2.2 2-Way Serial Link Module BMX NOM 0200

Subject of this Section

This section introduces the serial communications on the BMX NOM 0200 module.

What's in this Section?

This section contains the following topics:

Presentation of the BMX NOM 0200 2-Way Serial Link Module

General

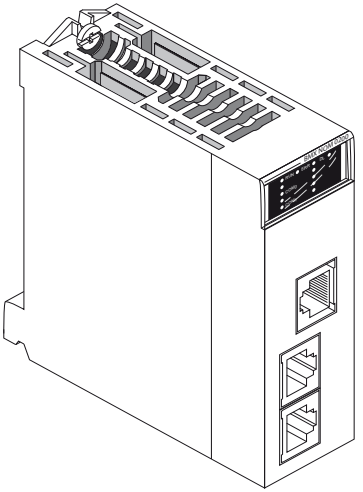
The BMX NOM 0200 and BMX NOM 0200H *(see page 30)* serial link modules are 2-way asynchronous serial line modules supporting Modbus Serial (master or slave) and Character Mode communications.

The BMX NOM 0200 is a simple-format, dedicated module, which can be installed on a Modicon M340 station rack.

NOTE: At the temperature extremes (-25... 0ºC and 60... 70ºC) (-13...32ºF) and (140...158ºF), the BMX NOM 0200H operating characteristics are the same as the BMX NOM 0200 characteristics within its (0…60ºC)(32...140ºF) temperature range.

Module introduction

The illustration below shows the physical characteristics of the BMX NOM 0200 module:



This BMX NOM 0200 module is composed of the elements in the following table:

Key	Description
1	Five indicator LEDs on the front of the module: • RUN and ERR show the module's status, • COM0 displays the traffic status on the port 0 or 1 (channel 0), • COM1 displays the traffic status on the port 2 (channel 1), • DL shows the firmware download status.

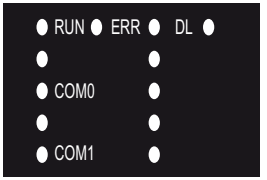
Key	Description
2	Integrated channel (channel 0) dedicated to the serial link with 2 serial ports: RS232 (port 0) and RS485 (port 1). **Note:** Only one port can be active at a time.
3	Integrated channel (channel 1) dedicated to the serial link with 1 serial port: RS485 (port 2).

NOTE: In some operating modes, **LEDs can indicate more specific information** *(see page 26)*.

Visual Diagnostics

Five LEDs are located on the front panel of the BMX NOM 0200 module. They display information about the module operating status and about the communication status of the built-in serial link.

LED Display:



- RUN = The module is powered and well configured.
- ERR = The module has detected an error and cannot function correctly.
- DL = The firmware is being downloaded.
- COM0 = Communication detected on port 0 or 1 (channel 0).
- COM1 = Communication detected on port 2 (channel 1).

LED meaning:

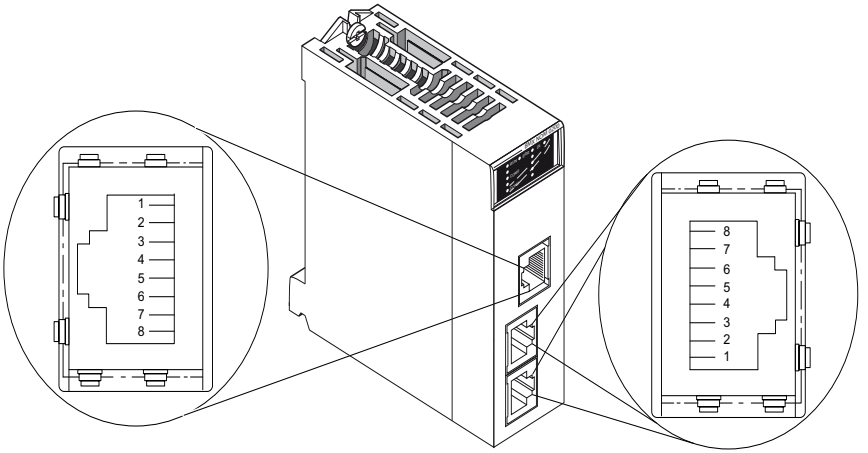
- Each LED can be in one of these states:
  - 1 = On
  - 0 = Off
  - B = Blinking
- During module startup all LEDs are powered ON and OFF, this verifies that the LEDs are functioning correctly.

RUN	ERR	COM0	COM1	DL	Diagnose
0	_	_	_	_	The module is not powered or non-operational.
0	B	_	_	_	The module is not configured.
1	1	_	_	_	The module improperly operative.
1	1	1	0	_	The module has detected a problem on the channel 0.

RUN	ERR	COM0	COM1	DL	Diagnose
1	1	1	B	_	The module has detected a problem on the channel 0, the channel 1 is exchanging data.
1	1	0	1	_	The module has detected a problem on the channel 1.
1	1	B	1	_	The module has detected a problem on the channel 1, the channel 0 is exchanging data.
1	0	B	_	_	The channel 0 is exchanging data.
1	0	_	B	_	The channel 1 is exchanging data.
B	B	_	_	0	The CPU is absent.
B	B	B	B	_	The module is performing self tests.
_	_	_	_	B	A module firmware is being downloaded.
_	_	_	_	1	The firmware is uploaded; the module must be reset.

Serial Ports Introduction

The illustration below shows the RJ45 serial ports on the BMX NOM 0200:



The table below shows the pin assignment for the serial port on the BMX NOM 0200:

Pin N°	RS485 channel 1 / port 1 or 2	RS232 channel 0 / port 0
1	_	RXD (Receive Data)
2	_	TXD (Transmit Data)
3	_	RTS (Request To Send)
4	D1 (B/B4)	DTR (Data Terminal Ready)

Pin N°	RS485 channel 1 / port 1 or 2	RS232 channel 0 / port 0
5	D0 (A/A4)	DSR (Data Set Ready)
6	_	CTS (Clear To Send)
7	_	DCD (Data Carrier Detect)
8	Potential serial link grounding (0 V)	Potential serial link grounding (0 V)

NOTE:

- The two RS485 lines are isolated. The isolation voltage between the two serial lines 500 V and between each isolated serial line and the backplane is up to 500V AC.
- The seven-wire RS232 and two-wire RS485 use the same female RJ45 connector. Only the signal cabling is different.

Channels Specifications

The channels of the BMX NOM 0200 module include:

- Two isolated RS485 Physical Interfaces,
- One non-isolated RS232 Physical Interface,
- Modbus Serial (ASCII and RTU) and Character Mode communication types.

The link specifications for the two protocols are:

	Modbus Serial / RS485	Modbus Serial / RS232	Character Mode / RS485	Character Mode / RS232
Type	Master/Slave	Master/Slave	Half Duplex	Full Duplex
Flow	19200 bauds. Parameters can be set from 300 bauds to 57600 bauds.	19200 bauds. Parameters can be set from 300 bauds to 115200 bauds.	9600 bauds. Parameters can be set from 300 bauds to 57600 bauds.	9600 bauds. Parameters can be set from 300 bauds to 115200 bauds
Number of devices	32	32	_	_
Authorized slave addresses	1 to 247	1 to 247	_	_
Max. length of Bus without branching	1000 m (15 m with Branching)	15 m	1000 m (15 m with Branching	15 m

	Modbus Serial / RS485	**Modbus Serial / RS232**	**Character Mode / RS485**	**Character Mode / RS232**
Message Size	Modbus Serial: • RTU: 256 bytes (252 bytes of data) • ASCII: 513 bytes (2x252 bytes of data)	Modbus Serial: • RTU: 256 bytes (252 bytes of data) • ASCII: 513 bytes (2x252 bytes of data)	1024 bytes	1024 bytes
Utilities	Read words/bits. Write words/bits. Diagnostics.	Read words/bits. Write words/bits. Diagnostics.	Send character strings. Receive character strings.	Send character strings. Receive character strings.
Hardware Flow Control	_	Optionally via RTS/CTS signals.	_	Optionally via RTS/CTS signals.

Consumption of the BMX NOM 0200 Module

This table shows the consumption of BMX NOM 0200 module:

Voltage	**Typical Current**	**Maximum Current**	**Typical Power Dissipation**	**Maximum Power Dissipation**
24 V DC	80 mA	130 mA	1.92 W	3.12 W

Modicon M340H (Hardened) Equipment

M340H

The Modicon M340H (Hardened) equipment is a ruggedized version of M340 equipment. It can be used at extended temperatures (-25...70ºC) (-13...158ºF) and in harsh chemical environments.

The M340H equipment, when within the standard temperature range (0…60ºC)(32...140ºF), has the same characteristics as the standard M340 equipment.

At the temperature extremes (-25... 0ºC and 60... 70ºC) (-13...32ºF) and (140...158ºF) the Hardened versions can have reduced power ratings that impact power calculations for Unity Pro applications.

If this equipment is operated outside the -25...70ºC (-13...158ºF) temperature range, the equipment can operate abnormally.

⚠ CAUTION

UNINTENDED EQUIPMENT OPERATION

Do not operate M340H equipment outside of its temperature range.

Failure to follow these instructions can result in injury or equipment damage.

Hardened equipment has a conformal coating applied to its electronic boards. This protection, when associated with appropriate installation and maintenance, allows it to be more robust when operating in harsh chemical environments.

Grounding of Installed Modules

General

The grounding of Modicon M340 modules is crucial to avoid electric shocks.

Grounding Processors and Power Supplies

⚠ DANGER

HAZARD OF ELECTRIC SHOCK, EXPLOSION OR ARC FLASH

Ensure ground connection contacts are present and not bent out of shape. If they are, do not use the module and contact your Schneider Electric representative.

Failure to follow these instructions will result in death or serious injury.

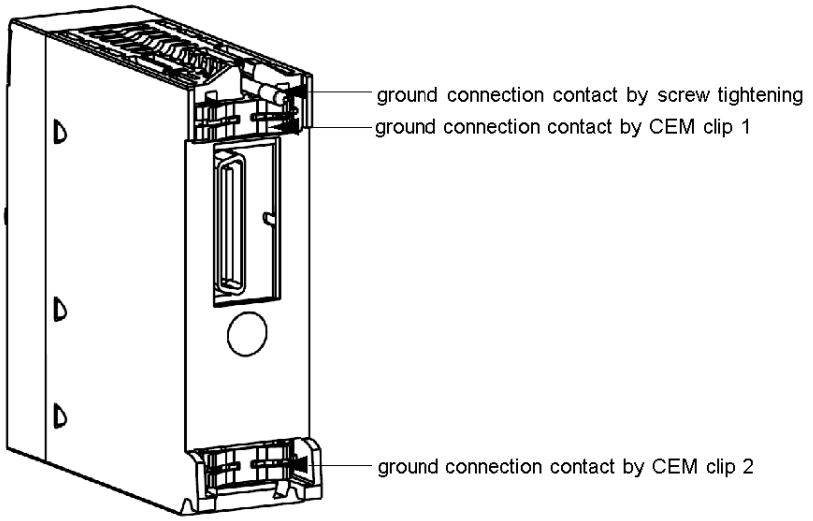
⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

Tighten the clamping screws of the modules to guarantee the system characteristics. A break in the circuit could lead to an unexcepted behavior of the system.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

All Modicon M340 modules are equipped with ground connection contacts at the rear for grounding purposes:



These contacts connect the grounding bus of the modules to the grounding bus of the rack.

Installation of the Module BMX NOM 0200

General

The BMX NOM 0200 module is installed in a Modicon M340 station rack and cannot use the slots required for the power supply and the processor. This installation must conform to the rack installation instructions.

The BMX NOM 0200 module requires the installation of a CPU with minimum OS version 02.10. This installation must conform to the CPU installation instructions.

An RJ45 connector can then be connected to the module according to the targeted network.

NOTE: The BMX NOM 0200 module can be installed in a rack while the application is running on the PLC.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

The application of this product requires expertise in the design and programming of control systems. Only persons with such expertise should be allowed to program, install, alter, and apply this products.

Follow all local and national safety codes and standards.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Number of Modules

Since the number of expert channels managed by a PLC station is related to the processor installed, the maximum number of BMX NOM 0200 modules in a station will therefore rely on:

- The number of channels configured on each BMX NOM 200 module (each channel counts as an expert channel),
- The type and version of processor installed,
- The number of expert channels already used.

When the application is built, Unity Pro checks that the limitation is not exceeded.

Connection/ Disconnection

The BMX NOM 0200 module can be connected or disconnected while the power is on. When the module is disconnected from the rack, its internal memory is erased. The module goes through an initialization phase once it is reconnected.

By default, the BMX NOM 0200 configuration is modbus slave at address 248, 19200 bits/s, RTU, 8bits, 1 stop, RS232 on channel 0 and RS485 on channel 1.

The address 248 is the point-to-point address to which any BMX NOM 0200 slave module answers. This functionality aims at finding any slave module whose address is unknown

Firmware Update

The firmware of the BMX NOM 0200 can be updated via the PLC backplane. Firmware update is defined in the Unity Pro Loader Manual.

BMX NOM 0200 Wiring Considerations

Operational Consideration

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

Although you can connect or disconnect the wires on the BMX NOM 0200 module and BMX P34 20x0 CPUs while the power to the BMX XBP station is on,doing so can interrupt the application in progress.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

The Link

The following situations can create a temporary disruption in the application or communications:

- The RJ45 connector is connected or disconnected when the power is on.
- Modules are re-initialized when the power is switched back on.

Modbus Serial Communication for BMX NOM 0200

7

Subject of this Chapter

This chapter presents the software implementation process for Modbus serial communication for BMX NOM 0200.

What's in this Chapter?

This chapter contains the following sections:

7.1 Generalities

Subject of this Section

This section presents the general points relating to Modbus serial communication and its services.

What's in this Section?

This section contains the following topics:

About Modbus Serial

Introduction

Communicating via Modbus enables data exchange between all devices connected to the bus. The Modbus Serial is a protocol that creates a hierarchical structure (one master and several slaves).

The master manages all exchanges in two ways:
- The master exchanges with the slave and awaits a response.
- The master exchanges with all the slaves without waiting for a response (general broadcast).

⚠ WARNING

CRITICAL DATA LOSS

Only use communications port for non-critical data transfers.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Performance

At a Glance

The tables that follow can be used to evaluate typical Modbus communication exchange times according to different criteria.

The results displayed correspond to the average operation period for the READ_VAR function in milliseconds.

Exchange Time Definition

The Exchange Time is the time between the creation of an exchange and the end of that exchange. It includes the serial link communication time.

The exchange is created when the communication function call is made.

The exchange ends when one of the following events occurs:
- Data is received.
- An anomaly occurs.
- Time-out expires.

Exchange Time for One Word

The table below shows exchange times for one word of Modbus communication on a BMX NOM 0200 module:

Baud rate of communication in bits per second	Cycle time in ms	Exchange time in ms Modbus Slave is a BMX P34 1000 cyclic
4800	Cyclic	65
4800	10	68
4800	50	100
9600	Cyclic	38
9600	10	47
9600	50	50
19200	Cyclic	29
19200	10	38
19200	50	50
38400	Cyclic	24
38400	10	30
38400	50	50
57600	Cyclic	17
57600	10	20

Baud rate of communication in bits per second	Cycle time in ms	Exchange time in ms Modbus Slave is a BMX P34 1000 cyclic
57600	50	50
115200	Cyclic	17
115200	10	20
115200	50	50

Exchange Time for 100 Words

The table below shows exchange times for 100 words of Modbus communication on a BMX NOM 0200 processor:

Baud rate of communication in bits per second	Cycle time in ms	Exchange time in ms Modbus Slave is a BMX P34 1000 cyclic
4800	Cyclic	560
4800	10	560
4800	50	600
9600	Cyclic	286
9600	10	295
9600	50	300
19200	Cyclic	152
19200	10	160
19200	50	200
38400	Cyclic	86
38400	10	90
38400	50	100
57600	Cyclic	56
57600	10	60
57600	50	100
115200	Cyclic	36
115200	10	40
115200	50	50

Measurement Accuracy

All exchange times listed above come from measures with an accuracy margin of +/- 10 ms.

How to Access the Serial Link Parameters

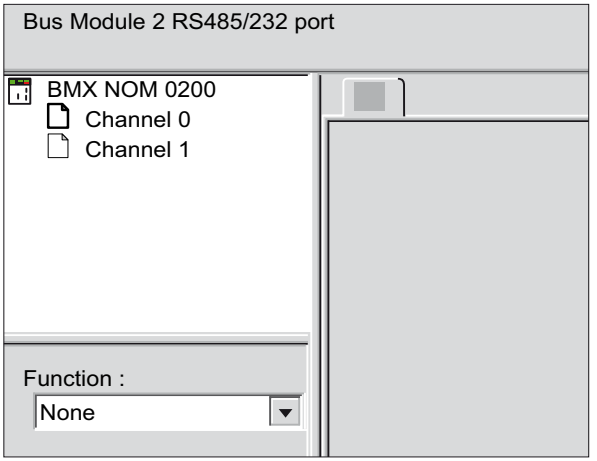
At a Glance

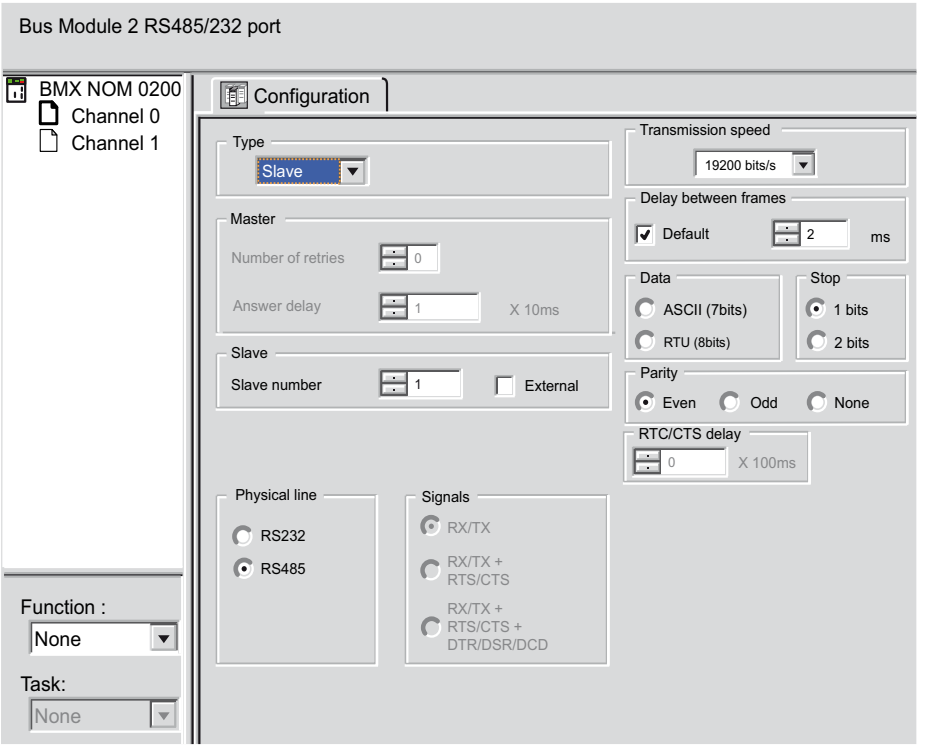
The following pages explain how to access the serial ports configuration screen for the BMX NOM 0200 module as well as the general elements of the Modbus and Character Mode link configuration and debug screens.

How to Access the Serial Link

The table below describes the procedure for accessing the serial link of a BMX NOM 0200 module:

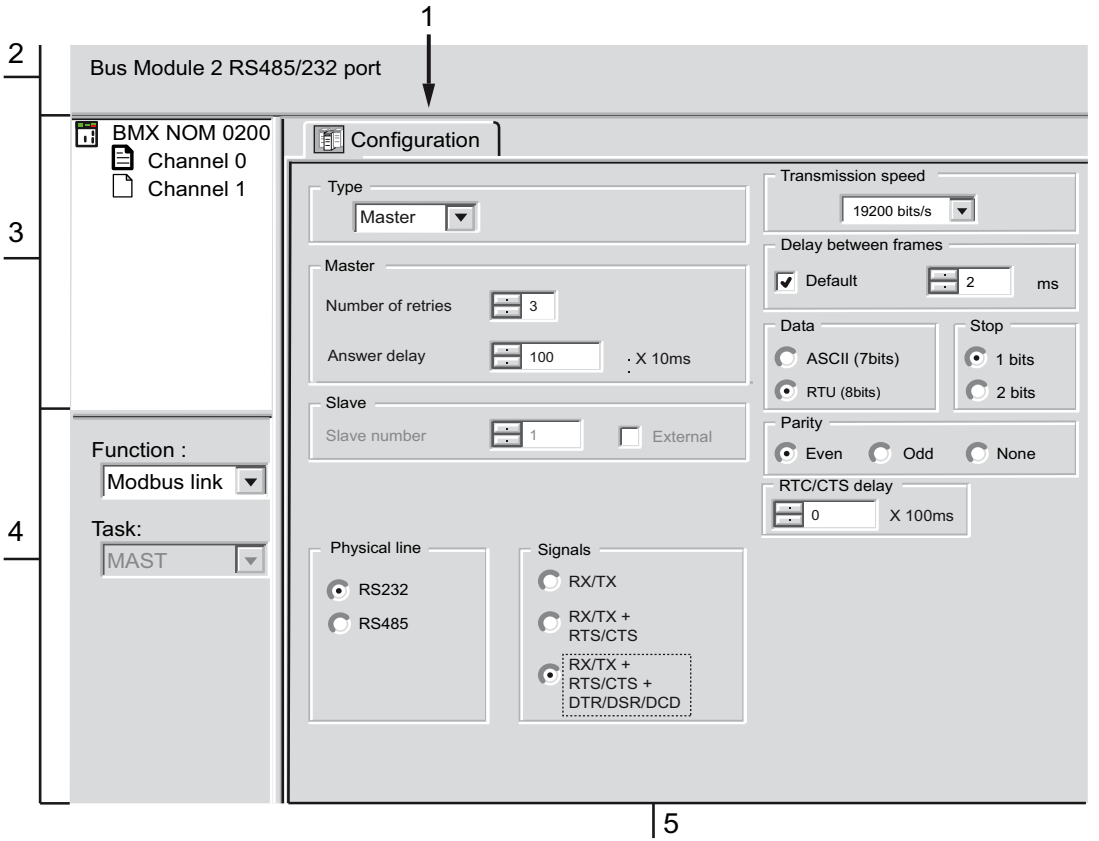
Step	Action
1	Open the hardware configuration editor.
2	Double-click on the BMX NOM 0200 module.

Step	Action
3	Select the channel to configure (Channel 0 or Channel 1). **Result with Channel 0 selected:** 

Step	Action
4	Select the **Modbus link** function. **Result with Channel 0 selected:** 

Description of the Configuration and Debug Screens

The figure below shows a configuration screen for Modbus Serial communication on Channel 0:


1
2
Bus Module 2 RS485/232 port
BMX NOM 0200
Channel 0
Channel 1
3
Configuration
Type
Master
Transmission speed
19200 bits/s
Delay between frames
Default
2
ms
Master
Number of retries
3
Answer delay
100
X 10ms
Data
ASCII (7bits)
RTU (8bits)
Stop
1 bits
2 bits
Slave
Slave number
1
External
Parity
Even
Odd
None
RTC/CTS delay
0
X 100ms
Function :
Modbus link
4
Task:
MAST
Physical line
RS232
RS485
Signals
RX/TX
RX/TX +
RTS/CTS
RX/TX +
RTS/CTS +
DTR/DSR/DCD
5

Description

The following table shows the different elements of the configuration and debug screens:

Key	Element	Function
1	Tabs	The tab in the foreground indicates the mode currently in use (**Configuration** in this example). Each mode can be selected using the corresponding tab. The available modes are: • **Configuration** • **Debug** (accessible in online mode only) • **Diagnostic** (accessible in online mode only)
2	Module Zone	Displays module reference and module LEDs status in online mode.

3	Channel zone	Enables you to:
		• Display the following tabs by clicking on **BMX NOM 0200**:
		  • "Overview", which gives the characteristics of the device.
		  • **"I/O Objects"**, which is used to presymbolize the input/output objects.
		  • "Fault", which shows the detected device faults (in online mode).
		• Display the following tabs by clicking on **Channel 0** or **Channel 1**:
		  • "Configuration"
		  • "Debugging"
		  • "Fault"
		• Display the channel name and symbol defined by the user (using the variables editor).

4	Genera parameters zone	This enables you to choose the general parameters associated with the channel: • **Function**: The available functions are "None", "Modbus link" and “Character mode link". By default, the "None" function is configured. • **Task**: Defines the master task in which the implicit exchange objects of the channel will be exchanged. This zone is grayed out and cannot be configured.
5	Configuration, debugging or fault zone	In configuration mode, this zone is used to configure the channel parameters. In debug mode, it is used to debug the communication channel. In diagnostic mode, it is used to display current detected errors either at module or at channel level.

7.2 Modbus Serial Communication Configuration

Subject of this Section

This section describes the software configuration process for Modbus serial communication.

What's in this Section?

This section contains the following topics:

Modbus Serial Communication Configuration Screen

General

The following pages provide an introduction to the configuration screen for Modbus serial communication.

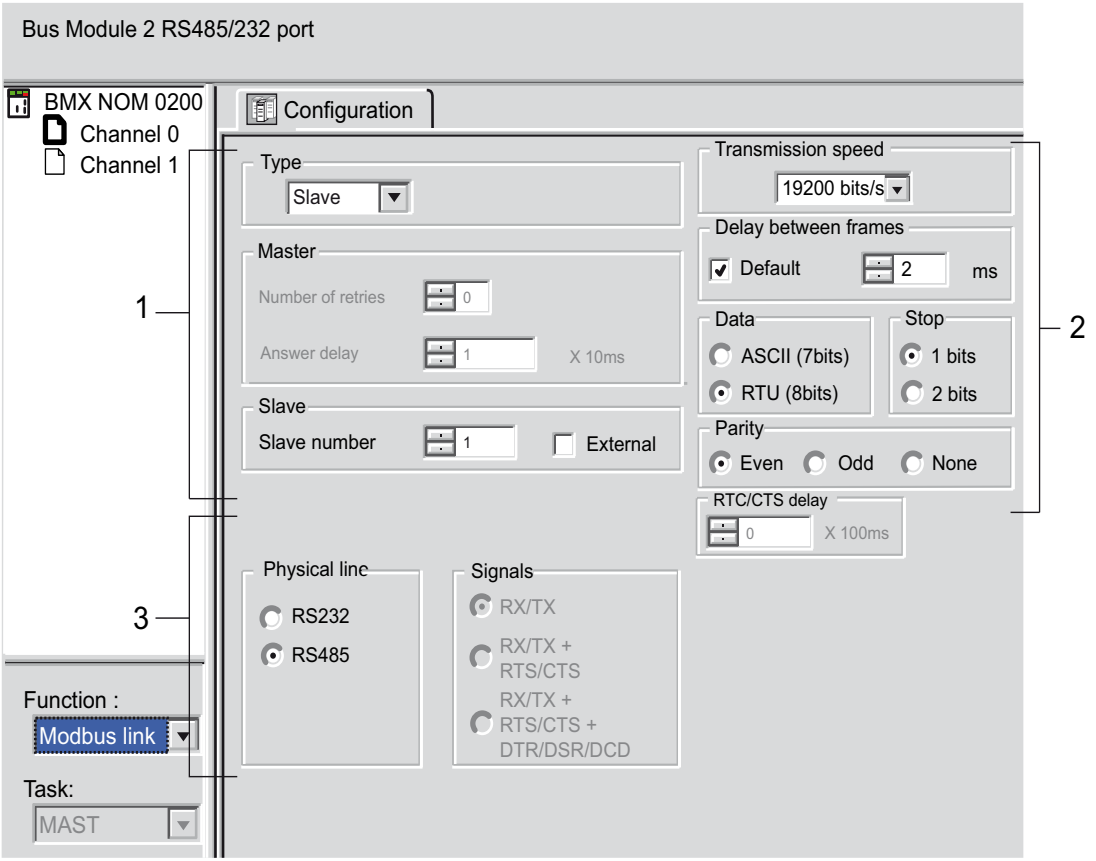
Access to the Configuration Screen

The following table describes the procedure for accessing the configuration screen for Modbus serial communication:

Step	Action
1	Open the BMX NOM 0200 sub-directory in the project browser *(see page 134)*.
2	Select the Channel to configure and "Modbus link" function on the screen that appears.

Illustration

The figure below shows the default configuration screen for Modbus serial communication on Channel 0:


Bus Module 2 RS485/232 port
BMX NOM 0200
Channel 0
Channel 1
Configuration
Type
Slave
Master
Number of retries
0
Answer delay
1
X 10ms
Slave
Slave number
1
External
Transmission speed
19200 bits/s
Delay between frames
Default
2
ms
Data
ASCII (7bits)
RTU (8bits)
Stop
1 bits
2 bits
Parity
Even
Odd
None
RTC/CTS delay
0
X 100ms
Physical linc
RS232
RS485
Signals
RX/TX
RX/TX + RTS/CTS
RX/TX + RTS/CTS + DTR/DSR/DCD
Function :
Modbus link
Task:
MAST
1
2
3

Description

This zone is used to configure channel parameters. In online mode, this zone is not accessible and will be grayed out. In offline mode, the zone is accessible but some parameters may not be accessible and will be grayed out.

The following table shows the different zones of the Modbus link configuration screen:

Key	Element	Comment
1	Application parameters *(see page 145)*	These parameters are accessible via three zones: ● Type, ● Master, ● Slave.
2	Transmission parameters *(see page 147)*	These parameters are accessible via five zones: ● Transmission speed, ● Delay between frames, ● Data, ● Stop bits, ● Parity.
3	Signal and physical line parameters *(see page 149)*	These parameters are accessible via three zones: ● Physical line, ● Signals, ● RTS/CTS delay.

NOTE: When configuring Modbus serial communication in Master mode, the "Slave" zone is grayed out and cannot be modified and vice-versa.

NOTE: In this example, the "Signals" and "RTS/CTS Delay" zones are grayed out because an RS485 physical line has been chosen.

Accessible Modbus Functions

At a Glance

Function accessibility for configuration of the serial link of a BMX NOM 0200 module using Modbus serial depends on the physical link being used.

Accessible Functions

The table below shows the different functions configurable according to the type of serial link used:

Function	RS485 Link (on Channel 0 or Channel 1)	RS232 Link (on Channel 0)
Master number of retries	X	X
Master answer delay	X	X
Slave number	X	X
Transmission speed	X	X
Delay between frames	X	X
Data	• ASCII (7 bits) • RTU (8 bits)	• ASCII (7 bits) • RTU (8 bits)
Stop	• 1 bit • 2 bits	• 1 bit • 2 bits
Parity	• Odd • Even • None	• Odd • Even • None
RX/TX signals	X	X
RTS/CTS signals	-	X
RTS/CTS delay	-	X
DTR/DSR/DCD Signals	-	X
Polarization	-	-

X Accessible Function
- Unaccessible Function

Default Values for Modbus Serial Communication Parameters

At a Glance

All Modbus serial communication parameters have default values.

Default Values

The table below shows the default values for Modbus serial communication parameters on Channel 0 and Channel 1 of the BMX NOM 0200 module:

Configuration parameter	Value
Mode	Slave
Physical Line	RS485
Slave number	1
Delay between frames	2 ms
Transmission speed	19200 bits/s
Parity	Even
Data Bits	RTU (8 bits)
Stop bits	1 bit

Application-linked Modbus Parameters

At a Glance

After configuring the communication channel, you need to enter the application parameters.

These parameters are accessible from three configuration zones:

- The Type zone,
- The Master zone,
- The Slave zone.

The Type Zone

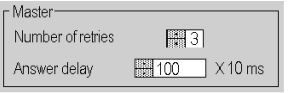
This configuration zone appears on the screen as shown below:

This zone enables you to select the role to be configured for the module in the Modbus serial communication:

- **Master**: When the module is the master.
- **Slave**: When the module is a slave.

The Master Zone

The configuration zone shown below is only accessible when "Master" is selected in the "Type" zone:

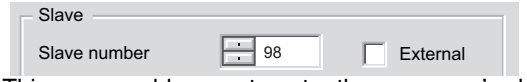


This zone enables you to enter the following parameters:

- **Number of retries**: number of connection attempts made by the master before defining the slave as absent.
  The default value is 3.
  Possible values range from 0 to 15.
  A value of 0 indicates no retries by the Master.
- **Answer delay**: the time between the Master's initial request and a repeated attempt if the slave does not respond. This is the maximum time between the transmission of the last character of the Master's request and the receipt of the first character of the request sent back by the slave.
  The default value is 1 second (100*10 ms).
  Possible values range from 10 ms to 10 s.

The Slave Zone

The configuration zone shown below is only accessible when "Slave" is selected in the "Type" zone:



This zone enables you to enter the processor's slave number:

The default value is 1.

Possible values range from 1 to 247.

Selection of **External** grays the **Slave number** field and makes the module use the value of the slave address saved into its internal FLASH memory.

NOTE: If the address stored into the FLASH is not into the MODBUS range address, then the default slave address 248 will be used.

When the firmware of the module is updated, the default slave address stored into the FLASH is set to 248. A new command has to be used to re-initialize the FLASH address.

Transmission-linked Modbus Parameters

At a Glance

After configuring the communication channel, you need to enter the transmission parameters.

These parameters are accessible from five zones:

- The Transmission Speed zone,
- The Delay Between Characters zone,
- The Data zone,
- The Stop zone,
- The Parity zone.

The Transmission Speed Zone

This configuration zone appears on the screen as shown below:

You can use it to select the transmission speed of the Modbus serial link. The selected speed has to be consistent with the other devices. The configurable values are 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600 and 115200 (only on channel 0 in RS232 mode) bits per second.

The Delay Between Frames Zone

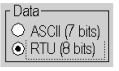
This configuration zone shown below is only accessible in RTU mode (it is grayed in ACSCII mode):

The Delay Between Frames is the minimum time separating two frames on reception. This delay is managed when the BMX NOM 0200 (master or slave) is receiving messages.

NOTE: The default value depends on the selected transmission speed.

The Data Zone

This configuration zone appears on the screen as shown below:



This zone allows you to enter the type of coding used to communicate using Modbus serial link. This field is set according to the other devices connected on the bus. There are two configurable modes:

- RTU mode:
  - The characters are coded over 8 bits.
  - The end of the frame is detected when there is a silence of at least 3.5 characters.
  - The integrity of the frame is checked using a word known as the CRC checksum, which is contained within the frame.
- ASCII mode:
  - The characters are coded over 7 bits.
  - The beginning of the frame is detected when the "**:**" character is received.
  - The end of the frame is detected by a carriage return and a line feed.
  - The integrity of the frame is checked using a byte called the LRC checksum, which is contained within the frame.

The Stop Zone

This configuration zone appears on the screen as shown below:

The Stop zone allows you to enter the number of stop bits used for communication. This field is set according to the other devices. The configurable values are:

- 1 bit
- 2 bits

The Parity Zone

This configuration zone appears on the screen as shown below:

This zones enables you to determine whether a parity bit is added or not, as well as its type. This field is set according to the other devices. The configurable values are:

- Even
- Odd
- None

Signal and Physical Line Parameters in Modbus

At a Glance

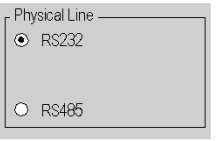
After configuring the communication channel, you need to enter the signal and physical line parameters.

These parameters are accessible via three zones:

- The Physical Line zone,
- The Signals zone,
- The RTS/CTS Delay zone.

The Physical Line Zone

This configuration zone shown below is accessible only on Channel 0 (it is grayed out and configured to RS485 on Channel 1):

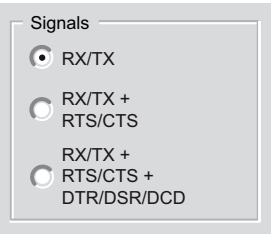


In this zone, you can choose between two types of physical line for the serial port on the BMX NOM 0200 module:

- The RS232 line,
- The RS485 line.

The Signals Zone

This configuration zone appears on the screen as shown below:



In this zone, you can select the signals supported by the RS232 physical line:

- RX/TX
- RX/TX + RTS/CTS (hardware flow management signals)
- RX/TX + RTS/CTS + DTR/DSR/DCD (Modem signals)

If the RS485 is configured, the entire zone will be grayed out and the default value will be RX/TX.

The RTS/CTS Delay Zone

This configuration zone appears on the screen as shown below:

RTS/CTS delay zone is available only when both RS232 and RX/TX+RTS/CTS or RX/TX+RTS/CTS+DTR/DSR/DCD check boxes are selected. An RTS/CTS hardware flow control is performed.

The RTS/CTS hardware flow control algorithm aims at preventing the overflow reception buffer (full duplex).

The RTS/CTS delay corresponds to the time out delay between the RTS rise up and the CTS rise up. A RTS/CTS delay value different from 0 also corresponds to the maximum waiting time between each character transmission after the rise of RTS and CTS signals. If the value is set to 0, UARTs can get stuck in a waiting state for an infinite time until the CTS rise up so the value 0 is used only in particular cases such as looping the RTS signal to the CTS signal in order to check that all connection are operating correctly.

NOTE: The default value is 0 ms.

7.3 Modbus Serial Communication Programming

Subject of this Section

This section describes the programming process involved in implementing Modbus serial communication.

What's in this Section?

This section contains the following topics:

Services Supported by a Modbus Link Master Module

At a Glance

When used as the master in a Modbus link, a BMX NOM 0200 module supports several services via the READ_VAR, WRITE_VAR and DATA_EXCH communication functions.

Data Exchanges

Reading or writing of variables are carried out by adressing following requests to the targeted slave device.

These requests use the READ_VAR and WRITE_VAR communication functions:

Modbus request	Function code	Communication function
Read bits	16#01 or 16#02	READ_VAR
Read words	16#03 or 16#04	READ_VAR
Write bits	16#0F	WRITE_VAR
Write words	16#10	WRITE_VAR

More generally, it is possible to send any Modbus requests to a slave device by using the DATA_EXCH communication function.

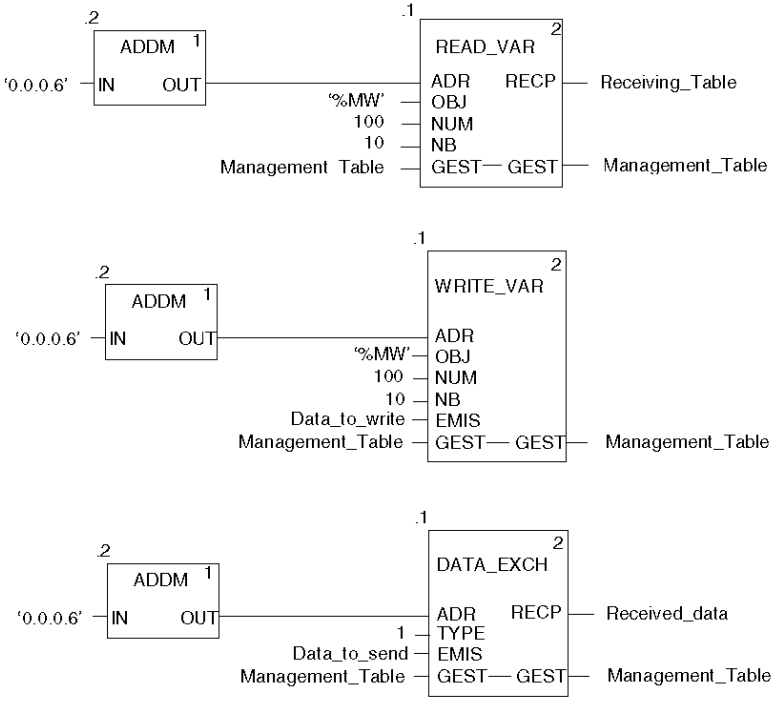
READ_VAR, WRITE_VAR and DATA_EXCH Communication Functions

Three specific communication functions are defined for sending and receiving data via a Modbus communication channel:

- READ_VAR: To read variables
- WRITE_VAR: To write variables
- DATA_EXCH: To send Modbus requests to another device over the selected protocol

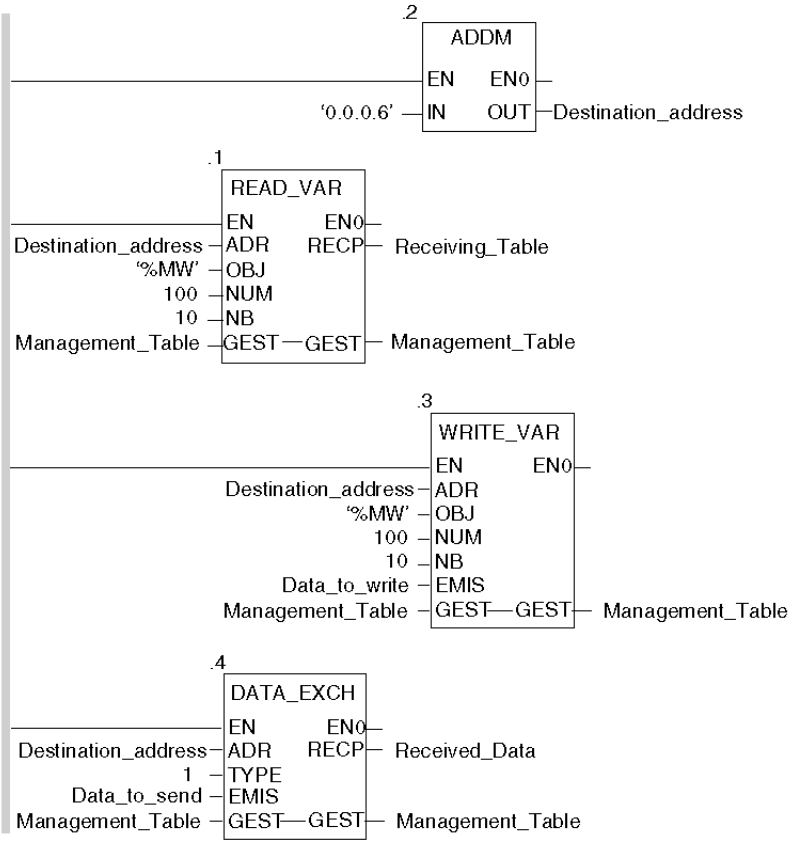
Programming Example in FBD

The diagram below represents an example of programming of the READ_VAR, WRITE_VAR and DATA_EXCH communication functions in the FBD language:



Programming Example in Ladder

The diagram below represents an example of programming of the READ_VAR, WRITE_VAR and DATA_EXCH communication functions in the Ladder language:



Programming Example in ST

The lines of code below represent an example of programming of the READ_VAR, WRITE_VAR and DATA_EXCH communication functions in the ST language:

```
READ_VAR(ADDM('0.0.0.6'), 'MW', 100, 10, Management_Table,
Receiving_Table);

WRITE_VAR(ADDM('0.0.0.6'), '%MW', 100, 10, Data_to_write,
Management_Table);

DATA_EXCH(ADDM('0.0.0.6'), 1, Data_to_send, Management_Table,
Received_data);
```

Cancelling an Exchange

An exchange executed by the READ_VAR, WRITE_VAR and DATA_EXCH functions can be cancelled with either ways of programming, which are both presented in ST language below:

- Using the CANCEL function:

```
IF (%MW40.0) THEN
   %MW200:=SHR(%MW40,8;)
   CANCEL(%MW200,%MW185);
END_IF;
```

  %MW40 is the GEST parameter (management table). %MW40.0 corresponds to the activity bit of the READ_VAR function and is set to 1 when the communication function is active. If this bit is set to 1, the program carries out the following instructions:
  - Moves the %MW40 bits one byte (8 bits) to the right and loads the byte corresponding to the communication's exchange number into the %MW200 word,
  - Cancels the exchange whose exchange number is contained within the %MW200 word using the CANCEL function.

- Using the communication function cancel bit:

```
IF (%MW40.0) THEN
   SET(%MW40.1);
   READ_VAR(ADDM('0.0.0.6'), '%MW', 100, 10, %MW40:4,
%MW10:10);
END_IF;
```

  %MW40 is the GEST parameter (management table). %MW40.0 corresponds to the activity bit of the READ_VAR function and is set to 1 when the communication function is active. If this bit is set to 1, the program sets the %MW40.1 bit, the function cancel bit, to 1. This stops communication of the READ_VAR function.

NOTE: When using the communication function cancel bit contained in the function exchange management word (%MW40 in this example), the function (READ_VAR in this example) must be called in order to activate the cancellation of the exchange.

NOTE: When using the communication function cancel bit, it is possible to cancel a communication from an animation table. This can be done by simply setting the function cancel bit to 1 (%MW40.1 in this example) and then start again the communication function.

NOTE: This example of programming concerns the READ_VAR function, but is equally applicable to the WRITE_VAR as well as the DATA_EXCH functions.

NOTE: The CANCEL function uses a report word for the CANCEL function (%MW185 in this example).

Description of ADDM Function Parameters

The following table outlines the various parameters for the ADDM function:

Parameter	Type	Description
IN	STRING	Address of device on bus or serial link. The syntax of the address is of the 'r.m.c.node' type. The address is made up of the following parameters: • **r**: Rack number of the module • **m**: Slot number of the module within the rack • **c**: Channel number of the module • **node**: Number of slave to which the request is being sent
OUT	ARRAY [0..7] OF INT	Array representing the address of a device. This parameter can be used as an input parameter for several communication functions.

Description of READ_VAR Function Parameters

The following table outlines the various parameters for the READ_VAR function:

Parameter	Type	Description
ADR	ARRAY [0..7] OF INT	Address of the destination entity given by the OUT parameter of the ADDM function.
OBJ	STRING	Type of object to be read. The available types are as follows: • %M: internal bit • %MW: internal word • %I: external input bit • %IW: external input word
NUM	DINT	Address of first object to be read.
NB	INT	Number of consecutive objects to be read.

GEST	ARRAY [0..3] OF INT	Exchange management table consisting of the following words:
		• Rank 1 word: A word managed by the system and consisting of two bytes:
		  • Most significant byte: Exchange number,
		  • Least significant byte: Activity bit (rank 0) and cancel bit (rank 1).
		• Rank 2 word: a word managed by the system and consisting of two bytes:
		  • Most significant byte: Operation report,
		  • Least significant byte: Communication report.
		• Rank 3 word: A word managed by the user which defines the maximum response time using a time base of 100 ms.
		• Rank 4 word: A word managed by the system which defines the length of the exchange.

RECP	ARRAY [n..m] OF INT	Word table containing the value of the objects read.

Description of WRITE_VAR Function Parameters

The following table outlines the various parameters of the WRITE_VAR function:

Parameter	Type	Description
ADR	ARRAY [0..7] OF INT	Address of the destination entity given by the OUT parameter of the ADDM function.
OBJ	STRING	Type of object to be written. The available types are as follows: • %M: internal bit • %MW: internal word **Note:** WRITE_VAR cannot be used for %I and %IW variables.
NUM	DINT	Address of first object to be written.
NB	INT	Number of consecutive objects to be written.
EMIS	ARRAY [n..m] OF INT	Word table containing the value of the objects to be written.

GEST	ARRAY [0..3] OF INT	Exchange management table consisting of the following words:
		• Rank 1 word: A word managed by the system and consisting of two bytes:
		  • Most significant byte: Exchange number,
		  • Least significant byte: Activity bit (rank 0) and cancel bit (rank 1).
		• Rank 2 word: A word managed by the system and consisting of two bytes:
		  • Most significant byte: Operation report,
		  • Least significant byte: Communication report.
		• Rank 3 word: A word managed by the user which defines the maximum response time using a time base of 100 ms.
		• Rank 4 word: A word managed by the system which defines the length of the exchange.

Description of DATA_EXCH Function Parameters

The following table outlines the various parameters of the DATA_EXCH function:

Parameter	Type	Description
ADR	ARRAY [0..7] OF INT	Address of the destination entity given by the OUT parameter of the ADDM function.
TYPE	INT	For Modicon M340 PLCs, the only possible value is **1**: Transmission of an **EMIS** array, then the PLC waits for the reception of a **RECP** array.
EMIS	ARRAY [n..m] OF INT	Integers table to be sent to the destination device of the request. **Note:** It is imperative that the length of the data to be sent (in bytes) be assigned to the fourth word of the management table before launching the function, in order for this to be correctly executed.

GEST	ARRAY [0..3] OF INT	Exchange management table consisting of the following words:
		• Rank 1 word: A word managed by the system and consisting of two bytes:
		  • Most significant byte: Exchange number.
		  • Least significant byte: Activity bit (rank 0) and cancel bit (rank 1).
		• Rank 2 word: A word managed by the system and consisting of two bytes,:
		  • Most significant byte: Operation report,
		  • Least significant byte: Communication report.
		• Rank 3 word: A word managed by the user which defines the maximum response time using a time base of 100 ms.
		• Rank 4 word: A word managed by the system which defines the length of the exchange.

RECP	ARRAY [n..m] OF INT	Integers table containing the data received. **Note:** The size of the data received (in bytes) is written automatically by the system in the fourth word of the management table.

Services Supported by a Modbus Link Slave Module

At a Glance

When used as a slave in a Modbus link, a BMX NOM 0200 module supports several services.

Data Exchanges

A slave module manages the following requests:

Modbus request	Function code	PLC object
Read n output bits	16#01	%M
Read n output words	16#03	%MW
Write n output bits	16#0F	%M
Write n output words	16#10	%MW

Diagnostics and Maintenance

The diagnostics and maintenance requests managed by a Modbus slave BMX NOM 0200 module are listed below:

Designation	Function code/sub-function code
Read exception status	16#07
Restart Communications Option	16#08 / 16#01
Return Diagnostic Register	16#08 / 16#02
Change ASCII Input Delimiter	16#08 / 16#03
Force Listen Only Mode	16#08 / 16#04
Clear Counters and Diagnostic Register	16#08 / 16#0A
Return Bus Message Count	16#08 / 16#0B
Return Bus Communication Error Count	16#08 / 16#0C
Return Bus Exception Error Count	16#08 / 16#0D
Return Slave Message Count	16#08 / 16#0E
Return Slave No Response Count	16#08 / 16#0F
Return Slave Negative Acknowledgements Count	16#08 / 16#10
Return Slave Busy Count	16#08 / 16#11
Return Bus Character Overrun Count	16#08 / 16#12
Get Communication event Counter	16#0B
Get Communication event Log	16#0C

Designation	Function code/sub-function code
Report Slave identification	16#11
Write Slave identification	16#11 / 16#01

7.4 Debugging Modbus Serial Communication

Modbus Serial Communication Debug Screen

General

The Modbus serial communication debug screen can only be accessed in online mode.

Accessing the Debug Screen

The following table describes the procedure for accessing the debug screen for Modbus serial communication:

Step	Action
1	Access the configuration screen for Modbus serial communication. *(see page 140)*
2	Select the "Debug" tab on the screen that appears.

Description of the Debug Screen

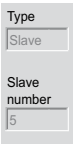
The debug screen is divided into two or three zones:

- The Type and Slave number zone,
- The Counters zone,
- The Signals zone (if RS232).

The Type and Slave number Zone

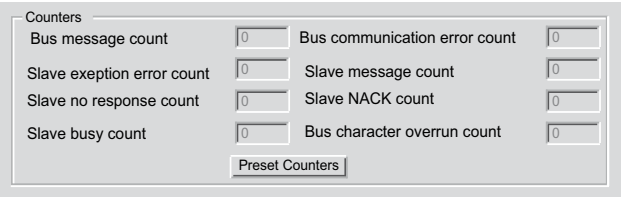
If the module has the function of Master in the Modbus link, this zone looks as following:

If the module has the function of Slave in the Modbus link, this zone looks as following:



The Counters Zone

This zone looks like this:



This zone shows the various debugging counters.

The Reset Counters button resets all the debug mode counters to zero.

Counter Operation

The Modbus serial communication debugging counters are:

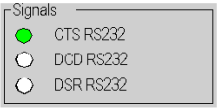
- **Bus message counter**: This counter indicates the number of messages that the module has detected on the serial link. Messages with a negative CRC check result are not counted.
- **Bus communication error counter**: This counter indicates the number of negative CRC check results counted by the module. If a character error (overflow, parity error) is detected, or if the message is less than 3 bytes long, the system that receives the data cannot perform the CRC check. In such cases, the counter is incremented accordingly.
- **Slave exception error counter**: This counter indicates the number of Modbus exception errors detected by the module.
- **Slave message counter**: This counter indicates the number of messages received and processed by the Modbus link.
- **Slave 'no response' counter**: This counter indicates the number of messages sent by the remote system for which it has received no response (neither a normal response, nor an exception response). It also counts the number of messages received in broadcast mode.
- **Negative slave acknowledgement counter**: This counter indicates the number of messages sent to the remote system for which it has returned a negative acknowledgement.
- **Slave busy counter**: This counter indicates the number of messages sent to the remote system for which it has returned a "slave busy" exception message.
- **Bus character overflow counter**: This counter indicates the number of messages sent to the module that it is unable to acquire because of character overflow on the bus. Overflow is caused by:
  - Character-type data that are transmitted on the serial port more quickly than they can be stored,
  - A loss of data due to a hardware event.

NOTE: For all counters, the count begins at the most recent restart, clear counters operation or module power-up.

The Signals Zone

This zone displays only if RS232 is selected in configuration screen. If RS485 is selected in configuration screen, this window is not displayed at all.

The Signals zone looks like this:



This zone indicates the activity of the signals:

- **CTS RS232**: shows the activity of the CTS signal.
- **DCD RS232**: shows the activity of the DCD signal.
- **DSR RS232**: shows the activity of the DSR signal.

Character Mode Communication for BMX NOM 0200

8

Subject of this Section

This chapter presents the software implementation of communication using Character Mode for BMX NOM 0200.

What's in this Chapter?

This chapter contains the following sections:

8.1 Generalities

About Character Mode Communication

Introduction

Communication in Character Mode enables dialog and communication functions to be carried out between the PLCs and the following devices:

- Regular peripherals (printer, keyboard-screen, workshop terminal, etc.),
- Specialized peripherals (barcode readers, etc.),
- Calculators (checking, production management, etc.),
- Heterogeneous devices (numerical commands, variable speed controllers, etc),
- External modem.

⚠ WARNING

CRITICAL DATA LOSS

Only use communication ports for non-critical data transfers.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

8.2 Character Mode Communication Configuration

Subject of this Section

This section describes the configuration process used when implementing Character Mode communication.

What's in this Section?

This section contains the following topics:

Character Mode Communication Configuration Screen

General

The following pages provide an introduction to the configuration screen for Character Mode communication.

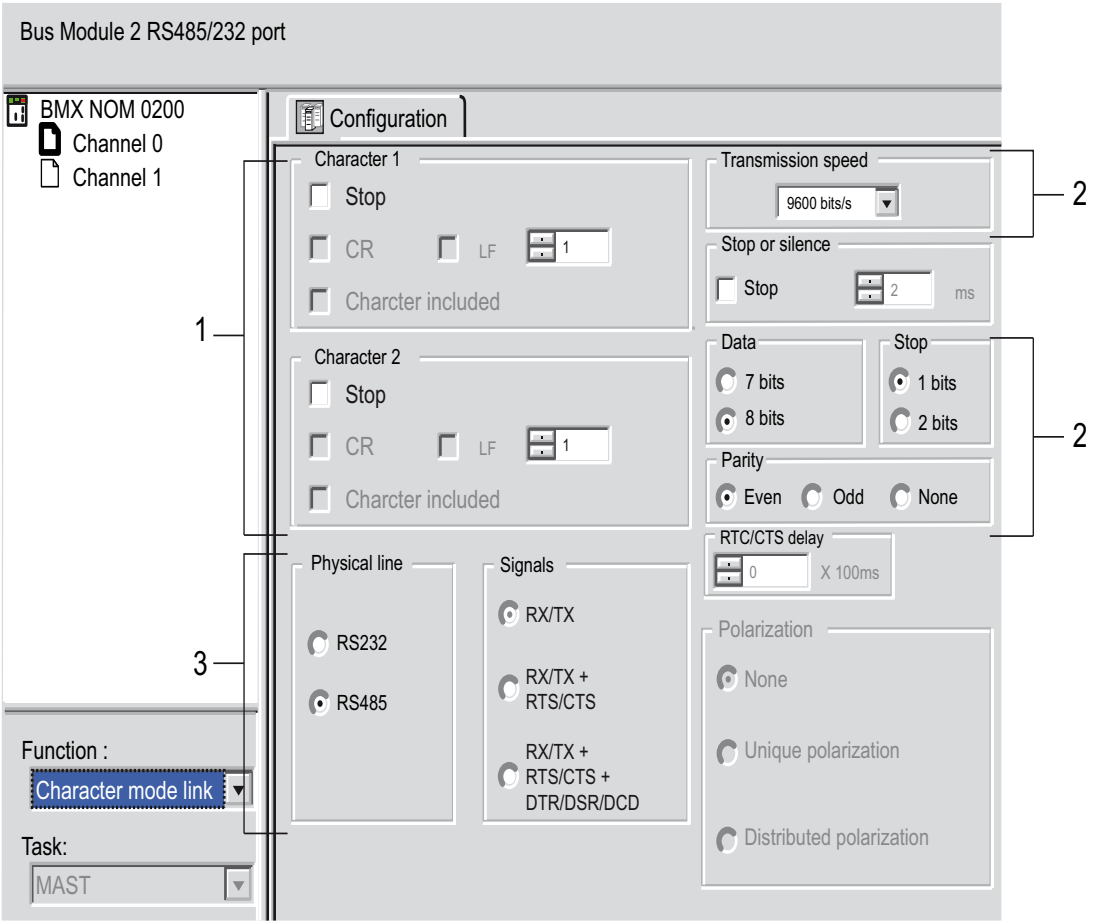
Accessing the Configuration Screen

The following table describes the procedure for accessing the configuration screen for Character Mode communication:

Step	Action
1	Open the BMX NOM 0200 sub-directory in the project browser *(see page 134)*.
2	Select the Channel to configure and the Character mode link function on the screen that appears.

Illustration

The figure below shows the default configuration screen for Character Mode communication on Channel 0:


Bus Module 2 RS485/232 port
BMX NOM 0200
Channel 0
Channel 1
Configuration
Character 1
Stop
CR
LF
1
Charcter included
Character 2
Stop
CR
LF
1
Charcter included
Transmission speed
9600 bits/s
Stop or silence
Stop
2
ms
Data
7 bits
8 bits
Stop
1 bits
2 bits
Parity
Even
Odd
None
RTC/CTS delay
0
X 100ms
Physical line
RS232
RS485
Signals
RX/TX
RX/TX + RTS/CTS
RX/TX + RTS/CTS + DTR/DSR/DCD
Polarization
None
Unique polarization
Distributed polarization
Function :
Character mode link
Task:
MAST
1
2
2
3

Description

This zone is used to configure channel parameters. In online mode, this zone is not accessible and will be grayed out. In offline mode, the zone is accessible but some parameters may not be accessible and will therefore be grayed out.

The following table shows the different zones of the Character Mode communication configuration screen:

Key	Element	Comment
1	Message end detection parameters *(see page 173)*	These parameters are accessible via two zones: ● Stop on reception, ● Stop on silence.
2	Transmission parameters *(see page 175)*	These parameters are accessible via four zones: ● Transmission speed, ● Data, ● Stop bits, ● Parity.
3	Signal and physical line parameters *(see page 177)*	These parameters are accessible via four zones: ● Physical line, ● Signals, ● RTS/CTS delay, ● Polarization.

NOTE: In this example, the "Polarization" and "RTS/CTS Delay" zones are grayed out respectively because an RS232 physical line and RX/TX signals have been chosen.

Accessible Functions in Character Mode

At a Glance

Function accessibility for configuration of the serial link of a BMX NOM 0200 using Character Mode protocol depends on the physical link being used.

Accessible Functions

The table below shows the different functions configurable according to the type of serial link used:

Function	RS 485 Link (Channel 0 or Channel 1)	RS 232 Link (Channel 0)
Transmission speed	X	X
Data	• 7 bits • 8 bits	• 7 bits • 8 bits
Stop	• 1 bit • 2 bits	• 1 bit • 2 bits
Parity	• Odd • Even • None	• Odd • Even • None
Stop on Reception	X	X
Stop on Silence	X	X
RX/TX Signals	X	X
RTS/CTS Signals	-	X
RTS/CTS delay	-	X
DTR/DSR/DCD Signals	-	X
Polarization	X	-

X Accessible Function
- Unaccessible Function

Default Values for Character Mode Communication Parameters

At a Glance

All Character Mode communication parameters have default values.

Default Values

The table below shows the default values for Character Mode communication parameters on Channel 0 and Channel 1 of BMX NOM 0200 module:

Configuration parameter	Value on Channel 0	Value on Channel 1
Physical Line	RS232	RS485
Signals	RX/TX	RX/TX (unique value)
Transmission speed	9600 bits/s	9600 bits/s
Parity	Odd	Odd
Data Bits	8 bits	8 bits
Stop bits	1 bit	1 bit
Polarization	None (unique value)	None

Message End Detection Parameters in Character Mode

At a Glance

After configuring the communication channel, you need to enter the message end detection parameters.

These parameters are accessible via two zones:

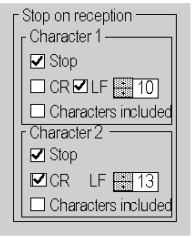
- The Stop on Reception Zone: stop on reception of a special character.
- The Stop on Silence Zone: stop on silence.

Conditions of Use

Selecting Stop on Silence means that Stop on Reception is deselected and vice versa.

The Stop on Reception Zone

This configuration zone appears on the screen as shown below:



A reception request can be terminated once a specific character is received.

By checking the Stop option, it is possible to configure Stop on Reception to be activated by a specific end-of-message character:

- CR: enables you to detect the end of the message by a carriage return.
- LF: enables you to detect the end of the message by a line feed.
- Data entry field: enables you to identify an end-of-message character other than the CR or LF characters, using a decimal value:
  - Between 0 and 255 if the data is coded over 8 bits
  - Between 0 and 127 if the data is coded over 7 bits
- Character included: enables you to include the end-of-message character in the reception table of the PLC application.

It is possible to configure two end-of-reception characters. In the window below, the end of reception of a message is detected by an LF or CR character.

The Stop on Silence Zone

This configuration zone appears on the screen as shown below:

This zone enables you to detect the end of a message on reception by the absence of message end characters over a given time.

Stop on Silence is validated by checking the Stop box. The duration of the silence (expressed in milliseconds) is set using the data entry field.

NOTE: The available values range from 1 ms to 10000 ms and depend on the transmission speed selected.

Transmission Parameters in Character Mode

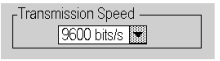
At a Glance

After configuring the communication channel, you need to enter the transmission parameters.

These parameters are accessible via four zones:
- The Transmission Speed zone,
- The Data zone,
- The Stop zone,
- The Parity zone.

The Transmission Speed Zone

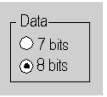
This configuration zone appears on the screen as shown below:



You can use this zone to select the transmission speed of the Character Mode protocol. The selected speed has to be consistent with the other devices. The configurable values are 300, 600, 1200, 2400, 4800, 9600, 19200, 57600 and 115200 (only on channel 0 in RS232 mode) bits per second.

The Data Zone

This configuration zone appears on the screen as shown below:



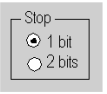
In this zone, you can specify the size of the data being exchanged on the link.

The available values are:
- 7 bits
- 8 bits

You are advised to adjust the number of data bits according to the remote device being used.

The Stop Zone

This zone looks like this:



The Stop zone allows you to enter the number of stop bits used for communication. You are advised to adjust the number of stop bits according to the remote device being used.

The configurable values are:
- 1 bit
- 2 bits

The Parity Zone

This configuration zone appears on the screen as shown below:

This zone enables you to determine whether a parity bit is added or not, as well as its type. You are advised to adjust parity according to the remote device being used.

The configurable values are:
- Even
- Odd
- None

Signal and Physical Line Parameters in Character Mode

At a Glance

After configuring the communication channel, you need to enter the physical line and signal parameters. These parameters are identical to the signal and physical line parameters for Modbus communication *(see page 149)* appart from an additional Polarization zone that is accessible only if the physical line selected is RS485.

The RTS/CTS Delay Zone

This configuration zone appears on the screen as shown below:

RTS/CTS delay zone is available only when both RS232 and RX/TX+RTS/CTS or RX/TX+RTS/CTS+DTR/DSR/DCD check boxes are selected. An RTS/CTS hardware flow control is performed.

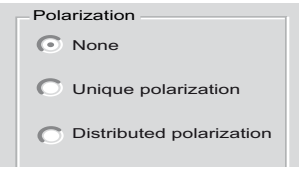
The RTS/CTS hardware flow control algorithm aims at preventing the overflow reception buffer (full duplex).

The RTS/CTS delay corresponds to the time out delay between the RTS rise up and the CTS rise up. A RTS/CTS delay value different from 0 also corresponds to the maximum waiting time between each character transmission after the rise of RTS and CTS signals. If the value is set to 0, UARTs can get stuck in a waiting state for an infinite time until the CTS rise up so the value 0 is used only in particular cases such as looping the RTS signal to the CTS signal in order to check that all connection cables are operating correctly.

NOTE: The default value is 0 ms.

The Polarization zone

This configuration zone shown below is accessible when "RS485" is selected in the "Physical Line" zone:



This zone gives the capability to choose between three types of configuration for the polarization on the channel:

- **None** to use no polarization in case you have your own termination.
- **Unique polarization** to use a low impedance polarization like in Modbus networks (the goal of this kind of polarization is to let the master maintain the default state).
- **Distributed polarization** to use a high polarization impedance (the goal of this kind of polarization is to let each device contribute to maintain the default state).

8.3 Character Mode Communication Programming

Character Mode Communication Functions

Available Functions

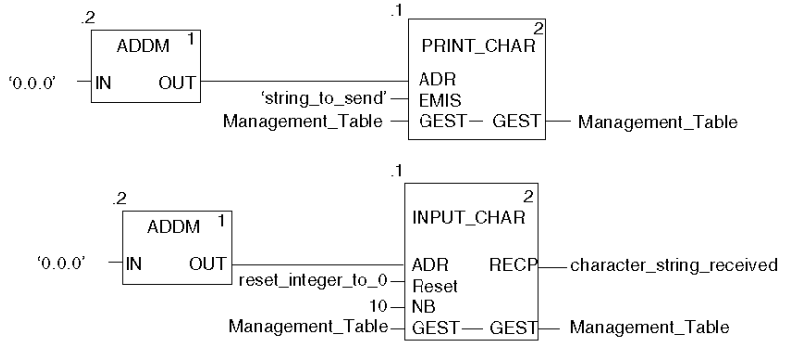
Two specific communication functions are defined for sending and receiving data via a communication channel in Character Mode:

- PRINT_CHAR: send a character string of a maximum of 16 x 1,024 bytes.
- INPUT_CHAR: read a character string of a maximum of 16 x 1,024 bytes.

NOTE: For INPUT_CHAR function, a configured time-out is necessary if the channel is configured without stop on silence, to acknowledge the activity bit of the function. For PRINT_CHAR function, it is advisable but not necessary to configure a time-out.

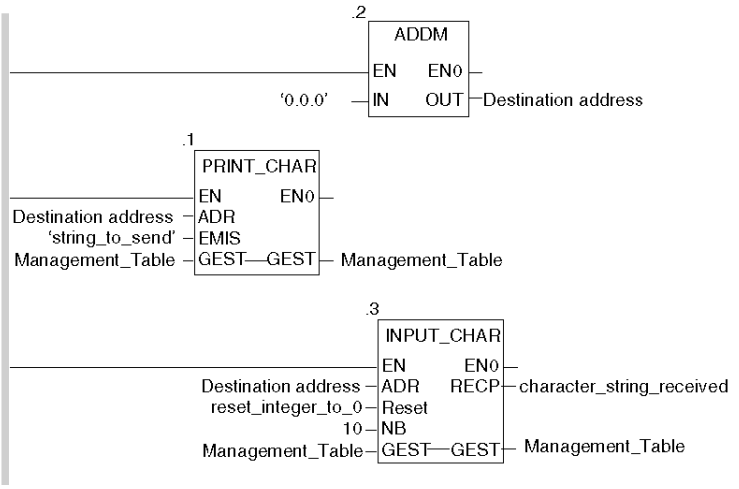
Example of Programming in FBD

The diagram below represents an example of programming of the PRINT_CHAR and INPUT_CHAR communication functions in FBD language:



Example of Programming in Ladder

The diagram below represents an example of programming of the PRINT_CHAR and INPUT_CHAR communication functions in Ladder language:



Example of Programming in ST

The lines of code below represent an example of programming of the PRINT_CHAR and INPUT_CHAR communication functions in ST language:

```
PRINT_CHAR(ADDM('0.1.0'), 'string_to_send',
Management_Table);
```

```
INPUT_CHAR(ADDM('0.1.0'), reset_integer_to_0, 10,
Management_Table, character_string_received);
```

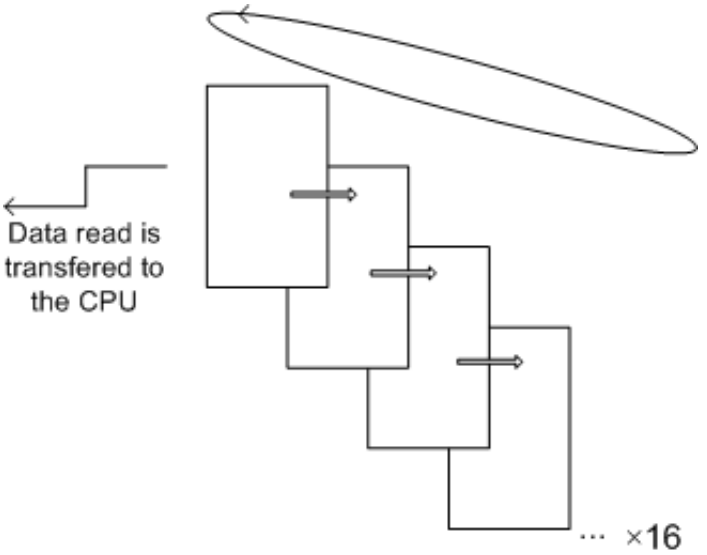
Feature of the INPUT_CHAR function

If the Reset input parameter is set to 1, all buffers are first reset then the module is waiting for the reception of data. Using this feature is advised in order to start properly a reception by removing old data that can remain in buffers.

Internal Mechanism of the BMX NOM 0200 Module

The data received is stored in a cycling set of 16 buffers in serial, each buffer containing 1024 bits.

The below figure represents this mechanism:



Two independent pointers allows access for reading and writing the data.

Each data reading access through the INPUT CHAR function erases the current buffer and moves the reading pointer to the following buffer.

When data is received:

- If no message end detection parameter has been configured, the data is writen in the current buffer until it is full, then the writing pointer moves to the following buffer.
- If any message end detection parameter (either a silence or a given character) has been configured, each time the late is reached, the writing pointer moves to the following buffer regardless of its position in the current buffer.
- When any bit of a buffer is overwriten due to the cycle behaviour of the mechanism, the full buffer is erased before the overwriting.

It is possible to launch up to 16 PRINT CHAR requests: the late will be sent in serial with a silence between each PRINT CHAR.

Cancelling an Exchange

There are two ways of programming that enable an exchange executed by the PRINT_CHAR and INPUT_CHAR functions to be cancelled. These are both presented in ST language below:

- Using the CANCEL function:

```
IF (%MW40.0) THEN
   %MW200:=SHR(%MW40,8;)
   CANCEL(%MW200,%MW185);
END_IF;
```

%MW40 is the GEST parameter (management table). %MW40.0 corresponds to the activity bit of the PRINT_CHAR function and is set to 1 when the communication function is active. If this bit is set to 1, the program carries out the following instructions:

  - Moves the %MW40 bits one byte (8 bits) to the right and loads the byte corresponding to the communication's exchange number into the %MW200 word.
  - Cancels the exchange whose exchange number is contained within the %MW200 word using the CANCEL function.

- Using the communication function's cancel bit:

```
IF (%MW40.0) THEN
   SET(%MW40.1);
   PRINT_CHAR(ADDM('0.1.0'), 'string_to_send', %MW40:4);
END_IF;
```

%MW40 is the GEST parameter (management table). %MW40.0 corresponds to the activity bit of the PRINT_CHAR function and is set to 1 when the communication function is active. If this bit is set to 1, the program sets the %MW40.1 bit, the function cancel bit, to 1. This stops communication of the PRINT_CHAR function.

NOTE: When using the communication function cancel bit, the function must be called in order to enable the cancel bit contained in the function exchange management word (%MW40 in this example).

NOTE: When using the communication function cancel bit, it is possible to cancel a communication from an animation table. This can be done by simply setting the function cancel bit to 1 (%MW40.1 in this example).

NOTE: This example of programming concerns the PRINT_CHAR function, but is equally applicable to the INPUT_CHAR function.

NOTE: The CANCEL function uses a report word for the CANCEL function (%MW185 in this example).

Description of ADDM Function Parameters

The following table outlines the various parameters for the ADDM function:

Parameter	Type	Description
IN	STRING	Address of device on bus or serial link. The syntax of the address is of the 'r.m.c.node' type. The address is made up of the following parameters: • **r**: rack number of the destination system, always = 0. • **m**: slot number of the destination system within the rack, always = 0. • **c**: channel number, always = 0 as the serial link of a remote system is always channel 0. • **node**: optional field that may be SYS or empty.
OUT	ARRAY [0..7] OF INT	Table showing the address of a device. This parameter can be used as an input parameter for several communication functions.

Description of PRINT_CHAR Function Parameters

The following table outlines the various parameters of the PRINT_CHAR function:

Parameter	Type	Description
ADR	ARRAY [0..7] OF INT	Address of the message receiving character mode channel given by the OUT parameter of the ADDM function.
EMIS	STRING	Character string to be sent.

GEST	ARRAY [0..3] OF INT	Exchange management table consisting of the following words:
		• Rank 1 word: a word managed by the system and consisting of two bytes:
		  • Most significant byte: exchange number
		  • Least significant byte: activity bit (rank 0) and cancel bit (rank 1)
		• Rank 2 word: a word managed by the system and consisting of two bytes:
		  • Most significant byte: operation report
		  • Least significant byte: communication report
		• Rank 3 word: a word managed by the user, which defines the maximum response time using a time base of 100 ms.
		• Rank 4 word: a word managed by the user which defines the length of the exchange.
		  • If this parameter length is set to 0 then the system sends the string entirely.
		  • If this parameter length is greater than the lenght of the string then the error 16#0A (Insufficient send buffer size) is returned into the 2nd management word and no character is sent.

Description of INPUT_CHAR Function Parameters

The following table outlines the various parameters of the INPUT_CHAR function:

Parameter	Type	Description
ADR	ARRAY [0..7] OF INT	Address of the message receiving character mode channel given by the OUT parameter of the ADDM function.
Reset	INT	This parameter may take two values: • Value 1: reset module reception memory to 0 • Value 0: do not reset module reception memory to 0
NB	INT	Length of character string to be received.

GEST	ARRAY [0..3] OF INT	Exchange management table consisting of the following words:
		• Rank 1 word: a word managed by the system and consisting of two bytes:
		  • Most significant byte: exchange number
		  • Least significant byte: activity bit (rank 0) and cancel bit (rank 1)
		• Rank 2 word: a word managed by the system and consisting of two bytes:
		  • Most significant byte: operation report
		  • Least significant byte: communication report
		• Rank 3 word: a word managed by the user which defines the maximum response time using a time base of 100 ms.
		• Rank 4 word: a word managed by the system which defines the length of the exchange.

RECP	STRING	Character string received. This string is saved in a character string.

8.4 Debugging Character Mode communication

Character Mode Communication Debug Screen

General

The Character Mode debug screen is accessible in online mode.

Accessing the Debug Screen

The following table describes the procedure for accessing the debug screen for Character Mode communication:

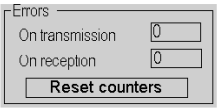
Step	Action
1	Access the configuration screen for Character Mode communication. *(see page 168)*
2	Select the "Debug" tab on the screen that appears.

Description of the Debug Screen

The debug screen consists of an Error zone and a Signals zone (if RS232).

The Error Zone

The Error zone looks like this:



This zone indicates the number of communication interruptions counted by the module:

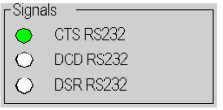
- **On transmission**: corresponds to the number of interruptions on transmission (image of `%MW4` word).
- **On reception**: corresponds to the number of interruptions on reception (image of `%MW5` word).

The Reset Counters button resets both counters to zero.

The Signals Zone

This zone is displayed only if RS232 is selected in configuration screen. If RS485 is selected in configuration screen, this window is not displayed at all.

The Signals zone looks like this:



This zone indicates the activity of the signals:

- **CTS RS232**: shows the activity of the CTS signal.
- **DCD RS232**: shows the activity of the DCD signal.
- **DSR RS232**: shows the activity of the DSR signal.

BMX NOM 0200 Module Diagnostics

9

9.1 BMX NOM 0200 Module Diagnostics

Subject of this Section

This section describes the diagnostics aspect in the implementation of a BMX NOM 0200 communication module.

What's in this Section?

This section contains the following topics:

Diagnostics of a BMX NOM 0200 Module

At a Glance

The module diagnostics function displays anomalies when they occur, classified according to their category:

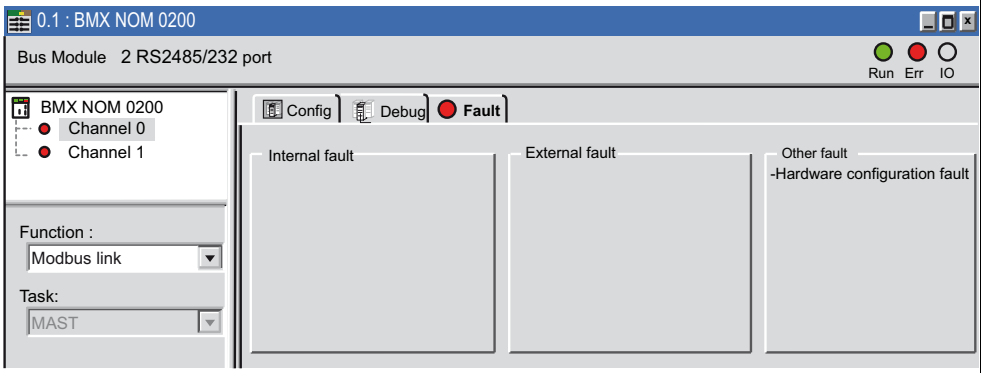
- **Internal detected error**:
  - module event
- **External event**:
  - Wiring control (broken-wire, overload or short-circuit)
- **Other anomalies**:
  - inoperative channel
  - configuration anomaly
  - module missing or off

A detected module error is indicated by a number of LEDs changing to red, such as:

- in the rack-level configuration editor:
  - the LED of the rack number
  - the LED of the slot number of the module on the rack
- in the module-level configuration editor:
  - the **Err** and **I/O** LEDs, depending on the type detected error
  - the **Channel** LED in the **Channel** field

Accessing the Module Diagnostic Screen

The table below shows the procedure for accessing the module diagnostic screen.

Step	Action
1	Open the module debugging screen.
2	Click on the module reference in the channel zone and select the **Fault** tab. **Result**: The list of module detected errors appears.  **Note:** It is not possible to access the module diagnostics screen if a configuration error, major breakdown error, or module missing error is detected. The following message then appears on the screen: " `The module is missing or different from that configured for this position.`"

Module Detected Errors List

The summary table below shows the various detected errors for a communication module:

Detected errors classification	Language objects
`Internal fault`: • `Module detected failure`	• `%MWr.m.MOD.2.0`
`External fault`: • `Terminal block`	• `%MWr.m.MOD.2.2`
`Other fault`: • `Faulty channel(s)` • `Hardware configuration fault` • `Module missing or off`	• `%MWr.m.MOD.2.1` • `%MWr.m.MOD.2.5` • `%MWr.m.MOD.2.6`

Detailed Diagnostics by Communication Channel

At a Glance

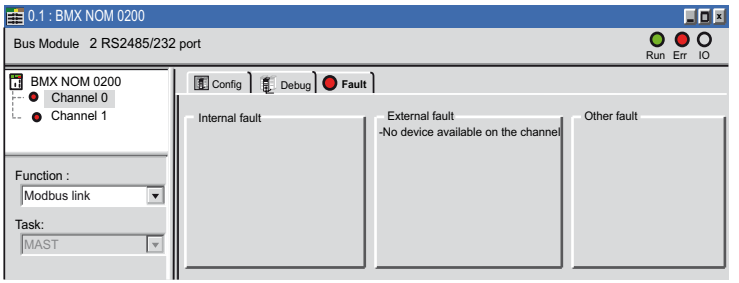
The channel Diagnostics function displays detected errors when they occur, classified according to their category:

- **Internal detected error**
  - self-tests in progress
- **External events**
  - device missing
  - device inoperative
  - serial-link communication time-out
- **Other detected errors**
  - line tool error
  - configuration error
  - communication loss
  - application error

A detected channel error is indicated in the **Debug** tab when the  LED, located in the **Error** column, turns red.

Accessing the Channel Diagnostic Screen

The table below shows the procedure for accessing the channel diagnostic screen.

Step	Action
1	Open the module debugging screen.
2	For the inoperative channel, click on the button  situated in the **Error** column. **Result**: The list of detected channel errors appears.  **Note:** Channel diagnostics information can also be accessed by program (instruction READ_STS).

Channel Detected Errors List

The summary table below shows the various detected errors for a configured serial link:

Detected errors classification	Language objects
Internal fault: • Self-tests in progress	• %MWr.m.c.2.4
External fault: • No device available on the channel • Device fault • Time-out error (CTS)	• %MWr.m.c.2.0 • %MWr.m.c.2.1 • %MWr.m.c.2.3
Other fault: • Line tool error • Hardware configuration fault • Problem communicating with the PLC • Application error	• %MWr.m.c.2.2 • %MWr.m.c.2.5 • %MWr.m.c.2.6 • %MWr.m.c.2.7