Rexroth Inline Terminal for Serial Data Transmission

R911170480
Edition 01

R-IB IL RS 232-PRO(-2MBD)-PAC

Firmware Version 1.10 or later
1 serial I/O channel, RS 232

1 Function

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

Parameterization and data exchange is carried out via the bus using process data.

Features

- A serial I/O channel (RS-232)
- DTR/CTS handshake supported
- Various protocols supported
- Transmission speed adjustable up to 38400 baud
- Number of data bits, stop bits, and parity can be set
- 4 KB receive buffer and 1 KB transmit buffer
- Parameterization and data exchange via the bus using process data
- Diagnostic and status indicators

☞ This data sheet is only valid in association with the application description for the Rexroth Inline system (see "Documentation" on page 2).

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

2 Ordering Data

Products

Description	Type	MNR	Pcs./ Pkt.
Inline terminal for serial data transmission; including accessories (connectors and labeling fields); Transmission speed 500 kbps	R-IB IL RS 232-PRO-PAC	R911170440	1
Inline terminal for serial data transmission; including accessories (connectors and labeling fields); Transmission speed 2 Mbps	R-IB IL RS 232-PRO-2MBD-PAC	R911170441	1

Accessories (as replacement parts)

Description	Type	MNR	Pcs./Pkt.
Connector set with a standard connector and a shield connector	R-IB IL AO/CNT-PLSET	R911289339	1

Documentation

Description	Type	MNR	Pcs./Pkt.
"Automation Terminals of the Rexroth-Inline Product Range" application description	DOK-CONTRL-ILSYSINS***-AW..-EN-P	R911317021	1

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 connectors)
Mode	Process data mode with12 byte
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 3,000 m above sea level)
Degree of protection	IP20 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	V.24 interface with DTR/CTS handshake Data terminal equipment (DTE) (Data Terminal Equipment; DTE) Electrical data according to EIA (RS) 232, CCITT V.28, DIN 66259, Part 1
Input impedance	5 kΩ typical
Permissible input voltage area	-30 V to +30 V
Switching thresholds	0.8 V to 2.4 V
Hysteresis	0.5 V typical
Output voltage "HIGH" (with 3 kΩ load)	6.7 V typical
Output voltage "LOW" (with 3 kΩ load)	-6.7 V typical
Output voltage "HIGH" (no-load operation)	≤ 25 V
Output voltage "LOW" (no-load operation)	≥ -25 V

Interfaces (continued)	
Permissible load capacity	2500 pF
Short-circuit protected against GND	Yes
Short-circuit current	±60 mA, maximum
Transmission Speed	
R-IB IL RS 232-PRO-PAC	500 kbps
R-IB IL RS 232-PRO-2MBD-PAC	2 Mbps

Power Consumption	**500 kbps**	**2 Mbps**
Communications power U_L	7.5 V	7.5 V
Current consumption at U_L	155 mA, typical; 225 mA, maximum*	165 mA, typical; 225 mA, maximum*
Total power consumption	Approximately 1.163 W, typical, 1.688 W, maximum*	Approximately 1.238 W, typical, 1.688 W, maximum*

* All serial interface connections short circuited.

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

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

Power Dissipation	**500 kbps**	**2 Mbps**
Power dissipation in the module	P_{EL} = 1.163 W	P_{EL} = 1.238 W
Power dissipation of the housing P_{HOU}	1.4 W, maximum (within the permissible operating temperature)	

Limitation of Simultaneity, Derating (500 kbps)

No limitation of simultaneity, no derating

Limitation of Simultaneity, Derating (2 Mbps)	
TA ≤ 50°C	No derating
TA ≥ 50°C	I_Q = 4 A, I_Q:Total diagonal routing current

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 Terminal/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-232 interface/7.5 V supply (bus logic)	500 V AC, 50 Hz, 1 min
RS-232 interface/24 V supply (I/O)	500 V AC, 50 Hz, 1 min
RS-232 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 Diagnostic and status indicators and terminal point assignment

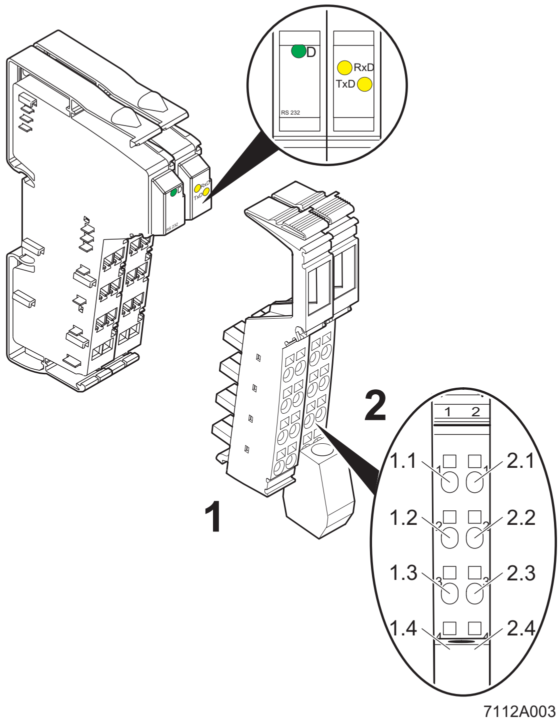


Fig. 1 *R-IB IL RS 232-PRO-PAC*

4.1 Local Diagnostic and Status Indicators

Des.	Color	Meaning
D	Green	Diagnostics
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 Point Assignment

Connector	Terminal Point	Signal	Assignment
1	**1.4, 2.4**	**FE**	Functional earth ground
	All other terminal points of this connector are not used.		
2	**1.1**	**RxD**	Serial data input
	2.1	**TxD**	Serial data output
	1.2	**+5 V**	Control output, internally wired to +5 V DC
	2.2	**CTS**	Control input for hardware handshake
	1.3	**DTR**	Control output for hardware handshake
	2.3	**GND**	GND for serial interface
	1.4, 2.4	**Shield**	Shield connection

☞ Observe the connection notes on page 9.

5 Internal Circuit Diagram

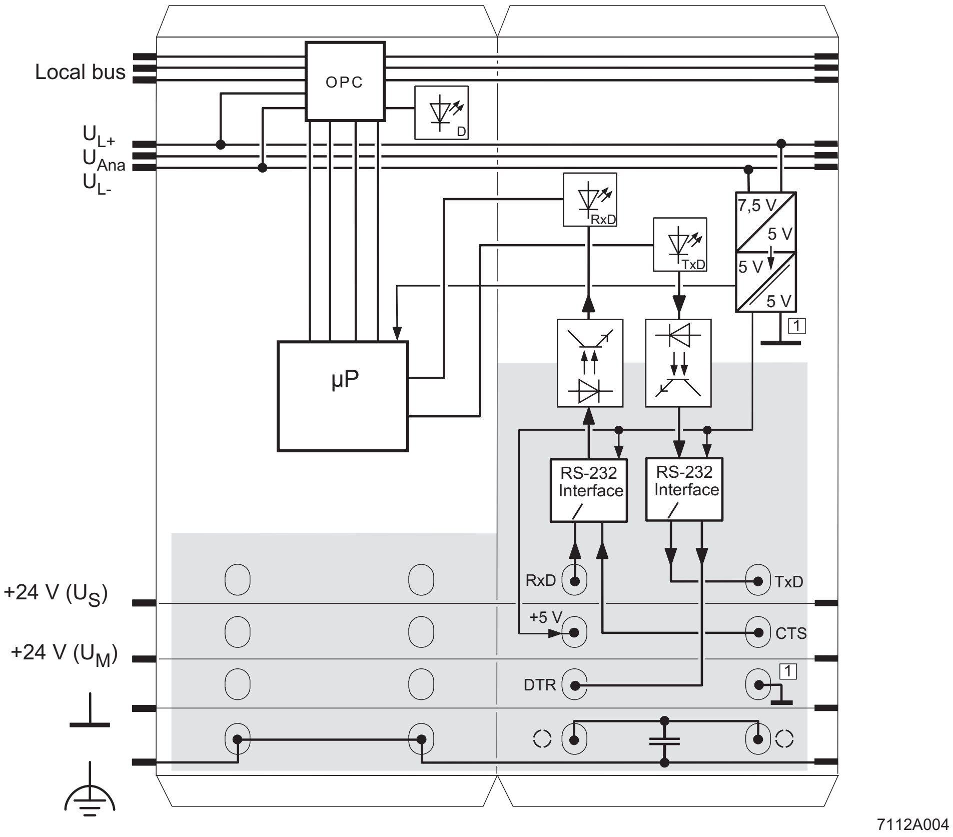


Fig. 2 Internal wiring of the terminal points

Key:

OPC	Protocol chip (bus logic including voltage conditioning)	µP	Microprocessor
	Diagnostic and status indicators with function information	RS-232 Interface	RS-232 interface
	Optocoupler		Capacitor
xxx	DC/DC converter with electrical isolation	1	Ground, electrically isolated from ground of the communications power U_L

Other symbols used are explained in the application description of the Rexroth Inline system (see "Documentation" on page 2).

6 V.24 Interface

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

The V.24 interface on the terminal represents some form of DTE (data termination equipment). This means that connector 2 terminal point 2.1 (TxD) is always used to transmit and connector 2 terminal point 1.1 (RxD) is always used to receive.

According to the standard, some form of DCE (data communication equipment) is connected to the V.24 interface as a peer. DTE can also be connected. Please refer to the connection notes under 6.2 and 6.3.

Measuring the voltage between the connection points for the TxD and GND signals in idle state will determine whether the device to be connected to the V.24 interface is a form of DTE or DCE. If the voltage measures approximately **-5 V**, the device is a form of **DTE**. If the voltage is approximately **0 V**, the device is a form of **DCE**.

Example

When using a 25-pos. standard connector (see Fig. 8 on page 9) the voltage between **pin 2 (TxD) and pin 7 (GND)** must be measured.

6.1 V.24 Module Handshake Signals

Any device with a V.24 interface can be connected to the V.24 interface on the terminal. Both the terminal and the device connected to the V.24 interface can act as transmitter **and** receiver for data exchange. As errors can occur during data exchange if both devices transmit or receive simultaneously, the **handshake** is used as a procedure for the mutual signaling of clear to receive and clear to transmit.

The terminal supports DTR and CTS handshake signals. Each uses one wire of the connecting cable.

The connecting signals are described from the point of view of the terminal, i.e., from the point of view of the DTE.

Handshake signals:

Signal	Meaning	Direction
CTS (Clear To Send)	The terminal receives the CTS signal from the connected device via the V.24 interface. If the CTS signal is set to *High*, the terminal can transmit data.	Input
	☞ The exception is 3964R, XON/ XOFF Protocol.	
DTR (Data Terminal Ready)	The DTR signal is transmitted from the terminal, i.e., set to *High*, once it is ready to receive. The peer connected to the V.24 interface is then able to transmit. After 4095 characters (4 KB) the terminal receive buffer is full, and the DTR signal is set to *Low*. As soon as more characters are read from the bus side, the DTR signal is set to *High* and the terminal is ready to receive.	Output
	☞ With the transparent, XON/ XOFF, and end-to-end protocols, DTR is set to „0“ if fewer than 15 characters are free in the receive FIFO.	

6.2 V.24 Interface Wiring With Four-Wire Handshake

The TxD, RxD, DTR, and CTS signals are used for a four-wire handshake connection between the terminal and the device to be connected. Each signal corresponds to one wire in the connecting cable. An Inline male connector is required on the terminal side. A 9 or 25-pos. socket is required on the opposite side depending on the device to be connected. Both GND pins are also wired.

In Figure 3 and Figure 4 the shield connector is connected on the right-hand side of the terminal. In this case, a capacitor is placed between the shield and FE. If the shield is to be placed directly on FE, the shield connector must be connected on the left-hand side of the terminal. Observe the connection notes on page 9.

In Figure 3 and Figure 4 it is assumed that the signal assignment of the connectors for the device to be connected corresponds to the assignment of a PC connector.

In individual cases, however, the signal assignment of the pins might be different because the DTE-DTE connections as well as the connections between 25-pos. and 9-pos. connectors and sockets are not standardized.

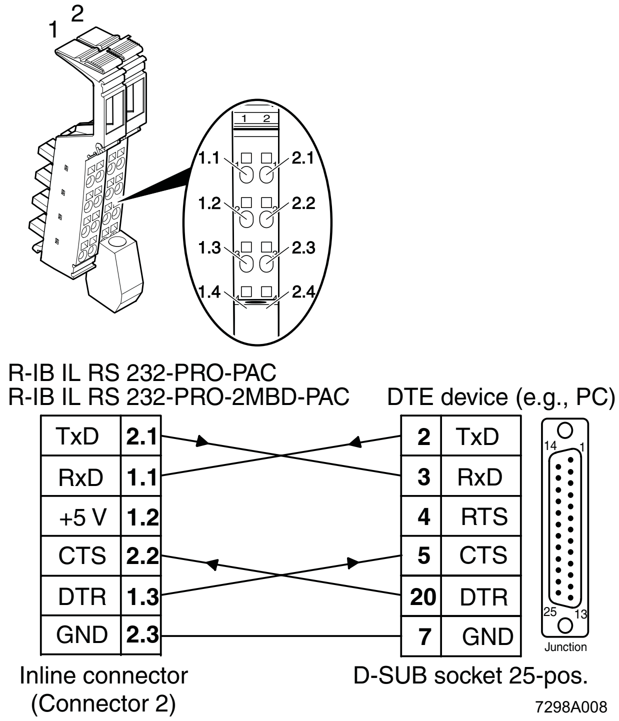


Fig. 3 V.24 interface wiring with handshake for ***DTE*** *(25-pos.)*

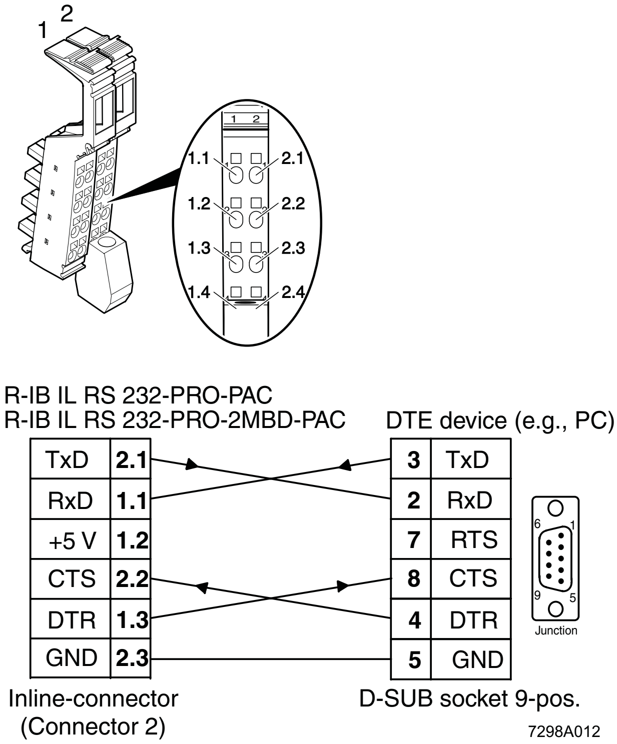


Fig. 4 V.24 interface wiring with handshake for ***DTE*** *(9-pos.)*

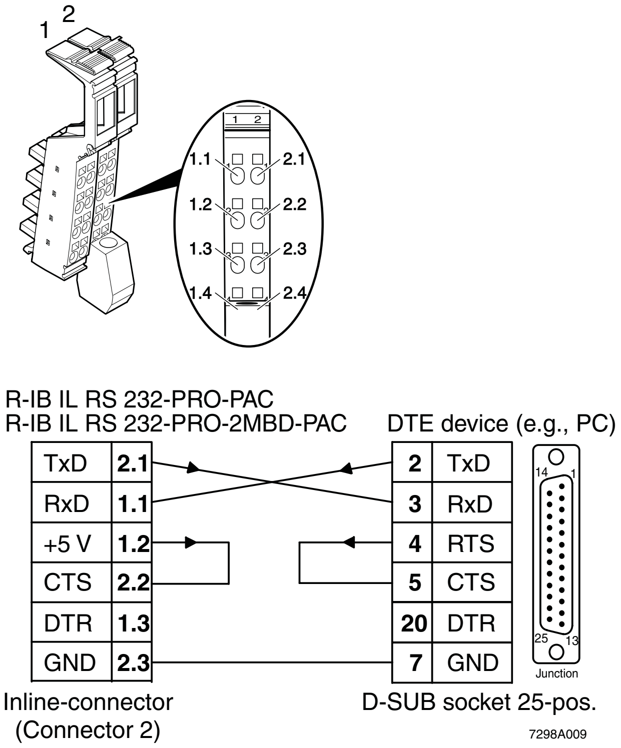
6.3 V.24 Interface Wiring Without Handshake

For wiring without handshake, the transmission can only be executed with the help of both TxD and RxD signals. Both wires for the TxD and RxD signals, in the same way as the GND contacts, are connected to the terminal male connector and are soldered to the socket on the side of the device to be connected.

In addition, a jumper is connected on the male connector between the terminal points for the +5 V and CTS signals and on the socket between the pins for the RTS and CTS signals.

In this way, permanent readiness to receive of the peer is simulated and the connected device is again able to transmit data via the V.24 interface.

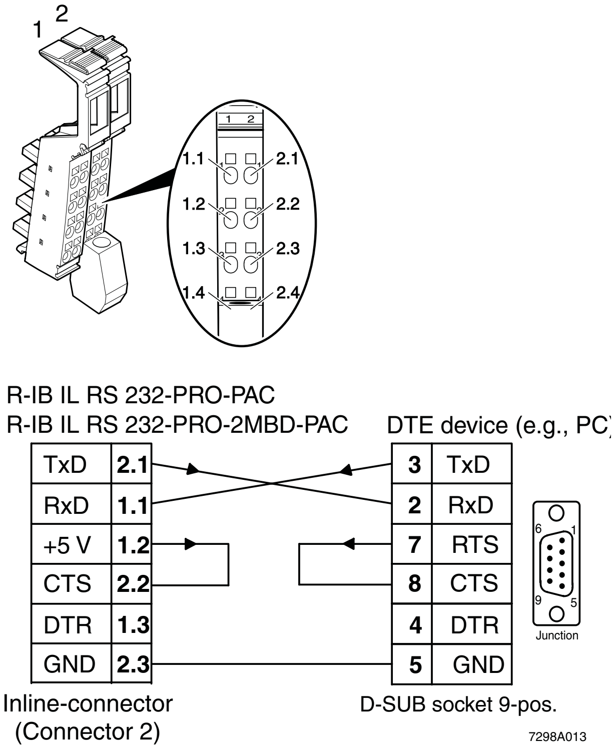
In Fig. 5 and Figure 6 the shield connector is connected on the right-hand side of the terminal. In this case, a capacitor is placed between the shield and FE.
If the shield is to be placed directly on FE, the shield connector must be connected on the left-hand side of the terminal. Observe the connection notes on page 9.
In Figure 5 and Figure 6 it is assumed that the signal assignment of the connectors for the device to be connected corresponds to the assignment of a PC connector.
In individual cases, however, the signal assignment of the pins might be different because the DTE-DTE connections as well as the connections between 25-pos. and 9-pos. connectors and sockets are not standardized.
The terminal sets the DTR signal to *Low* before the receive FIFO overflows. As the DTR signal is not evaluated for wiring without handshake, some of the data sent to the module via the V.24 interface may be lost until the module is ready to receive again.



TxD	2.1
RxD	1.1
+5 V	1.2
CTS	2.2
DTR	1.3
GND	2.3

2	TxD
3	RxD
4	RTS
5	CTS
20	DTR
7	GND

Fig. 5 V.24 interface wiring without handshake for ***DTE*** *(25-pos.)*



TxD	2.1
RxD	1.1
+5 V	1.2
CTS	2.2
DTR	1.3
GND	2.3

3	TxD
2	RxD
7	RTS
8	CTS
4	DTR
5	GND

Fig. 6 V.24 interface wiring without handshake for ***DTE*** *(9-pos.)*

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.
If you connect the shield via connector 1, you must connect the shield connector on the left-hand side of the terminal. All wires must be connected to connector 2.
Ensure that on connector 2, terminal point 1.2 (+5 V) is exclusively used to provide the 5 V signal for the CTS input (terminal point 2.2), in the event of communication without handshake. In this case insert a jumper between the terminal points.
Any other use is not permitted.

Use a connector with shield connection when installing the I/O device. Fig. 7 and Fig. 8 show the connection schematically (without shield connector).

7.1 Capacitor Between Shield and FE

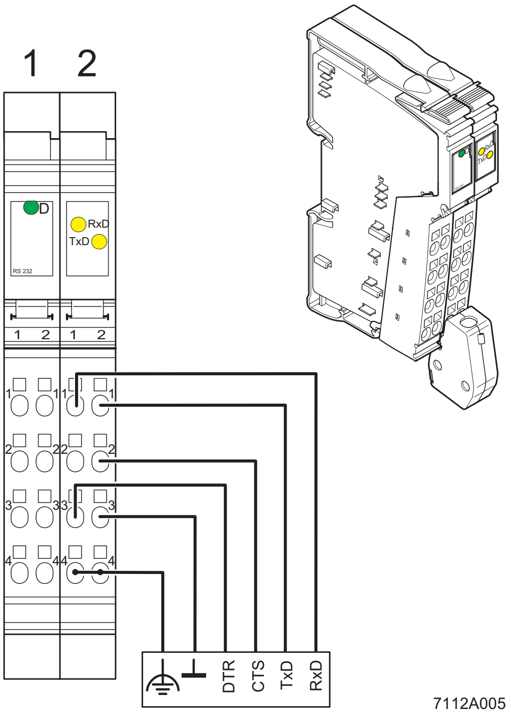


Fig. 7 Connection of an I/O device with a serial interface

In this example the V.24 interface wiring for communication with 4-wire handshake is shown.

7.2 Shield Connected Directly to FE

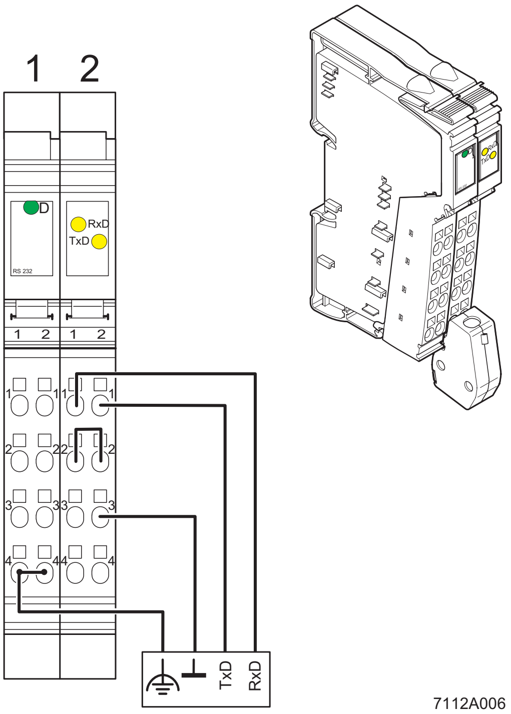


Fig. 8 Connection of an I/O device with a serial interface

In this example the V.24 interface wiring for communication without handshake is shown. You should insert a jumper between connection points 1.2 (+5 V) and 2.2 (CTS).

8 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.

8.1 Overview of the Supported Protocols

protocol	Receive Memory	Transmit Memory	Special Features When Receiving
Transparent	4096 bytes	1023 bytes	
End-to-end	3 buffers each with 330 bytes	1023 bytes (including end characters)	Two end characters are filtered out
Dual buffer	2 buffers each with 330 bytes	1023 bytes (including end characters)	Only stores the most recently received data, end characters are filtered out
3964R	3 buffers each with 330 bytes	3 buffers each with 330 bytes	Data exchange with software handshake, time monitoring, and checksum
XON/XOFF	4096 bytes	1023 bytes	Software handshake

8.2 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 KB) and the receive FIFO can store 4096 bytes (4 KB). 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.

8.3 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 330 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 3 buffers available, each with 330 bytes. If the buffer size of 330 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.

8.4 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 330 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.

8.5 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 3 buffers available for transmission and 3 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).

8.6 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 15 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.

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 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	Command/	Data	Data	Data	Data	Data	Data	Data	Data	Data	Data	Data
IN	Status	Data	Data	Data	Data	Data	Data	Data	Data	Data	Data	Data

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

10.1 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
E:	Command			IN parameter				x	x	x	x	x	x	x	x

E: Error

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

Byte 0, Bit 7 = error: Is not valid for the command "Read characters". For the meaning of Bits for this command please refer to chapter ""Read Characters" Command" on page 15.

Code (bin)	Code (hex) (Where Byte 1, Bit 7 = 0)	Command
000	0	Read status bits. The input bytes 0 and 1 contain the number of characters received.
001	1	Transmit characters
010	2	Store characters temporarily
011	3	Read characters Parameter OUT = C_{hex}: Read firmware version, Parameter OUT = 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, Byte 0, Bit 6 is used for toggling. If you want to transmit character strings subsequently, use the command code 001_{hex} for the first transmission and then use 101_{hex} and 001_{hex} alternately.

10.2 "Read Status Bits" Command

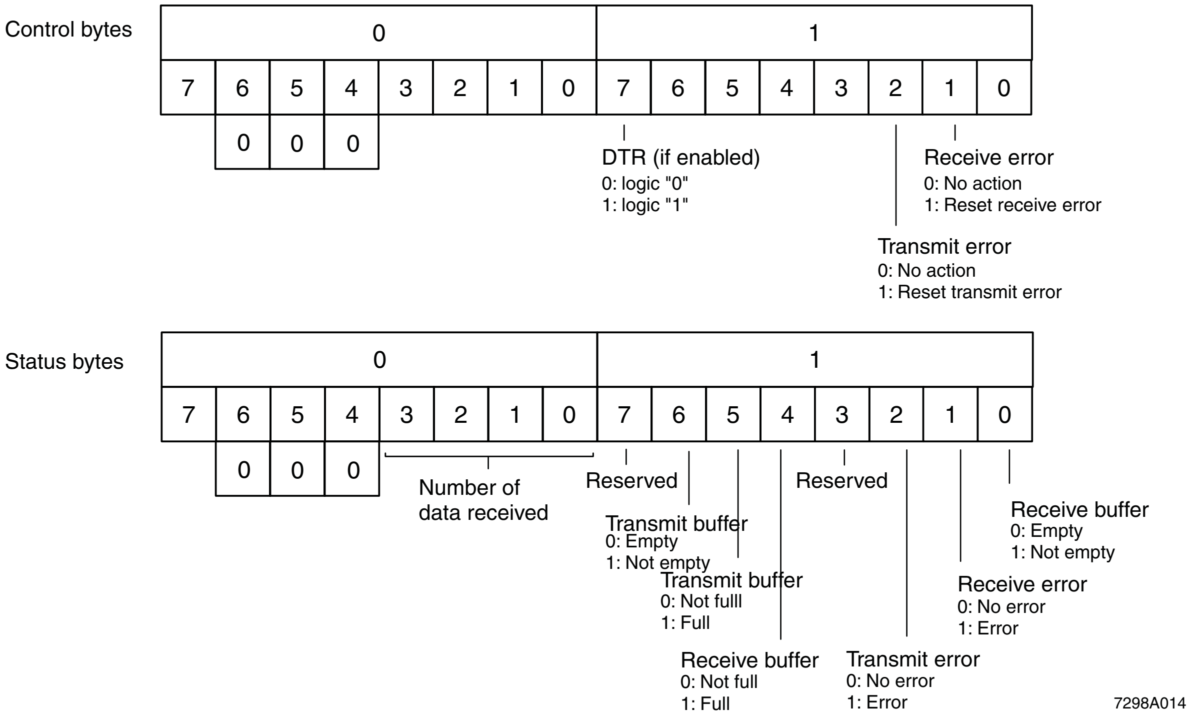


Fig. 9 Format of the process data bytes 0 and 1

The DTR signal can only be generated when "DTR control via process data" is enabled.

Byte	Bit/Status	Effect	protocol
Byte 1	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	
Byte 0	Bit 0 to 3	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 1contain 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.

10.3 "Transmit Characters" Command

Process data is stored in the transmit memory and then directly transmitted via RS 232. 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 17 characters (C1 - C17)

Byte	0	1	2	3	4	5	6	7	8	9	10	11
OUT	1B$_{hex}$	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11
IN	1B$_{hex}$	–	–	–	–	–	–	–	–	–	–	–
OUT	56$_{hex}$	C12	C13	C14	C15	C16	C17	–	–	–	–	–
IN	56$_{hex}$	–	–	–	–	–	–	–	–	–	–	–

For protocols with end identifiers the end identifier is added after every transmitted block (11 characters, maximum)

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

10.4 "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.

Process data assignment for the "Save characters" command and subsequent "Transmit characters" command with 41 characters (C1 - C41)

Byte	0	1	2	3	4	5	6	7	8	9	10	11
OUT	2B$_{hex}$	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11
IN	2B$_{hex}$	–	–	–	–	–	–	–	–	–	–	–
OUT	6B$_{hex}$	C12	C13	C14	C15	C16	C17	C18	C19	C20	C21	C22
IN	6B$_{hex}$	–	–	–	–	–	–	–	–	–	–	–
OUT	2B$_{hex}$	C23	C24	C25	C26	C27	C28	C29	C30	C31	C32	C33
IN	2B$_{hex}$	–	–	–	–	–	–	–	–	–	–	–
OUT	18$_{hex}$	C34	C35	C36	C37	C38	C39	C40	C41	–	–	–
IN	18$_{hex}$	–	–	–	–	–	–	–	–	–	–	–

For protocols with end identifiers the end identifier is added after every transmitted block (41 characters)

Reasons for an error bit set:

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

10.5 "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 eleven characters (C1 - C11)

Byte	0	1	2	3	4	5	6	7	8	9	10	11
OUT	30_{hex}	–	–	–	–	–	–	–	–	–	–	–
IN	$3B_{\text{hex}}$	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11

For the protocols end-end, dual buffers and 3964R, Bit 7 in the status byte 0 is not used as error bit but indicates whether there are still other characters from the received block that remain to be read.

For the indicated protocols a data transfer is handled as block that is identified by a special end-identification. When reading the characters received, it is important to know whether the signs provided with this command all belong to one single block or whether there are still characters that remain to be read. The status bit "Receive buffer is not empty" does not provide this information because it also indicates whether further blocks have been received. For this reason, bit 7 in the status byte 0 is used in this case. A maximum block length of 330 characters is supported.

Status byte 0, Bit 7

Bit/ Status	**Effect**	**protocol**
Bit 7 = '0'	The characters read are the last ones in the received block.	End-to-end, Dual buffer, 3964R
Bit 7 = '1'	There are still some more characters to be read from the received block.	

☞ Together with the toggle bit, the upper nibble of the status byte 0 might have the values Bhex (1011bin) and Fhex (1111bin).

Example

In the 3964R protocol a block with 20 characters was received. For reading, the command "Read characters" must be given twice. The second time the command is given, the toggle bit is set.

Process data assignment for the "Transmit Characters" command with 20 characters (C1 - C20)

Byte	0	1	2	3	4	5	6	7	8	9	10	11
OUT	30_{hex}	–	–	–	–	–	–	–	–	–	–	–
IN	BB_{hex}	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11
OUT	70_{hex}	–	–	–	–	–	–	–	–	–	–	–
IN	79_{hex}	C12	C13	C14	C15	C16	C17	C18	C19	C20	–	–

10.6 "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	protocol	Baud rate Data width	1. delimiter	2. delimiter	Direct baud rate	Reserved	Reserved		Reserved	
IN	40_{hex}	–	–	–	–	–	–	–	–	–	–	–

Output byte 2 to 3 for the "Write Configuration" command

Byte 2							
7	6	5	4	3	2	1	0
		DTR		protocol			

Byte 3							
7	6	5	4	3	2	1	0
Baud rate				Data width			

Element Value Range

☞ The options in **bold** are default settings.

	DTR Control	
Code	**Meaning**	**Representation in CMD/PC WORX**
0_{hex}	**Automatic**	***Automatic***
1_{hex}	Via process data	*Via process data*

	Protocol *(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 *(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 byte 3, see "Direct Baud Rate (DBC)" on page 18.

	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*
$\mathbf{02_{hex}}$	**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**
$\mathbf{24_{hex}}$	**$**
xx_{hex}	Any character

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

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

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 send and receive data that have not yet been processed will be deleted.

Reasons for an error bit set:

- Using a reserved code
- Setting a reserved bit

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

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

Direct Baud Rate (DBC)

Choose direct programming of the baud rate in the output byte 3 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 seven 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.

10.7 "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	protocol	Baud rate Data width	1. delimiter	2. delimiter	Direct baud rate	00	00	00	00	00

10.8 "Read Firmware Version" Command

With a control byte 0 = $3C_{hex}$ and a control byte 1 = 00_{hex} the input bytes 2 and 3 supply 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: 8_{hex}			

Type code 8_{hex} is identical with the type code of R-IB IL RS 232 PRO-PAC and R-IB IL RS 232-PRO-2MBD-PAC.

Notes:

Notes:

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

The data specified above only serve to describe the product. They do not indicate any specific condition or suitability for a certain application. The information provided does not release the user from the obligation of own judgement and verification. It must be remembered that our products are subject to natural wear and aging.