

IB IL RS 485/422 ...

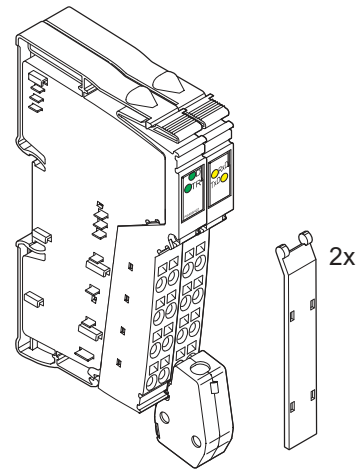
Inline terminal for serial data transmission

Firmware Version 1.20 or later

AUTOMATION

Data sheet
6878_en_04

© PHOENIX CONTACT - 03/2009



6199A002

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.

Features

- A serial input and output channel (RS-485 or RS-422)
- Various protocols supported
- Transmission speed can be set up to 38,400 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 PCP services
- Diagnostic and status indicators



This data sheet only is valid with the IL SYS INST UM E user manual (see "Ordering data" on page 4).



Make sure you always use the latest documentation.
It can be downloaded at www.phoenixcontact.net/download.



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

Table of contents

1	Description.....	1
2	Ordering data	4
3	Technical data	4
4	Internal basic circuit diagram	7
5	Diagnostic and status indicators and terminal point assignment.....	8
5.1	Local diagnostic and status indicators.....	8
5.2	Function identification.....	8
5.3	Terminal point assignment in half duplex mode (RS-485)	8
5.4	Terminal point assignment in full duplex mode (RS-422)	8
6	Installation instructions	9
7	Serial interface.....	9
7.1	RS-485	9
7.2	RS-422	9
8	Connection notes.....	9
8.1	Capacitor between shield and FE.....	9
8.2	Shield connected directly to FE.....	10
9	Connection examples	10
9.1	Comparison of the wiring examples in Figure 5 to Figure 9	10
9.2	RS-485: IB IL RS 485/422 terminal as the end point of a network	11
9.3	RS-485: IB IL RS 485/422 terminal in the center of a network	11
9.4	RS-485: IB IL RS 485/422 terminal as the end point of a network	12
9.5	RS-422: Shield connected using a capacitor to the IB IL RS 485/422 terminal.....	12
9.6	RS-422: Shield connected directly to the IB IL RS 485/422 terminal	13
10	Programming data/configuration data.....	14
10.1	Local bus (INTERBUS).....	14
10.2	Other bus systems	14
11	Data storage and transmission	14
11.1	Overview of supported protocols.....	14
11.2	Transparent protocol	15
11.3	End-to-end protocol.....	15
11.4	Dual buffer protocol	15
11.5	3964R protocol.....	15
11.6	XON/XOFF protocol	16
11.7	Modbus RTU (Remote Terminal Unit)	16
11.8	Modbus ASCII	16
11.9	Time monitoring for Modbus.....	16

12	PCP communication	17
12.1	Object dictionary	17
12.2	Object description	17
12.3	Error messages in PCP mode	24
13	Process data.....	25
13.1	Assignment of the process data output word (~control word).....	25
13.2	Format of the process data output word (~control word)	25
13.3	Assignment of the process data input word (~status word)	26
13.4	Format of the process data input word (~status word).....	27

2 Ordering data

Products

Description	Type	Order No.	Pcs./Pkt.
Inline terminal for serial data transmission; transmission speed of 500 kbps; including connectors and labeling fields	IB IL RS 485/422-PAC	2861933	1
Inline terminal for serial data transmission; transmission speed of 500 kbps; without accessories	IB IL RS 485/422	2836793	1
Inline terminal for serial data transmission; transmission speed of 2 Mbps; including connectors and labeling fields	IB IL RS 485/422-2MBD-PAC	2862097	1
Inline terminal for serial data transmission; transmission speed of 2 Mbps; without accessories	IB IL RS 485/422-2MBD	2855839	1



The connector set listed below is needed for the complete fitting of the IB IL RS 485/422 and IB IL RS 485/422-2MBD terminals.

Accessories

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

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
"Peripherals Communication Protocol (PCP)" user manual	IBS SYS PCP G4 UM E	2745169	1

3 Technical data

General data

Housing dimensions (width x height x depth)	24.4 mm x 120 mm x 72 mm (with connectors)
Weight	90 g (without connectors), 135 g (with connectors)
Operating mode	Process data mode with 1 word PCP mode with 2 words
Transmission speed	
IB IL RS 485/422-PAC	500 kbps
IB IL RS 485/422	500 kbps
IB IL RS 485/422-2MBD-PAC	2 Mbps
IB IL RS 485/422-2MBD	2 Mbps
Permissible temperature (operation)	-25°C to +55°C
Permissible temperature (storage/transport)	-25°C to +85°C
Permissible humidity (operation/storage/transport)	75% to 85% according to DIN EN 61131-2
Permissible air pressure (operation)	80 kPa to 106 kPa (up to 2000 m above sea level)
Permissible air pressure (storage/transport)	70 kPa to 106 kPa (up to 3000 m above sea level)
Degree of protection	IP20 according to IEC 60529
Protection class	Class 3 according to EN 61131-2, IEC 61131-2
Connection data for Inline connectors	
Connection method	Spring-cage terminals
Conductor cross-section	0.08 mm ² to 1.5 mm ² (solid or stranded), 28 - 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
Cable termination resistance	120 Ω, typical
Permissible input differential voltage	±5.7 V, maximum



Owing to the 120 Ω termination resistor, 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

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	Approximately 1.275 W, typical; 1.950 W, maximum*	Approximately 1.388 W, typical; 1.950 W, maximum*

* All serial interface connections short circuited.



NOTE: Damage to the device

A continuous short circuit can damage the device.



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

Supply of the module electronics through the bus coupler

Connection method	Potential routing
-------------------	-------------------

Power dissipation

	500 kbps	2 Mbps
Power dissipation of the electronics P_{TOT}	$P_{TOT} = 1.2$ W	$P_{TOT} = 1.4$ W
Power dissipation of the housing P_{HO}	1.2 W, maximum*	1.4 W, maximum*

* Within the permissible operating temperature

Limitation of simultaneity, derating

	500 kbps	2 Mbps
$T_A \leq 50^\circ\text{C}$	No limitation of simultaneity, no derating	No limitation of simultaneity, no derating
$T_A > 50^\circ\text{C}$	No limitation of simultaneity, no derating	$I_Q = 4$ A I_Q : Total diagonal routing current $I_M/I_S/GND$

Safety equipment

None

Electrical isolation/isolation of the voltage areas

The 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 cables 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

5 V supply incoming remote bus/7.5 V supply (bus logic)

5 V supply outgoing remote bus/7.5 V supply (bus logic)

RS-485/422 interface/7.5 V supply (bus logic)

RS-485/422 interface/24 V supply (I/O)

RS-485/422 interface/functional earth ground

7.5 V supply (bus logic)/24 V supply (I/O)

7.5 V supply (bus logic)/functional earth ground

24 V supply (I/O)/functional earth ground

- Test voltage

500 V AC, 50 Hz, 1 min.

500 V AC, 50 Hz, 1 min.

500 V AC, 50 Hz, 1 min.

500 V AC, 50 Hz, 1 min.

500 V AC, 50 Hz, 1 min.

500 V AC, 50 Hz, 1 min.

500 V AC, 50 Hz, 1 min.

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.phoenixcontact.net/catalog.

4 Internal basic circuit diagram

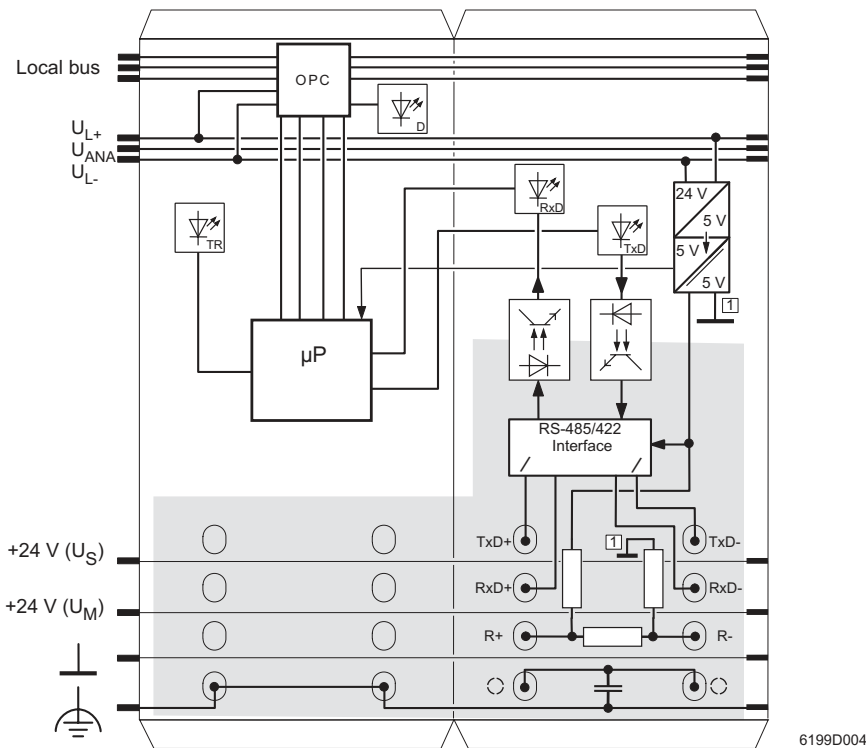












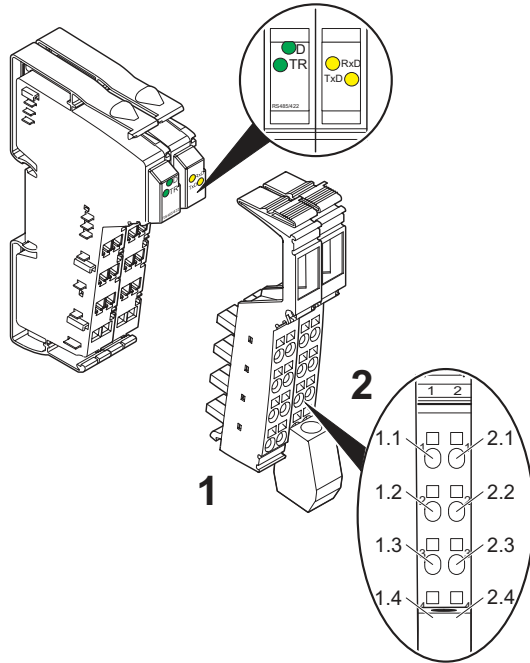
Figure 1 Internal wiring of the terminal points

Key:

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

 Other symbols used are explained in the IL SYS INST UM E user manual.

5 Diagnostic and status indicators and terminal point assignment



61990003

Figure 2 Diagnostic and status indicators and terminal point assignment

5.1 Local diagnostic and status indicators

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

5.2 Function identification

Orange

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

5.3 Terminal point assignment in half duplex mode (RS-485)

Connector	Terminal point	Signal	Assignment	Data direction
1	1.4, 2.4	FE	Functional earth ground	
	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	

5.4 Terminal point assignment in full duplex mode (RS-422)

Connector	Terminal point	Signal	Assignment	Data direction
1	1.4, 2.4	FE	Functional earth ground	
	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	

6 Installation instructions



Please observe the "Connection notes" on page 9.

7 Serial interface

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

7.1 RS-485

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

Use a twisted pair, common shielded data cable to connect the devices. Fit this data cable with a termination resistor at the two furthest points of the RS-485 network. For connection to the Inline terminal, use the termination resistor installed in the device (see Figure 1).

Connection examples are shown in Figure 5 to Figure 7 on page 11 to page 12.

This operating mode supports half duplex transmission. Make sure that only one device is transmitting data at a time. To ensure a defined idle cable state, data cable polarization is included in the IB IL RS 485/422 terminal.

7.2 RS-422

In RS-422 mode, a point-to-point connection can be established.

Use a twisted pair, common shielded data cable to connect the devices. Fit this data cable with a termination resistor at each device. For connection to the Inline terminal, use the termination resistor installed in the device (see Figure 1).

Connection examples are shown in Figure 8 on page 12 and Figure 9 on page 13.

This operating mode supports full duplex transmission.

8 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).

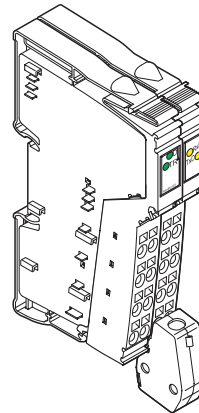
With the two connection options, you can connect one side of the cable shield directly and one side using a capacitor to FE without any additional effort. In this way, you can prevent ground loops, which can occur if a shield is connected directly to FE on both sides.

Regardless of the side to which the shield is connected, all wires must be wired to connector 2.



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

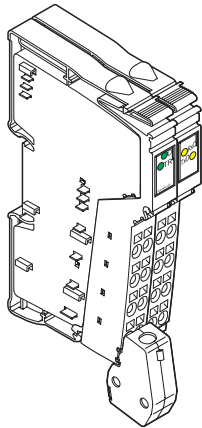
8.1 Capacitor between shield and FE



6878B005

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

8.2 Shield connected directly to FE



6878B006

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

9 Connection examples

9.1 Comparison of the wiring examples in Figure 5 to Figure 9

Operating mode	Special feature	Shield connection	Remark	Connectors required per IB IL RS 485/422	Example in
RS-485	IB IL RS 485/422 as the end point of a network	Capacitive	Termination resistor required	Connector set	Figure 5
RS-485	IB IL RS 485/422 in the center of a network	Capacitive/ direct	–	2 shield connectors	Figure 6
RS-485	IB IL RS 485/422 as the end point of a network	Direct	Termination resistor required	Connector set	Figure 7
RS-422	Shield connected using a capacitor	Capacitive	Termination resistor required	Connector set	Figure 8
RS-422	Shield connected directly	Direct	Termination resistor required	Connector set	Figure 9



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

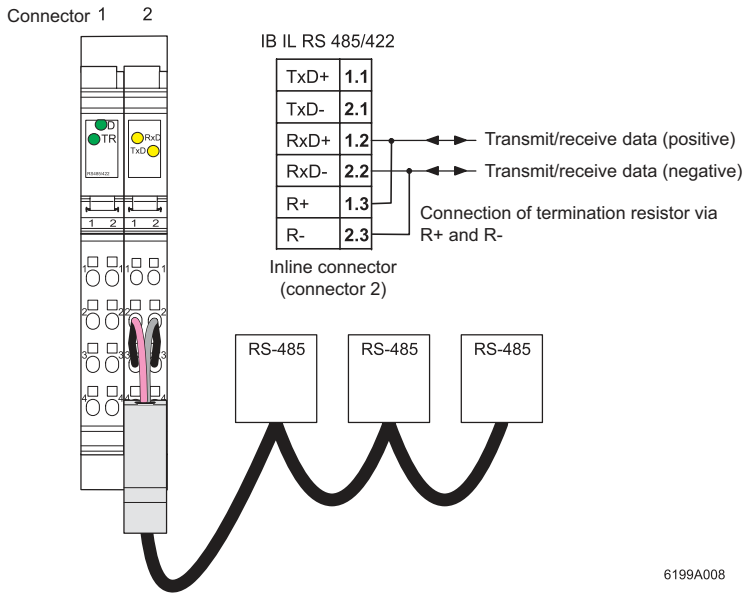
When wiring, please observe the following notes:

- 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 a termination resistor at both ends.
- For an RS-422 point-to-point connection, the receive signals on every device must have a termination resistor.

9.2 RS-485: IB IL RS 485/422 terminal as the end point of a network



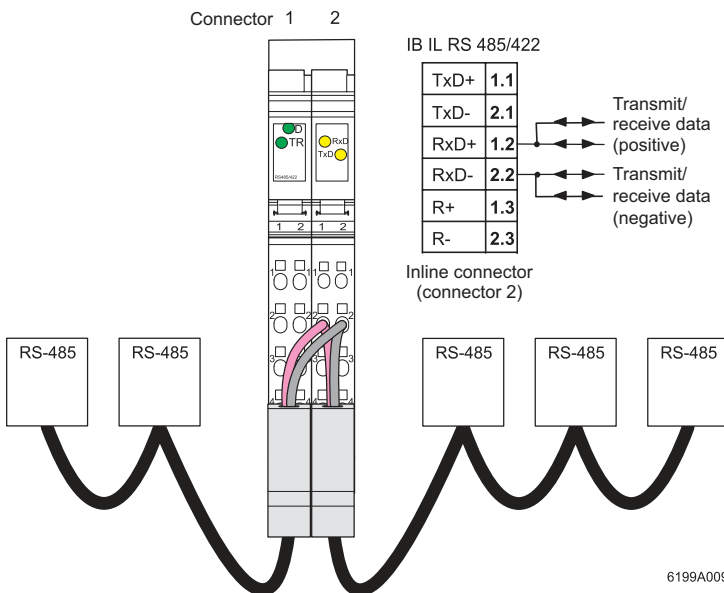
Connect the shield to FE using a capacitor.



6199A008

Figure 5 RS-485 interface wiring: IB IL RS 485/422 as the end point of a network, capacitor between data cable shield and FE

9.3 RS-485: IB IL RS 485/422 terminal in the center of a network



6199A009

Figure 6 RS-485 interface wiring: IB IL RS 485/422 in the center of a network, data cable shield connected directly and using a capacitor to FE

9.4 RS-485: IB IL RS 485/422 terminal as the end point of a network



Connect the shield directly to FE.

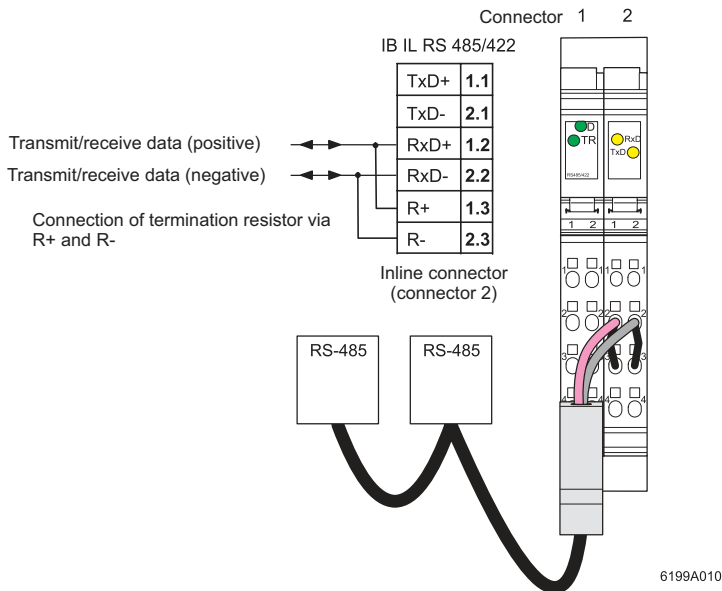


Figure 7 RS-485 interface wiring: IB IL RS 485/422 as the end point of a network, data cable shield connected directly to FE

9.5 RS-422: Shield connected using a capacitor to the IB IL RS 485/422 terminal

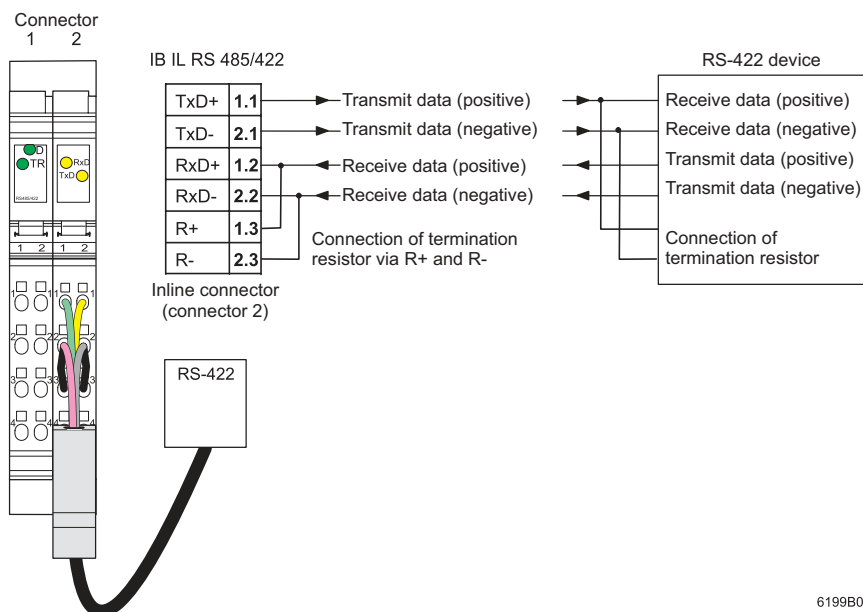
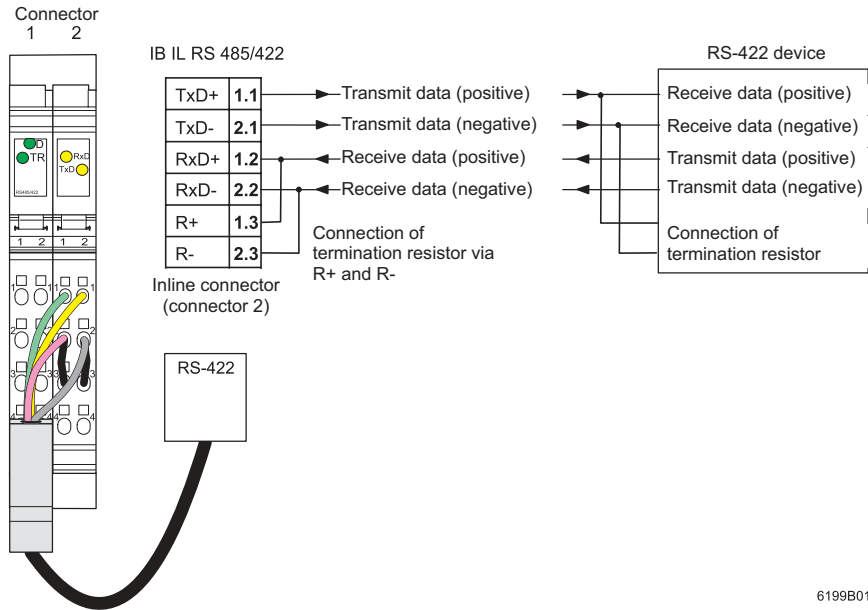


Figure 8 RS-422 interface wiring: Data cable shield connected using a capacitor to FE

9.6 RS-422: Shield connected directly to the IB IL RS 485/422 terminal



6199B012

Figure 9 RS-422 interface wiring: Data cable shield connected directly to FE

10 Programming data/configuration data

10.1 Local bus (INTERBUS)

ID code	DC _{hex} (220 _{dec})
Length code	01 _{hex}
Process data channel	16 bits
Input address area	2 bytes
Output address area	2 bytes
Parameter channel (PCP)	4 bytes
Register length (bus)	6 bytes

10.2 Other bus systems



For the programming data of other bus systems, please refer to the corresponding electronic device data sheet, e.g., GSD, EDS.

11 Data storage and transmission

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

11.1 Overview of supported protocols

Protocol	Receive memory	Transmit memory	Special features when receiving
Transparent	4096 bytes	1023 bytes	
End-to-end	25 buffers with 58 bytes each	1023 bytes (including end characters)	Two end characters are filtered out
Dual buffer	2 buffers with 58 bytes each	1023 bytes (including end characters)	Only stores the most recently received data, end characters are filtered out
3964R	25 buffers with 58 bytes each	15 buffers with 58 bytes each	Data exchange with software handshake, time monitoring, and checksum
XON/XOFF	4096 bytes	1023 bytes	Software handshake
Modbus RTU	25 buffers with 58 bytes each	15 buffers with 58 bytes each	Time monitoring and checksum
Modbus ASCII	25 buffers with 58 bytes each	15 buffers with 58 bytes each	Time monitoring and checksum

11.2 Transparent protocol

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

The transmit FIFO (First In, First Out 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 subsequent characters are ignored.

11.3 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 when the terminal is configured using the INIT-TABLE object.

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 delimiters confirm that the serial data has been received without error and the maximum data length of 58 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 of 58 bytes each are available. If the buffer size of 58 bytes is exceeded without the two delimiters being detected, the buffer is overwritten again. Depending on the INIT-TABLE object, subindex 0C_{hex} (rotation switch), there are two variants.

Variant 1 (0C_{hex} = 0; default setting):

Only the **new** data is available in the overwritten buffer, i.e., any existing data from the previous cycle is rejected (no rotation).

Variant 2 (0C_{hex} = 1): The buffer is re-written character by character. If the two delimiters are detected, the **new** characters **and** the **remaining** characters from the previous cycle are available in the overwritten buffer (rotation).

The transmit FIFO is 1023 bytes in size. The delimiters are attached to, and stored with, the data to be sent.

11.4 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, as in the end-to-end protocol.

As soon as a new data block is received, the previous one is overwritten. This is achieved using two buffers, which are written to alternately. This means that one buffer will always be available to receive serial data, while the other stores 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 58 bytes is exceeded without the two end characters (delimiters) being detected, the buffer is overwritten again. Depending on the INIT-TABLE object, subindex 0C_{hex} (rotation switch), there are two variants.

Variant 1 (0C_{hex} = 0): **Only** the **new** data is available in the overwritten buffer, i.e., any existing data from the previous cycle is rejected (no rotation).

Variant 2 (0C_{hex} = 1): The buffer is re-written character by character. If the two delimiters are detected, the **new** characters **and** the **remaining** characters from the previous cycle are available in the overwritten buffer (rotation).

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.

11.5 3964R protocol

This protocol, developed by Siemens, is the most complex. It uses start and end identifiers, a checksum, and time monitoring.

15 buffers are available for transmitting and 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).

11.6 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}. These characters can also be defined when the terminal is configured using the INIT-TABLE object.

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

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

Serial data is not filtered when it is transmitted. So any characters that occur with the code defined for XON and XOFF are 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.

11.7 Modbus RTU (Remote Terminal Unit)

Telegram structure in RTU mode:

Address	Function	Data	CRC check
1 byte	1 byte	0 ... 56 bytes	2 bytes (EXOR)

A frame (telegram) is finished if there is a break in transmission of at least 3.5 characters (referred to as t3.5). If there is a break in frame transmission of more than 1.5 characters (referred to as t1.5), the message is considered incomplete and is thus rejected.

In the Modbus environment, two timeouts are always required: t1.5 and t3.5. For baud rates above 19,200 baud, the timeouts are specified as t1.5 = 1.5 ms and t3.5 = 3.0 ms.

11.8 Modbus ASCII

Telegram structure in ASCII mode:

Start	Address	Function	Data	CRC check	End
1 ASCII character (:)	2 ASCII characters	2 ASCII characters	2 x (0 ... 56) ASCII characters	2 ASCII characters	2 ASCII characters (CR, LF)

In Modbus ASCII mode, each hex value is converted into two ASCII characters. In this way, the data throughput is halved compared to RTU mode.

Example: The number 5B_{hex} is split into ASCII characters 5 and B. The corresponding ASCII codes are 35_{hex} (5) and 42_{hex} (B).

Both end characters can be modified using the first delimiter and second delimiter entries in the INIT-TABLE object.

11.9 Time monitoring for Modbus

Time monitoring is valid for Modbus RTU as well as for Modbus ASCII. After a telegram is sent, a response is expected within the response time of 2.5 s. After a timeout, the transmit error bit is set in the process data.

The broadcast telegram is an exception where no slave responds and thus the response timeout is not maintained. Instead, the turnaround delay of 200 ms is observed. The next telegram may not be sent until this time has elapsed.

12 PCP communication



For information about PCP communication, please refer to the IBS SYS PCP G4 UM E user manual.

By default upon delivery, the terminal is configured for data transmission according to the parameters on page 22. You can configure the terminal to meet the requirements of your application.

The terminal is configured in PCP mode using the INIT-TABLE object.



The IBS CMD (for standard controller boards) and PC WorX (for Field Controllers (FC) and Remote Field Controllers (RFC)) programs are available for the configuration and parameterization of your INTERBUS system.

For additional information, please refer to the "IBS CMD SWT G4 UM E" user manual and the Quick Start Guide for your version of PC WorX.

Parameter records or text strings are transmitted to or from a connected I/O device in PCP mode using the SERIAL-DATA object.

12.1 Object dictionary

Index	Data type	N	L	Meaning	Object name	Rights
5FC1 _{hex}	Var of unsigned 8	1	1	Module start indicator	START-IND	rd/wr
5FE0 _{hex}	String var of octet string	1	58	Send/receive serial data	SERIAL-DATA	rd/wr
5FFF _{hex}	Array of unsigned 8	20	1	IB IL RS 485/422 terminal configuration	INIT-TABLE	rd/wr

N: Number of elements rd: Read access permitted
L: Length of an element in bytes wr: Write access permitted

12.2 Object description

START-IND object

The object indicates whether or not the terminal was restarted.

Following a power up, the byte will always have a value of 01_{hex}. For a restart to be detected, the byte must be set to 00_{hex} by the application. If it is then set to 01_{hex} again **by the terminal**, this indicates that the terminal has been restarted.



The object is of no significance to the function of the terminal.

Object description:

Object	START-IND	
Access	Read, write	
Data type	Var of unsigned 8	1 byte
Index	5FC1 _{hex}	
Subindex	00 _{hex}	
Length (bytes)	01 _{hex}	
Data	Module start indicator	

Index	Data type	N	L	Meaning	Object name	Rights
5FC1 _{hex}	Var of unsigned 8	1	1	Module start indicator	START-IND	rd/wr
5FE0 _{hex}	String var of octet string	1	58	Send/receive serial data	SERIAL-DATA	rd/wr
5FFF _{hex}	Array of unsigned 8	20	1	IB IL RS 485/422 terminal configuration	INIT-TABLE	rd/wr

Value range of the object:

Module start indicator		
Code	Meaning	Representation in CMD/PC WorX
00 _{hex}	Reset power up message	<i>Reset power up message</i>
01 _{hex}	Power up was carried out	<i>Power up completed</i>



If **you** set the object to 01_{hex}, it has **no effect** on the function of the terminal.

A power up cannot be triggered in this way.

However, it is then not possible to detect a terminal restart.

Here, the parameters are:

Error_Class: 8
 Error_Code: 0
 Additional_Code: 0022_{hex}
 Meaning: No character of the serial data will be accepted from this service.

A write service without user data (length = 0) will be answered with a negative confirmation.

Here, the parameters are:

Error_Class: 8
 Error_Code: 0
 Additional_Code: 0030_{hex}
 Meaning: Value is out of range

SERIAL-DATA object

This object is used to send and receive serial data.

Object description:

Object	SERIAL-DATA	
Access	Read, write	
Data type	String var of octet string	1 x 58 bytes
Index	5FE0 _{hex}	
Subindex	00 _{hex} 00 _{hex} : 3A _{hex}	(Only access to all data possible)
Length (bytes)	00 _{hex} : 3A _{hex}	Amount of data in the buffer : Maximum length of the object
Data	Send/receive serial data	

If no serial data is available on a read access, a read response (read service response) is generated with the result (+) and length of 0.

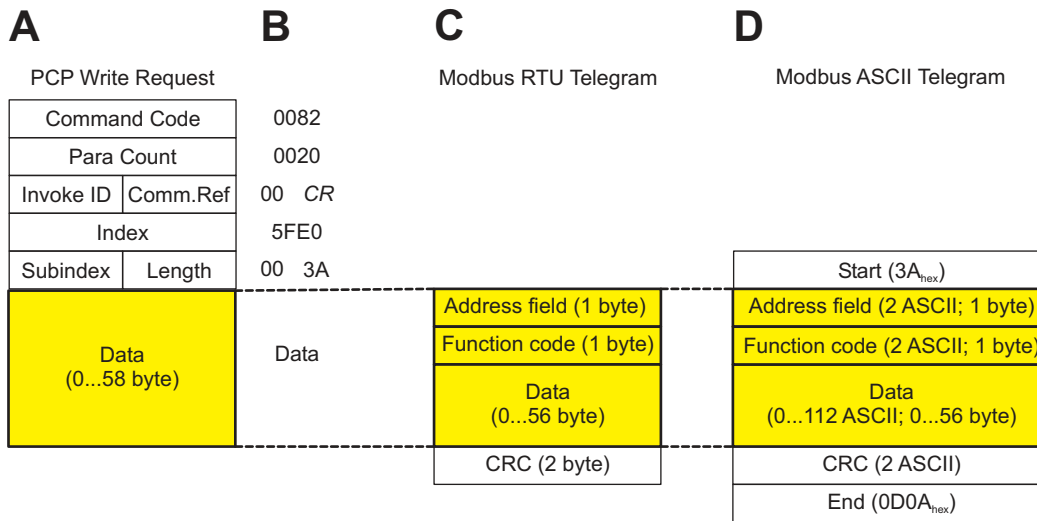
With a protocol data unit (PDU) size of 64 bytes, a maximum of 58 characters can be transmitted.

The length of the read or write service depends on the number of characters to be transmitted. If, for example, 20 characters are to be read, the read response is 24 bytes long (20 bytes of data + 4-byte PCP overhead).

Error messages for the SERIAL-DATA object:

If, during a write service, there is not enough transmit memory available for serial data, the service will be answered with a negative confirmation.

Mapping a write request to the Modbus telegrams:



6199A014

Figure 10 Write request

Key:

- A General representation of a write request
- B Mapped example call for 58 bytes of data
- C Transmitted data for Modbus RTU
Data transmitted using the service is indicated by the color background and bold outline.
CRC is only added to the Modbus device.
- D Transmitted data for Modbus ASCII
Data transmitted using the service is indicated by the color background and bold outline.
Start, CRC, and End are only added to the Modbus device.

A read response service is mapped accordingly.

INIT-TABLE object

The terminal is configured by writing the INIT-TABLE object using a write service.

Object description:

Object	INIT-TABLE	
Access	Read, write	
Data type	Array of unsigned 8	20 x 1 byte
Index	5FFF _{hex}	
Subindex	00 _{hex}	Write all elements
	01 _{hex}	Protocol
	02 _{hex}	Baud rate
	03 _{hex}	Data width
	04 _{hex}	Reserved
	05 _{hex}	Reserved
	06 _{hex}	Error pattern
	07 _{hex}	First delimiter
	08 _{hex}	Second delimiter
	09 _{hex}	3964R priority
	0A _{hex}	Output type
	0B _{hex}	Reserved
	0C _{hex}	Rotation switch
	0D _{hex}	XON pattern
	0E _{hex}	XOFF pattern
	0F _{hex}	Baud rate direct
	10 _{hex}	Basic baud rate direct
	11 _{hex}	Reserved
	:	:
	14 _{hex}	Reserved
Length (bytes)	14 _{hex} subindex 00 _{hex} 01 _{hex} subindex 01 _{hex} to 14 _{hex}	
Data	IB IL RS 485/422 terminal configuration	

INIT-TABLE object elements:

Element		Meaning	Default setting		Data type
dec	hex		Code	Meaning	
1	1	Protocol	00 _{hex}	Transparent	Unsigned 8
2	2	Baud rate	07 _{hex}	9600 baud	Unsigned 8
3	3	Data width	02 _{hex}	8 data bits, even parity, 1 stop bit	Unsigned 8
4	4	Reserved	00 _{hex}		Unsigned 8
5	5	Reserved	00 _{hex}		Unsigned 8
6	6	Error pattern	24 _{hex}	(\$)	Unsigned 8
7	7	First delimiter	0D _{hex}	Carriage return (CR)	Unsigned 8
8	8	Second delimiter	0A _{hex}	Line feed (LF)	Unsigned 8
9	9	3964R priority	00 _{hex}	Low	Unsigned 8
10	A	Output type	01 _{hex}	RS-485	Unsigned 8
11	B	Reserved	00 _{hex}		Unsigned 8
12	C	Rotation switch	00 _{hex}	No rotation	Unsigned 8
13	D	XON pattern	11 _{hex}		Unsigned 8
14	E	XOFF pattern	13 _{hex}		Unsigned 8
15	F	Baud rate direct	00 _{hex}		Unsigned 8
16	10	Basic baud rate direct	00 _{hex}		Unsigned 8
17 - 20	11 - 14	Reserved	00 _{hex}		Unsigned 8



Set all reserved elements to 00_{hex}.

Element value range:



The options in bold are default settings.

Protocol		
Code	Meaning	Representation in CMD/PC WorX
00_{hex}	Transparent	<i>Transparent</i>
01 _{hex}	End-to-end	<i>End-End</i>
02 _{hex}	Dual buffer	<i>Dual buffer</i>
03 _{hex}	3964R	<i>3964R</i>
04 _{hex}	XON/XOFF	<i>XON/XOFF</i>
05 _{hex}	Modbus RTU	<i>Modbus RTU</i>
06 _{hex}	Modbus ASCII	<i>Modbus ASCII</i>

Baud rate	
Code	Value
00 _{hex}	110
01 _{hex}	300
02 _{hex}	600
03 _{hex}	1200
04 _{hex}	1800
05 _{hex}	2400
06 _{hex}	4800
07_{hex}	9600
08 _{hex}	19200
09 _{hex}	38400
FF _{hex}	Direct

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

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

3964R priority		
Code	Meaning	Representation in CMD/PC WorX
00 _{hex}	Low priority	<i>Low priority</i>
01 _{hex}	High priority	<i>High priority</i>

Output type	
Code	Meaning
00 _{hex}	Invalid
01 _{hex}	RS-485
02 _{hex}	RS-422

Rotation switch		
Code	Meaning	Representation in CMD/PC WorX
00 _{hex}	No rotation	No rotation
01 _{hex}	Rotation	<i>Rotation</i>

XON pattern	
Code	Meaning
11 _{hex}	
xx _{hex}	Any character (not the same as XOFF pattern)

Baud rate direct	
Code	Meaning
00 _{hex}	Baud rate = Basic baud rate
xx _{hex}	$n = (\text{Basic baud rate}/\text{baud rate}) - 1$

XOFF pattern	
Code	Meaning
13 _{hex}	
xx _{hex}	Any character (not the same as XON pattern)

Basic baud rate direct	
Code	Meaning
00 _{hex}	500 kbaud
01 _{hex}	62.5 kbaud
02 _{hex}	15625 baud

If the **baud rate** contains the value FF_{hex}, **direct baud rate programming** is selected. Thus the baud rate can be freely

assigned. Element 15 indicates the desired baud rate. Element 16 indicates the basic baud rate.

Example:

The baud rate should be 25,000 baud Element 2 = FF_{hex}

500 kbaud is selected as the basic baud rate Element 16 = 00_{hex}

$n = (500,000 \text{ baud} / 25,000 \text{ baud}) - 1 = 19 = 13_{\text{hex}}$ Element 15 = 13_{hex}

When element 16 is written, direct baud rate programming takes effect.

In theory, direct baud rate programming supports a maximum value of 500 kbaud. Correct operation of the module is tested and ensured up to 38,400 baud. Operation at a higher baud rate depends on the application.

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, for example, be the result of parity errors, exceeded value ranges or noise interference. In the transparent and XON/XOFF protocols, this pattern is also used if the receive FIFO is full and additional characters are received.

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

The value in the **3964R priority** element defines the priority of a device if there is an initialization conflict (several devices attempting to send data simultaneously). The device with priority level 1 has priority over the device with priority level 0.

The **rotation switch** determines how the buffer is re-written when it is full and the two end characters (delimiters) have not been detected.

No rotation:

Only the **new** data is available in the overwritten buffer, i.e., any existing data from the previous cycle is rejected.

Rotation:

The buffer is re-written character by character. If the two delimiters are detected, the **new** characters **and** the **remaining** characters from the previous cycle are available in the overwritten buffer.



If at least one element of the INIT-TABLE object is written, the pointers for the transmit and receive FIFOs will be reset. This means that all transmit and receive data that has not yet been processed will be lost.

XON pattern and **XOFF pattern** contain the control characters for the XON/XOFF protocol. The characters must not be the same.

In the case of the INIT-TABLE object, if one of the elements is written, the transmit and receive buffers of the affected

channel are set to empty. This means that all transmit and receive data that has not yet been processed will be lost.

Error messages for the INIT-TABLE object

- If an element with an invalid value is written during a write service, the service will be acknowledged with a negative confirmation.

Here, the parameters are:

Error_Class: 8
 Error_Code: 0
 Additional_Code: xx30_{hex}
 Meaning: Value is out of range

The high byte of the Additional_Code (xx) contains the number of the affected element. If several elements are affected, the highest number is given. If, for example, the output type element is written with the value 0, an error message with the Additional_Code 0A30_{hex} will be displayed, because the 10th element is faulty.

- For a write request with the subindex 0 (write entire table) in which the XON/XOFF protocol is to be set and the XON pattern is the same as the XOFF pattern, an error message will be generated.

Here, the parameters are:

Error_Class: 8
 Error_Code: 0
 Additional_Code: 0E30_{hex}
 Meaning: Parameterization error

- If a reserved element is written, the value must equal 0, otherwise an error message is generated.

Here, the parameters are:

Error_Class: 8
 Error_Code: 0
 Additional_Code: xx30_{hex}
 Meaning: Parameterization error

12.3 Error messages in PCP mode

The terminal error messages have the parameters:
Error_Class = 8 (device-specific error) and Error_Code = 0
(no communication error).

The exact error cause is indicated by the Additional_Code.
The low byte of the Additional_Code indicates the error
cause. The high byte of the Additional_Code (xx) contains
the number of the affected element. If several elements are
affected, the highest number is given.

The following Additional_Codes can occur on this terminal:

0022_{hex} No character of the serial data will be accepted
from this service.
xx30_{hex} Value is out of range or parameterization error
0000_{hex} Hardware fault



For additional information about error messages
in PCP mode, please refer to the
IBS SYS PCP G4 UM E PCP manual and the
user manual for your controller board.

13 Process data



For the assignment of the (word.bit) or (byte.bit) view illustrated under 13.1 and 13.3 to your **INTERBUS** control or computer system, please refer to the DB GB IBS SYS ADDRESS data sheet.

13.1 Assignment of the process data output word (~control word)

(Word.bit) view	Word	Word 0															
	Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
(Byte.bit) view	Byte	Byte 0								Byte 1							
	Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Assignment		Reserved								Reserved	Reserved	Reserved	Reserved	Execute re-initialization	Reset transmit error	Reset receive error	Reserved



Set all reserved bits to 0.

13.2 Format of the process data output word (~control word)

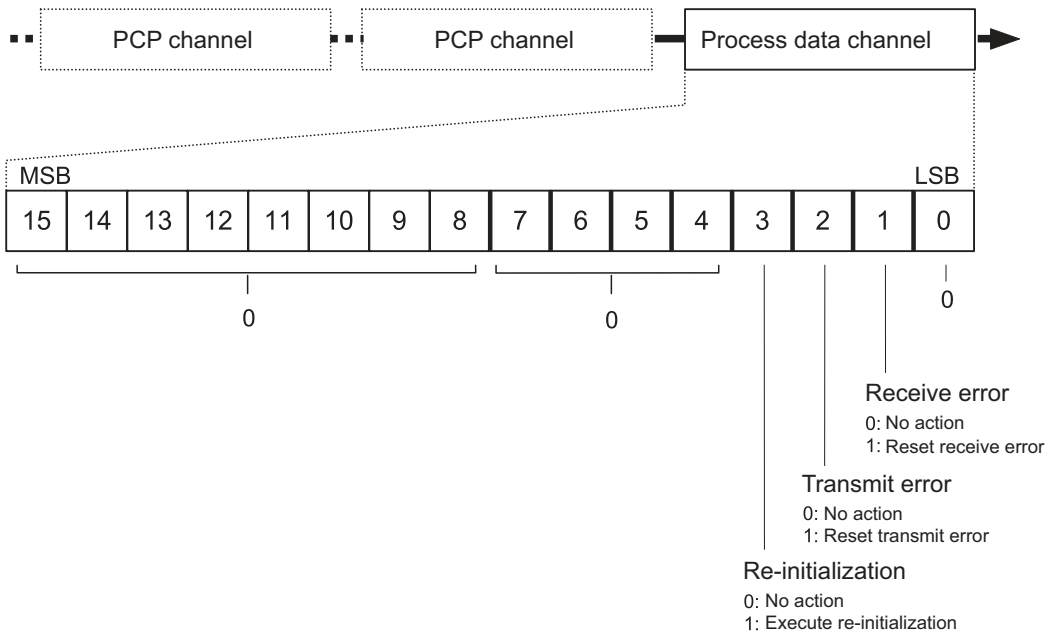


Figure 11 Format of the process data output word

If the control word contains 3C00_{hex}, then the status word returns the firmware version and the module ID.

Example:

Status word (hex)	1	2	0	7
	Firmware Version 1.20			Module ID

Using the module ID, it is possible to clearly identify different terminals with the same ID code.

Terminal	ID code	Module ID
IB IL RS 485/422	DC _{hex} (220 _{dec})	7
IB IL RS 232	DC _{hex} (220 _{dec})	8

13.3 Assignment of the process data input word (~status word)

(Word.bit) view	Word	Word 0															
	Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
(Byte.bit) view	Byte	Byte 0								Byte 1							
	Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Assignment		Number of received characters								Reserved	Transmit buffer not empty	Transmit buffer full	Receive buffer full	Re-initialization executed	Transmit error	Receive error	Receive buffer not empty

13.4 Format of the process data input word (~status word)

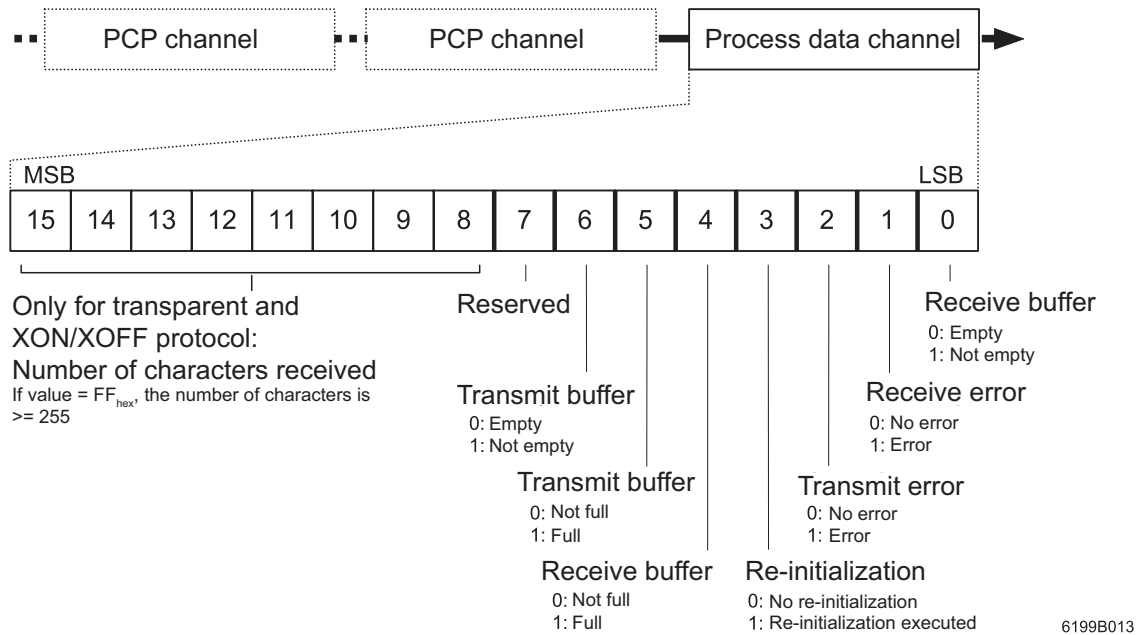


Figure 12 Format of the process data input word

Bit/status	Effect	Protocol
Bit 0 = '1'	The receive buffer is not empty , characters to be read are available.	All
Bit 1 = '1'	Receive error In the case of the 3964R protocol, this bit indicates that a telegram could not be received without error after six transmit attempts by the serial peer or after the block waiting time had elapsed. For Modbus RTU and Modbus ASCII, this bit indicates that the checksum comparison was faulty.	3964R, Modbus RTU, Modbus ASCII
Bit 2 = '1'	Transmit error In the case of the 3964R protocol, this bit indicates that a telegram could not be transmitted by the terminal to the serial peer after six transmit attempts. The telegram was rejected. For Modbus RTU and Modbus ASCII, this bit indicates that the timeout of 2.5 s elapsed without any response being received.	3964R, Modbus RTU, Modbus ASCII
Bit 3 = '1'	A re-initialization was executed; the transmit and receive buffers are now empty.	All
Bit 4 = '1'	The receive buffer is full :	
	Residual capacity: < 15 characters	Transparent, XON/XOFF
	Residual capacity: None	End-to-end, 3964R, Modbus RTU, Modbus ASCII
Bit 5 = '1'	The transmit buffer is full :	
	Residual capacity: ≤ 30 characters	Transparent, XON/XOFF, end-to-end, dual buffer
	Residual capacity: None	3964R, Modbus RTU, Modbus ASCII
Bit 6 = '1'	The transmit buffer is not empty , characters to be sent are available.	All
Bit 7 = '1'	Reserved	



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