IB IL RS 485/422-PRO ...
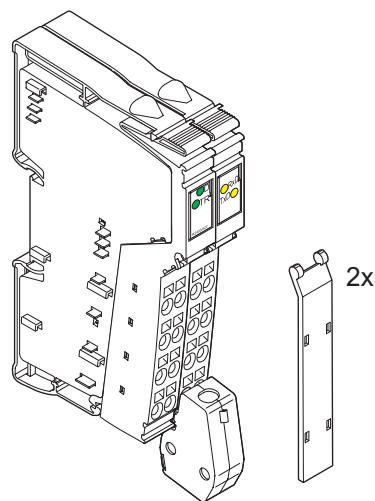
Inline terminal for serial data transmission

AUTOMATION

Data Sheet
6985_en_05

2x

1 Description

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.

Difference between IB IL RS 485/422 ... and IB IL RS 485/422-PRO ...:

IB IL RS 485/422 ...
Parameterization and data exchange are carried out via the bus using PCP services.

IB IL RS 485/422-PRO ...
Parameterization and data exchange are carried out via the bus using process data. This allows faster communication times than with the IB IL RS 485/422.

Features

– A serial I/O channel (RS-485 or RS-422)
– Various protocols supported
– Transmission speed can be set 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

ℹ️ This data sheet is only valid in association with the IL SYS PRO INST UM E user manual.

ℹ️ Make sure you always use the latest documentation.
It can be downloaded at www.download.phoenixcontact.com.
A conversion table is available on the Internet at www.download.phoenixcontact.com/general/7000_en_00.pdf.

ℹ️ This data sheet is valid for the products listed on the following page:

PHŒNIX CONTACT

INSPIRING INNOVATIONS

2 Ordering data

Products

Description	Type	Order No.	Pcs./Pkt.
Inline terminal for serial data transmission, complete with accessories (connector and labeling fields); transmission speed 500 kbaud	IB IL RS 485/422-PRO-PAC	2863627	1
Inline terminal for serial data transmission, without accessories, transmission speed 500 kbaud	IB IL RS 485/422-PRO	2863707	1
Inline terminal for serial data transmission, complete with accessories (connector and labeling fields); transmission speed 2 Mbaud	IB IL RS 485/422-PRO-2MBD-PAC	2878887	1

> **i** The listed connector set is needed for the complete fitting of the IB IL RS 485/422-PRO terminal.
>
> When using the terminal according to the example
> "RS-485: Terminal in the network center" on page 9 two shield connectors are required for connecting the cables.

Accessories

Description	Type	Order No.	Pcs./Pkt.
Connector set with a standard connector and a shield connector	IB IL AO/CNT-PLSET	2732664	1 set
Shield connector	IB IL SCN-6 SHIELD	2736353	5

Documentation

Description	Type	Order No.	Pcs./Pkt.
"Automation Terminals of the Inline Product Range" user manual	IL SYS INST UM E	2698737	1
"INTERBUS Addressing" data sheet	DB GB IBS SYS ADDRESS	9000990	1

3 Technical data

General data	
Housing dimensions (width x height x depth)	24.4 mm x 136 mm x 72 mm (with connectors)
Weight	90 g (without connectors), 135 g (with connectors)
Operating mode	Process data mode with 6 words
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	IP20 according to IEC 60529
Class of protection	Class 3 according to EN 61131-2, IEC 61131-2
Connection data for connectors	
Connection type	Spring-cage terminals
Conductor cross-section	0.2 mm^2 to 1.5 mm^2 (solid or stranded), 24 - 16 AWG

Interfaces

Bus

Local bus	Through data routing

Serial interfaces

Type	RS-485 half duplex or RS-422 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

> ℹ️ Owing to the 120 Ω 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

IB IL RS 485/422-PRO	500 kbaud
IB IL RS 485/422-PRO-PAC	500 kbaud
IB IL RS 485/422-PRO-2MBD-PAC	2 Mbaud

Supply of the module electronics through the bus coupler

Connection method	Potential routing

Power consumption

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

* All connections of the serial interface are short-circuited.

> ⚠️ **ATTENTION: Defective module when continuously short-circuited**
>
> A continuous short circuit can damage the device. Therefore rule out continuous short circuits.

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

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 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 or computer system

None

Approvals

For the latest approvals, please visit www.download.phoenixcontact.com or www.eshop.phoenixcontact.com.

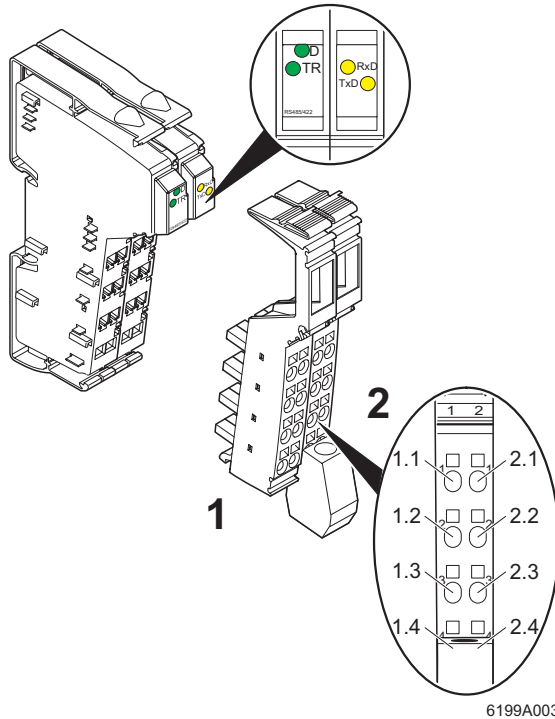# 4 Diagnostic/status indicators and terminal point assignment

Figure 1   Diagnostic/status indicators and terminal point assignment

4.1 Local diagnostic and status indicators

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

4.2 Function identification

Orange

2 Mbps: white stripe in the vicinity of the D LED

4.3 Terminal point assignment in the half-duplex operating mode (RS-485)

Connec-tor	Terminal point	Signal	Assignment	Data direction
1	1.4, 2.4	FE	Functional earth ground	
	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 IB IL RS 485/422-PRO (-PAC) terminal, hardware version 01 or later. In all previous versions terminal point 2.3 is not used.
For the IB IL RS 485/422-PRO-2MBD-PAC terminal, hardware version 00 or later.

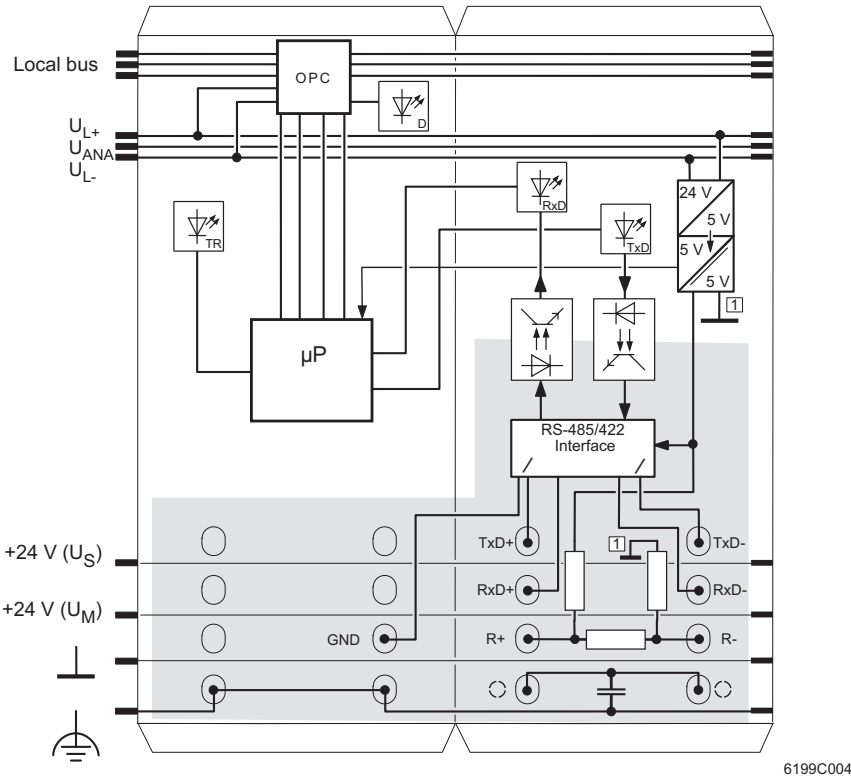### 4.4 Terminal point assignment in the full-duplex operating mode (RS-422)

Connec-tor	Terminal point	Signal	Assignment	Data direction
1	1.4, 2.4	FE	Functional earth ground	
	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 IB IL RS 485/422-PRO (-PAC) terminal, hardware version 01 or later. In all previous versions terminal point 2.3 is not used.
For the IB IL RS 485/422-PRO-2MBD-PAC terminal, hardware version 00 or later.

ℹ️ Observe the connection notes on page 7.

5 Internal circuit diagram

6199C004

Figure 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		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 IB IL RS 485/ 422-PRO (-PAC) terminal. In all previous versions, this terminal point is not used. For the 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 IL SYS INST UM E user manual or in the Inline system manual for your bus system.

6   Serial interface

The terminal has an RS-485 and an RS-422 interface. The interfaces can only be operated alternatively.

6.1   RS-485

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

Use a data cable with commonly shielded twisted pairs to connect the devices. Equip this data cable with a termination resistor at both most remote points of the RS-485 network. To connect them to the Inline terminal, use the termination resistor installed in the device (see Figure 2).

Connection examples are shown in Figures 5 to 7 on pages 9 to 10.

This operating mode supports the half-duplex transmission. Make sure that only one device is transmitting data at a time.

In order to be able to guarantee a defined cable idle state, the data cable polarization is contained in the terminal.

6.2   RS-422

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

Use a data cable with commonly shielded twisted pairs to connect the devices. Equip this data cable with a termination resistor at each device. To connect them to the Inline terminal, use the termination resistor installed in the device (see Figure 2).

Connection examples are shown in Figures 8 to 9 on pages 10 to 11.

This operating mode supports full-duplex transmission.

7   Connection notes

ℹ️ 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).

The two connection options allow to connect one cable shield side to FE directly and the other capacitively - without additional effort. In this way, you can prevent ground loops occurring if a shield is connected directly to FE on both sides.

No matter at what side you connect the shield, you must connect all wires to connector 2.

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

7.1   Shield connected capacitively to FE
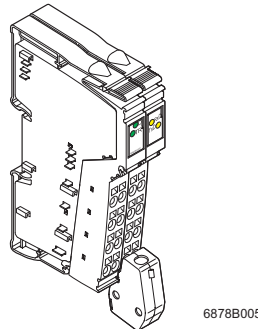

6878B005

Figure 3   Position of the shield connector to connect the shield capacitively to FE

7.2   Shield connected directly to FE

6878B006
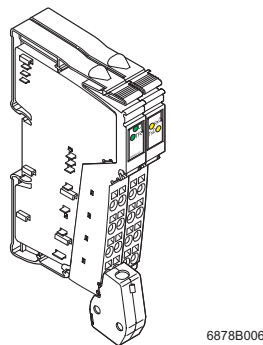
Figure 4   Position of the shield connector to connect the shield directly to FE

8 Connection examples

8.1 Comparison of the connection examples in Figure 5 to Figure 9

Operating mode	Special remark	Shield connection	Comment	Required connectors per **IB IL RS 485/422-PRO**	Example in
RS-485	IB IL RS 485/422-PRO terminal as network end point	Capacitive	Termination resistor required	Connector set	Figure 5
RS-485	IB IL RS 485/422-PRO in the network center	Capacitive/ direct	–	2 shield connectors	Figure 6
RS-485	IB IL RS 485/422-PRO terminal as network end point	Direct	Termination resistor required	Connector set	Figure 7
RS-422	Shield connected capacitively to FE	Capacitive	Termination resistor required	Connector set	Figure 8
RS-422	Shield connected directly	Direct	Termination resistor required	Connector set	Figure 9

> **i** The connector set consists of a shield connector and a standard connector.

For connection, please observe the following notes:

– 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 equipped with termination resistors at both ends.
– For an RS-422 point-to-point connection, the receive signals on every device must be fitted with termination resistors.

8.2 RS-485: Terminal as network end point

ℹ️ Connect the shield to FE using a capacitor.
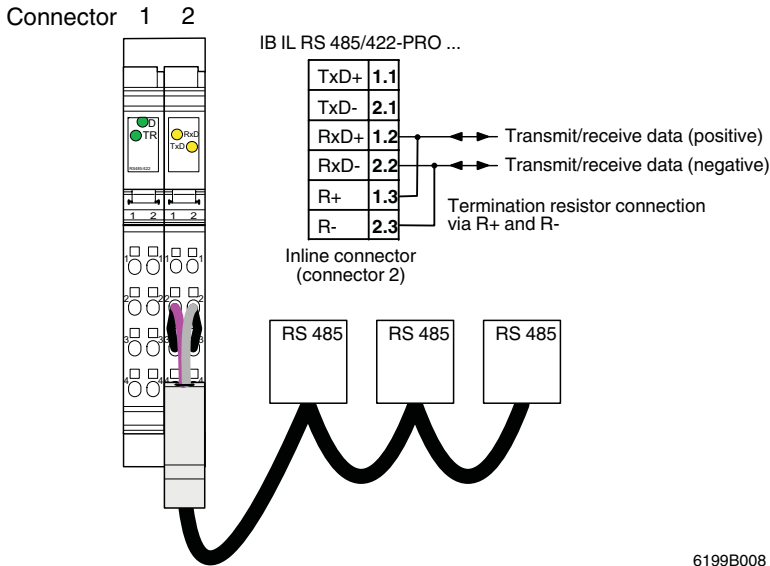
Figure 5   RS-485 interface wiring: IB IL RS 485/422-PRO ... as the end point of a network, data line shield connected to FE using a capacitor

8.3 RS-485: Terminal in the network center
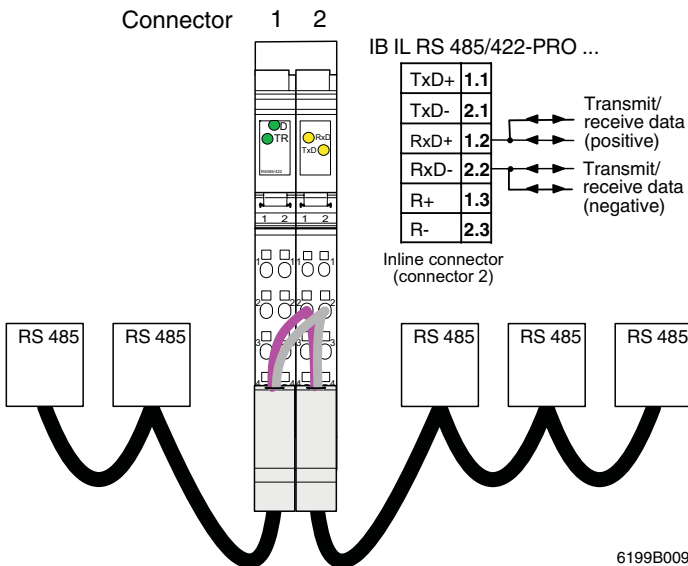
Figure 6   RS-485 interface wiring: IB IL RS 485/422-PRO ... in the center of a network, data line shield directly and connected capacitively to FE

8.4 RS-485: Terminal as network end point

ℹ️ Connect the shield directly to FE.
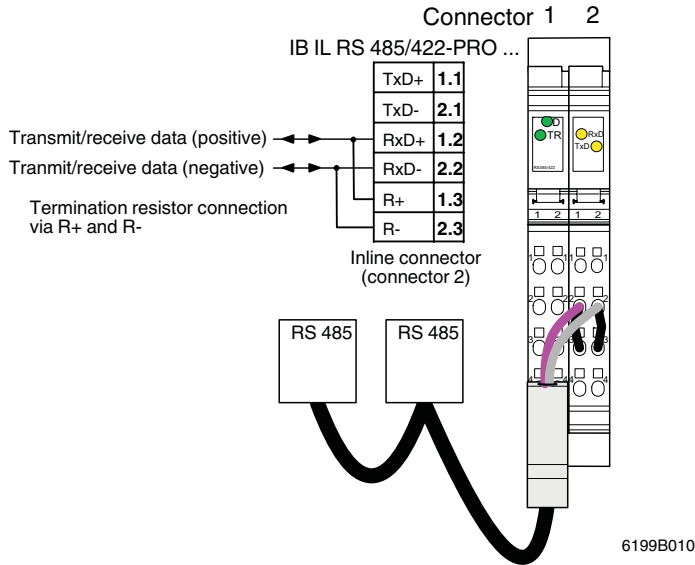
Figure 7   RS-485 interface wiring: IB IL RS 485/422-PRO ... as the end point of a network, data line shield connected directly to FE

8.5 RS-422: Shield connected capacitively to the terminal
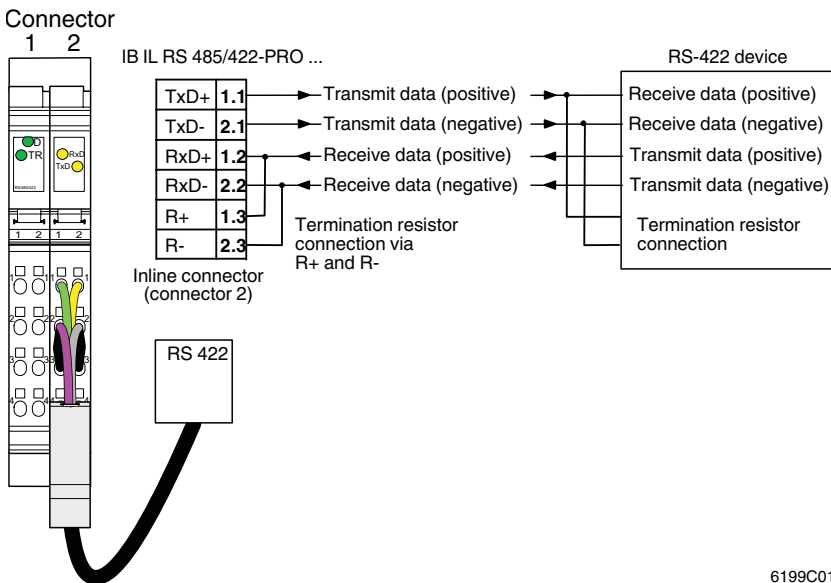
Figure 8   RS-422 interface wiring: Data line shield connected capacitively to FE

8.6   RS-422: Shield directly connected to the terminal
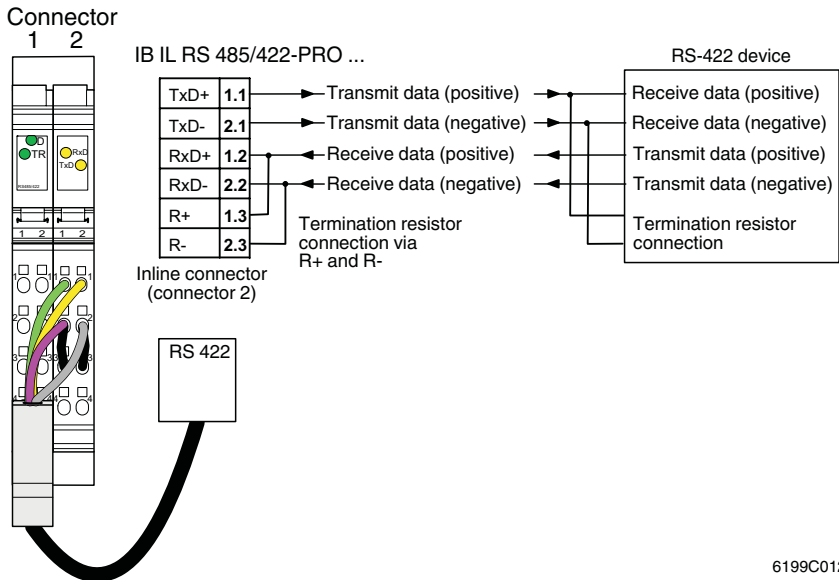
Figure 9   RS-422 interface wiring: Data line shield directly connected to FE

9   Programming data/configuration data

9.1   Local bus (INTERBUS)

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

$\mathbf{i}$	For the programming data/configuration data of other bus systems, please refer to the corresponding 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 peers.

10.1   Overview of supported protocols

Protocol	Receive buffer	Transmit memory	Special features when receiving
Transpar-ent	4096 bytes	1023 bytes	
End-to-end	25 buffers with 11 bytes each	1023 bytes (including end characters)	Two end characters are filtered out
Dual buffer	2 buffers with 11 bytes each	1023 bytes (including end characters)	Only stores the most recently received data, end characters are filtered out
3964R	25 buffers with 11 bytes each	5 buffers with 330 bytes each	Data exchange with software handshake, time monitoring, and checksum
XON/ XOFF	4096 bytes	1023 bytes	Software handshake
Movilink	11 bytes	11 bytes	Data exchange with time monitoring and check sum

ℹ️ The Movilink protocol is available for firmware Version 1.10 or later.

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 further 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. Both delimiters confirm that the serial data has been received without error 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. 25 buffers with 11 bytes each are available. If the buffer size of 11 bytes is exceeded, without the two delimiters being detected, the buffer is overwritten again.

The transmit FIFO consists of 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. In this way, there is always one buffer ready to receive serial data while the second buffer stores the most recently 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 delimiters being detected, the buffer is overwritten again.

When transmitting serial data, the same is valid as 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.

10.5 3964R protocol

This protocol, developed by Siemens, is the most complex. It uses start and end identification, checksum and time monitoring.
5 buffers are available for transmitting, 25 buffers are available for receiving.

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).

> **i** As the RS-485 interface 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 like the transparent protocol, however, using a software handshake.

Data transmission with this protocol is controlled by the XON and XOFF characters. XON is set to 11_{hex} and XOFF to 13_{hex}.

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.

Transmission of serial data is not filtered. Any characters, which occur with the code defined for XON and XOFF, are thus transmitted 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 interface. It can be discussed if using the protocol for the RS-485 interface is useful as the latter 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.

10.7 Movilink protocol

The terminal is the Movilink master in the Movilink protocol. After transmission of a telegram the terminal waits for a maximum of 500 ms for the response. After the response was received, the response data is available in the input data. Only then can the next telegram be sent.

A delimiter is placed ahead of the process data when a Movilink telegram is being sent. The calculated checksum is added behind the process data.

When the data is received, the delimiter and the checksum are filtered out before the data is transmitted over the bus.

Transmit or receive errors are reported when a transmission timeout or a negative checksum comparison occurs.

Transmit and receive buffer have a size of 11 bytes each.

11 Process data

The terminal process image comprises six data words each in input and output direction.

ℹ️ For the assignment of the illustrated (word.bit) or (byte.bit) view to your INTERBUS control system or computer system, please refer to the DB GB IBS SYS ADDRESS data sheet.

The terminal has six process data words.

Word	0		1		2		3		4		5	
Byte in the Motorola format	0	1	2	3	4	5	6	7	8	9	10	11
Byte in the Intel format	1	0	3	2	5	4	7	6	9	8	11	10
OUT	Command/parameter	Data	Data	Data	Data	Data	Data	Data	Data	Data	Data	Data
IN	Status parameter	Data	Data	Data	Data	Data	Data	Data	Data	Data	Data	Data

ℹ️ The byte representation in the Motorola format, also called Big Endian (high order byte at starting address) corresponds to the INTERBUS standard representation. All byte representations in the data sheet have this format.

The byte representation in the Intel format is also called Little Endian (low order byte at starting address).

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

11.1 Word 0 general

Control word

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	Command			OUT parameter				x	x	x	x	x	x	x	x

Status word

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Err	Command			IN parameter				x	x	x	x	x	x	x	x

Err: error

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

Code (Bin)	Code (hex) (with bit 15 = 0)	Command
000	0	Read status bits. Input word 1 contains the number of characters received.
001	1	Transmit characters
010	2	Buffer characters (not in the Movilink protocol)
011	3	Read characters. Parameter = C_{hex}: Read FW version, Parameter = D_{hex}: Read configuration
100	4	Write configuration
101	5	Toggle command 1: transmit characters
110	6	Toggle command 2: Buffer characters (not in the Movilink protocol)
111	7	Toggle command 3: Read characters (not in the Movilink protocol)

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
– Buffer characters (not in the Movilink protocol)
– Read characters (not in the Movilink protocol)

Here, bit 14 is used for toggling.

11.2 "Read Status Bits" command

Control word

	MSB															LSB
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		0	0	0												

Receive error
0: No action
1: Reset receive error

Transmit error
0: No action
1: Reset transmit error

Status word

	MSB															LSB
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		0	0	0												

Number of received data

Reserved

Transmit buffer
0: Empty
1: Not empty

Transmit butter
0: Not full
1: Full

Receive buffer
0: Not full
1: Full

Reserved

Receive buffer
0: Empty
1: Not empty

Receive error
0: No error
1: Error

Transmit error
0: No error
1: Error

6985D001

Figure 10 Format of the process data word 0

> **i** In the Movilink protocol bits 8 to 11 in the status word have no meaning.

Bit/status	Effect	Protocol
Bit 0 = '1'	The receive buffer is not empty, there are characters to be read.	All excluding Movilink
Bit 1 = '1'	The receive error indicates that a 3964R telegram could not be received without error after six transmit attempts by the serial peer or after the block waiting time had elapsed.	3964R
	The receive error indicates that – a parity error occurred while the response was received, – the first character received is not SD2, – the checksum test was negative, – the number of received characters is unequal to the number of transmitted characters.	Movilink
Bit 2 = '1'	The transmit error indicates that a 3964R telegram could not be transmitted from the module to the serial peer without error after six transmit attempts. The telegram was rejected.	3964R
	The transmit error indicates, that the set timeout of 500 ms elapsed and a response has not yet been received.	Movilink
Bit 3 = '1'	Reserved	
Bit 4 = '1'	The receive buffer is full:	Transparent, end-to-end, 3964R, XON/ XOFF
	Transparent and XON/XOFF protocol:	Residual capacity: <15 characters	
	3964R and end-to-end protocol:	Residual capacity: none	

Bit/status	Effect		Protocol
Bit 5 = '1'	The transmit buffer is full: 3964R protocol: Dual buffer, transparent, end-to-end, and XON/XOFF protocol:	Residual capacity: none Residual capacity: ≤30 characters	All excluding Movilink
Bit 6 = '1'	The transmit buffer is not empty, there are characters to be transmitted.		All excluding Movilink
	The transmit buffer is not empty, there are characters to be transmitted. This bit is set during the entire data transmission.		Movilink
Bit 7 = '1'	Reserved		
Bits 8 to 11	Number of characters received. If the code = F_{hex}, more than 14 characters have been received.		All excluding Movilink

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

$\boxed{\mathbf{i}}$ Both error bits (bits 1 and 2) are not automatically reset. They can only be reset by the process data output word.

$\boxed{\mathbf{i}}$ In the transparent and XON/XOFF protocols, the input word 1 contains the total number of characters received.

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 eleven characters (C1 - C11)

Word	0		1		2		3		4		5	
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	$1B_{hex}$	–	–	–	–	–	–	–	–	–	–	–

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

"Transmit Characters" command for the Movilink protocol

After the command 1_{hex} or 5_{hex} was received over the bus, the terminal generates a Movilink telegram. The process data length is specified in the first byte (in the example A_{hex} = 10 bytes). The SD1 start delimiter is placed ahead of the process data in the telegram. The calculated checksum is added behind the process data. The checksum is calculated over the complete telegram.

Process data assignment for the "Transmit Characters" command with 10 characters (C1 - C10)
Output data

Word	0		1		2		3		4		5	
Byte	0	1	2	3	4	5	6	7	8	9	10	11
OUT	$1A_{hex}$	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	–

Movilink telegram

Delimiter	Address	Type	User data	Check-sum

6985A003

The command is only acknowledged over the bus, when either the response of the Movilink peer was received or the timeout period of 500 ms elapsed. When there is a positive acknowledgment (error bit = 0) the process input data contains the response data. Start delimiter and checksum have been filtered out from the response.

The start delimiter of the request telegram (SD1) and the response telegram (SD2) are defined as follows:
SD1 = 02_{hex}, SD2 = $1D_{hex}$.

The error bit (bit 15 in the status word, i.e. in the first word received (Wort 0)) indicates the acknowledgment:

Error bit = 0:
Data transmission is okay, the data behind byte 1 is response data of the Movilink peer.

Error bit = 1:
An error has occurred. It may have several causes:
– The length information in the output data is equal to 0 or larger than 11.
– The Movilink response is invalid;
  The options are as follows:
  – The first character is unequal SD2.
  – The checksum comparison is negative.
  – The number of received characters is unequal to the number of transmitted characters.
– Parity errors occurred while the response was received.
– The timeout period elapsed, which means that no response was received within 500 ms.

Set the toggle bit for the next transfer in the command. If command 1 was used before, command 5 will be sent. Then command 1 again.

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 eleven characters (C1 - C11)

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

11.6 "Write Configuration" command

Process data assignment for the "Write Configuration" command

Output words 0 to 5

Word	0		1		2		3		4		5	
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	1st delimiter	2nd delimiter	Direct baud rate	Reserved	Reserved		Reserved	
IN	40_{hex}	–	–	–	–	–	–	–	–	–	–	–

Output word 1

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

Output type:
0: RS 485
1: RS 422

6985B002

Figure 11 Output word 1 in the "Write Configuration" command

Element value range

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

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

The **error pattern** contains the character that is written into the FIFO, if a character was received with an error (not valid for the 3964R protocol). Reasons are e.g. parity errors, exceeded value range, superimposed noise. In the transparent and XON/XOFF protocols, the pattern is also used if the receive FIFO is full and further characters are received.

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

	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
06_{hex}	Movilink

i	The Movilink protocol is available for firmware Version 1.10 or later.

	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
$\mathbf{07_{hex}}$	**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

i	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 $0D_{hex}$, $0E_{hex}$ and $0F_{hex}$ baud rate codes in the output word 1, see "Direct Baud Rate (DBC)" on page 19.

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	None	1	*8 data bits, without parity, 1 stop bit*
05_{hex}	7	None	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	None	2	*8 data bits, without parity, 2 stop bits*
$0B_{hex}$	7	None	2	*7 data bits, without parity, 2 stop bits*

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

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

ℹ	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.

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

Direct Baud Rate (DBC)

Choose direct programming of the baud rate in the output word 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:

$$\text{Baud rate = basic clock / (DBC + 1)}$$

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

$$\text{DBC = basic clock / baud rate - 1}$$

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

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

An example for the first four output words is:
40 00 00 D2 00 00 1F $_{hex}$.

ℹ	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.

Example for a configuration with transparent protocol

– 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

Word	0		1		2		3		4		5	
Byte	0	1	2	3	4	5	6	7	8	9	10	11
Meaning	40_{hex}	Error pattern	Output type/protocol	Baud rate/data width	1st delimiter	2nd delimiter	Direct baud rate	Reserved	Reserved		Reserved	
OUT	40	00	00	83	00	00	00	–	–	–	–	–
IN	40	–	–	–	–	–	–	–	–	–	–	–

Example for a configuration with Movilink protocol

– Default: Movilink protocol
– Baud rate: 9600 baud
– Data width: 8 data bits with odd parity and one stop bit
– Configuration (in hex): 4000 0672 0000 0000

11.7 "Read Configuration" command

Process data assignment for the "Read Configuration" command

Word	0		1		2		3		4		5	
Byte	0	1	2	3	4	5	6	7	8	9	10	11
OUT	$3D_{hex}$	×	×	×	×	×	×	×	×	×	×	×
IN	$3D_{hex}$	Error pattern	Output type/protocol	Baud rate/data width	1st delimiter	2 nd delimiter	Direct baud rate	00	00	00	00	00

11.8 "Read Firmware Version" command

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

15	14	13	12	11	10	9	8	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 IB IL RS 485/422 ....