Coupler/Power Terminal and I/O Terminal

- Test Distance	- Test Voltage
5 V supply incoming remote bus / 7.5 V supply (bus logic)	500 V AC, 50 Hz, 1 min
5 V supply outgoing remote bus / 7.5 V supply (bus logic)	500 V AC, 50 Hz, 1 min
RS-485/422 interface / 7.5 V supply (bus logic)	500 V AC, 50 Hz, 1 min
RS-485/422 interface / 24 V supply (I/O)	500 V AC, 50 Hz, 1 min
RS-485/422 interface / functional earth ground	500 V AC, 50 Hz, 1 min
7.5 V supply (bus logic) / 24 V supply (I/O)	500 V AC, 50 Hz, 1 min
7.5 V supply (bus logic) / functional earth ground	500 V AC, 50 Hz, 1 min
24 V supply (I/O) / functional earth ground	500 V AC, 50 Hz, 1 min

Error Messages to the Higher-Level Control System

None

Approvals

The latest approvals can be downloaded at www.boschrexroth.com.

4 Local Diagnostic and Status Indicators and Terminal Point Assignment

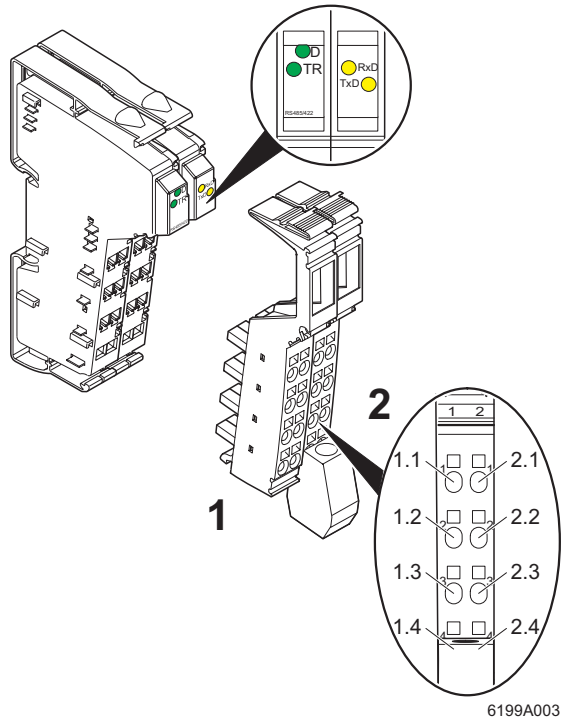

Fig. 1 Diagnostic and status indicators and terminal point assignment

Observe the connection notes on page 8.

4.1 Local Diagnostic and Status Indicators

Des.	Color	Meaning
D	Green	Diagnostics
TR	–	–
Serial Interface:		
RxD	Yellow	Terminal receiving data from the connected device
TxD	Yellow	Terminal transmitting data to the connected device

4.2 Function Identification

Orange

2 Mbps: white stripe in the D LED area

4.3 Terminal Assignment in Half Duplex (RS-485) Mode

Connector	Terminal Point	Signal	Assignment	Data Direction
1	**1.4, 2.4**	**FE**	Functional earth ground	
	2.3	**GND**	GND*	
	2.3	**GND**	GND*	
	All other terminal points of this connector are not used.			
2	**1.1**	**TxD+**	Reserved	
	2.1	**TxD-**	Reserved	
	1.2	**RxD+**	Receive/transmit data (positive)	Input/ Output
	2.2	**RxD-**	Receive/transmit data (negative)	Input/ Output
	1.3	**R+**	Termination resistor (positive)	
	2.3	**R-**	Termination resistor (negative)	
	1.4, 2.4	**Shield**	Shield connection	

* for the R-IB IL RS 485/422-PRO-PAC terminal, hardware version 01 or later

for the R-IB IL RS 485/422-PRO-2MBD-PAC terminal, hardware version 00 or later

4.4 Terminal Assignment in Full Duplex (RS-422) Mode

Connector	Terminal Point	Signal	Assignment	Data Direction
1	**1.4, 2.4**	**FE**	Functional earth ground	
	2.3	**GND**	GND* (	
	2.3	**GND**	GND*	
	All other terminal points of this connector are not used.			
2	**1.1**	**TxD+**	Transmit data (positive)	Output
	2.1	**TxD-**	Transmit data (negative)	Output
	1.2	**RxD+**	Receive data (positive)	Input
	2.2	**RxD-**	Receive data (negative)	Input
	1.3	**R+**	Termination resistor (positive)	
	2.3	**R-**	Termination resistor (negative)	
	1.4, 2.4	**Shield**	Shield connection	

for the R-IB IL RS 485/422-PRO-PAC terminal, hardware version 01 or later

for the R-IB IL RS 485/422-PRO-2MBD-PAC terminal, hardware version 00 or later

5 Internal Circuit Diagram

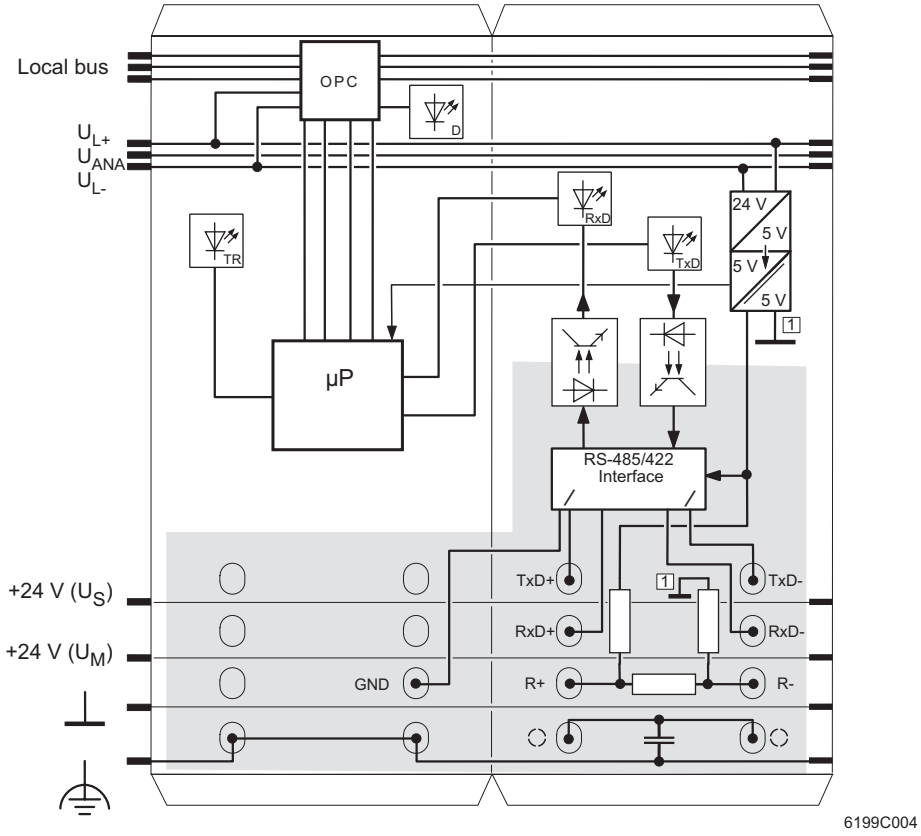

Fig. 2 Internal wiring of the terminal points

Key:

OPC	Protocol chip (bus logic including voltage conditioning)	RS-485/422 Interface	RS-485/422 interface
	Diagnostic and status indicators with function information		Capacitor
	Optocoupler		Resistor
	DC/DC converter with electrical isolation	1	Ground, electrically isolated from ground of the communications power U_L
µP	Microprocessor		

Hardware version 01 or later: GND is assigned to terminal point 2.3 of the R-IB IL RS 485/422-PRO-PAC terminal. In all previous versions, this terminal point is not occupied. For the R-IB IL RS 485/422-PRO-2MBD-PAC terminal with hardware version 00 or later, the terminal point 2.3 is assigned to GND.

Other symbols used are explained in the application descriptions of the Rexroth Inline system (see "Documentation" on page 3).

6 Serial Port

The terminal has one RS-485 and one RS-422 interface. The interfaces cannot be operated simultaneously.

6.1 RS-485

In RS-485 operating mode, an RS-485 network with several devices can be created.

Use a twisted-pair, common shielded data line to connect the devices. Fit this data line with termination resistors at the two most remote points of the RS-485 network. Use the termination resistor integrated in the device when connecting the Inline terminal (see Fig. 2).

Connection examples are shown in Figures 5 to 7 on pages 10 to 11.

This operating mode supports half duplex transmission. Ensure that data is not sent simultaneously by several devices.

To ensure a defined idle cable state, the terminal contains a polarized data line.

6.2 RS-422

In RS-422 operating mode, it is possible to establish a point-to-point connection.

Use a twisted-pair, common shielded data line to connect the devices. Fit this data line with termination resistors at every device. Use the termination resistor integrated in the device when connecting the Inline terminal (see Fig. 2).

Connection examples are shown in Figures 8 and 9 on pages 12 and 12.

This operating mode supports full duplex transmission.

7 Connection Notes

By assigning terminal points 1.4 and 2.4 of both connectors you can connect the cable shield either using a capacitor (connector 2) or directly (connector 1) to functional earth ground (FE). Using the two connection options you can connect one side of the cable shield directly and one side using a capacitor to FE without additional effort. In this way, you can prevent the creation of ground loops that would occur if a shield with two direct connections were placed on FE.Regardless of which side the shield is connected to, all wires must be wired on connector 2.

Use at least one connector with shield connection when connecting the I/O device.

7.1 Capacitor Between Shield and FE

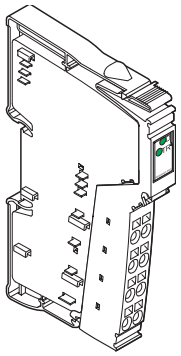

Fig. 3 Position of the shield connector to connect the shield to FE using a capacitor

7.2 Shield Connected Directly to FE

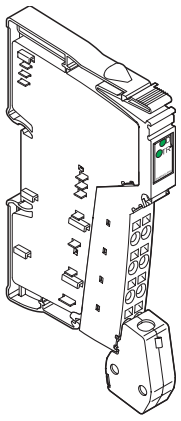

Fig. 4 Position of the shield connector to connect the shield directly to FE

8 Connection Examples

8.1 Comparison of Wiring Examples in Fig. 5 to Fig. 9

Mode	Special Remark	Shield Connection	Remark	Connectors Required per terminal	Shown in
RS-485	Terminal as the end point of a network	With a capacitor	Termination resistor required	Connector set	Fig. 5
RS-485	Terminal in the center of a network	With a capacitor/directly	–	2 shield connectors	Fig. 6
RS-485	Terminal as the end point of a network	Directly	Termination resistor required	Connector set	Fig. 7
RS-422	Shield connected with a capacitor	With a capacitor	Termination resistor required	Connector set	Fig. 8
RS-422	Shield connected directly	Directly	Termination resistor required	Connector set	Fig. 9

The connector set consists of a shield connector and a standard connector.

When Wiring, Please Observe the Following:

- Always connect the shield between two devices on one side using a capacitor and on the other side directly to FE.
- An RS-485 network must be fitted with termination resistors at the beginning and at the end.
- For an RS-422 point-to-point connection, the receive signals on every device must be fitted with termination resistors.

8.2 RS485: Terminal as the End Point of a Network

Connect the shield to FE using a capacitor.

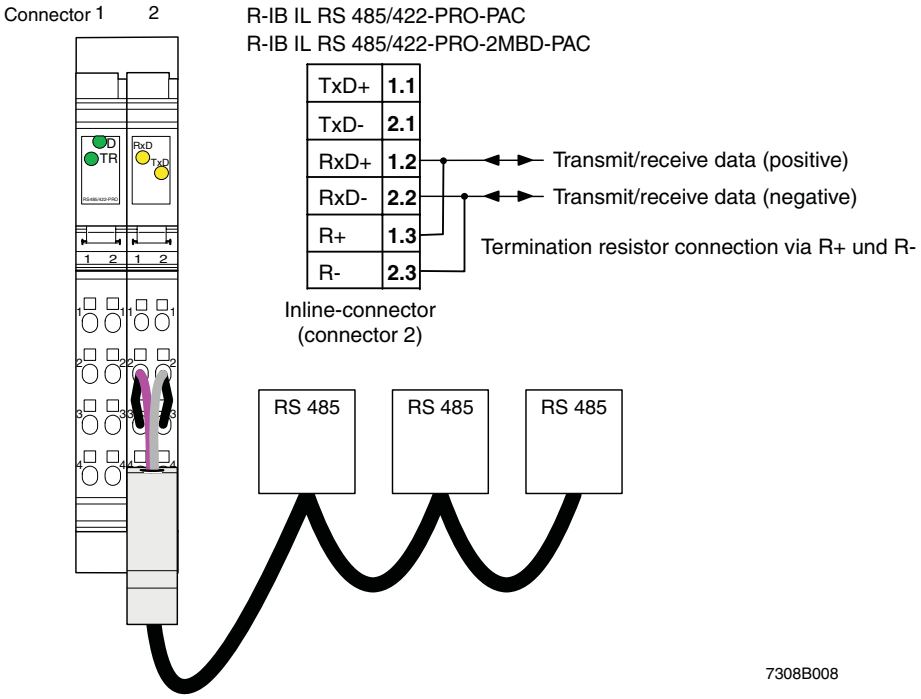

Fig. 5 Interface Wiring RS485: terminal as the end point of a network data line shield connected with a capacitor to FE

8.3 RS485: Terminal in the Center of a Network

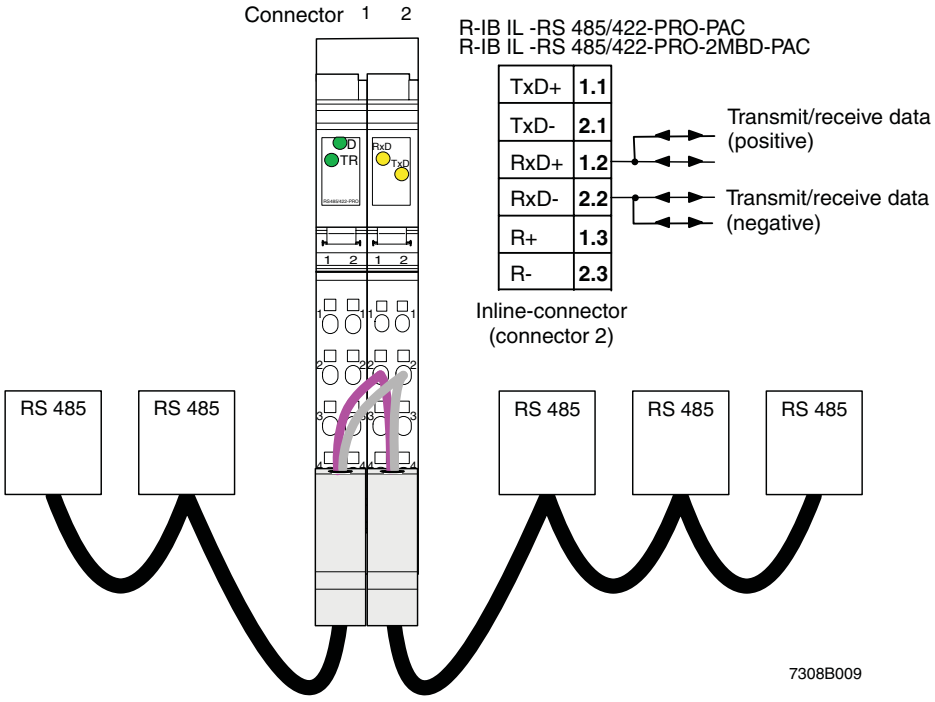

Fig. 6 RS485 interface wiring: in the center of a network, data line shield connected directly and using a capacitor to FE

8.4 RS485: Terminal as the End Point of a Network

Connect the shield directly to FE.

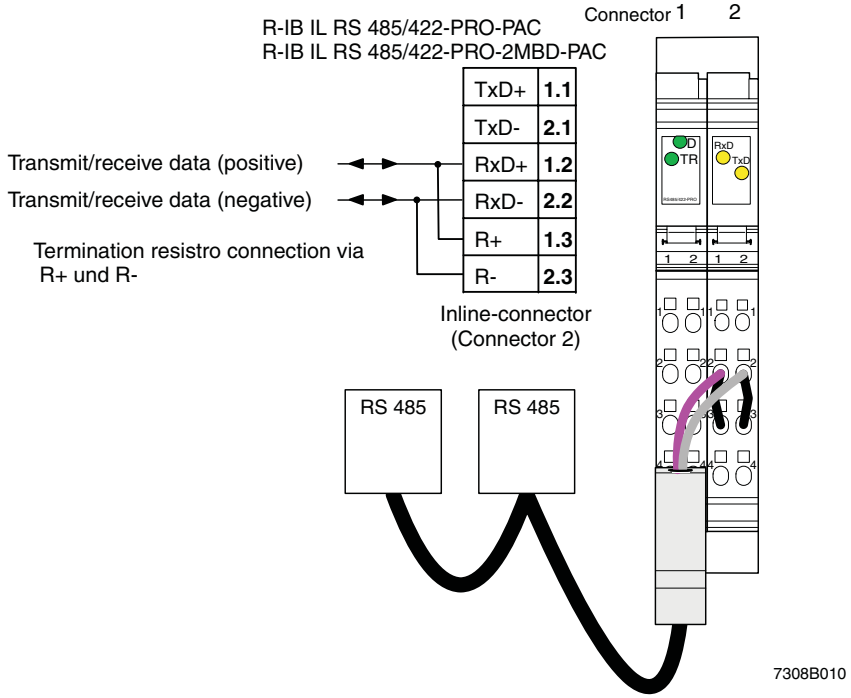

Fig. 7 Interface Wiring RS485: terminal as the end point of a network data line shield directly connected to FE

8.5 RS422: Shield Connected With a Capacitor to the Terminal

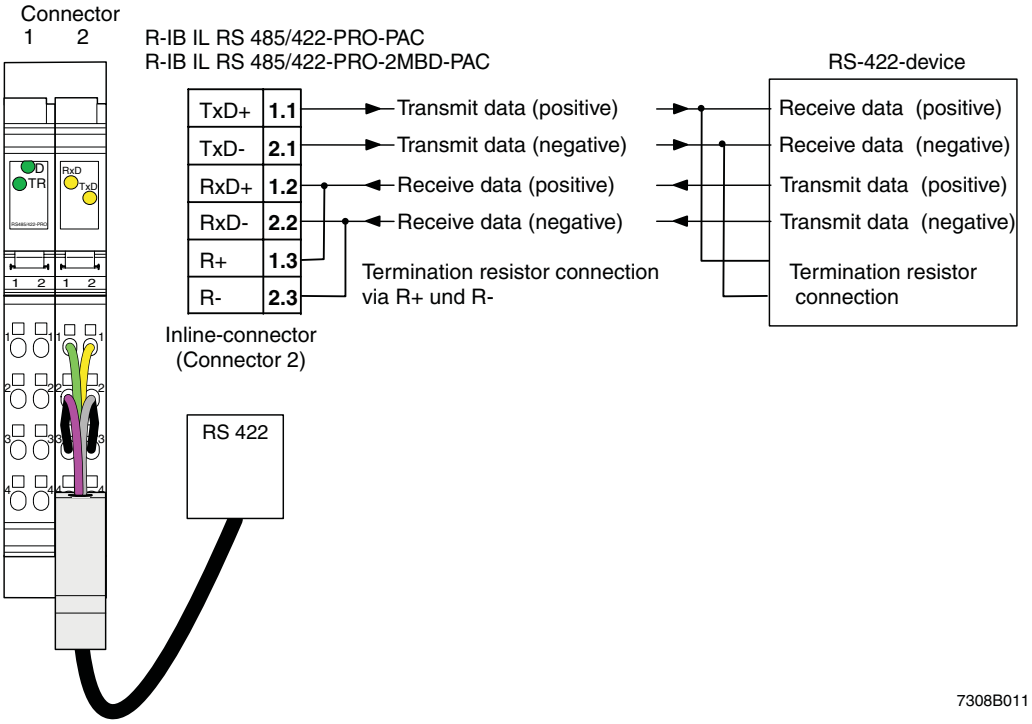

Fig. 8 RS422 interface wiring: data line shield connected with a capacitor to FE

8.6 RS422: Shield Connected Directly to the Terminal

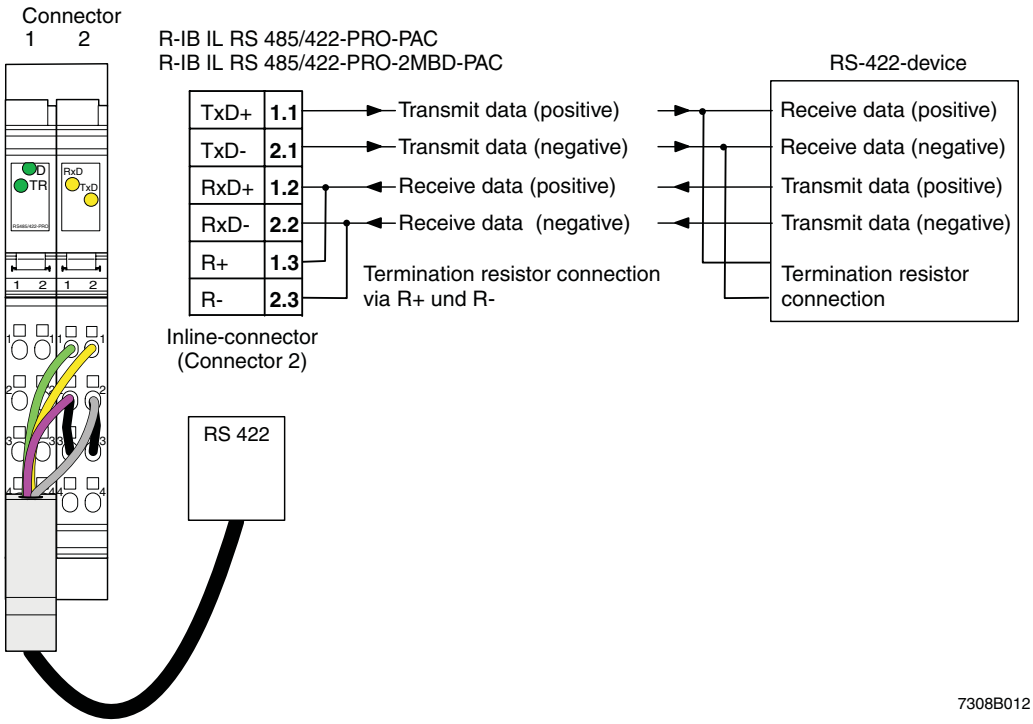

Fig. 9 RS422 interface wiring: data line shield directly connected to FE

9 Programming Data/Configuration Data

9.1 Localbus

ID code	BF_{hex} (191_{dec})
Length code	06_{hex}
Process data channel	96 bits
Input address area	12 bytes
Output address area	12 bytes
Parameter channel (PCP)	0 bytes
Register length (bus)	12 bytes

9.2 Other Bus Systems

For the programming/configuration data of other bus systems, please refer to the appropriate electronic device data sheet (e.g GSD, EDS).

10 Data Storage and Transmission

The terminal stores the received serial data in an intermediate buffer until it is fetched from the serial interface by the bus controller board or the device. Serial data traffic can be managed using various protocols. The protocol used depends on the type of protocol supported by the peer.

10.1 Overview of the Supported Protocols

Protocol	Receive Memory	Transmit Memory	Special Features When Receiving
Transparent	4096 bytes	1023 bytes	
End-to-end	25 buffers each with 11 bytes	1023 bytes (including end characters)	Two end characters are filtered out
Dual buffer	2 buffers each with 11 bytes	1023 bytes (including end characters)	Only stores the most recently received data, end characters are filtered out
3964R	25 buffers each with 11 bytes	5 buffers each with 330 bytes	Data exchange with software handshake, time monitoring, and checksum
XON/XOFF	4096 bytes	1023 bytes	Software handshake

10.2 Transparent Protocol

If the transparent protocol is used, serial data is transmitted through the terminal in the same format it was received from the serial interface or the bus side.

The transmit FIFO (**F**irst-**I**n-**F**irst-**O**ut memory) can store 1023 bytes (1 kbyte) and the receive FIFO can store 4096 bytes (4 kbytes). If the terminal receives another character after the 4095th character, the error pattern is stored in the receive FIFO. All other subsequent characters are ignored.

10.3 End-to-End Protocol

The serial data is conditioned for the end-to-end protocol.

If serial data is sent from the bus side, two additional characters, the first and second delimiters, are attached for transmission to the serial interface. The first and second delimiters are defined upon terminal configuration.

Serial data sent from the serial interface can only be read by the user if the terminal has received the first and second delimiters. The two end characters confirm that the serial data has been received without errors and the maximum data length of 11 bytes has been observed. The delimiters are filtered out when the data is read by the bus side.

Unlike in the transparent protocol, the receive memory is not organized as a FIFO but as a buffer. There are 25 buffers available, each with 11 bytes. If the buffer size of 11 bytes is exceeded, without the two delimiters being detected, the buffer is overwritten again.

The transmit FIFO can store 1023 bytes. The delimiters are attached to, and stored with, the data to be sent.

10.4 Dual Buffer Protocol

With this protocol, the **last** received data block is stored. A data block is defined as a sequence of characters with the first and second delimiter end characters, as in the end-to-end protocol.

As soon as a new data block is received, the previous one is overwritten. This is achieved by means of two buffers, which are written alternately. This means that one buffer will always be available to receive serial data, while the other will be storing the last received data block. A data block is only regarded as complete once both delimiters have been detected, one after the other. It can then be read from the bus side.

If the buffer size of 11 bytes is exceeded, without the two end characters (delimiters) being detected, the buffer is overwritten again.

The same conditions as in the end-to-end protocol apply to sending serial data. If serial data is sent from the bus side, two additional characters, the first and second delimiters, are attached for transmission to the serial interface.

10.5 3964R Protocol

This protocol, developed by Siemens, is the most complex. It uses beginning and end identifiers, a checksum and a time monitoring function.
There are 5 buffers available for transmission and 25 buffers for reception.

Character delay time:	220 ms
Acknowledgment delay time:	2 s
Block waiting time:	10 s
Number of attempts to establish a connection:	6

The optional 3964 priority defines which device may send first (high priority) if there is an initialization conflict (several devices attempting to send data simultaneously).

As the RS-485 terminal is a half duplex interface a waiting time (slave response time) is kept between data transmission and reception. It is 11 bits and depends on the baud rate.

10.6 XON/XOFF Protocol

This protocol operates in the same way as the transparent protocol, but uses a software handshake.

Data transmission with this protocol is controlled by the XON and XOFF characters. XON is preset to 11_{hex} and XOFF to 13_{hex}. You can also define these characters upon terminal configuration.

If the terminal receives an XOFF, no more serial data will be sent until an XON is received.

The terminal itself will transmit an XOFF if the available space in the receive memory is less than 5 bytes. As soon as more memory becomes available again, the module will transmit a single XON.

Serial data is not filtered when it is sent. Any characters, which occur with the code defined for XON and XOFF, are thus sent and may trigger undesirable events at the receiver. When serial data is received, the XON and XOFF characters are filtered and are not available as data. Any characters with the XON or XOFF code are lost. Ensure that characters with these codes do not appear in the data stream.

With the XON/XOFF protocol this function is only completely available for the RS-422. It can be discussed if using the protocol for the RS-485 is useful as the RS-485 is a half duplex connection, which enables either to transmit or to receive data. However, with the XON/XOFF protocol the receiver can respond with a XOFF at any time when transmitting characters. Reception of this control character by the transmitter is thus not ensured.

11 Process Data

The terminal process image comprises 12 byte process data each in input and output direction.

The terminal has 12 byte process data

Byte	0	1	2	3	4	5	6	7	8	9	10	11
OUT	Kommando/ Parameter	Daten	Daten	Daten	Daten	Daten	Daten	Daten	Daten	Daten	Daten	Daten
IN	Status Parameter	Daten	Daten	Daten	Daten	Daten	Daten	Daten	Daten	Daten	Daten	Daten

The command is used to determine the function. The actually transmitted data depends on the command.

11.1 Byte 0 and 1 General

Control byte

Control byte 0								Control byte 1							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
0	command			OUT parameter				x	x	x	x	x	x	x	x

Status byte

Status byte 0								Status byte 1							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
d	command			IN parameter				x	x	x	x	x	x	x	x

d: disturbance

x = 0 or 1; the assignment depends on the command.

Code (bin)	**Code (hex) (Where Byte 0, Bit 7= 0)**	**Command**
000	0	Read status bits. The input bytes 0 and 1 contains the number of characters received.
001	1	Transmit characters
010	2	Store characters temporarily
011	3	Read characters Parameter = C_{hex}: Read firmware version, Parameter = D_{hex}: Read Configuration
100	4	Write configuration
101	5	Toggle command 1: Transmit characters
110	6	Toggle command 2: Store characters temporarily
111	7	Toggle command 3: Read characters

Command Toggling

Command toggling is used to execute a command on a terminal again. In this way a second command code is available for the same function. This applies for the following commands:

- Transmit characters
- Store characters temporarily
- Read characters

Here, bit 14 is used for toggling.

11.2 "Read Status Bits" Command

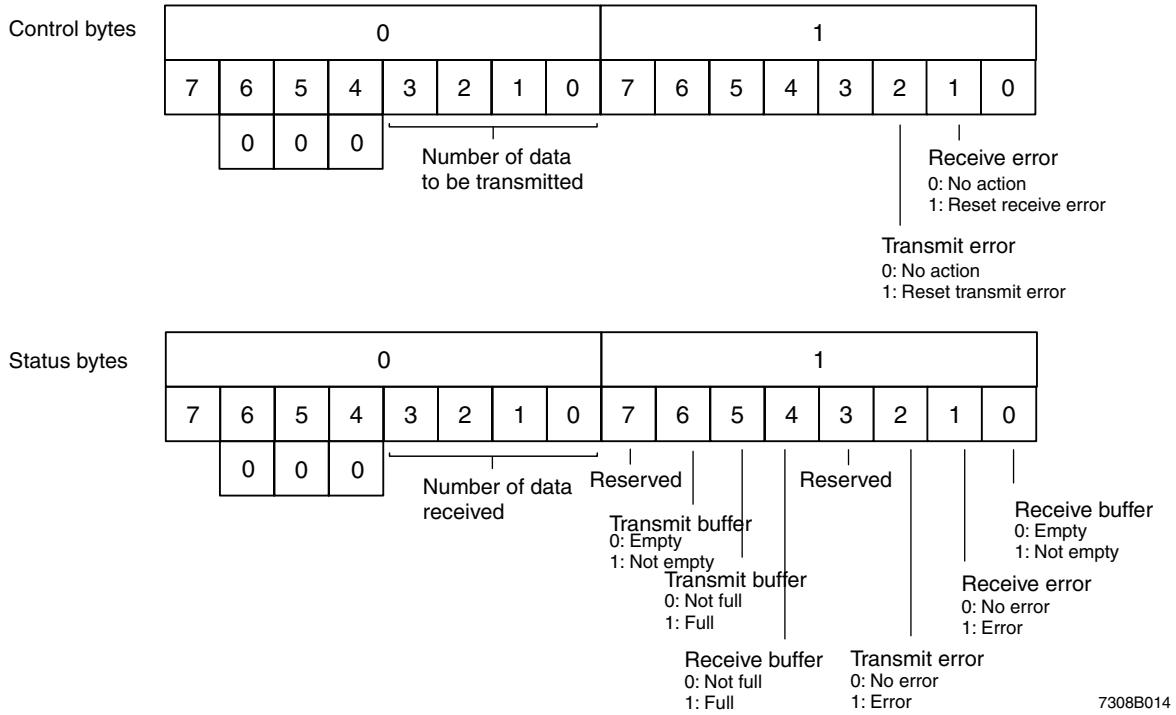

Fig. 10 Format of the process data bytes 0 and 1

Bit/Status	Effect	protocol
Bit 0 = '1'	The receive buffer is not empty, characters to be read are available.	All
Bit 1 = '1'	The receive error indicates that a 3964R telegram could not be received without error after six transmit attempts of the serial peer or after the block waiting time had been exceeded.	3964R
Bit 2 = '1'	The send error indicates that a 3964R telegram could not be transmitted from the module to the serial peer without errors after six transmit attempts. The telegram was rejected.	3964R
Bit 3 = '1'	Reserved	
Bit 4 = '1'	The receive buffer is full: Transparent and XON/XOFF protocol: Residual capacity: < 15 characters 3964R and end-to-end protocol: Residual capacity: none	Transparent, end-to-end, 3964R, XON/XOFF
Bit 5 = '1'	The transmit buffer is full: 3964R protocol: Residual capacity: none Dual buffer, transparent, end-to-end, XON/XOFF protocol: Residual capacity: ≤30 characters	All
Bit 6 = '1'	The transmit buffer is not empty, characters to be sent are available.	All
Bit 7 = '1'	Reserved	
Bits 8 to 11	Number of characters received. If the code = F_{hex}, more than 14 characters have been received.	

☞ Both error bits (bits 1 and 2) are not automatically reset. They can only be reset by the process data output byte.

☞ In the transparent and XON/XOFF protocols, the input bytes 0 and 1 contain the total number of characters received.

With the "Read status bits" command the content of the input data is continuously updated. Unlike with other commands toggling is not required.

11.3 "Transmit Characters" Command

Process data is stored in the transmit memory and then directly transmitted via RS-485 or RS-422. A maximum of eleven characters can be transmitted. The OUT parameter determines the number of characters to be transmitted. Characters stored in the intermediate buffer are transmitted first. After the command has been executed successfully, the intermediate buffer is cleared.

Process data assignment for the "Transmit Characters" command with 11 characters (C1 - C11)

Byte	0	1	2	3	4	5	6	7	8	9	10	11
OUT	$1B_{hex}$	Z1	Z2	Z3	Z4	Z5	Z6	Z7	Z8	Z9	Z10	Z11
IN	$1B_{hex}$	–	–	–	–	–	–	–	–	–	–	–

Reasons for an error bit set:

- OUT parameter = 0 **and** intermediate buffer empty
- OUT parameter >11
- Not enough space in the transmit memory
- Not enough space in the intermediate buffer

11.4 "Store Characters Temporarily" Command

The transmit data is stored in an intermediate buffer, which can store 330 characters. No characters are transmitted. The OUT parameter determines the number of characters. The "Transmit Characters" command is used for transmitting the data stored temporarily. In this way character blocks of up to 330 characters can be transmitted. They are divided over 20 telegrams with 11 characters each.

Reasons for an error bit set:

- OUT parameter = 0 **or** >11
- Not enough space in the intermediate buffer

11.5 "Read Characters" Command

This command is used to read a maximum of eleven characters. The IN parameter contains the number of valid characters available in the input data.

Process data assignment for the "Read Characters" command with 11 characters (C1 - C11)

Byte	0	1	2	3	4	5	6	7	8	9	10	11
OUT	30_{hex}	–	–	–	–	–	–	–	–	–	–	–
IN	$3B_{hex}$	Z1	Z2	Z3	Z4	Z5	Z6	Z7	Z8	Z9	Z10	Z11

11.6 "Write Configuration" Command

Process data assignment for the "Write Configuration" command

Output bytes 0 to 11

Byte	0	1	2	3	4	5	6	7	8	9	10	11
OUT	40_{hex}	Error Pattern	Output type Protocol	Baud rate Data width	1. delimiter	2. delimiter	Direct baud rate	Reserved	Reserved		Reserved	
IN	40_{hex}	–	–	–	–	–	–	–	–	–	–	–

Output bytes 2 to 3 for command „write configuration“

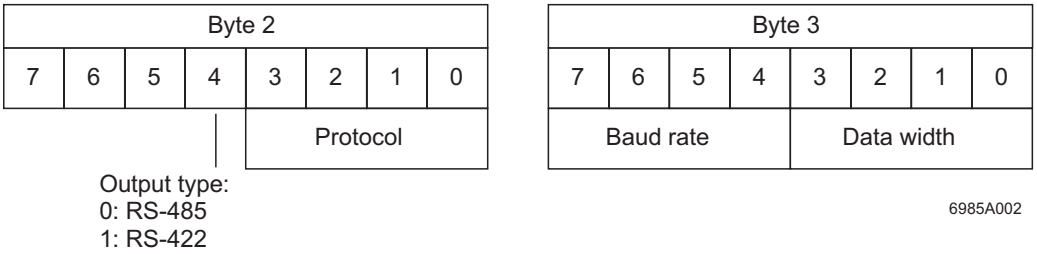

Fig. 11 Output byte 2 and 3 in the "Write Configuration" command

Element Value Range

The options in bold are default settings.

	Protocol
Code	**Meaning**
00_{hex}	**Transparent**
01_{hex}	End-to-end
02_{hex}	Dual buffer
03_{hex}	3964R with low priority
04_{hex}	3964R with high priority
05_{hex}	XON/XOFF

	Baud Rate
Code	**Value**
00_{hex}	110 baud
01_{hex}	300 baud
02_{hex}	600 baud
03_{hex}	1200 baud
04_{hex}	1800 baud
05_{hex}	2400 baud
06_{hex}	4800 baud
07_{hex}	**9600** baud
08_{hex}	19200 baud
09_{hex}	38400 baud
$0D_{hex}$	Directly, basis 500 kbaud
$0E_{hex}$	Directly, basis 62.5 kbaud
$0F_{hex}$	Directly, basis 15625 baud

The specified baud rates of 110 baud to 38400 baud are adequate for most applications. However, you can freely choose the baud rate by means of direct programming. For this, use the 0Dhex, 0E hex and 0Fhex baud rate codes in the output bytes 1 to 3, see "Direct Baud Rate (DBC)" on page 20.

	Data Width			
Code	**Meaning**			**Representation in CMD/PC WORX**
	Data Bits	**Parity**	**Stop Bits**	
00_{hex}	7	Even	1	*7 data bits, even parity, 1 stop bit*
01_{hex}	7	Odd	1	*7 data bits, odd parity, 1 stop bit*
02_{hex}	**8**	**Even**	**1**	***8 data bits, even parity, 1 stop bit***
03_{hex}	8	Odd	1	*8 data bits, odd parity, 1 stop bit*
04_{hex}	8	Without	1	*8 data bits, without parity, 1 stop bit*
05_{hex}	7	Without	1	*7 data bits, without parity, 1 stop bit*
06_{hex}	7	Even	2	*7 data bits, even parity, 2 stop bits*
07_{hex}	7	Odd	2	*7 data bits, odd parity, 2 stop bits*
08_{hex}	8	Even	2	*8 data bits, even parity, 2 stop bits*
09_{hex}	8	Odd	2	*8 data bits, odd parity, 2 stop bits*
$0A_{hex}$	8	Without	2	*8 data bits, without parity, 2 stop bits*
$0B_{hex}$	7	Without	2	*7 data bits, without parity, 2 stop bits*

	Error Pattern
Code	**Meaning**
24_{hex}	**$**
xx_{hex}	Any character

	First Delimiter
Code	**Meaning**
$0D_{hex}$	**Carriage Return (CR)**
xx_{hex}	Any character

Second Delimiter	
Code	**Meaning**
$0A_{hex}$	**Line Feed (LF)**
xx_{hex}	Any character

	Output Type
Code	**Meaning**
0_{hex}	**RS-485**
1_{hex}	**RS-422**

Reasons for an error bit set:

- Using a reserved code
- Setting a reserved bit
- Baud rate 110 baud or 300 baud in the 3964 protocol

Example

Default: transparent protocol
Baud rate: 19200 baud
Data width: 8 data bits with odd parity and one stop bit
Configuration (in hex): 4000 0083 0000 0000

The **error pattern** contains the character that is written to the FIFO if a character was received with errors (this does not apply to the 3964R protocol). This can be the result of, for example, parity errors, exceeded value ranges or noise interference. In the transparent and XON/XOFF protocols, the pattern is also used if the receive FIFO is full and further characters are received.

The **first delimiter** and the **second delimiter** contain the end characters for the dual buffer and the end-to-end protocols.

After successful configuration the characters for the receive and transmit FIFO are reset. In this way, all transmit and receive data that have not yet been processed will be deleted.

Direct Baud Rate (DBC)

Byte	0	1	2	3	4	5	6	7	8	9	10	11
Meaning	40_{hex}	Error Pattern	Output type	Baud rate	1. delimiter	2. delimiter	Direct baud rate	Reserved	Reserved		Reserved	
OUT	40	00	00	83	00	00	00	–	–	–	–	–
IN	40	–	–	–	–	–	–	–	–	–	–	–

Choose direct programming of the baud rate in the output byte 1 by means of the $0D_{hex}$, $0E_{hex}$ and F_{hex} baud rate codes. You can select a basic clock for the baud rate. The actual baud rate is calculated according to the following formula:

Baud rate = basic clock / (DBC + 1)

Specify DBC in the output byte 6. To determine DBC, change the equation to read:

DBC = basic clock / baud rate - 1

Example

The baud rate is 15625 baud. A basic baud rate of 500 kbaud (code $0D_{hex}$) is chosen. Determine the direct baud rate:

DBC = (500000 baud / 15625 baud) - 1 = $31_{dec} = 1F_{hex}$

Here is an example for the first 4 output bytes:
40 00 00 D2 00 00 1F $_{hex}$.

Programming of the direct baud rate theoretically enables a maximum value of 500 kbaud. Proper operation of the terminal is tested and guaranteed for up to 38400 baud. Operation with higher baud rates depends on the application.

11.7 "Read Configuration" Command

Process data assignment for the "Read Configuration" command

Byte	0	1	2	3	4	5	6	7	8	9	10	11
OUT	$3D_{hex}$	x	x	x	x	x	x	x	x	x	x	x
IN	$3D_{hex}$	Error Pattern	Output type	Baud rate	1. delimiter	2. delimiter	Direct baud rate	00	00	00	00	00

11.8 "Read Firmware Version" Command

With a control byte $3C00_{hex}$ the second input byte supplies the firmware version and the type code.

Byte 2								Byte 3							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Firmware version, e.g.,100_{hex}												Type code: 7_{hex}			

Type code 7_{hex} is identical with the type code of R-IB IL RS 485/422-PRO-PAC and
R-IB IL RS 485/422-PRO-2MBD-PAC.

DOK-CONTRL-ILRS485*P**-KB02-EN-P

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

7308_en_01