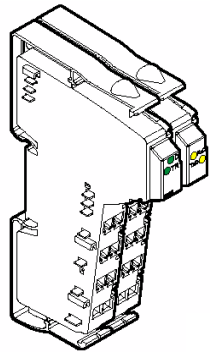
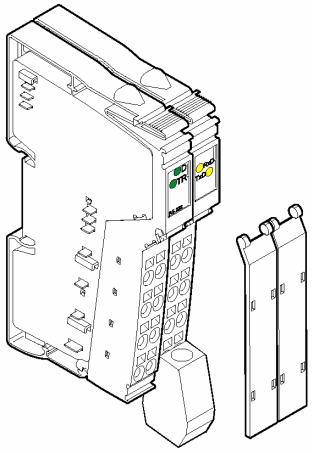


VersaPoint Module

RS-232 Communications Module
IC220BEM232

GFK-2394 *March 2010*

The VersaPoint RS-232 Communications Module IC220BEM232 interfaces serial I/O devices to a VersaPoint I/O Station.



Module with Attached Terminal Strips

Module IC220BEM232 requires a standard and shielded Terminal Strip. See the ordering information below.

Features

- Serial I/O channel (RS-232)
- Supports DTR/CTS handshake
- Supports various protocols
- Baud rate adjustable up to 38400 baud
- Adjustable number of data bits, stop bits, and parity
- Configuration and data exchange using PCP communications services.
- 4 kbyte receive buffer, 1 kbyte transmit buffer
- LED diagnostic and status indicators

Module Specifications

Housing dimensions (width x height x depth)	24.4mm x 120mm x 71.5mm (0.96in. x 4.8in. x 2.86in.)
Operating temperature	-25°C to +55°C, (-13°F to +131°F)
Storage temperature	-25°C to +85°C (-13°F to +185°F)
Operating humidity	75% average, 85% occasionally. Take appropriate measures against increased humidity.
Storage humidity	75% average, 85% occasionally
Degree of protection	IP 20 according to IEC 60529

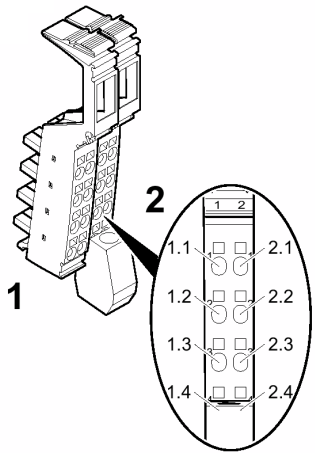
Power Consumption

Communications power U_L	7.5V
Current consumption from local bus U_L	155mA maximum, 225mA maximum
Nominal current consumption at U_S	None
Total power consumption	0.163W typical, 1.688W maximum

Ordering Information

IC220BEM232	Communications Module, RS-232
IC220TBK203	Terminal Strip Set

Terminal Assignments



Terminal Strip 1 (left) Terminal Point Assignments		
1.4, 2.4	FE	Functional earth ground
All other terminal points on this Terminal Strip not used.		
Terminal Strip 2 (right) Terminal Point Assignments		
1.1	RxD	Serial data input
2.1	TxD	Serial data output
1.2	+5V	Control output, internally connected to +5VDC
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

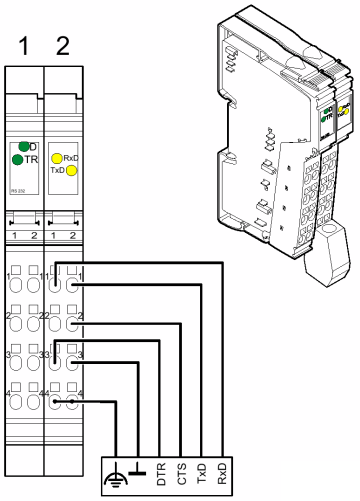
Example Connections

By using points 1.4 and 2.4 on both Terminal Strips, the cable shield can be connected either using a capacitor (Terminal Strip 2), or directly to functional earth ground (Terminal Strip 1). Connecting one side of the cable shield directly and one side using a capacitor to FE can prevent the creation of ground loops that would occur if a shield with two direct connections were placed on FE.

When connecting to the shield via Terminal Strip 1, the shield connector must be connected to the left side of the Terminal Strip. All wires must be connected to Terminal Strip 2.

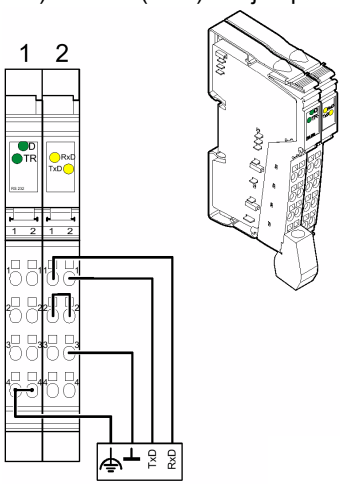
On Terminal Strip 2, point 1.2 (+5V) must only be used to provide the 5-volt signal for the CTS input (point 2.2) for communications without handshaking. A jumper must be inserted between the points 1.2 and 2.2. No other use is permitted.

Example Wiring for 4-Wire Handshake



Example Wiring for No Handshake

Points 1.2 (+5V) and 2.2 (CTS) are jumpered.



RS-232 Interface

The RS-232 interface on the RS-232 Communications Module provides a DTE (data terminal equipment) connection. Terminal Strip 2 point 2.1 (TxD) is always used to transmit, and point 1.1 (RxD) is always used to receive.

Some type of DCE (data communications equipment) is connected to the RS-232 interface as a peer. A DTE can also be connected. Measuring the voltage between the connection points for the TxD and GND signals in the idle state will determine whether the device to be connected is a DTE or DCE device. If the voltage measures approximately –5V, it is a DTE device. If the voltage is approximately 0V, it is a DCE device.

The diagrams that follow show cabling between the RS-232 Communications Module and a typical DTE device with a D-Sub 9-pin or 25-pin female connector. Actual pin assignments of an attached device may be different than those shown, depending on the type of device used. Both GND pins are wired.

In the examples, the shield is connected with a capacitor to FE and the Terminal Strip with the shield is installed on the right side of the RS-232 Communications Module. If the shield is to be directly connected to FE, the Terminal Strip with the shield must be installed on the left side of the module.

Handshake Signals

Any device with an RS-232 interface can be connected to the VersaPoint RS-232 Communications Module. Both the module itself and the connected device can act as transmitted and receiver for exchanging data. Because errors can occur when both devices are sending and receiving simultaneously, the handshake is used to mutually-signal clear- to-receive and clear- to-send. The VersaPoint RS-232 Communications Module supports DTR and CTS handshake signals. Each uses one wire of the connecting cable. The handshake signals are described below.

SIgnal	*Direction*	*Meaning*
CTS	Input	The RS-232 Communications Module receives the CTS signal from the connected device. If CTS is set to High, the module can send data. Exception: 3964R, XON/XOFF protocol.
DTR	Output	The RS-232 Communications Module sends the DTR signal set to high when it is ready to receive. After the connected device sends 4095 characters (4 kbytes), the module's Transmit/Receive Buffer is full, and it sets the DTR signal to low. After more characters are read from by the NIU, the DTR signal is again set to High and the module is ready to receive. 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 memory.

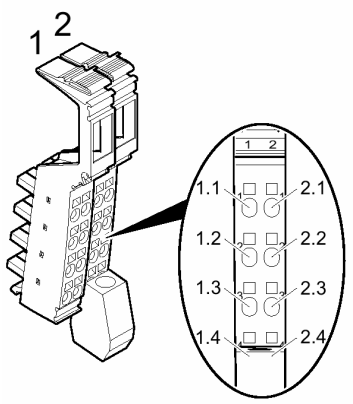
RS-232 Wiring with Handshake for DTE

The TxD, RxD, DTR, and CTS signals are used for a four-wire handshake connection between the RS-232 Communications Module and the device to be connected. Each signal corresponds to one wire in the connecting cable.

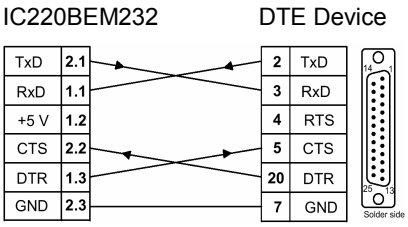
A Terminal Strip is required on the VersaPoint RS-232 Communications module. A 9- or 25-position female connector is required on the opposite side depending on the device to be connected. Both GND pins are wired.

In the diagrams below, the shield connector is connected on the right side of the terminal. In this case, the shield is connected with a capacitor to FE.

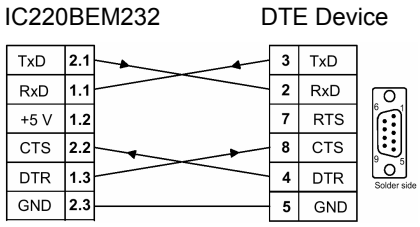
If the shield is to be directly connected to FE, the shield connector must be connected on the left side of the terminal.



Typical Wiring for 25-Pin Connector

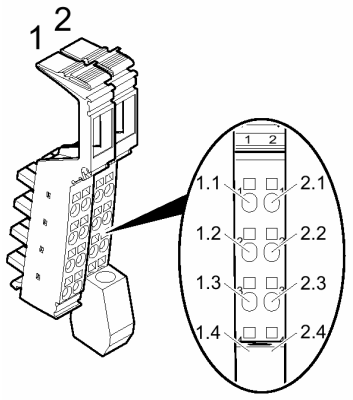


Typical Wiring for 9-Pin Connector

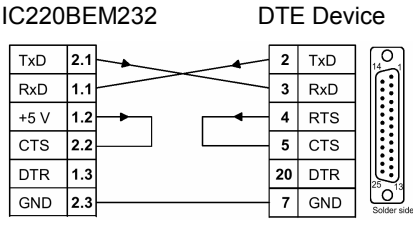


RS-232 Wiring with No Handshake for DTE

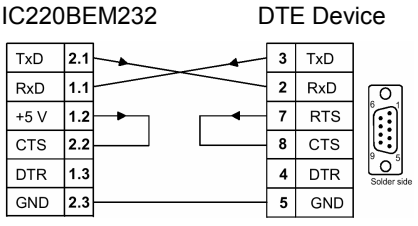
In the case of wiring without handshake, the wires for the TxD and RxD signals and the GND contacts are connected to the RS-232 Communications Module Terminal Strip, and soldered to the female connector on the side of the device to be connected. In addition, a jumper is connected on the Terminal Strip between the points for +5V and CTS signals and on the female connector between the pins for the RTS and CTS signals. That simulates the ready-to-receive state for the peer, and the connected device will always be able to send via the RS-232 interface. The terminal sets the DTR signal to *Low* before the receive FIFO memory overflows. As the DTR signal is not evaluated using 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.



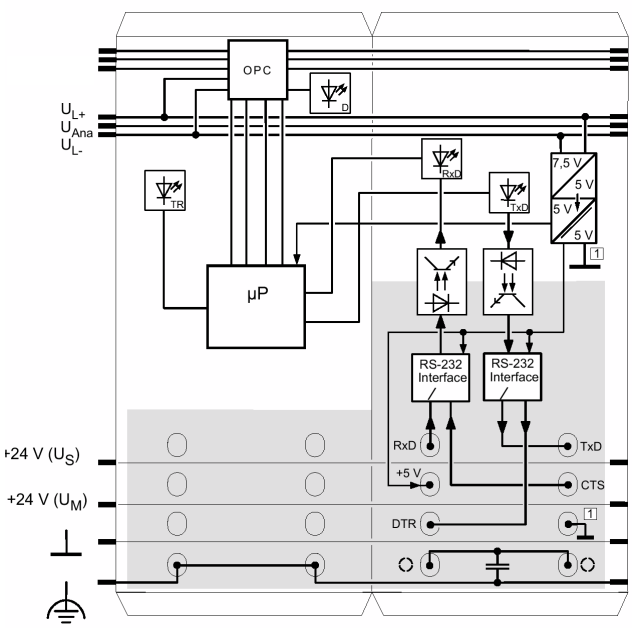
Typical Wiring for 25-Pin Connector



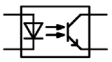
Typical Wiring for 9-Pin Connector



Internal Circuit Diagram of the RS-232 Communications Module




LEDs


Optocoupler


DC/DC converter with electrical isolation

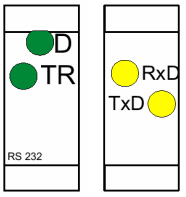
Microprocessor

RS-232 interface


Capacitor

Ground, electrically-isolated from ground of the potential jumper

RS-232 Module LEDS



LED	*Color*	*Meaning*
D	Green	Bus diagnostics
TR	Green	I/O Station communications active
RxD	Yellow	Receiving data from connected device
TxD	Yellow	Transmitting data to connected device

Programming Data

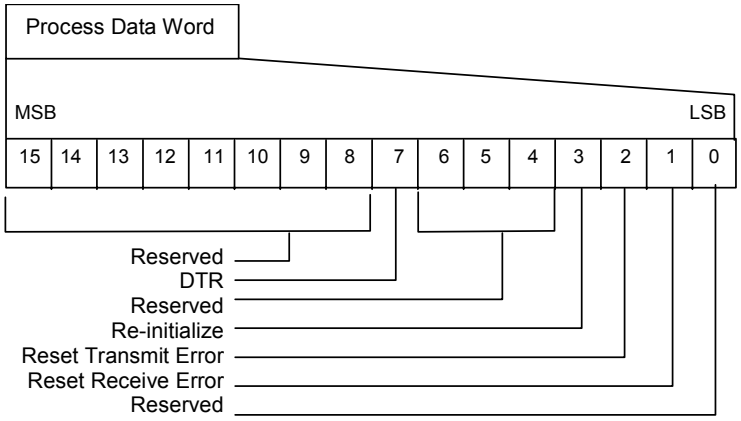
The VersaPoint RS-232 Communications Module uses Process Data outputs and Process Data Inputs to monitor and reset the status of the RS-232 communications.

The module uses PCP Communications, described next, to configure the communications parameters and to exchange the RS-232 data with the I/O Station's NIU.

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

Process Data: Output (Control) Word

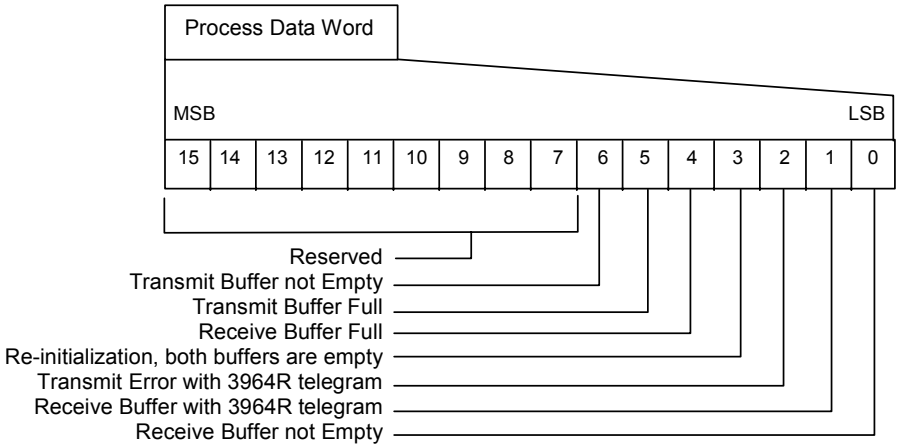
The controller sends the RS-232 Communications Module one word of output data. The basic format of this data is:



The DTR signal can only be generated when "DTR control via process data " is enabled. If the control word contains the value $3C00_{hex}$, the Process Data Input Word (see next page) returns the firmware version.

Process Data: Input (Status) Word

The RS-232 Communications Module sends the controller one word of input data. The basic format of this data is:



Status Bit Definitions

Bit	*Meaning*	*Protocol*
0 = 1	The Receive Buffer is not empty; characters to be read are available	All
1 = 1	The receive error indicates that a 3964R telegram could not be received without error after six attempts by the connected device or after the block waiting time had elapsed.	3964R
2 = 1	The transmit error indicates that a 3964R telegram could not be transmitted by the RS-232 Communications Module to the connected device without error after six sent attempts. The telegram was rejected.	
3 = 1	Reinitialization was executed, Transmit and Receive Buffers are now empty.	All
4 = 1	The Receive Buffer is full. Transparent and XON/XOFF protocol: residual capacity is < 15 characters. 3964 and end-to-end protocol: residual capacity is ≤ 30 characters	Transparent, end-to-end, 3964R, XON/XOFF
5 = 1	The Transmit Buffer is full. 3964R protocol: residual capacity is none Dual buffer, transparent, end-to-end, XON/XOFF protocol: residual capacity is ≤ 30 characters:	All
6 = 1	The Transmit Buffer is not empty, characters to be sent are available.	All
7	Reserved	

PCP Communications

The VersaPoint RS-232 Communications Module uses PCP communications for configuration and data transfer. The module's PCP communications characteristics are described here. Details of using PCP communications are provided in the *NIU User's Manual* for the type of Network Interface Unit being used in the I/O Station.

Object Dictionary of the RS-232 Communications Module

Index (hex)	Data Type	Number of Elements	Length (bytes)	Meaning	Object Name	Access Permitted
5FC1	Variable of Unsigned 8	1	1	Module start indicator	START-IND	read/write
5FE0	String Variable of Octet String	1	58	Transmit/ receive RS-232 data	V24-DATA	read/write
5FFF	Array of Unsigned 8	20	1	Module configuration	INIT-TABLE	read/write

START-IND Object

The START-IND object indicates whether or not the module was restarted. After the voltage has been switched on (power up), the byte always has a value of 01_{hex}. For a restart to be detected, the application must have previously set the byte to 00_{hex}. If it is then reset by the terminal to 01_{hex}, a restart is triggered.

The START-IND object has no meaning as far as the module functions are concerned.

If the application sets the Module Start Indicator to 01(Powerup Completed), it has no effect on the module. Setting this data does not trigger a restart. In addition, if the application sets the Module Start Indicator to 01, it is then not possible for the application to detect a module restart.

START-IND Object Description

Object	START-IND
Access	Read, write
Data Type	Simple variable 1 byte
Index	$5FC1_{hex}$
Subindex	00_{hex}
Length (byte)	01_{hex}
Data	Module start indicator: 00 = Reset powerup message 01 = powerup completed

V24-DATA Object

This object is used for sending and receiving RS-232 data.

V24-DATA Object Description

Object	V24-DATA	
Access	Read, write	
Data Type	String Var of Octet String 1 x 58 bytes	
Index	$5FE0_{hex}$	
Subindex	00_{hex}	(only access to all data possible)
Length (byte)	00_{hex} : $3A_{hex}$	Amount of data present in the buffer : Maximum length of the object
Data	Send/receive RS-232 data	

If no V.24 data is available on a read access, a read response (read service reply) is generated with result (+) and length = 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 the RS-232 characters to be transmitted. If, for example, 20 RS-232 characters are to be read, the read response will be 24 bytes long (20 bytes of RS-232 data + 4 bytes PCP overhead).

Error Messages for the V24-DATA Object

If, during a write service, there is not enough transmit memory available for RS-232 data, the service is answered with a negative confirmation. The parameters are:

Error_Class	8
Error_Code	0
Additional_Code	0022_{hex}
Meaning	No character from the RS-232 data will be accepted from this service.

A write service without user data (length =0) is answered with a negative confirmation. The parameters are:

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

INIT-TABLE Object

The module is configured in PCP mode with a write service using the INIT-TABLE object.

If at least one element of the INIT-TABLE object is written, the pointers for the send and receive FIFO memory are reset. This means that all send and receive data that has not yet been processed is lost.

INIT-TABLE Object Description

Object	INIT-TABLE			
Access	Read, write			
Data Type	Array of Unsigned 8			
Index	$5FFF_{hex}$			
Subindex	00_{hex}	Write all elements	09_{hex}	3964R Priority
	01_{hex}	Protocol	$0A_{hex}$	Output type
	02_{hex}	Baud rate	$0B_{hex}$	DTR control system
	03_{hex}	data width	$0C_{hex}$	Rotation switch
	04_{hex}, 05_{hex}	Reserved	$0D_{hex}$	XON Pattern
	06_{hex}	Error Pattern	$0E_{hex}$	XOFF Pattern
	07_{hex}	First Delimiter	$0F_{hex}$ to 14_{hex}	Reserved
	08_{hex}	Second Delimiter		
Length (bytes)	14_{hex} Subindex 00_{hex} 01_{hex} Subindex 01_{hex} to 14_{hex}			
Data	VersaPoint RS-232 module configuration			

INIT-TABLE Object Elements

Element		*Meaning*	*Default Setting*		*Data Type*
Decimal	*Hex*		*Code (hex)*	*Meaning*	
1	1	Protocol	00	Transparent	Unsigned 8
2	2	Baud rate	07	9600 baud	Unsigned 8
3	3	Data width	02	8 data bits, even parity, 1 stop bit	Unsigned 8
4	4	Reserved	00		Unsigned 8
5	5	Reserved	00		Unsigned 8
6	6	Error pattern	24	($)	Unsigned 8
7	7	First delimiter	0D	Carriage return (CR)	Unsigned 8
8	8	Second delimiter	0A	Line Feed (LF)	Unsigned 8
9	9	3964R priority	00	Low	Unsigned 8
10	A	Output type	00	RS-232	Unsigned 8
11	B	DTR control system	00	Automatic	Unsigned 8
12	C	Rotation switch	00	No rotation	Unsigned 8
13	D	XON pattern	11		Unsigned 8
14	E	XOFF pattern	13		Unsigned 8
15-20	F-14	Reserved	00		Unsigned 8

INIT-TABLE Element Values

Element	*Value (hex)*	*Protocol*
Protocol	00	Transparent
	01	End-to-end
	02	Dual buffer
	03	3964R
	04	XON/XOFF

Element	*Value (hex)*	*Baud*	*Value (hex)*	*Baud*	*Value (hex)*	*Baud*
Baud rate	00	110	04	1800	08	19200
	01	300	05	2400	09	38400
	02	600	06	4800	0A	
	03	1200	07	9600	0B	

Element	*Value (hex)*	*Data Width*	*Value (hex)*	*Data Width*
Data Width	00	7 data bits, even parity, 1 stop bit	06	7 data bits, even parity, 2 stop bits
	01	7 data bits, odd parity, 1 stop bit	07	7 data bits, odd parity, 2 stop bits
	02	8 data bits, even parity, 1 stop bit	08	8 data bits, even parity, 2 stop bits
	03	8 data bits, odd parity, 1 stop bit	09	8 data bits, odd parity, 2 stop bits
	04	8 data bits, without parity, 1 stop bit	0A	8 data bits, without parity, 2 stop bits
	05	7 data bits, without parity, 1 stop bit	0B	7 data bits, without parity, 2 stop bits

Element	*Value (hex)*	*Error Pattern*
Error Pattern	24	$
	(xx)	Any character

The *error pattern* character is written to the FIFO memory if an error occurs while receiving an RS-232 character (this does not apply to the 3964R protocol). This can be the result of, for example, parity errors, exceeded value ranges or noise interferences. In the transparent and XON/XOFF protocols, the pattern is also used if the receive FIFO memory is full and further characters are received.

Element	*Value (hex)*	*First Delimiter*
First Delimiter	0D	Carriage Return (CR)
	(xx)	Any character

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

Element	*Value (hex)*	*Second Delimiter*
Second Delimiter	0A	Lie Feed (LF)
	(xx)	Any character

Element	*Value (hex)*	*3964R Priority*
3964R Priority	00	Low priority
	01	High Priority

3964R priority defines the priority of a device if more than one device attempts to send data simultaneously at initialization. The device with priority 1 has priority over the device with priority 0.

Element	*Value (hex)*	*Output Type*
Output Type	0D	RS-232

Element	*Value (hex)*	*DTR Control*
DTR Control	00	Automatic
	01	Via process data

Element	*Value (hex)*	*Rotation Switch*
Rotation Switch	00	No rotation
	01	Rotation

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

*No rotation***:** In the re-written buffer only the new data is available, i.e., 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 re-written buffer.

Element	*Value (hex)*	*XON Pattern*
XON Pattern	11	
	(xx)	Any Character (must be different than XOFF pattern)

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

Element	*Value (hex)*	*XOFF Pattern*
XOFF Pattern	13	
	(xx)	Any Character (must be different than XON pattern)

INIT-TABLE Object Error Messages

- If an element with an invalid value is written during a write service, the service is answered with a negative confirmation. 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 DTR control element is written with the value 2, an error message with the ADDITIONAL_CODE $0B30_{hex}$ is generated, because the 11th element is faulty.

- An error message is generated for a write request with the subindex 0 (write entire table) in which the XON/XOFF protocol is to be set and the XON and XOFF patterns are the same. The parameters are:

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

- If a reserved element is to be written, the value must be equal to 0, otherwise an error message is generated. The parameters are:

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

PCP Mode Error Messages

The RS-232 Communications module's error messages have the following parameters:

Error_Class 8	(device-specific error)
Error_Code 0	(no communication error).

The exact error cause is given by the Additional_Code. The ADDITIONAL_CODE low byte 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 may occur with the RS-232 Communications module:

0022_{hex}	No character from the RS-232 data will be accepted from this service
$xx30_{hex}$	Value is out of range or parameterization error
0000_{hex}	Hardware fault

For additional information on error messages in PCP mode, please refer to the *NIU User's Manual* for the NIU in the I/O Station.

Data Lengths and Protocols

The VersaPoint RS-232 Communications Module stores the data it exchanges with the connected serial device in a Transmit buffer area and a Receive buffer area. The amount of data these buffers will store depends on the type of protocol being used by the connected device.

Protocol	Transmitted Data	Received Data	When Receiving Data
Transparent	1023 bytes	4096 bytes	
XON/XOFF	1023 bytes	4096 bytes	Software handshake
End-to-end	1023 bytes (including end characters)	25 buffers, each with 58 bytes	Two end characters are filtered out
Dual buffer	1023 bytes (including end characters)	2 buffers, each with 58 bytes	Only stores the most recently-received data. End characters are filtered out
3964R	15 buffers each with 58 bytes	2 buffers, each with 58 bytes	Data exchange with software handshake, time monitoring, and checksum

Protocol Characteristics

The VersaPoint RS-232 Communications Module must be configured to match the type of protocol supported by the connected device. The default protocol is XON/XOFF.

Transparent Protocol

With Transparent Protocol, data is transmitted in the same format it was received. Transparent Protocol supports a CTS hardware handshake.

The module stores data to be transmitted and data received from the attached device in FIFO (First-In-First-Out) memory. The module can store up to 1023 bytes (1 kbyte) of data for transmission, and store up to 4096 bytes (4 kbytes) of data received from the connected device. If the module receives another character after the 4095th, the error pattern is stored in the receive FIFO. All other subsequent characters are ignored.

XON/XOFF Protocol

XON/XOFF Protocol works in the same way as Transparent Protocol, but uses software handshake instead of hardware handshake. Data transmission 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 module is configured using the INIT-TABLE object, as described in the *NIU User's Manual*.

If the module receives an XOFF character, it stops transmitting data. When the module receives an XON character, it starts transmitting data again.

The RS-232 Communications module sends an XOFF when the available space in the Received Data FIFO is less than 5 bytes. As soon as more memory becomes available, the module sends a single XON. Transmission is not dependent on the CTS input.

The module does not filter serial data before transmission. Any characters that occur with the code defined for XON and XOFF are sent, and may trigger undesirable events at the receiver. When the module receives serial data, the XON and XOFF characters are filtered and are not made available as data. Any characters with the XON or XOFF code are lost. It is your responsibility as the user to ensure that characters with these codes do not appear in the data stream.

End-to-End Protocol

When the VersaPoint I/O Station NIU, provides serial data to the RS-232 Module for transmission to the attached device, the NIU attaches two additional characters, the first and second delimiters. The first and second delimiters are defined when the module is configured using the INIT-TABLE object, as described in previously.

The VersaPoint I/O Station NIU can only read serial data from the RS-232 Communications Module if the module has received the first and second delimiters. The two end characters confirm that the data has been received without error and the maximum data length of 58 bytes has been observed. The delimiters are filtered out when the NIU reads the data.

For End-to-End Protocol, the area of internal memory used for received data is organized as a series of 25 buffers; it is not organized as a FIFO. Each buffer contains 58 bytes. 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), data can be overwritten in two different ways.

- If the Rotation Switch parameter ($0C_{hex}$) is set to 0, the defaut, only new data in the re-written buffer is available; data from the previous cycle is rejected.
- If the Rotation Switch parameter ($0C_{hex}$) is set to 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 re-written buffer (rotation).

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

Dual Buffer Protocol

With Dual Buffer Protocol, the RS-232 Module stores the last data block it receives. The data block is a string of characters with the first and second delimiter end characters, as in the End-to-End Protocol.

When the module receives a new data block, the previous one is overwritten. This is done by using two buffers, which are written alternately. One buffer is always available to receive new serial data, while the other buffer stores the last received data block. A data block is only regarded as complete once both delimiters have been detected, one after the other. When the data block is complete, it can be read by the VersaPoint I/O Station NIU.

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), data can be overwritten in two different ways.

- If the Rotation Switch parameter ($0C_{hex}$) is set to 0, the defaut, only new data in the re-written buffer is available; data from the previous cycle is rejected.
- If the Rotation Switch parameter ($0C_{hex}$) is set to 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 re-written buffer (rotation).

When the VersaPoint I/O Station NIU, provides serial data to the RS-232 Module for transmission to the attached device, the NIU attaches two additional characters, the first and second delimiters. The first and second delimiters are defined when the module is configured using the INIT-TABLE object.

3964R Protocol

This protocol uses beginning and end identifiers, a checksum and a time-monitoring function. There are 15 buffers available for transmitting and 25 buffers for receiving.

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

The optional 3964 priority defines which device may transmit first (high priority) if there is an initialization conflict (more than one device attempting to transmit data simultaneously).

Specifications for Communications Module IC220BEM232

Serial RS-232 Interface	
Type	RS-232 interface with DTR/CTS handshake Data terminal equipment (DTE) version Electrical data according to EIA (RS) 232 CCITT V,28, DIN 66259 Part 1
Input Impedance	5k Ohms, typical
Permissible input voltage range	-30V to +30V
Switching thresholds	0.8V to 2.4V
Hysteresis	0.5V typical
Output voltage, with 3 Ohm load)	High: 6.7V typical. Low: -6.7V typical
Output voltage, no load operation	High: $\leq$ 25V. Low: $\geq$ -25V
Permissible load capacity	2500 pF
Short circuit protected against GND	Yes
Short circuit current	$\pm$60mA maximum

Power Dissipation
Power dissipation of the electronics: 1.163W
Power dissipation of the housing: 1.2W maximum

Electrical Isolation / Isolation of the Voltage Range	
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 lines are electrically-isolated on the same voltage. FE is a separate potential area.	
Separate System Potentials Consisting of Module / Power Terminal and I/O Terminal	
5V supply incoming remote bus / 7.5V supply (bus logic)	500 VAC, 50 Hz, 1 min
5V supply outgoing remote bus / 7.5V supply (bus logic)	500 VAC, 50 Hz, 1 min
RS-232 interface / 7.5V supply (bus logic)	500 VAC, 50 Hz, 1 min
RS-232 interface / 7.5V supply (bus logic)	500 VAC, 50 Hz, 1 min
7.5V supply (bus logic) / 24V supply (I/O)	500 VAC, 50 Hz, 1 min
7.5V supply (bus logic) / functional earth ground	500 VAC, 50 Hz, 1 min
24V supply (I/O) / functional earth ground	500 VAC, 50 Hz, 1 min