Modicon M340 for Ethernet

Communications Modules and Processors

User Manual

(Original Document)

12/2018

31007131.16

www.schneider-electric.com

All pertinent state, regional, and local safety regulations must be observed when installing and using this product. For reasons of safety and to help ensure compliance with documented system data, only the manufacturer should perform repairs to components.

When devices are used for applications with technical safety requirements, the relevant instructions must be followed.

Failure to use Schneider Electric software or approved software with our hardware products may result in injury, harm, or improper operating results.

Failure to observe this information can result in injury or equipment damage.

Table of Contents

Safety Information 11
About the Book 15
Part I Modicon M340 Hardware and Communication Requirements 19
Chapter 1 Modicon M340 Modules for Ethernet Communications . 21
1.1 External Features 22
BMX NOE 01x0 Physical Description 23
BMX P34 2020 Physical Description 24
BMX P34 2030/20302 Physical Description 25
1.2 Common Features of Modicon M340 Modules and Processors 26
Module Dimensions 27
Ethernet LED Indicators 28
10/100 BASE-T Interface 33
Chapter 2 Modicon M340 Ethernet Module Overview 35
General Presentation of an Ethernet Network 36
Rack Position: BMX NOE 01x0 and BMX P34 20x0x 37
Chapter 3 Choosing an Ethernet Communications Module or Processor for Modicon M340 39
Communication Module Features and Selection Guide 40
BMX P34 xxxxx Processors Catalog 41
Ethernet Service Selection Table 42
Compatibility: BMX NOE 01x0 and BMX P34 20x0 44
Chapter 4 Hardware Installation 45
Assembling a Modicon M340 Station 46
Grounding of Installed Modules 48
Modicon M340 Memory Cards 50
Memory Card Features 55
Wiring Considerations 57

Part II	Ethernet Communications	59
Chapter 5	IP Parameters	61
	Methods for IP Addressing	62
	Modicon M340 Rotary Switches	64
	Deriving IP Parameters from the MAC Address	66
	The IP Address Assignment Process	67
	Ethernet Port Status	70
Chapter 6	Multi-Module Communication	73
	Multi-Module Communication	73
Chapter 7	Description of Ethernet Communications Services	81
7.1	I/O Scanning Service	82
	I/O Scanning Service	83
	Read and Write Zones	85
7.2	Address Assignment through DHCP and BOOTP	86
	Address Assignment through DHCP/BOOTP	87
	Modicon M340 DHCP/BOOTP Example	88
	BMX NOE 01x0 as a DHCP Server	89
7.3	SNMP	91
	SNMP and Schneider Private MIB Overview	92
	SNMP Communication	93
	SNMP Operations Example	95
7.4	Global Data	96
	Global Data	96
7.5	Fast Device Replacement	99
	Replacing Devices	99
7.6	Bandwidth Monitoring	100
	Bandwidth Monitoring	101
	Static and Dynamic Bandwidth Monitoring	102
7.7	TCP/IP Messaging	104
	Data Exchange	105
	Transparent Device Access	106
7.8	Time Synchronization Service	108
	Introducing the Time Synchronization Service	109
	Using the Time Synchronization Service	112
	Using the R_NTPC Block for Time Synchronization	113

7.9 Electronic Mail Notification Service . . . 115
Introducing the Electronic Mail Notification Service . . . 116
Using the Electronic Mail Notification Service . . . 117
Using the SEND_EMAIL Block for Electronic Mail Notification . . . 118
Electronic Mail Notification Service Error Codes . . . 120
Part III Modicon M340 Module Descriptions and Specifications . . . 121
Chapter 8 The BMX NOE 01x0 Communication Module . . . 123
BMX NOE 01x0 Ethernet Services . . . 124
BMX NOE 01x0 Electrical Characteristics . . . 127
Standards and Certifications . . . 128
Chapter 9 The BMX P34 2020 and BMX P34 2030 CPUs . . . 129
BMX P34 20x0 Ethernet Services . . . 130
BMX P34 20x0x Electrical Characteristics . . . 132
Part IV Ethernet Configuration with Control Expert. . . . 133
Chapter 10 Software Configuration Parameters . . . 135
10.1 The Configuration Screen . . . 136
The Module Configuration Screen . . . 136
10.2 Security . . . 137
Security Features . . . 137
10.3 IP Configuration Parameters . . . 139
The IP Configuration Tab . . . 140
Configuration Parameters for IP Addresses . . . 142
Ethernet Frame Format . . . 143
10.4 Messaging Configuration Parameters . . . 144
The Messaging Configuration Tab . . . 145
Messaging Configuration Parameters . . . 147
10.5 I/O Scanner Configuration Parameters . . . 148
The I/O Scanning Tab . . . 149
I/O Scanning Contextual Menu for Copy/Cut/Paste . . . 154
I/O Scanning with Multiple Lines . . . 156
Introduction to Configuring Advantys from Control Expert . . . 158
Introduction to Configuring the PRM Master DTM. . . . 162
Introduction to Configuring a BMX PRA 0100 from Control Expert . . . 164
Property Box . . . 168
Saving an Advantys Configuration in an Control Application . . . 174
Managed Variables . . . 175
I/O Scanner Concepts . . . 177

	Scanning Period	178
	Configuration of Parameters Linked to the I/O Scanner Utility	179
	Configuration of General Parameters for the I/O Scanner	180
	I/O Scanner: Health Block Bits	181
	I/O Scanner: Device Control Block	183
10.6	Global Data Configuration Parameters	185
	The Global Data Configuration Tab	186
	Configuration of Global Data Variables	187
	Configuration of General Parameters for Global Data	189
	Configuration of Parameters Linked to Global Data	190
10.7	SNMP Configuration Parameters	191
	Configuring SNMP as an Agent	192
	SNMP Configuration Parameters	194
10.8	Address Server Configuration Parameters	196
	Address Server	196
10.9	Bandwidth Checking	198
	Bandwidth Checking Configuration	198
10.10	Electronic Mail Notification Service Configuration Parameters	200
	The SMTP Tab	200
10.11	Time Synchronization Service Configuration Parameters	203
	The NTP Tab	203
Chapter 11	**Configuring an Ethernet Network**	**209**
	Configuration Methodology for an Ethernet Network	210
	Adding the Module to an Ethernet Network	212
Chapter 12	**Debugging with Control Expert**	**215**
	Module Debugging Screen	216
	General Debugging Parameters	218
	Debugging Parameters for TCP/IP Utilities	221
	Debugging Parameters for I/O Scanning	222
	Debugging Parameters for Global Data	223
Chapter 13	**Ethernet Language Objects**	**225**
13.1	Language Objects and IODDTs of Ethernet Communication	226
	Language Objects and IODDTs of Ethernet Communication	227
	Details of T_COM_EIP IODDT	228
	Implicit Exchange Language Objects Associated with the Application-Specific Function	230
	Explicit Exchange Language Objects Associated with the Application-Specific Function	231

13.2 Exchange Objects of Type T_COM_ETH_BMX 233
Details of Implicit Exchange Objects of the IODDT Type T_COM_ETH_BMX 234
Details of Explicit Exchange Objects of the IODDT Type T_COM_ETH_BMX 236
Details of Explicit Exchange Objects for an Ethernet Function 238
13.3 Language Objects Associated with BMX NOE 01x0 and BMX P34 20x0 CPU Configuration 239
Language Objects for Configuration 240
Language Objects for Implicit Exchange 241
Language Objects for Explicit Exchange 243
13.4 Language Objects and Generic IODDT Applicable to Communication Protocols 246
Details of the Language Objects of the IODDT of Type T_GEN_MOD 246

Chapter 14 M340 Ethernet Communications Quick Start 249
Hardware Installation 250
Modicon M340 Configuration with Control Expert 251
Ethernet Network Configuration with Control Expert 254
Assign the BMX NOE 0100 IP Address 255
Configure Ethernet Communication Service (I/O Scanning) 256
Associate the Network with the Module 260
Build a Program 261
Connect the System and Download the Configuration 262
Debugging the Module 263

Part V Embedded Web Pages 265

Chapter 15 Embedded Web Pages 267
15.1 M340 Web Pages 268
Introduction to Web Services 269
Embedded HTTP Server 271
BMX NOE 0100 Home Page 272
BMX NOE 0100 Monitoring Page 273
BMX NOE 0100 Diagnostics Page 274
Server Rack Display Page 275
BMX NOE 0100 Setup Page 276
FTP Security Page 277
HTTP and Data Editor (Write) Page 278
Upload MIB File 280
Properties 281

15.2	Ethernet Service Diagnostics Pages	282
	Global Data	283
	I/O Scanning	284
	Messaging	285
	Statistics	286
	Bandwidth Monitoring	287
15.3	Class C Services	288
	Introduction to Class C Services	289
	Graphic Editor Overview	290
	Top Window User Functions	292
	Display Window User Functions	297
	Property Sheet	300
	Security	302
	Graphic Editor Applet Parameters	303
	Graphic Objects	304
	Extended Graphic Objects	321
Appendices		**331**
Appendix A	**TCP/IP Technical Characteristics**	**333**
	Summary of TCP/IP Characteristics	334
	Address Management for Ethernet Modules	336
	Modbus Communication on the TCP/IP Profile	337
	Managing TCP Connections for Modbus	340
	Opening a TCP/IP Connection	341
	Closing a TCP/IP Connection	343
	Broken TCP/IP Connections	344
Appendix B	**IP Address Details**	**347**
	IP Addresses and Classes	348
	Multicasting Considerations	353
Appendix C	**Transparent Ready Service Classes**	**355**
	Service Classes	355
Appendix D	**Schneider Private MIB**	**359**
	The Schneider Private MIB	360
	Schneider Private MIB Tree Structure	362
	MIB Subtree Description	370
	Switch Subtree Description	371
	Port 502 Messaging Subtree Description	372
	I/O Scanning Subtree Description	373
	Global Data Subtree Description	374

Web Subtree Description . . . 375
Address Server Subtree Description . . . 376
Equipment Profile Subtree Description . . . 377
Time Management Subtree Description . . . 379
Email Subtree Description . . . 380
Transparent Factory MIB Version . . . 381
Private Traps and MIB Files . . . 382
Glossary . . . 385
Index . . . 395

Safety Information

Important Information

NOTICE

Read these instructions carefully, and look at the equipment to become familiar with the device before trying to install, operate, service, or maintain it. The following special messages may appear throughout this documentation or on the equipment to warn of potential hazards or to call attention to information that clarifies or simplifies a procedure.

The addition of this symbol to a “Danger” or “Warning” safety label indicates that an electrical hazard exists which will result in personal injury if the instructions are not followed.

This is the safety alert symbol. It is used to alert you to potential personal injury hazards. Obey all safety messages that follow this symbol to avoid possible injury or death.

DANGER
DANGER indicates a hazardous situation which, if not avoided, **will result in** death or serious injury.

WARNING
WARNING indicates a hazardous situation which, if not avoided, **could result in** death or serious injury.

CAUTION
CAUTION indicates a hazardous situation which, if not avoided, **could result** in minor or moderate injury.

NOTICE
NOTICE is used to address practices not related to physical injury.

PLEASE NOTE

Electrical equipment should be installed, operated, serviced, and maintained only by qualified personnel. No responsibility is assumed by Schneider Electric for any consequences arising out of the use of this material.

A qualified person is one who has skills and knowledge related to the construction and operation of electrical equipment and its installation, and has received safety training to recognize and avoid the hazards involved.

BEFORE YOU BEGIN

Do not use this product on machinery lacking effective point-of-operation guarding. Lack of effective point-of-operation guarding on a machine can result in serious injury to the operator of that machine.

⚠ WARNING
UNGUARDED EQUIPMENT ● Do not use this software and related automation equipment on equipment which does not have point-of-operation protection. ● Do not reach into machinery during operation. **Failure to follow these instructions can result in death, serious injury, or equipment damage.**

This automation equipment and related software is used to control a variety of industrial processes. The type or model of automation equipment suitable for each application will vary depending on factors such as the control function required, degree of protection required, production methods, unusual conditions, government regulations, etc. In some applications, more than one processor may be required, as when backup redundancy is needed.

Only you, the user, machine builder or system integrator can be aware of all the conditions and factors present during setup, operation, and maintenance of the machine and, therefore, can determine the automation equipment and the related safeties and interlocks which can be properly used. When selecting automation and control equipment and related software for a particular application, you should refer to the applicable local and national standards and regulations. The National Safety Council's Accident Prevention Manual (nationally recognized in the United States of America) also provides much useful information.

In some applications, such as packaging machinery, additional operator protection such as point-of-operation guarding must be provided. This is necessary if the operator's hands and other parts of the body are free to enter the pinch points or other hazardous areas and serious injury can occur. Software products alone cannot protect an operator from injury. For this reason the software cannot be substituted for or take the place of point-of-operation protection.

Ensure that appropriate safeties and mechanical/electrical interlocks related to point-of-operation protection have been installed and are operational before placing the equipment into service. All interlocks and safeties related to point-of-operation protection must be coordinated with the related automation equipment and software programming.

NOTE: Coordination of safeties and mechanical/electrical interlocks for point-of-operation protection is outside the scope of the Function Block Library, System User Guide, or other implementation referenced in this documentation.

START-UP AND TEST

Before using electrical control and automation equipment for regular operation after installation, the system should be given a start-up test by qualified personnel to verify correct operation of the equipment. It is important that arrangements for such a check be made and that enough time is allowed to perform complete and satisfactory testing.

⚠ WARNING

EQUIPMENT OPERATION HAZARD

- Verify that all installation and set up procedures have been completed.
- Before operational tests are performed, remove all blocks or other temporary holding means used for shipment from all component devices.
- Remove tools, meters, and debris from equipment.

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

Follow all start-up tests recommended in the equipment documentation. Store all equipment documentation for future references.

Software testing must be done in both simulated and real environments.

Verify that the completed system is free from all short circuits and temporary grounds that are not installed according to local regulations (according to the National Electrical Code in the U.S.A, for instance). If high-potential voltage testing is necessary, follow recommendations in equipment documentation to prevent accidental equipment damage.

Before energizing equipment:

- Remove tools, meters, and debris from equipment.
- Close the equipment enclosure door.
- Remove all temporary grounds from incoming power lines.
- Perform all start-up tests recommended by the manufacturer.

OPERATION AND ADJUSTMENTS

The following precautions are from the NEMA Standards Publication ICS 7.1-1995 (English version prevails):

- Regardless of the care exercised in the design and manufacture of equipment or in the selection and ratings of components, there are hazards that can be encountered if such equipment is improperly operated.
- It is sometimes possible to misadjust the equipment and thus produce unsatisfactory or unsafe operation. Always use the manufacturer’s instructions as a guide for functional adjustments. Personnel who have access to these adjustments should be familiar with the equipment manufacturer’s instructions and the machinery used with the electrical equipment.
- Only those operational adjustments actually required by the operator should be accessible to the operator. Access to other controls should be restricted to prevent unauthorized changes in operating characteristics.

About the Book

At a Glance

Document Scope

This manual describes the implementation of the BMX NOE 01x0 communication module and the BMX P34 20x0 CPU platform with embedded ports on Ethernet networks.

Validity Note

This document is valid for EcoStruxure™ Control Expert 14.0 or later.

The technical characteristics of the devices described in the present document also appear online. To access the information online:

Step	Action
1	Go to the Schneider Electric home page *www.schneider-electric.com*.
2	In the **Search** box type the reference of a product or the name of a product range. ● Do not include blank spaces in the reference or product range. ● To get information on grouping similar modules, use asterisks (*).
3	If you entered a reference, go to the **Product Datasheets** search results and click on the reference that interests you. If you entered the name of a product range, go to the **Product Ranges** search results and click on the product range that interests you.
4	If more than one reference appears in the **Products** search results, click on the reference that interests you.
5	Depending on the size of your screen, you may need to scroll down to see the data sheet.
6	To save or print a data sheet as a .pdf file, click **Download XXX product datasheet**.

The characteristics that are presented in the present document should be the same as those characteristics that appear online. In line with our policy of constant improvement, we may revise content over time to improve clarity and accuracy. If you see a difference between the document and online information, use the online information as your reference.

Related Documents

Title of documentation	Reference number
Modicon M340, Processors, Setup Manual	*35012676 (English)*, *35012677 (French)*, *35013351 (German)*, *35013352 (Italian)*, *35013353 (Spanish)*, *35013354 (Chinese)*
EcoStruxure™ Control Expert, Operating Modes	33003101 (English), 33003102 (French), 33003103 (German), 33003104 (Spanish), 33003696 (Italian), 33003697 (Chinese)
Modicon M580, M340, and X80 I/O Platforms, Standards and Certifications	EIO0000002726 (English), EIO0000002727 (French), EIO0000002728 (German), EIO0000002730 (Italian), EIO0000002729 (Spanish), EIO0000002731 (Chinese)
EcoStruxure™ Control Expert, I/O Management, Block Library	33002531 (English), 33002532 (French), 33002533 (German), 33003684 (Italian), 33002534 (Spanish), 33003685 (Chinese)
EcoStruxure™ Control Expert, Communication, Block Library	33002527 (English), 33002528 (French), 33002529 (German), 33003682 (Italian), 33002530 (Spanish), 33003683 (Chinese)
Transparent Ready User Guide	31006929 (English), 31006930 (French), 31006931 (German), 31006932 (Spanish)
FactoryCast for Modicon M340 User Manual	35015192 (English), 35015193 (French), 35015194 (German), 35015195 (Spanish), 35015196 (Italian)

You can download these technical publications and other technical information from our website at *www.schneider-electric.com/en/download*.

Product Related Information

⚠ 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 product. Follow all local and national safety codes and standards. **Failure to follow these instructions can result in death, serious injury, or equipment damage.**

Part I
Modicon M340 Hardware and Communication Requirements

About this Part

This part contains an overview of hardware and communications requirements for the BMX NOE 01x0 modules and BMX P34 20x0 CPUs on Ethernet networks.

For Modicon M340 system installation and specifications, see the book *Modicon M340 using Unity Pro: Processors, Racks and Power Supply Modules*.

NOTE: The BMX NOE 01x0 modules have an embedded Web server. You can access the website for monitoring, diagnosing or setting up the module. Refer to the *FactoryCast for Modicon M340 User Manual* for a description of the website.

What Is in This Part?

This part contains the following chapters:

Chapter	Chapter Name	Page
1	Modicon M340 Modules for Ethernet Communications	21
2	Modicon M340 Ethernet Module Overview	35
3	Choosing an Ethernet Communications Module or Processor for Modicon M340	39
4	Hardware Installation	45

Chapter 1
Modicon M340 Modules for Ethernet Communications

Introduction

This chapter provides graphic representations of the Modicon M340 modules that can be used for communications between Ethernet networks and Modicon M340 rack assemblies. The appropriate modules are:

- **BMX NOE 0100** and **BMX NOE 0110**: These network option modules are dedicated to Ethernet communications.
- **BMX P34 2020** and **BMX P34 2030**: In addition to their functionality as processors, these CPU modules have embedded ports for Ethernet communications.

Elsewhere in this guide is information about selecting the appropriate hardware for your system and application needs *(see page 39)*.

What Is in This Chapter?

This chapter contains the following sections:

Section	Topic	Page
1.1	External Features	22
1.2	Common Features of Modicon M340 Modules and Processors	26

Section 1.1
External Features

About this Section

This section identifies the external features and ports on the BMX NOE 01x0 modules and BMX P34 20x0 CPUs.

What Is in This Section?

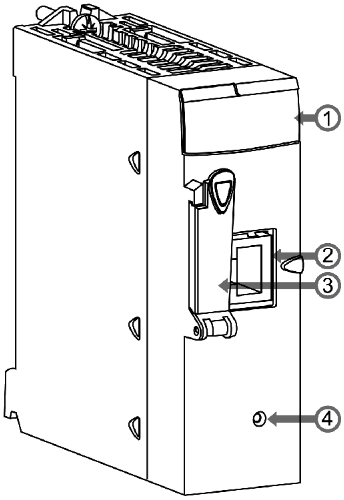
This section contains the following topics:

Topic	Page
BMX NOE 01x0 Physical Description	23
BMX P34 2020 Physical Description	24
BMX P34 2030/20302 Physical Description	25

BMX NOE 01x0 Physical Description

The Module

This illustration shows the BMX NOE 01x0 communication modules:



Legend:

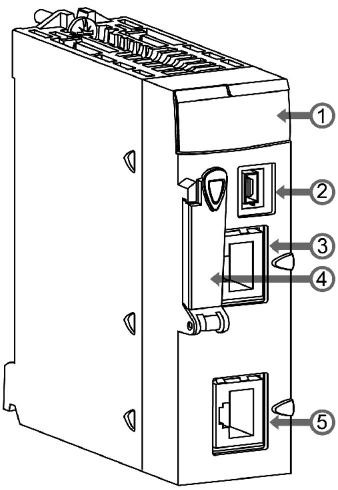
- 1 *LED display* *(see page 28)*
- 2 *Ethernet port* *(see page 33)*
- 3 *memory card slot*
  A memory card can be used to store files, such as Web pages and log files. Elsewhere in this guide are performance characteristics of standard and optional memory cards *(see page 50)*.
- 4 *reset button*
  Press this button to cold start the module. (After the reset, the network recognizes the module as a new device.)

NOTE: Confirm that the door of the memory card slot is closed when the module is powered.The memory card may experience electrostatic discharge if the door remains open.

BMX P34 2020 Physical Description

The Module

This illustration shows the BMX P34 2020 CPU:



Legend:

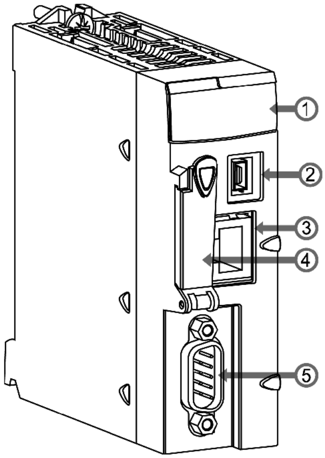
- 1 *LED display (see page 28)*
- 2 *USB port*
- 3 *Ethernet port (see page 33)*
- 4 *memory card slot*
  Used to:
  - store files, such as Web pages and log files
  - back up code
  - back up application

  Elsewhere in this guide are performance characteristics of standard and optional memory cards *(see page 50)*.
- 5 *Modbus serial port*

BMX P34 2030/20302 Physical Description

The Module

This illustration shows the BMX P34 2030/20302 CPUs:



Legend:
- 1 *LED display* *(see page 28)*
- 2 *USB port*
- 3 *Ethernet port* *(see page 33)*
- 4 *memory card slot*
  Used to:
  - store files, such as Web pages and log files
  - back up code
  - back up application

  Elsewhere in this guide are performance characteristics of standard and optional memory cards *(see page 50)*.
- 5 *CANopen port*

Section 1.2
Common Features of Modicon M340 Modules and Processors

About this Section

This section describes the common physical features of the BMX NOE 01x0 modules and BMX P34 20x0 CPUs.

What Is in This Section?

This section contains the following topics:

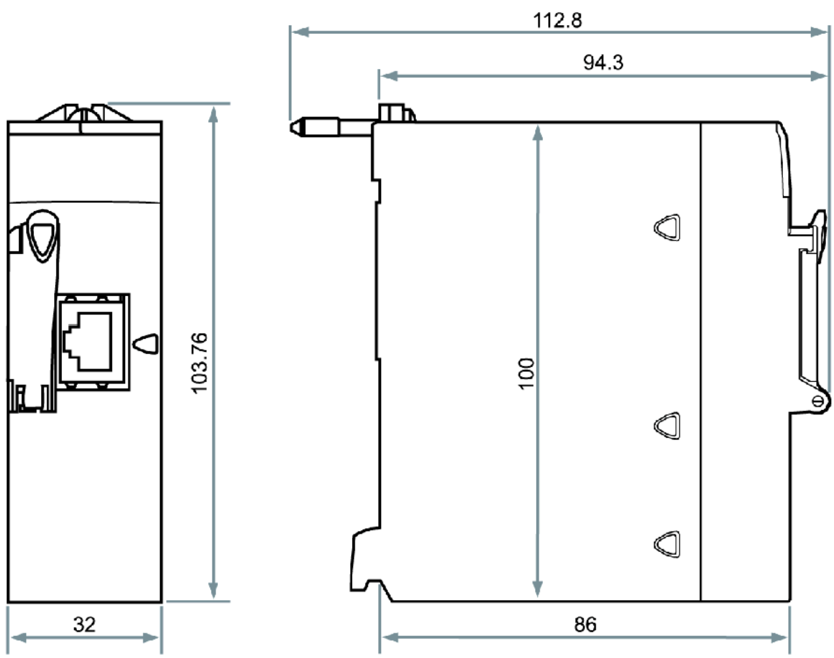
Topic	Page
Module Dimensions	27
Ethernet LED Indicators	28
10/100 BASE-T Interface	33

Module Dimensions

Dimensions

The dimensions of the Modicon M340 modules conform to the characteristics of the rack.

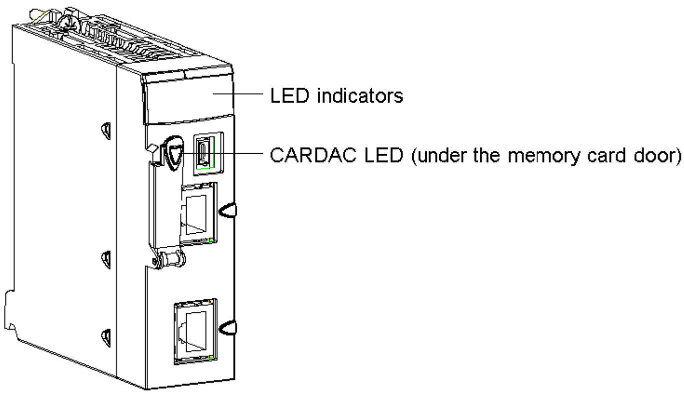
This figure shows the dimensions (in mm) for the M340 Ethernet communication modules and M340 CPUs:



Ethernet LED Indicators

Introduction

There are several LEDs available on the front panel of each Modicon M340 module or processor, enabling rapid diagnosis of the PLC status:

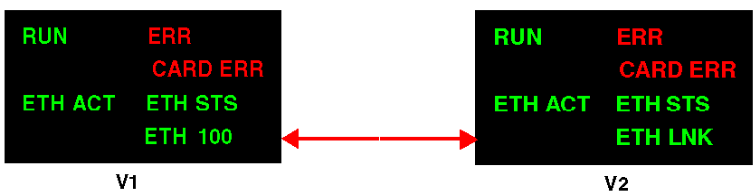


These LEDs provide information on:

- PLC functioning
- the memory card
- communication with the modules
- serial communication
- communication on the CANopen network
- communication on the Ethernet network

BMX NOE 01x0 LEDs

The following diagram shows the diagnostic LEDs on the BMX NOE 01x0 modules. Note that two displays exist, depending on whether you are using firmware V1 or V2 (or greater) of the module.

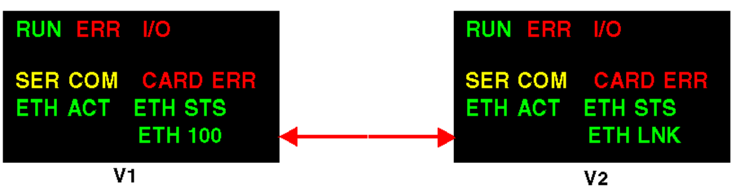


The colors and blink patterns of the LEDs indicate the status and operating conditions of Ethernet communications on the module:

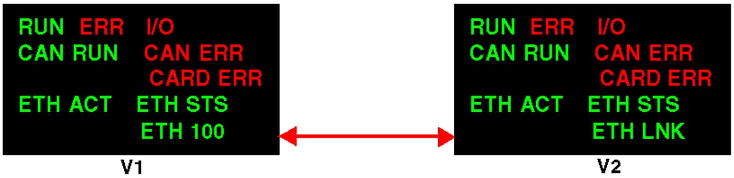
Label	Pattern	Indication
RUN (green): operational state	on	● Module is operating and configured.
	flashing	● Module is blocked by a software detected error.
	off	Module is not configured (application is absent, invalid, or incompatible).
ERR (red): detected error	on	Processor, system, or configuration detected error
	flashing	● Module is not configured (application is absent, invalid, or incompatible). ● Module is blocked by a software detected error.
	off	Normal (no detected errors)
ETH STS (green): Ethernet communication status	on	Communication OK
	2 flashes	Invalid MAC address
	3 flashes	Link not connected
	4 flashes	Duplicate IP address
	5 flashes	Waiting for a server IP address
	6 flashes	Secure and safe mode (with default IP address)
	7 flashes	Configuration conflict between rotary switches and internal configuration
CARDERR (red): memory card detected error	on	● Memory card is missing. ● Memory card is not usable (bad format, unrecognized type). ● Memory card had been removed and reinserted.
	off	● Memory card is valid and recognized.
Note 1: Rapid flashing is defined as ON for 50 ms and OFF for 50 ms. Note 2: Slow flashing is defined as ON for 200 ms and OFF for 200 ms.		

BMX P34 2020, BMX P34 2030/20302 Processor LEDs

The following diagram shows the diagnostic LEDs on the BMX P34 2020 processor. Note that two displays exist, depending on whether you are using firmware V1 or V2 (or greater) of the processor.



The following diagram shows the diagnostic LEDs on the BMX P34 2030/20302 processor. Note that two displays exist, depending on whether you are using firmware V1 or V2 (or greater) of the processor.



The colors and blink patterns of the LEDs indicate the status and operating conditions of Ethernet communications on the module:

Label	Pattern	Indication
RUN (green): operational state	on	● PLC hardware and PLC program operations are normal. ● Module is in RUN state.
	flashing	● PLC is in STOP mode or a blocking error in the application has been detected. ● Processor is configured but not in RUN state.
	off	PLC is not configured (application is absent, invalid, or incompatible).
ERR (red): detected error	on	Processor, system, or configuration detected error
	flashing	● PLC is not configured (application is absent, invalid, or incompatible). ● PLC is in STOP mode or a blocking error in the application has been detected.
	off	Normal (no detected errors)
ETH STS (green): Ethernet communication status	on	Communication OK
	2 flashes	Invalid MAC address
	3 flashes	Link not connected
	4 flashes	Duplicate IP address
	5 flashes	Waiting for a server IP address
	6 flashes	Secure and safe mode (with default IP address)
	7 flashes	Configuration conflict between rotary switches and internal configuration

Label	Pattern	Indication
CARDERR (red): memory card detected error	on	• Memory card is missing. • Memory card not usable (bad format, unrecognized type). • Memory card content is inconsistent with internal RAM application.
	off	• Memory card is valid and recognized. • Application on card is consistent with the internal RAM application.
I/O (red): input/output status	on	• Error detected on a configured module or CPU channel • Configuration mismatch with the application (module missing...)
	off	Normal (no detected errors)
SER COM (yellow): serial data status	flashing	Data exchange (send/receive) on the serial connection in progress
	off	No data exchange on the serial connection
CAN RUN (green): CANopen operations	on	CANopen network operational
	rapid flashing (note 1)	Automatic detection of data flow or LSS services in progress (alternates with CAN ERR).
	slow flashing (note 2)	CANopen network is pre-operational.
	1 flash	CANopen network is stopped.
	3 flashes	Downloading CANopen firmware.
CAN ERR (red): CANopen detected error	on	CANopen bus is stopped.
	rapid flashing (note 1)	Automatic detection of data flow or LSS services in progress (alternates with CAN RUN).
	slow flashing (note 2)	CANopen configuration is not valid.
	1 flash	At least one detected error counter has reached or exceeded alert level.
	2 flashes	A guard event (NMT slave or NMT master) or a heartbeat event has occurred.
	3 flashes	The SYNC message was not received before the end of the communication cycle period.
	off	No error detected on CANopen.

Label	Pattern	Indication
CARDAC (green): memory card access **Note:** This LED is located under the memory card door *(see page 23)*.	on	Access to the card is enabled.
	flashing	Activity on the card: during each access, the card LED is set to OFF, then back to ON.
	off	Access to the card is disabled. You can remove the card after you disable card access by setting system bit %S65 to 0.
Note 1: Rapid flashing is defined as ON for 50 ms and OFF for 50 ms. Note 2: Slow flashing is defined as ON for 200 ms and OFF for 200 ms.		

LED Differences Between Firmware V1 and V2 Modules for both BMX NOE 01x0 and BMX P34 20x0x

The following table describes the meaning of the ETH ACT and ETH 100 LEDs on the front panel for firmware V1 NOE and CPU modules.

Label	Pattern	Indication
ETH ACT (green): Ethernet communication (transmission/reception activity)	on	Ethernet link detected: no communications activity.
	off	No Ethernet link detected.
	flashing	Ethernet link detected: receiving or sending packets.
ETH 100 (green): Ethernet transmission speed	on	Ethernet transmission at 100 Mbit/s (Fast Ethernet).
	off	Ethernet transmission at 10 Mbit/s (Ethernet) or no link detected.

The following table describes the meaning of the ETH ACT and ETH LNK LEDs on the front panel for firmware V2 NOE and CPU modules.

Label	Pattern	Indication
ETH ACT (green): Ethernet communication (transmission/reception) activity	on	Communications activity detected.
	off	No communications activity detected.
ETH LNK (green): Ethernet link status	on	Ethernet link detected.
	off	No Ethernet link detected.

NOTE:
- Rapid flashing is defined as ON for 50 ms and OFF for 50 ms.
- Slow flashing is defined as ON for 200 ms and OFF for 200 ms.

10/100 BASE-T Interface

General

The module's 10/100 BASE-T interface is a standard RJ45 connector. In an industrial environment, use a cable with the following characteristics:

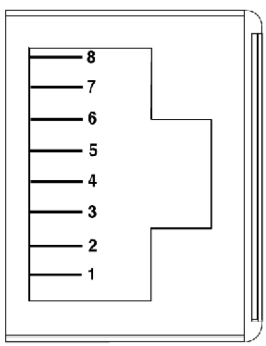
- shielded twisted double pair
- impedance 100 Ω±15 Ω (from 1 to 16 MHz)
- maximum attenuation 11.5 dB/100 meters
- maximum length 100 meters

The following straight-through ConneXium cables fit these requirements for connecting terminal devices:

Description	Reference		Length, m (ft)
	Low Smoke Zero Halogen	UL/CSA CMG	
Straight-through cable with RJ45 ends	490 NTW 000 02	490 NTW 000 02 U	2 (6.6)
	490 NTW 000 05	490 NTW 000 05 U	5 (16.4)
	490 NTW 000 12	490 NTW 000 12 U	12 (39.4)
	490 NTW 000 40	490 NTW 000 40 U	40 (131.2)
	490 NTW 000 80	490 NTW 000 80 U	80 (262.5)

Pin Assignment

The connector:



Pinout assignment table:

Pin	Signal
1	TD+
2	TD-
3	RD+
4	not connected
5	not connected
6	RD-
7	not connected
8	not connected

NOTE: If there is a connection via a shielded cable, the connector casing on the module is linked up to the ground connection.

Line Speed

The different line speeds that are available for the BMX NOE 01x0 and the integrated Ethernet port of the BMX P34 2020/2030/20302 CPUs are:
- 100 Mb in half duplex
- 100 Mb in full duplex
- 10 Mb in half duplex
- 10 Mb in full duplex

The user can not configure the line speed. Characteristics of speed adaptation are:
- Auto-sensing and auto-negotiation allow the Ethernet module to quickly configure itself to the local Ethernet switch's speed and duplex mode.
- The negotiated speed between two Ethernet devices is limited to the speed of the slower device.

Port Status

Elsewhere in this guide is a discussion of the Ethernet port status *(see page 70)*.

Chapter 2
Modicon M340 Ethernet Module Overview

Introduction

This chapter contains an overview of the BMX NOE 01x0 modules and BMX P34 20x0 CPUs on Ethernet networks.

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
General Presentation of an Ethernet Network	36
Rack Position: BMX NOE 01x0 and BMX P34 20x0x	37

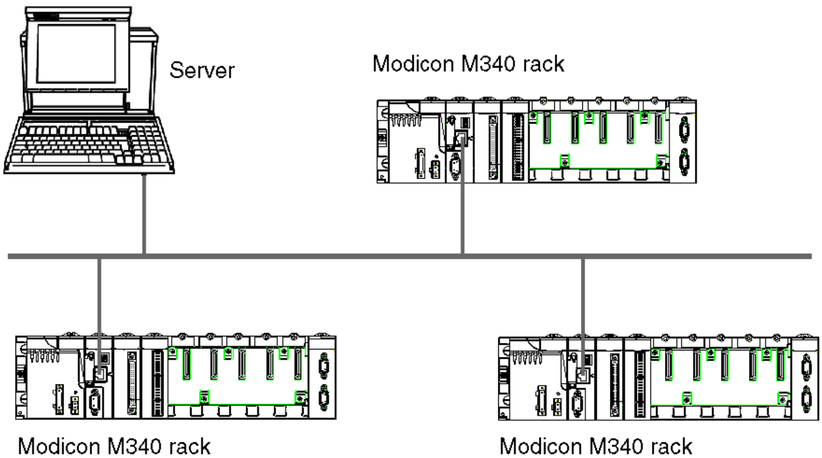
General Presentation of an Ethernet Network

Overview

In general, Ethernet is designed to facilitate:

- coordination between programmable controllers
- local or centralized supervision
- communication with the business data processing of production
- communication with remote inputs/outputs

An Ethernet network:



NOTE: Be careful routing the Ethernet cable through the factory as the cable may be damaged by other equipment within the factory.

Rack Position: BMX NOE 01x0 and BMX P34 20x0x

Introduction

This topic describes the appropriate rack positions of the BMX NOE 01•0 modules and BMX P34 20x0x CPUs on a station assembly during installation *(see page 45)*.

Available Modules

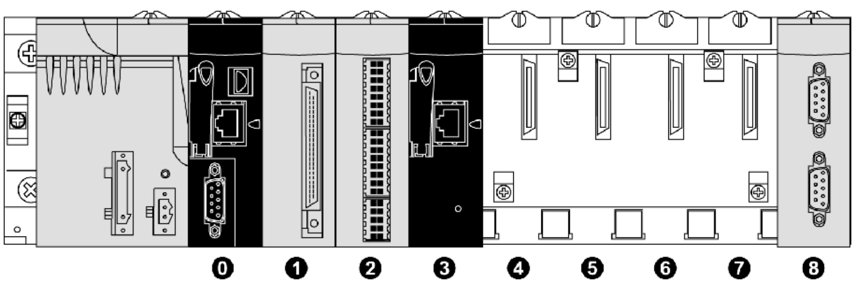
A Modicon M340 CPU can manage an entire rack. These three CPUs have Ethernet communication ports:
- BMX P34 2020
- BMX P34 2030
- BMX P34 20302

The BMX P34 20x0x CPUs can manage a station composed of:
- discrete I/O modules
- analog I/O modules
- function modules (counting, communication, etc.)

NOTE: Refer to the *Modicon M340 Using Unity Pro -- Processors, Racks, and Power Supply Modules Setup Manual* for specific part numbers.

Rack Position

The following rack assembly includes a Modicon M340 CPU (in this case a BMX P34 2030) and a BMX NOE 01•0 module (in this case a BMX NOE 0100). Rack positions 0 to 8 are indicated. (The double-wide power supply is mounted at the beginning of the rack.)



0 BMX P34 2030 CPU at rack position 0
1 discrete I/O module at rack position 1
2 counter module at rack position 2
3 BMX NOE 0100 Ethernet communications module at rack position 3
4-7 available rack positions
8 Modicon M340 extension module at rack position 8

Module Positioning

Mechanically, it is possible to position the BMX P34 CPU or the I/O modules in any slot. However, we recommend placing the modules in these positions:

- **BMX P34 20x0•**: Place the CPU at position 0. Position 0 also conforms to the CPU form factor.
- **BMX NOE 01x0**: Modules that conform to the Modicon M340 I/O form factor, such as the BMX NOE 01•0 modules, can be placed in any other available slot.
- **BMX CPS 2000**: The double-wide rack power supply is mounted at the beginning of the assembly on the left.

BMX P34 20•0• Processors

The following table shows the rack operations and communications details for the BMX P34 20x0x CPUs:

CPU	Physical Format	Maximum Number of I/O*		Maximum Memory Size	Integrated Port		
		Discrete	Analog		CANopen	Ethernet	Modbus Serial
BMX P34 2020	simple	1024	256	4096 Kb	—	X	X
BMX P34 2030/ 20302	simple	1024	256	4096 Kb	X	X	—
*: per rack							
X: available							
—: not available							

Chapter 3
Choosing an Ethernet Communications Module or Processor for Modicon M340

Introduction

This section helps you select the hardware that is most appropriate for your Ethernet application and system requirements.

Ethernet networks can be complex. Therefore, the BMX NOE 01x0 communication modules and BMX P34 20x0 CPUs support a variety of network services.

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Communication Module Features and Selection Guide	40
BMX P34 xxxxx Processors Catalog	41
Ethernet Service Selection Table	42
Compatibility: BMX NOE 01x0 and BMX P34 20x0	44

Communication Module Features and Selection Guide

Introduction

The Modicon M340 PLCs can communicate with Ethernet networks using:
- Ethernet communication modules (BMX NOE 01x0 *(see page 123)*)
- embedded ports on the Modicon M340 CPUs (BMX P34 20x0x *(see page 129)*)

The interfaces for these communications modules are described below. When making your selection, consider each module's services *(see page 42)*.

Embedded Ports on Modicon M340 CPUs

The BMX P34 20x0x CPUs have these ports:

CPU	Ports
BMX P34 2020	• USB • Ethernet • Modbus serial
BMX P34 2030/20302	• USB • Ethernet • CANopen

The locations of the ports are shown at External Features *(see page 22)*.

BMX P34 xxxxx Processors Catalog

Introduction

The choice of BMX P34 xxxxx processor is made, primarily, according to its characteristics and possibilities.

BMX P34 xxxxx Processors Catalog

The following table describes the important maximum characteristics of the BMX P34 xxxxx processors.

Characteristic		BMX P34 1000	BMX P34 2000	BMX P34 2010/ 20102	BMX P34 2020	BMX P34 2030/ 20302
Maximum Number of channels	Discrete rack inputs/outputs	512	1024	1024	1024	1024
	Analog inputs/outputs	128	256	256	256	256
	Expert channels (counting, PTO, MPS, NOM, etc.)	20	36	36	36	36
Maximum Number of modules	Embedded Serial port	1	1	1	1	-
	Embedded Ethernet port	-	-	-	1	1
	Embedded CANopen port	-	-	1	-	1
	Network communication (TCP/IP)	2	3	3	3	3
	AS-i fieldbus[1] communication	2	4	4	4	4
Memory size	User application	2048 Kb	4096 Kb	4096 Kb	4096 Kb	4096 Kb
Legend	**1** The AS-i field bus requires at least PLC Operating System V2.10.					

Ethernet Service Selection Table

Available Services

This tables summarizes the services that are available for the different Ethernet communications modules.

Service	Ethernet Modules	Embedded Ports on CPUs	
	BMX NOE 01x0	BMX P34 2030/20302	BMX P34 2020
Connection at 10 Mbits/s	X	X	X
Connection at 100 Mbits/s	X	X	X
TCP/IP	X	X	X
SNMP:			
• Standard MIB	X	X	X
• MIB Transparent Factory	X	X	X
I/O Scanner	X	—	—
Address Server (BOOTP/DHCP server)	X	—	—
BOOTP/DHCP client	X	X	X
Modbus Messaging	X	X	X
Firmware update via Unity Loader	X	X	X
Embedded HTTP server	X	X	X
Global Data	X	—	—
NTP	X	—	—
SMTP	—	X	X
Fast Device Replacement (FDR server)	X	—	—
FDR client	X	X	X
Diagnostics from Web pages	X	X	X
User-customizable Web pages	with an installed class C memory card (BMX NOE 0110 modules only)	—	—
Additional interface	N/A	CANopen	Serial
Legend			
X: service is present			
—: service is not available			

See the detailed descriptions for:
- Ethernet services *(see page 81)*
- Schneider’s Transparent Ready service classes *(see page 355)*
- Class C services for the BMX NOE 01x0 modules *(see page 355)*

Compatibility: BMX NOE 01x0 and BMX P34 20x0

M340 NOE and CPU Version Compatibility

Note the following compatibility issues when plugging an BMX NOE 01x0 module with a BMX P34 20x0 CPU in the rack. Certain combinations of firmware V1 and V2 modules are supported. The following table shows compatible module combinations.

	NOE Firmware V1	NOE Firmware V2 (with an application for an NOE V1)	NOE Firmware V2 (with an application for an NOE V2)
CPU Firmware V1	Compatible	Compatible	Incompatible
CPU Firmware V2 (with an application for a CPU V1)	Compatible	Compatible	Incompatible
CPU Firmware V2 (with an application for a CPU V2)	*Incompatible	Incompatible	Compatible

* In this case, the NOE module will be not recognized as correct by the CPU when the application starts. An I/O error message will be displayed, as if a module other than an NOE were inserted in the slot.

Chapter 4
Hardware Installation

Introduction

This chapter describes the installation of the BMX NOE 01x0 modules and BMX P34 20x0 CPUs.

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Assembling a Modicon M340 Station	46
Grounding of Installed Modules	48
Modicon M340 Memory Cards	50
Memory Card Features	55
Wiring Considerations	57

Assembling a Modicon M340 Station

Introduction

⚠ WARNING
MODULE DESTRUCTION - LOSS OF APPLICATION
Disconnect all power to the rack before the installation of the BMX P34 20x0 CPUs
Failure to follow these instructions can result in death, serious injury, or equipment damage.

This topic provides steps for installing the Modicon M340 communication modules and CPUs on the rack.

Modicon M340 modules and processors are powered by the backplane bus.

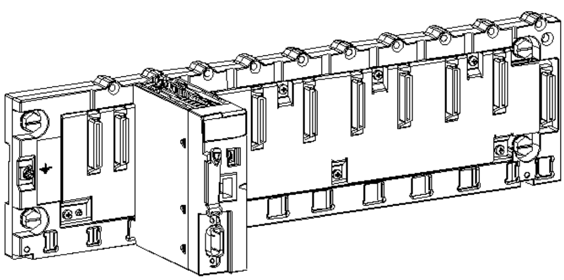
Fitting operations (installation, assembly, and disassembly) are described below.

Elsewhere in this guide is a discussion of the proper address locations on the backplane for BMX NOE 01x0 and Modicon M340 CPUs *(see page 37)*.

Installing a Processor

A BMX P34 20x0 processor is always installed on the rack in slot marked **00** (address 0).

The following diagram shows a BMX P34 20x0 processor mounted on a BMX XBP rack in the slot marked **00** (address 0):

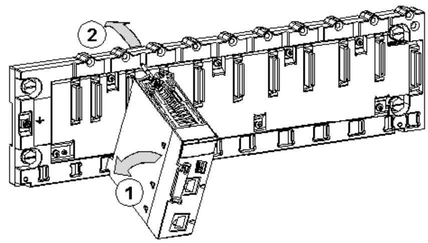
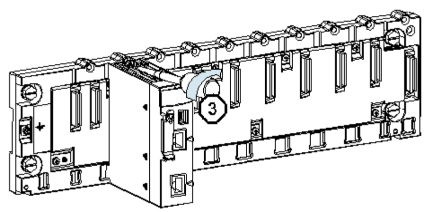


NOTE: Before installing a module, take off the protective cap from the module connector located on the backplane.

Mounting Instructions

NOTE: The mechanical assembly instructions for the BMX NOE 01x0 and the BMX P34 20x0 CPUs are identical.

To mount a module or processor on the rack:

Step	Action	Illustration
1	Position the two pins on the reverse side of the module or CPU (at the bottom) in the corresponding slots on the rack. Note: Before positioning the pins, make sure you have removed the protective cover.	The following diagram describes steps 1 and 2: 
2	Incline the module or CPU towards the top of the rack so that the module sits flush with the back of the rack. It is now set in position.	
3	Tighten the safety screw to ensure that the module or CPU is held in place on the rack. The recommended tightening torque is between 0.4 and 1.5 Nm .	The following diagram describes step 3: 

Grounding of Installed Modules

General

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

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.

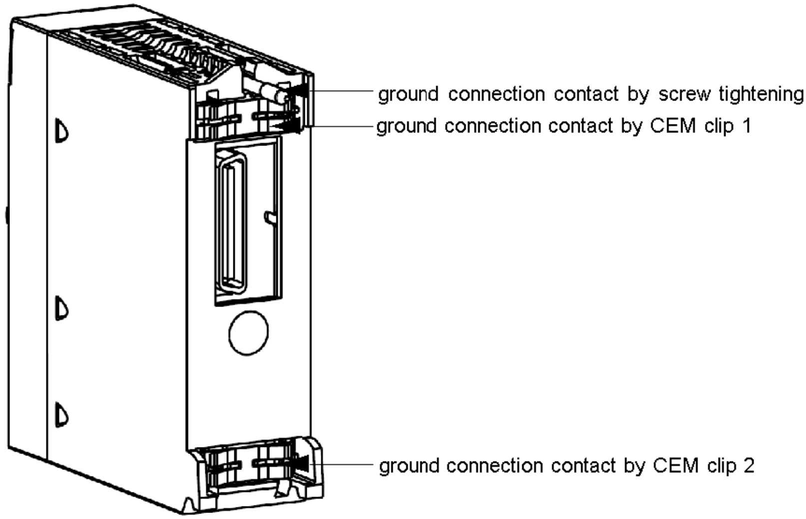
WARNING

UNINTENDED EQUIPMENT OPERATION

Tighten the clamping screws of the modules. A break in the circuit could lead to an unexpected 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.

Modicon M340 Memory Cards

Introduction

⚠ WARNING
RISK OF LOST APPLICATION Do not remove the memory card from the module while the PLC is running. Remove the memory card only when the power is off. **Failure to follow these instructions can result in death, serious injury, or equipment damage.**

This topic discusses the Schneider memory cards that are available for Modicon M340 CPUs and the BMX NOE 01x0 modules. Elsewhere in this guide is a discussion of the card location on the modules *(see page 22)*.

Card Functionality

The following table describes the functionality of the different memory cards when inserted in Modicon M340 CPUs and BMX NOE 01x0 modules:

Memory Card Part	Functionality		
	BMX P34 CPUs	BMX NOE 01x0	Comment
BMX RMS 008MP	• application backup • Web server activation on the embedded Ethernet port (Transparent Ready class B)	—	Delivered with the BMX P34 CPUs
BMX RMS 008MPF	• application backup • Web server activation on the embedded Ethernet port (Transparent Ready class B) • 8 MB file storage (usability)	—	Order separately
BMX RMS 128MPF	• application backup • Web server activation on the embedded Ethernet port (Transparent Ready class B) • 128 MB file storage (usability)	—	Order separately
BMXRWSB000M	—	• services for Transparent Ready class B30	Delivered with the BMX NOE 0100 module

Memory Card Part	Functionality		
	BMX P34 CPUs	BMX NOE 01x0	Comment
BMX RWSF016M	—	• services for Transparent Ready class C30 • Web page memory (16 MB)	No longer sold. Nevertheless, it can be updated to BMX RWSFC016M.
BMX RWSFC016M	—	• services for Transparent Ready class C30 • FactoryCast services • Web page memory (64 MB)	Delivered with the BMX NOE 0110. This card is required for user-customizable Web pages and FactoryCast services.
Note: See the detailed discussions for: • Ethernet service classes A, B, C, and D *(see page 355)* • Class C services for the BMX NOE 0100 module *(see page 355)*			

Card Services

NOTICE

INOPERABLE MEMORY CARD

Do not format the memory card with a non-Schneider tool. The memory card needs a structure to contain program and data. Formatting with another tool destroys this structure.

Failure to follow these instructions can result in equipment damage.

⚠ CAUTION

UNINTENDED EQUIPMENT OPERATION

Do not use a write-protected memory card with the module. Write-protected cards prevent some services from operating properly.

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

The following table lists the services that are available when the memory card is inserted in various Modicon M340 modules:

Memory Card	Module	Program Backup	File Storage	FDR Server	Web Server Class	FactoryCast
BMX RMS 008MP	BMX P34 1000	Yes	No	—	—	—
	BMX P34 2000	Yes	No	—	—	—
	BMX P34 2010	Yes	No	—	—	—
	BMX P34 20102	Yes	No	—	—	—
	BMX P34 2020	Yes	No	No	B	—
	BMX P34 2030	Yes	No	No	B	—
	BMX P34 20302	Yes	No	No	B	—
BMX RMS 008MPF	BMX P34 1000	Yes	No	—	—	—
	BMX P34 2000	Yes	Yes	—	—	—
	BMX P34 2010/ 20102	Yes	Yes	—	—	—
	BMX P34 2020	Yes	Yes	No	B	—
	BMX P34 2030/ 20302	Yes	Yes	No	B	—
BMX RMS 128MPF	BMX P34 1000	Yes	No	—	—	—
	BMX P34 2000	Yes	Yes	—	—	—
	BMX P34 2010/ 20102	Yes	Yes	—	—	—
	BMX P34 2020	Yes	Yes	No	B	—
	BMX P34 2030/ 20302	Yes	Yes	No	B	—
BMXRWSB000M	BMX NOE 0100	—	—	Yes	B	—
BMXRWSC016M	BMX NOE 0100	—	—	Yes	C	—
BMXRWSFC016M	BMX NOE 01x0	—	—	Yes	C	Yes
B = embedded web page C = user-customized web page Elsewhere in this guide is a description of the Web server classes *(see page 355)*.						

NOTE: The NOE module works only with a memory card that is present at boot-up time. A memory card that is inserted during NOE operations is not recognized.
Although operation is possible without a valid memory card inserted in the module, a valid memory card should be present at all times in the module to ensure correct operation of the module and all its services.

Card Compatibility with NOE Versions

The following table lists the compatibility issues between memory card versions and BMX NOE 01x0 versions:

Memory Card →	BMX RWS C016M Class C V1	BMX RWS C016M upgraded to FactoryCast V1.1	BMX RWS C016M upgraded to FactoryCast V2	BMX RWSF C032M V1.1	BMX RWSF C032M V2	BMX_RWS B 000M Class B
NOE ↓						
NOE 0100 Firmware V1	Compatible	*Incompatible	*Incompatible	*Incompatible	Not supported	Compatible
NOE 0100 Firmware V2	Compatible (with a V1 application)	*Incompatible	*Incompatible	Not supported	Not supported	Compatible (with a V1 application)
NOE 0110 Firmware V1	Not supported	Compatible	Not supported	Compatible	Not supported	Compatible
NOE 0110 Firmware V2	Not supported	Not supported	Compatible	Not supported	Compatible	Compatible (with a V1 application)

* In these cases, the Web server is embedded in the memory card, then loaded into the NOE module. At start-up, the NOE will block this memory card.

Card Compatibility with CPU Versions

The following table lists the compatibility issues between memory card versions and BMX P34 20•0 versions:

Memory Card →	BMX RMS 008MP / MPF Web Pages V1 or V1.1	BMX RMS 008MP / MPF Web Pages V2
CPU ↓		
CPU Firmware V1	Compatible	Compatible (with a V1 application)
CPU Firmware V2 (with an application for a CPU V1)	Compatible (with a V1 application)	Compatible (with a V1 application)
CPU Firmware V2 (with an application for a CPU V2)	Compatible	Compatible

Precautions

NOTICE

MEMORY CARD DESTRUCTION

To keep the memory card in normal working order, the following precautions should be taken:

- Do not remove the memory card from its slot when the module is accessing it (green access LED on or flashing).
- Do not touch the memory card connections.
- Keep the memory card away from electrostatic and electromagnetic sources as well as heat, sunlight, water and moisture.
- Keep the door to the memory card slot closed while the module is running.
- Avoid impacts to the memory card.
- Check the postal service security policy before sending a memory card by postal service. In some countries the postal service exposes mail to high levels of radiation, as a security measure. These high levels of radiation may erase the contents of the memory card and render it unusable.

Failure to follow these instructions can result in equipment damage.

Memory Card Features

Introduction

This topic discusses the features and services provided by Schneider memory cards for Modicon M340 CPUs and the BMX NOE 01x0 modules.

Card Features

The following table describes the features and services of the different memory cards when inserted in Modicon M340 CPUs and the BMX NOE 01x0:

	Memory Card	BMXRMS008MP / BMXRMS008MPF / BMXRMS128MPF	BMXRWSB000M	BMXRWSFC032M
	Module	BMX P34 2020/2030/20302	BMX NOE 0100	BMX NOE 0110
	TR Class level	B	B	C
Monitoring	Data Editor	X	X	X
	Data Editor Lite	X	X	X
	Graphic Editor	-	-	X
	Graphic Viewer	-	-	X
	Symbol, unlocated access	-	-	X
	Custom Web pages	-	-	X
Diagnostic	Ethernet services Lite	X	X	-
	Ethernet services	-	-	X
	Rack Viewer Lite	X	X	-
	Multi-Rack Viewer	-	-	X
	Zoom module in rack	X	-	X
	Zoom bus CANopen	X	-	X
	Alarm Viewer	-	-	X
Setup	HTTP security	X	X	X
	FTP security	-	-	X

	Memory Card	BMXRMS008MP / BMXRMS008MPF / BMXRMS128MPF	BMXRWSB000M	BMXRWSFC032M
FactoryCast	SOAP XML support	-	-	X
	RDE on Pocket PC	-	-	X
	Widget library	-	-	X
	NTP	-	-	-
	FTP	-	X	X
	Web Designer support	-	-	X
Legend				
X: service is present				
—: service is not available				

NOTE: You can update the content of the BMXRWSC016M card to include the features and services of the BMXRWSFC032M card using Unity Loader.

Without Memory Card

If no memory card is inserted in the module, you cannot access the website. The following error message appears: "Access Error: Site temporary unvailable. Try again. No SD card present."

Also, if no memory card is inserted in the module:
- The address server will not work (NOE modules only).
- NTP has no time zone files (NOE modules only).
- Unity Loader will not work (NOE and CPU modules).

Wiring Considerations

The Link

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

- The RJ45 10/100 BASE-T interface connector gets connected or disconnected when the power is on.
- Modules are re-initialized when the power is switched back on.

Part II
Ethernet Communications

About this Part

This part describes Ethernet communications.

What Is in This Part?

This part contains the following chapters:

Chapter	Chapter Name	Page
5	IP Parameters	61
6	Multi-Module Communication	73
7	Description of Ethernet Communications Services	81

Chapter 5
IP Parameters

About this Chapter

This chapter describes the assignment of IP parameters to the BMX NOE 01x0 and BMX P34 20x0 CPUs. Each network address must be valid and unique on the network.

Elsewhere in this guide you can find information about:

- address management for Ethernet modules *(see page 336)*
- IP address classes *(see page 348)*

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Methods for IP Addressing	62
Modicon M340 Rotary Switches	64
Deriving IP Parameters from the MAC Address	66
The IP Address Assignment Process	67
Ethernet Port Status	70

Methods for IP Addressing

Addressing Methods

The Modicon M340 modules and CPUs with Ethernet functionality can have their addresses set through the rotary switches *(see page 64)*, the Control Expert IP Configuration tab *(see page 140)*, and combinations of the two:

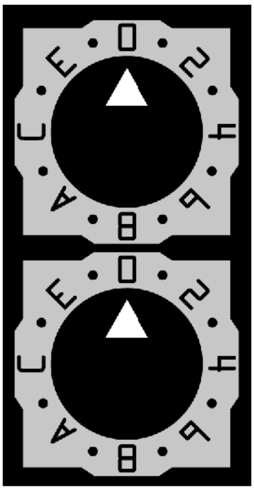
Address Method	Description
STORED	The lower rotary switch is set to STORED (manufacturer default setting), and the module uses the application's configured parameters.
device name (over DHCP)	There are two components of the device name: ● default device name of the module: ❍ BMX_2020_*xxy* (BMX P34 2020 CPU) ❍ BMX_2030_*xxy* (BMX P34 2030/20302 CPU) ❍ BMX_0100_*xxy* (BMX NOE 0100 and BMX PRA 0100 modules) ❍ BMX_0110_*xxy* (BMX NOE 0110 module) ● numeric value between 00 and 159 set on the rotary switches *(see page 64)* (For the default device name, *xx* is the value of the upper rotary switch and *y* is the value of the lower rotary switch.) **Example:** For a BMX NOE 0100 module, values of 120 (12 x 10) and 6 (6 x 1) on the respective upper and lower rotary switches indicate a value of 126. The value is appended to the default device name (BMX_0100_*xxy*) to create the valid DHCP device name of BMX_0100_126.
CLEAR IP	The lower rotary switch is set to **CLEAR IP**, and the module uses its MAC-based default IP address *(see page 66)*.
BOOTP	Set the lower rotary switch *(see page 64)* to one of its **BOOTP** positions to get an address over BOOTP (see note). **Note:** To configure the module in the application to get its address from a BOOTP server, see "from a server," below.

<table>
<tr><th>Address Method</th><th>Description</th></tr>
<tr><td>from a server (STORED)</td><td>A server-assigned IP address can then be obtained from either a BOOTP or DHCP server (see page 86).
BOOTP:
Set the lower rotary switch to one of its STORED positions.
Select From a server on the IP Configuration tab (see page 140).
Leave the Device Name field empty.
DHCP:
Set the lower rotary switch to one of its STORED positions.
Select From a server on the IP Configuration tab (see page 140).
Enter a valid device name in the Device Name field.
Note: The M340 Ethernet modules will not receive an IP address from a BOOTP/DHCP server on application download if the IP configuration has not changed.</td></tr>
<tr><td>disabled</td><td>Communications are disabled.</td></tr>
<tr><td colspan="2">Note: An error can occur when the assigned address is a mismatch for the address in the application. Details of these errors are at Ethernet Port Status (see page 70).</td></tr>
</table>

Modicon M340 Rotary Switches

Introduction

The BMX NOE 01•0 or BMX P34 20x0 operates as a single node on an Ethernet LAN and possibly other networks. The module must have a unique IP address. The two rotary switches on the back of the module provide a simple way to select an IP address:



NOTE: Set the arrow firmly into the desired position. If you do not feel the switch click into place, the value of the switch may be incorrect or undetermined.

Summary of Valid IP Address Settings

Each rotary switch position that you can use to set a valid IP address is marked on the module. The following information summarizes the valid address settings:

- **device name**: For a switch-set device name, select a numeric value from 00 to 159. You can use both switches:
  - On the upper switch (Tens digit), the available settings are 0 to 15.
  - On the lower switch (Ones digit), the available settings are 0 to 9.

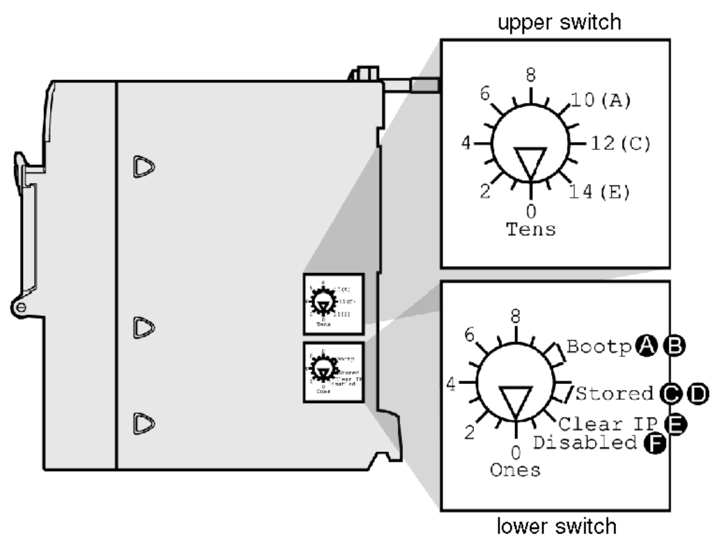
  The device name is calculated from the sum of the two switch values. For example, a BMX P34 2020 CPU with the switch setting in the above figure is assigned the DHCP device name **BMX_2020_123**.
  The selection on the lower switch of any non-numeric (**BOOTP**, **STORED**, **CLEAR IP**, **DISABLED**) makes the setting on the upper switch inconsequential.
- **BOOTP**: To get an IP address from a BOOTP server, select either of the two BOOTP positions on the bottom switch.
- **STORED**: The device uses the application's configured (stored) parameters.

- **CLEAR IP**: The device uses the default IP parameters.
- **DISABLED**: The device does not respond to communications.

The functionality of the rotary switch when used in conjunction with the Control Expert IP Configuration tab *(see page 140)* is discussed throughout the IP Address chapter *(see page 61)*.

Switch Labels

To assist you in setting the rotary switches to their proper positions, a label is affixed to the right side of the module. The switch settings are described in this table:



Upper Switch
0 to 9: Tens value for the device name (0, 10, 20 . . . 90)
10(A) to 15(F): Tens value for the device name (100, 110, 120 . . . 150)
Lower Switch
0 to 9: Ones value for the device name (0, 1, 2 . . . 9)
BOOTP: Set the switch to A or B to receive an IP address from a BOOTP server.
Stored: Set the switch to C or D to use the application's configured (stored) parameters.
Clear IP: Set the switch to E to use the default IP parameters.
Disabled: Set the switch to F to disable communications.

Deriving IP Parameters from the MAC Address

Introduction

If no IP parameters are received from the application when the rotary switch *(see page 64)* is set to **Stored** or **Clear IP** positions, the module is configured at power-up with its default IP address. The default IP address for the module is derived from its hardware MAC address in accordance with a default IP address format.

Default IP Address Format

The default IP address format is `84.x.y.z`:

- `84`: a fixed value
- `x.y.z`: The last three fields in the default IP address correspond to the decimal equivalents of the last three bytes in the MAC address.

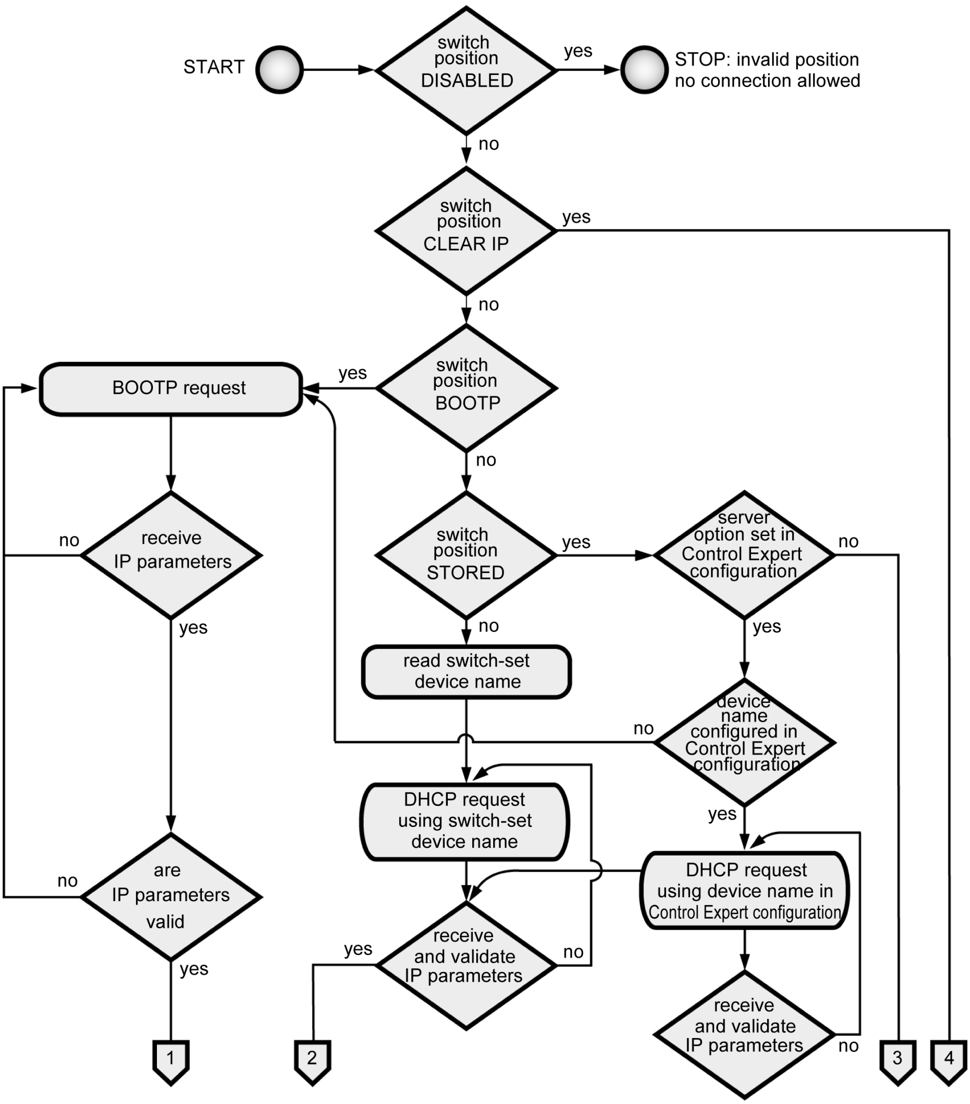
Example

For example a device with `00-00-53-12-01-C4` MAC address has the following default IP address: `84.18.1.196` (12 hex = 18 dec, 01 hex = 1 dec, C4 hex = 196 dec).

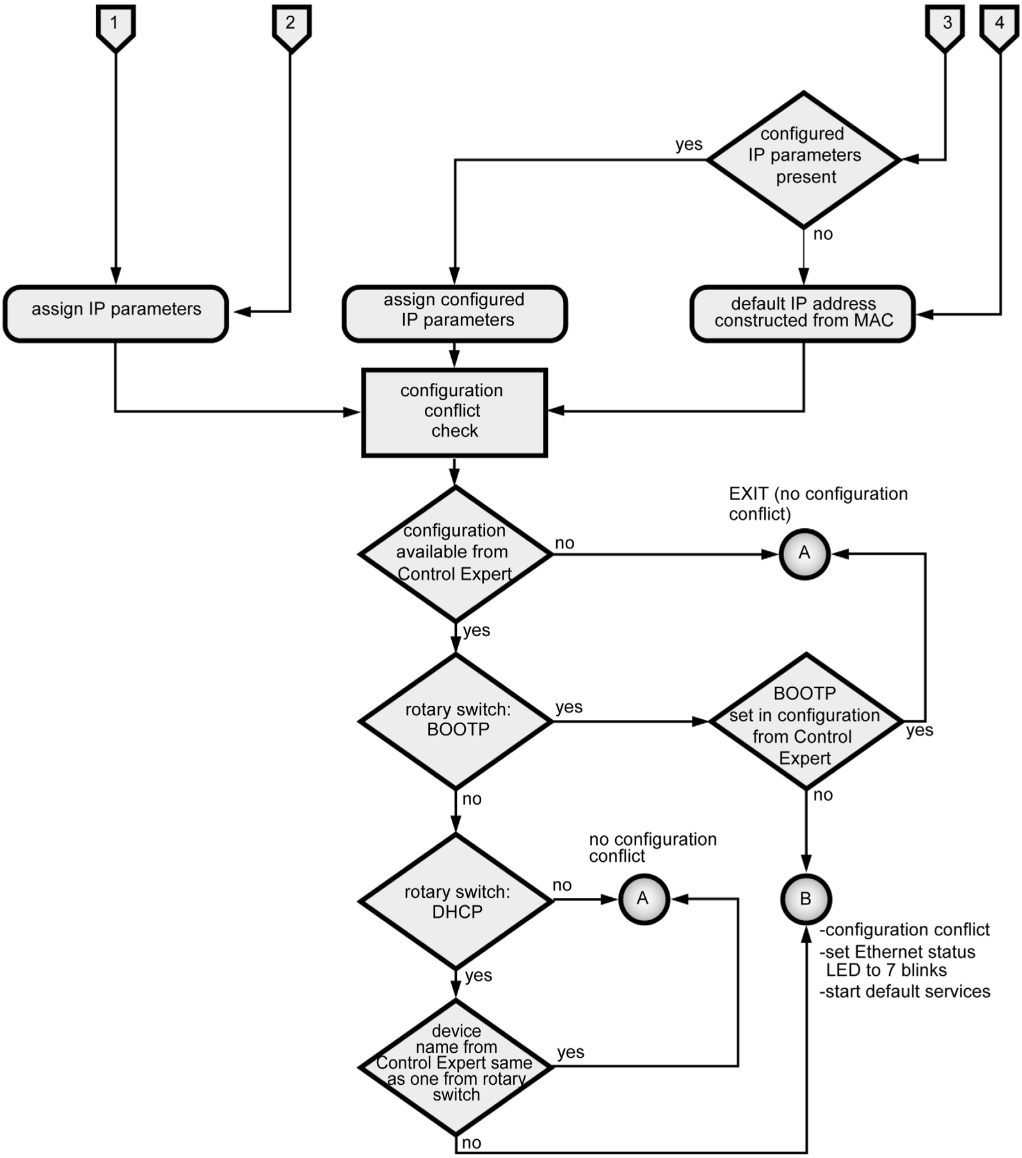
The IP Address Assignment Process

Determining the IP Address

The IP addressing methods for the Modicon M340 module are shown in the figure:



IP address assignment (continued):



Frame Format Priorities

The module supports communications in the Ethernet II and 802.3 frame formats. (Ethernet II is the default.) When communicating with a BOOTP or DHCP server, the module makes 4 requests based on the configured frame type. To avoid flooding the server with simultaneous requests, the transmission time of each request is base on the backoff algorithms defined by the respective BOOTP and DHCP RFCs.

NOTE: In addition, modules that communicate with DHCP servers make requests in this order:

- first: 4 requests in the DHCP format defined by RFC 2132
- second: 4 requests in the DHCP format defined by RFC 1533

Ethernet Port Status

Port Status

The status of the Ethernet port depends on the IP address configuration in the application and the setting on the rotary switches *(see page 64)*:

<table>
<tr><th rowspan="2">Application</th><th colspan="5">Switch Setting</th></tr>
<tr><th>DHCP (device name from rotary switches)</th><th>BOOTP</th><th>STORED</th><th>CLEAR IP</th><th>DISABLED</th></tr>
<tr><td>None</td><td>DEFAULT: Get IP address. (1)</td><td>DEFAULT: Get IP address. (1)</td><td>DEFAULT: Use default IP address. (1)</td><td rowspan="4">DEFAULT: This switch setting implements device’s default IP parameters regardless of the application type.</td><td rowspan="4">LED off. This switch setting stops communications between the module and the network, so the application type is inconsequential.</td></tr>
<tr><td>Configured</td><td>MISMATCH: Get IP address. (1)</td><td>MISMATCH: Get IP address. (1)</td><td>NO ERROR: Get IP address from application. (2)</td></tr>
<tr><td>From a server without device name (3)</td><td>NO ERROR: Get IP address. (2)</td><td>NO ERROR: Get IP address. (2)</td><td>NO ERROR: Get IP address with BOOTP. (2)</td></tr>
<tr><td>From a server with device name</td><td>NO ERROR: Get IP address. Device name derived from switches (2) (5)</td><td>MISMATCH: Get IP address. (1)</td><td>NO ERROR: Get IP address with DHCP (device name from application). (2)(4)</td></tr>
<tr><td colspan="6">Note 1: start default services only
Note 2: start application-configured services
Note 3: Leaving the From a server/Device Name field empty on the IP Configuration tab (see page 140) facilitates communications that are compatible with any rotary switch setting.
Note 4: To use the device name from the IP Configuration tab (see page 140) in the Control Expert application, the lower rotary switch much be set to one of its STORED positions.
Note 5: When the server returns a name that matches the one you configure on the IP Configuration tab (see page 140), communications are OK. When there is a mismatch, there is a communications error.
Note 6: The M340 Ethernet modules will not receive an IP address from a BOOTP/DHCP server on application download if the IP configuration has not changed.</td></tr>
</table>

Elsewhere in this guide is a detailed discussion about deriving the default IP address *(see page 66)*.

Device Name

⚠ CAUTION

UNINTENDED EQUIPMENT OPERATION

The communications port can enter the idle state when the position of the rotary switches does not match the port network configuration in the application. For example, the switches may be set to BOOTP or DHCP while the application calls for a fixed IP address.

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

The device name generated by the rotary switch setting follows this format:
- BMX P34 2020 CPU: BMX_2020_*xxy*
- BMX P34 2030 CPU: BMX_2030_*xxy*
- BMX P34 20302 CPU: BMX_20302_*xxy*
- BMX NOE 0100 module: BMX_0100_*xxy*
- BMX NOE 0110 module: BMX_0110_*xxy*

(Note that *xx* is the value of the upper rotary switch and *y* is the value of the lower rotary switch.)

NOTE: For information about how the module prioritizes IP addressing options, refer to the discussion on IP address assignment *(see page 61)*.

Chapter 6
Multi-Module Communication

Multi-Module Communication

At a Glance

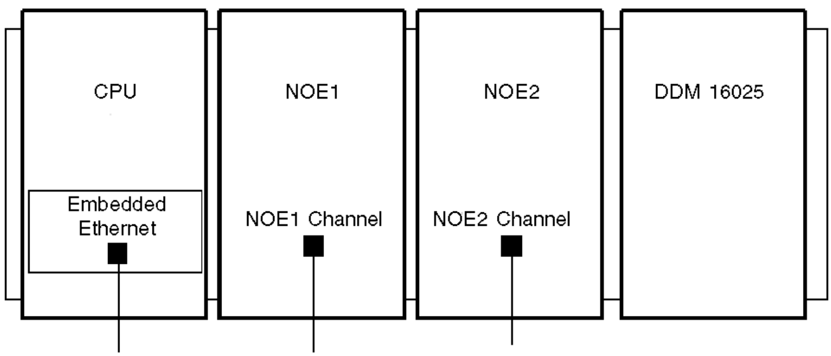
This topic explains communication capabilities across modules and devices in a PLC configuration. Different modules and devices are available providing access to USB, Modbus and/or Ethernet communication:
- CPU devices with embedded USB, Modbus and/or Ethernet ports
- NOE modules for Ethernet communication

Overview

A maximum of two NOE modules can be installed on a PLC configuration.

Therefore, the maximum number of configured Ethernet links is three:
- one link via the CPU with embedded Ethernet port
- two links via the NOE modules (more are not allowed, even if the CPU configured has no Ethernet port)

The following illustration shows a possible configuration:



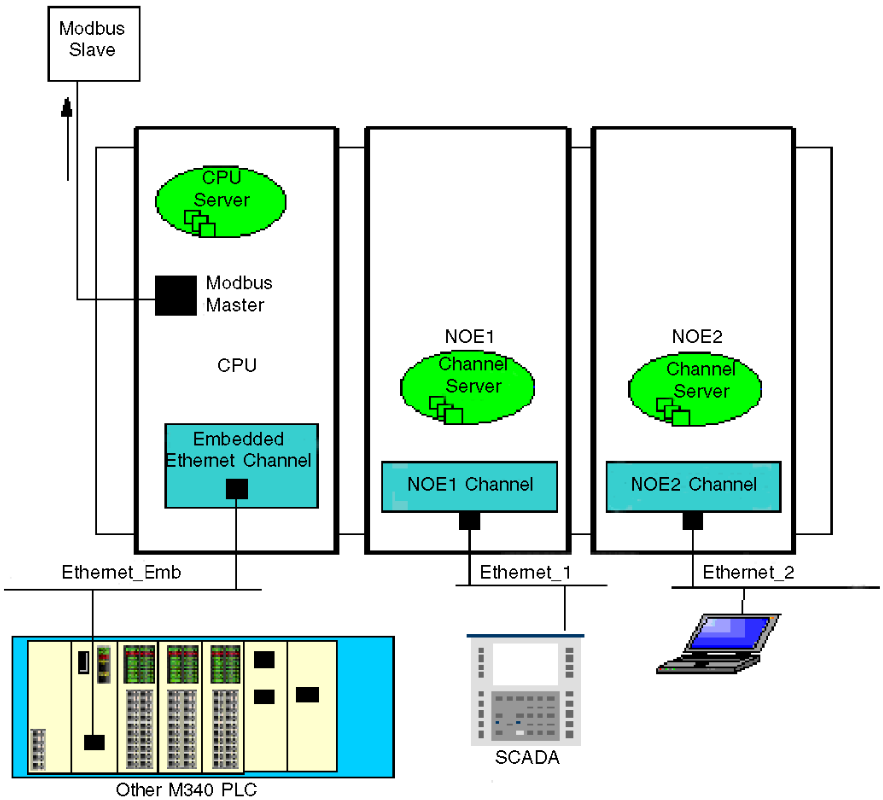
This example describes a rack with four modules:
- the CPU at rack module address [0.0]
- the NOE1 at rack module address [0.1]

- the NOE2 at rack module address [0.2]
- an I/O module or function module (in this example, we show a DDM 16025 discrete module at rack module address [0.3]).

NOTE: To define the rack/slot combination [x.x]:
- first digit = rack number
- second digit = slot number

Communication Overview and Definitions

The following illustration presents the overall installation:



In this example, there are:
- two NOE modules that are connected to two different Ethernet networks
- a CPU that is connected to a third Ethernet network and to a Modbus slave

The following table defines the terms that are used to describe communication methods using multiple communication modules:

Term	Definition
Channel	Represents the communication channel itself, able to manage the dispatching of the messaging flow.
Channel Server	A server dedicated to the management of the communication channel itself. For example, it provides statistical information on an Ethernet link. Control Expert accesses this server to display information related to an NOE module, for example.
CPU Server	This server answers all requests sent to the CPU, whatever the communication media used to access it.

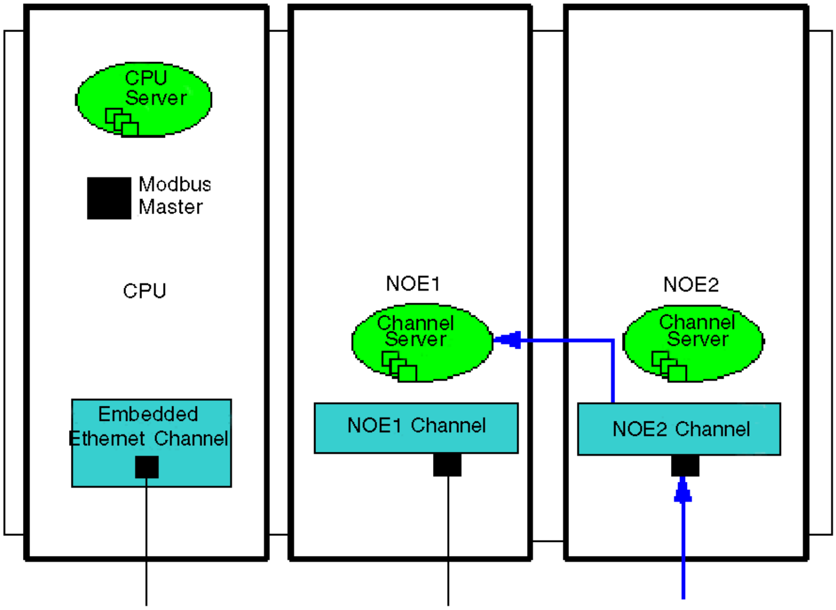
Routing Table

The following table describes the communication capabilities across the different modules:

Destination →	NOE1 Channel Server	NOE1 Channel	NOE2 Channel Server	NOE2 Channel	CPU Ethernet Channel	CPU Modbus Master
Source ↓						
NOE1 Channel	supported	not supported	supported	not supported	not supported	supported
NOE2 Channel	supported	not supported	supported	not supported	not supported	supported
CPU Ethernet	supported	not supported	supported	not supported	not supported	supported
CPU USB	supported	supported	supported	supported	supported	supported
CPU Modbus Slave	supported	supported	supported	supported	supported	not supported

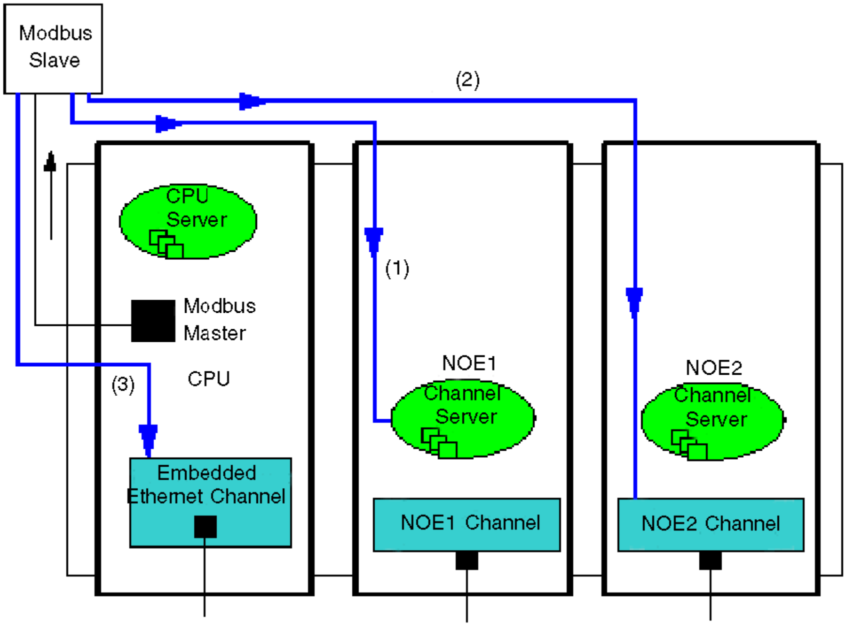
Example 1: NOE2 Channel to NOE1 Channel Server

The following illustration describes the connection from an NOE2 channel (source) to an NOE1 channel server (destination). The same methodology can be implemented from an NOE1 channel to an NOE2 channel server:



Example 2: CPU Modbus Slave to NOE Channel, NOE Server Channel, CPU Ethernet Channel

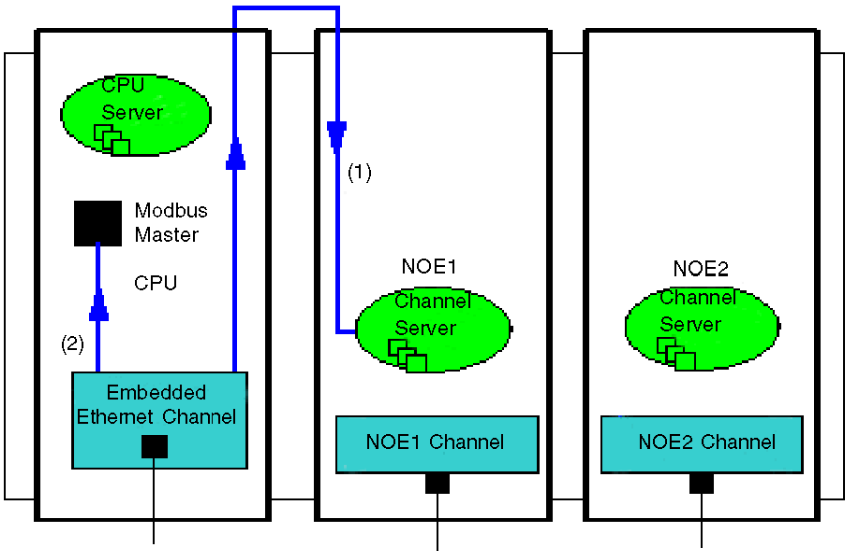
This example describes the connection from a CPU Modbus slave to:
- an NOE1 channel server (1)
- an NOE2 channel (2)
- a CPU Ethernet channel (3)



Example 3: CPU Ethernet to NOE Channel Server, CPU Modbus Master

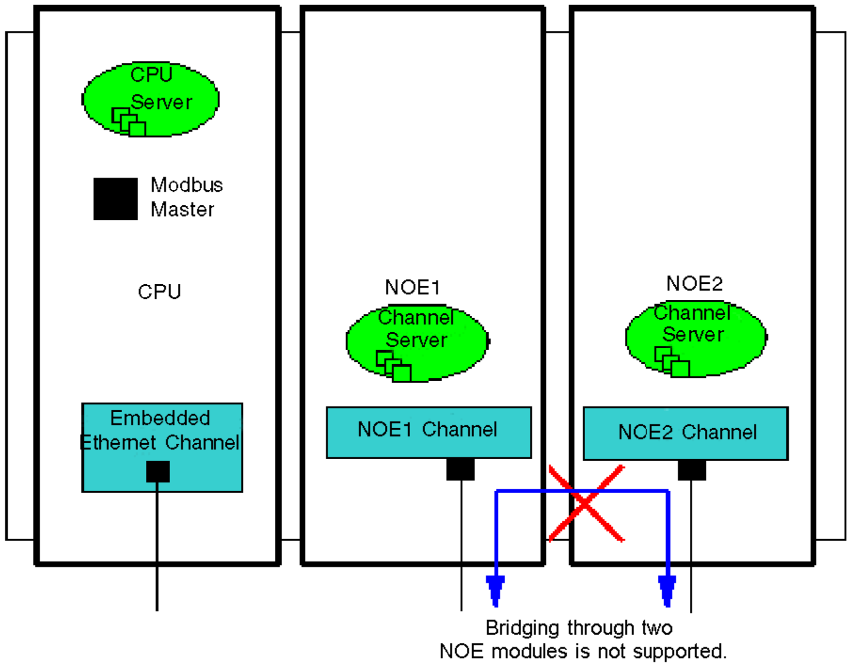
This example describes the connection from a CPU Ethernet to:

- an NOE1 channel server (1)
- a CPU Modbus Master (2)



Example 4: Not Supported - NOE1 Channel to NOE2 Channel Not Supported

This example illustrates a configuration that is not supported, with communication from an NOE1 channel to an NOE2 channel:



Chapter 7
Description of Ethernet Communications Services

About this Chapter

This chapter briefly describes the services that are available for Modicon M340 modules that support Ethernet communications (BMX NOE 01x0 and BMX P34 20x0 CPUs).

NOTE: Not all services are available for all modules at all times. Elsewhere in this guide is a list of the services that are available for each Ethernet communications module *(see page 42)*.

What Is in This Chapter?

This chapter contains the following sections:

Section	Topic	Page
7.1	I/O Scanning Service	82
7.2	Address Assignment through DHCP and BOOTP	86
7.3	SNMP	91
7.4	Global Data	96
7.5	Fast Device Replacement	99
7.6	Bandwidth Monitoring	100
7.7	TCP/IP Messaging	104
7.8	Time Synchronization Service	108
7.9	Electronic Mail Notification Service	115

Section 7.1
I/O Scanning Service

About this Section

This section presents some functions, characteristics, and configuration options for the I/O scanning service.

What Is in This Section?

This section contains the following topics:

Topic	Page
I/O Scanning Service	83
Read and Write Zones	85

I/O Scanning Service

Introduction

The I/O Scanner is used to periodically:

- read remote inputs
- write remote outputs
- read/write remote outputs

The I/O Scanner is used to periodically read or write remote inputs and outputs on the Ethernet network without specific programming.

Configure the I/O Scanner with Control Expert *(see page 148)*.

NOTE: I/O scanning is not supported by the Ethernet port of BMX P34 20x0 CPUs.

Characteristics

This service comprises the following essential elements:

- **read zone**: the values of remote inputs
- **write zone**: the values of remote outputs
- **scanning periods**: independent of the PLC cycle and specific to each remote device

During operation, the module:

- manages the TCP/IP connections with each remote device
- scans inputs and copies their values into the configured %MW word zone
- scans outputs and copies their values from the configured %MW word zone
- reports status words so that correct operation of the service can be monitored by the PLC application
- applies the preconfigured fallback values (in the event of a communication problem)
- enables or disables each entry in the I/O scanner table in accordance with its application

Recommended Use

Scanning is performed only when the PLC is in Run mode.

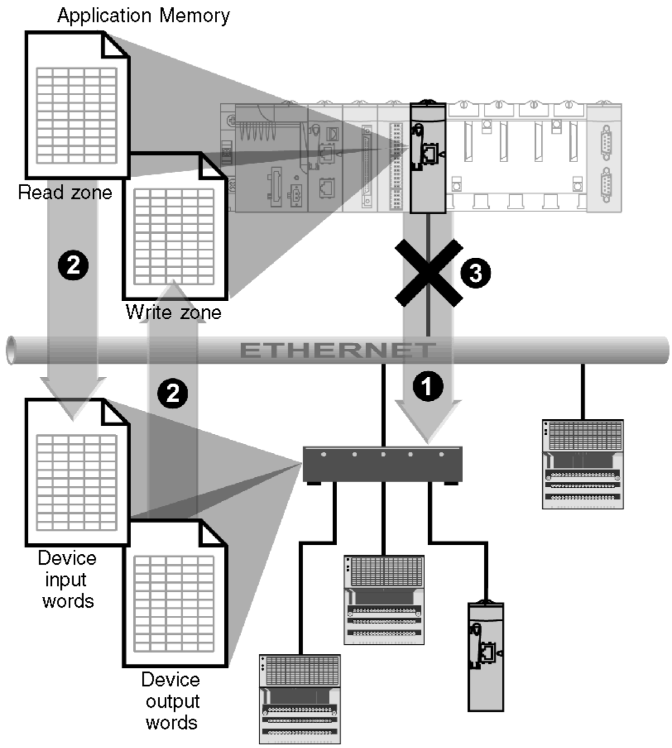
This service operates with all devices that support Modbus TCP/IP server mode.

The exchange mechanism, which is transparent to the user, is executed with these request types (as long as the remote devices support them):

- read
- write
- read and write

I/O Scanner Operation

This figure illustrates the scanning of remote inputs and outputs:



1 As soon as the PLC switches to Run mode, the module successfully opens a connection for each scanned device. (One connection is made for each line entered in the table of scanned devices.)
2 Then the module periodically reads input words and reads/writes output words for each device.
3 If the PLC switches to Stop mode, the connections to all devices are closed.

Summary of Functionality

The I/O scanning service:

- manages the connection with each remote device (one connection per remote device)
- scans the inputs/outputs of the device using Modbus read/write requests on the TCP/IP server mode
- updates the read and write zones in the application memory
- refreshes the status bits for each remote device

Each I/O Scanner device can be enabled/disabled *(see page 183)*.

NOTE: These status bits show whether the module's input/output words have been refreshed.

Read and Write Zones

The Zones

Within the application memory, the I/O scanning service defines:
- %MW word zone: reserved for reading inputs
- %MW word zone: reserved for writing outputs
- refresh periods: independent of the PLC scan

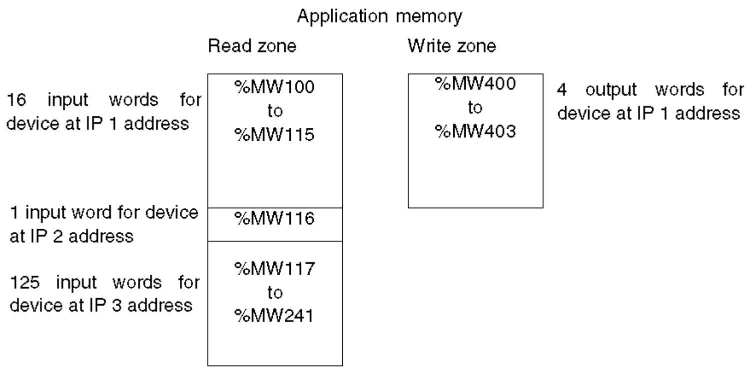
The read and write zones associated with the Ethernet module are tables of internal words (%MW) that contiguously group all input and output word values for connected remote devices. Remote input and output devices are supplied with:
- input words: used to send back the values of the inputs to the module
- output words: used to assign the value of the outputs to the remote device

NOTE: Refer to the documentation for each device for the number and details of input and output words to be managed.

Example

In this example, the Ethernet module scans three devices:
- a Momentum module at address IP1 (type 170 AA1 140 00: 16 analogue inputs; this module has 16 input words and 4 output words)
- a Momentum module at address IP2 (type 170 AA1 340 00: 16 discrete inputs; this module has one input word)
- a Premium PLC with 125 input words at address IP3

The read zone begins at %MW100 and the write zone at %MW400.



NOTE: The fields dedicated to remote devices must not have any overlap. Equally, the read and write zones must not have any overlap.

NOTE: Not all devices can be write-scanned by multiple modules. Check the remote device's documentation to see if it can be accessed by the I/O Scanner.

Section 7.2
Address Assignment through DHCP and BOOTP

About this Section

This section explains how the BMX NOE 01x0 modules or the Ethernet ports on the BMX P34 20x0 CPUs get their IP addresses through DHCP or BOOTP, network protocols that can dynamically assign IP addresses to network devices.

What Is in This Section?

This section contains the following topics:

Topic	Page
Address Assignment through DHCP/BOOTP	87
Modicon M340 DHCP/BOOTP Example	88
BMX NOE 01x0 as a DHCP Server	89

Address Assignment through DHCP/BOOTP

Introduction

This topic discusses the manner in which a BMX NOE 01x0 modules or the Ethernet port of a BMX P34 20x0 CPUs get an IP address from a DHCP/BOOTP server.

NOTE: Before attempting to get an IP address from a DHCP/BOOTP server, obtain the appropriate network address and subnetwork mask from your system administrator.

NOTE: The BMX NOE 01x0 module will not get an IP address from a DHCP/BOOTP server while downloading an application if the IP configuration has not changed.

Address Request and Reply

This table explains how a DHCP/BOOTP client (like an Ethernet module or CPUs with Ethernet ports) gets an IP address:

Step	Action
1	The module supports communications in the Ethernet II and 802.3 frame formats. (Ethernet II is the default.) When communicating with a DHCP or BOOTP server, the module makes 4 requests based on the configured frame type. To avoid flooding the server with simultaneous requests, the transmission time of each request is based on the backoff algorithms defined by the respective DHCP and BOOTP RFCs. **Note**: In addition, modules that communicate with DHCP servers make requests in this order: ● first: 4 requests in the DHCP format defined by RFC 2132 ● second: 4 requests in the DHCP format defined by RFC 1533
2	The remote device acting as the DHCP/BOOTP address server responds to this request and assigns to the client module: ● an IP address ● the gateway IP address ● the corresponding subnetwork mask
Note: The display of this information from a browser connected on the HTML pages of the rack display on the web server is not constantly updated. Therefore, connect a Control Expert application on the correct PLC to obtain the actual values.	

In the absence of an address server or if there is no server response, the client module can not get a stable IP address. Therefore, its services are not available. The client continues to issue periodic BOOTP/DHCP requests until it receives a response from the server.

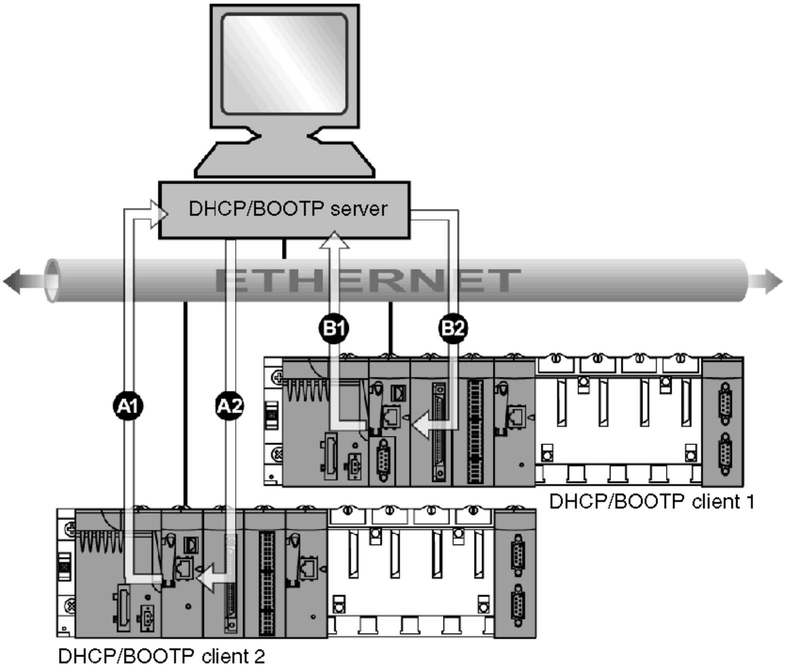
NOTE: Setting the rotary switch position to Clear IP is one way to force the client module to use its default IP address to start basic services, including Modbus TCP messaging.

These concepts are graphically represented in the IP address assignment process flowchart *(see page 67)*.

Modicon M340 DHCP/BOOTP Example

Example

This figure shows the routing of requests during start-up of a device on a network:



A1 request from BMX NOE 0100 at device name 2
A2 response from DHCP/BOOTP server
B1 request from BMX P34 2030 at MAC1 address
B2 response from DHCP/BOOTP server

The DHCP server in the above figure contains the following table:

MAC Address	IP Address
MAC1 Address	IP1 Address
device name 2	IP2 Address
MAC3 Address	IP3 Address
device name 4	IP4 Address
...	...

BMX NOE 01x0 as a DHCP Server

About DHCP

DHCP (Dynamic Host Configuration Protocol) manages network parameters for network devices. Individual devices can get network IP configurations from a DHCP server. To do so, the device includes its device name in the request to the server.

The BMX NOE 01x0 modules (with memory card *(see page 50)* required) can be configured as a DHCP server only after the station power is cycled. It can also have its address configured by the user or dynamically allocated from an address server when the device is configured as a DHCP/BOOTP client.

NOTE:

- Before attempting to get an IP address from a DHCP server, obtain the appropriate network address and subnet mask from your system administrator.
- Because DHCP is built on BOOTP functionality *(see page 87)*, a DHCP server can respond to BOOTP protocol requests.

Server Response

⚠ CAUTION
UNINTENDED EQUIPMENT OPERATION
Do not configure the PLC to automatically start in RUN mode when a module acts as a server for starting scanned devices. Doing so risks that some devices (such as Momentum 170s) will not obtain IP addresses when the DHCP server restarts.
Failure to follow these instructions can result in injury or equipment damage.

The server responds to client requests and sends them their IP address configurations.

To do this, the server device has a table, which groups:

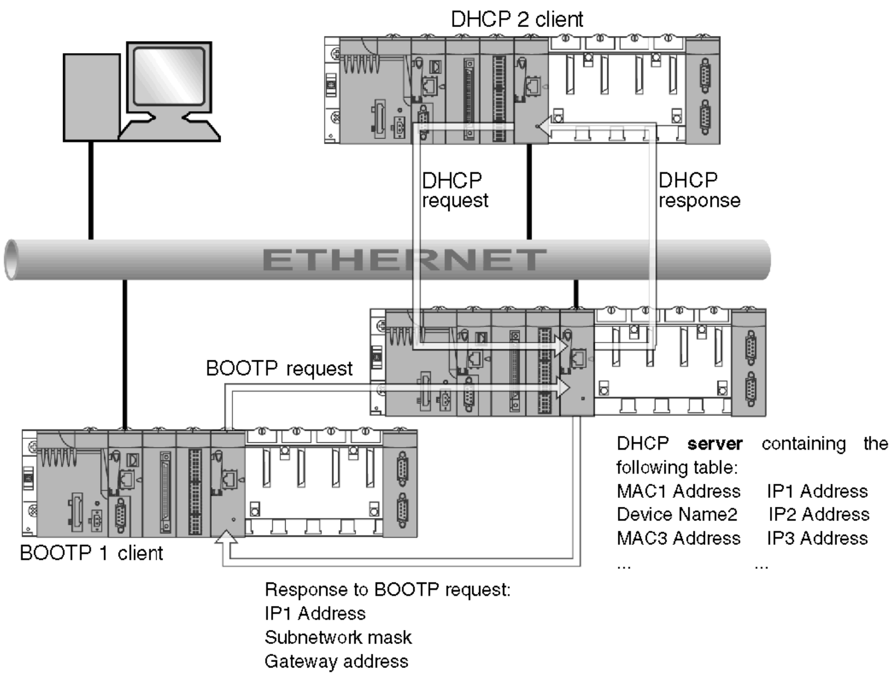
- correspondence between the MAC addresses or the Names (device names) of the client stations and the IP addresses
- the Netmask and the Gateway
- the names and access paths to the parameter files

NOTE: The server on the BMX NOE 01x0 supplies network configuration data only to client stations connected to the LAN. Talk to your system administrator about LAN connections on your network.

You may experience a time delay for DHCP server responses if your DHCP server module is busy with other higher priority services (such as IO Scanning, Global Data, or Modbus Messaging):
- For general DHCP server use (such as IP address configuration), the maximum server performance is achieved through the use of a dedicated DHCP server.
- If you plan to use the Fast Device Replacement (FDR) service, we recommend that you configure a lightly utilized Schneider Ethernet module with an FDR server for maximum performance.

Example of DHCP Server

This figure shows request routing during the response to a start-up request from the client:



NOTE: The subnetwork mask and the address of the gateway are provided by the DHCP server.

Section 7.3
SNMP

About this Section

This section describes the Simple Network Management Protocol (SNMP).

What Is in This Section?

This section contains the following topics:

Topic	Page
SNMP and Schneider Private MIB Overview	92
SNMP Communication	93
SNMP Operations Example	95

SNMP and Schneider Private MIB Overview

Introduction

An SNMP agent runs on:
- Ethernet communication modules
- CPUs with embedded Ethernet communications ports

Network management systems use SNMP to monitor and control Ethernet architecture components for the rapid network diagnosis.

Network management systems allows a network manager to:
- monitor and control network components
- isolate troubles and find their causes
- query devices, such as host computer(s), routers, switches, and bridges, to determine their status
- obtain statistics about the networks to which they are attached

NOTE: Network management systems are available from a variety of vendors. Schneider Electric provides an SNMP-based diagnostics tool called ConneXview.

Simple Network Management Protocol

Ethernet communication modules support SNMP, the standard protocol for managing local area networks (LANs). SNMP defines exactly how a manager communicates with an agent. SNMP defines the format of:
- requests that a manager sends to an agent
- replies that the agent returns to the manager

The MIB

The set of objects that SNMP can access is known as a Management Information Base (MIB). Ethernet monitoring and management tools use standard SNMP to access configuration and management objects included in the device's MIB, providing that:
- objects that SNMP can access are defined and given unique names
- manager and agent programs agree on the names and meanings of fetch and store operations

Transparent Ready products support two SNMP network management levels:
- **Standard MIB II**: This first level of network management can be accessed via this interface. It lets the manager identify the devices that create the architecture and retrieve general information on the configuration and operation of the Ethernet TCP/IP interface.
- **MIB Transparent Ready interface**: Schneider has obtained a private MIB, groupeschneider (3833) *(see page 359)*. This MIB includes a set of data that enables the network management system to supervise the Ethernet services. The Transparent Ready private MIB can be downloaded from the Web server on any Transparent Ready module in a PLC.

SNMP Communication

Overview

SNMP defines network management solutions in terms of network protocols and the exchange of supervised data.

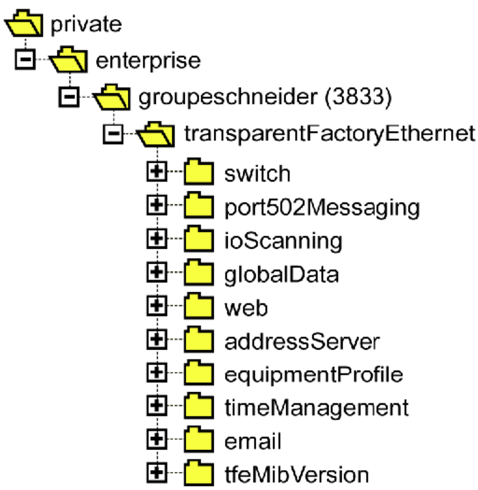
The SNMP structure relies on the following essential elements:

- **Manager:** The manager allows entire or partial network supervision.
- **Agents:** Each supervised device has one or more software modules named "Agent" that are used by the SNMP protocol.
- **MIB:** The Management Information Base is a database or collection of objects.

The SNMP agent is implemented on the Modicon M340 modules and on the Ethernet port of the processors. This allows a manager to access MIB-II standardized objects from the Modicon M340 agent through the SNMP protocol. The MIB-II allows management of TCP/IP communication layers.

On the modules that support Ethernet communications, it is possible to access objects from the MIB Transparent Factory, which provides specific information about Global Data, I/O Scanning, and Messaging.

The following figure shows the tree structure of the TFE Ethernet MIB:



The source file of the TFE private MIB *(see page 359)* is available on modules and CPUs that support Ethernet communications. The MIB can be uploaded from the module's web page by selecting Upload MIB File *(see page 280)*. This file may be compiled by the major SNMP managers on the market.

The SNMP Protocol

The SNMP protocol defines 5 types of messages between the agent and the manager. These messages are encapsulated in UDP datagrams.

Messages from the manager to an agent:
- `Get_Request`: message used to obtain the value of one or more variables
- `Get_Next_Request`: obtains the value of the next variables
- `Set_Request` : sets the value of a variable

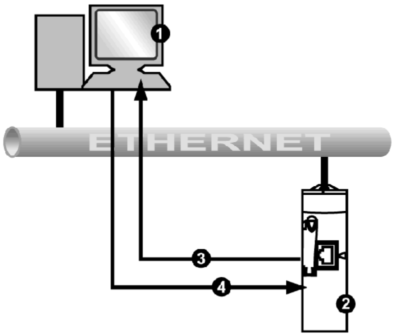
Messages from an agent to the manager:

- `Get_Response`: allows the agent to re-send the value of the requested variable.
- `Trap`: allows asynchronous event signaling by the agent.

SNMP Operations Example

Modicon M340 Example

The SNMP manager transmits read or write requests (`Set_Request`, `Get_Request`, `Get_Next_Request`, etc.) for objects defined in the MIB - II SNMP and the SNMP agent of the Modicon M340 module responds.



1 SNMP manager
2 SNMP agent (Modicon M340)
3 Get_Response trap
4 Set_Request, Get_Request, Get_Next_Request

The module's SNMP agent transmits events (traps) to the Manager. The managed traps systems are as follows:

- `Coldstart Trap`:
  - On the BMX NOE 01x0 modules and on the Ethernet port of the BMX P34 20x0 CPUs, the event is transmitted following a module supply Reset, a processor Reset, or the downloading of an application to the PLC.
- `Authentication Failure Trap`: event transmitted following an authentication problem. The **Community Name** field in the received message is different to the one configured on the module. This trap can be enabled during module configuration.

Section 7.4
Global Data

Global Data

Introduction

Global Data, which is supported by the BMX NOE 01x0 modules, provides an automatic data exchange for the coordination of PLC applications.

NOTE: Global Data is not supported by the Ethernet port of BMX P34 20x0 CPUs.

Global Data

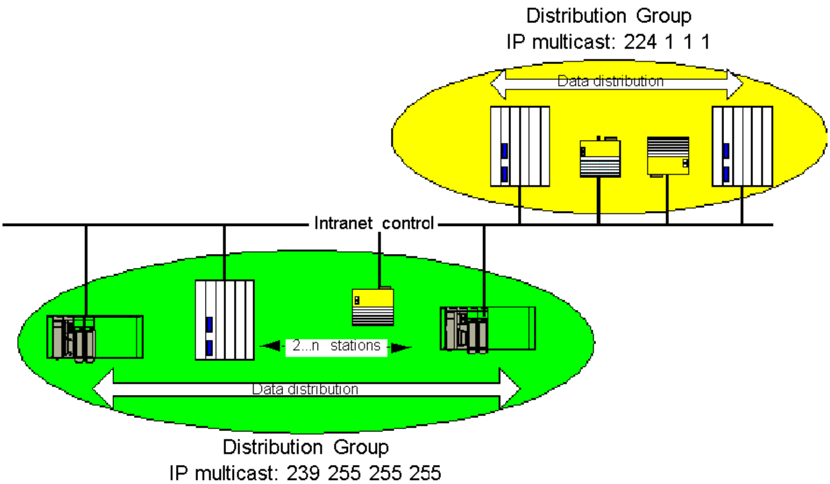
Key features of Global Data:
- Data exchanges are based on a standard producer/consumer protocol, giving optimal performance while maintaining a minimum network load.
- The Global Data service provides real-time exchanges between stations in the same distribution group, which share variables used for PLC coordination.
- Global Data can synchronize remote applications or share a common database among several distributed applications.
- A maximum of 64 stations can participate in Global Data within the same distribution group.

Operation

An application variable is a group of contiguous words from a PLC. Communication module stations can publish or subscribe to application variables:
- **publish:** Communication modules can publish one 1024-byte local application variable for other communication modules in the time-based distribution group. The publication period can be configured from 10 ms to 15000 ms (in 10 ms increments).
- **subscribe:** A communication module can subscribe to between 1 and 64 application variables that are published by other modules in their distribution group, regardless of their location. The validity for each variable is controlled by health status bits, linked to a refresh timeout configurable between 50 ms and 15 s (15000 ms). Access to a variable element is not possible. The maximum size of the subscribed variables is 4 Kbytes.

Through Global Data configuration you can define:
- the number of valid published and subscribed variables
- the group with which these variables are associated for the communications module

Once the module is configured, exchanges between the communication modules sharing the same Distribution Group are automatically carried out when the PLC is in RUN mode.



A distribution group is a group of communication modules identified by the same multicast IP address. Multicast exchanges *(see page 353)* distribute Global Data. Several independent distribution groups can co-exist on the same subnetwork with different multicast addresses.

Important points:
- A publish/subscribe protocol on UDP/IP is used for data distribution.
- The content of the publication variable is synchronized during the out section of the MAST task.
- The content of the subscribed variables are copied to the PLC application memory during the in section of the MAST task.
- The PLC memory zones that receive the various subscription variables should not overlap. Do not use these variables for any other purpose.

Health Bits

A health bit (status bit) is associated with each application variable.

This bit indicates the validity of each subscription variable. It is 1 if the variable has been published and the subscriber has received it in the configured validity time. Otherwise, it is 0.

Multicast Filtering

Global Data uses multicast addressing. In accordance with the nature of broadcast messages, the multicast frames transmitted by a module are repeated on all switch ports, generating network congestion.

When the switches possess this function, multicast filtering allows multicast frame spreading to be limited to only the ports that require it.

The GMRP protocol establishes the list of ports involved in data traffic.

Operating Modes

The operating modes are as follows:
- Stopping the PLC stops Global Data exchanges.
- The use of the I/O forcing system bits (%S9, %SW8, %SW9) does not stop Global Data exchanges *(see EcoStruxure™ Control Expert, Operating Modes)*.

Limits

There are no theoretical limits for the number of stations that share a distribution group. There is a limit on the number of variables (64) that can be exchanged within the distribution group. For more information, refer to the table describing the parameters for the BMX NOE 01x0 modules *(see page 125)* later in this user manual.

NOTE: It is recommend to use fewer than 200 modules in any single distribution group.

Section 7.5
Fast Device Replacement

Replacing Devices

Introduction

This service provides automatic recovery of I/O module parameters or intelligent modules connected to a Transparent Factory Ethernet sub-segment. The recovery of this information is important during the replacement of a module with another module.

This service:
- supplies an IP address to a device from the name given to this station (device name)
- gives a remote station the capacity to store parameters and (when required) recover them

Operation

This service requires the use of the DHCP server *(see page 89)* and the FTP/TFTP server of the BMX NOE 01x0 modules. It can manage up to 64 remote stations.

Operating principles:
- initially:
  - (1) The valid device obtains an IP address from the name that has been given to it (device name).
  - (2) The device gives its configuration parameters to the server.
- secondly: This station breaks down. It is replaced with a non-configured device of the same type, which has the same name as the station it replaces:
  - (1) It then transmits a DHCP request to the server.
  - (2) It receives its configuration file, which was saved first on the server.
  - (3) The device restarts automatically.

NOTE: The Device Name is limited to 16 ASCII characters.

NOTE:
Schneider Electric supports FTP communication between BMX NOE 01x0 modules and the following tools:
- Schneider Electric firmware upgrade utilities
- Microsoft Windows command-line FTP client
- FTP clients in device Web pages
- FTP client access by approved devices that use the Schneider Electric FDR server service

Section 7.6
Bandwidth Monitoring

About this Section

This section describes the bandwidth monitoring function of the BMX NOE 01x0 communication modules and the Ethernet channel of the BMX P34 20x0 CPUs.

Bandwidth monitoring tracks the module allocation for each of these services:
- messaging *(see page 104)*
- I/O scanning *(see page 82)*
- Global Data *(see page 96)*

With the workload data that it retrieves, the bandwidth monitoring service reports:
- the available resources for the module
- the current working capacity of the module

What Is in This Section?

This section contains the following topics:

Topic	Page
Bandwidth Monitoring	101
Static and Dynamic Bandwidth Monitoring	102

Bandwidth Monitoring

Introduction

Transparent Ready offers two types of services:

- real time: High-performance and predictable behavior are expected from services such as:
  - periodic services: I/O Scanner and Global Data
  - non-periodic services: Messaging on port 502 (Modbus)
- other: for Web and network management

After you configure the I/O Scanner, it is possible to estimate the load percentage of the module for this service.

The Bandwidth Monitoring function carries out this estimate during configuration.

The actual distribution of service loads is shown on the module’s Control Expert diagnostics screen and on the diagnostics services bandwidth monitoring Web page.

Operation

In order to estimate this load, Control Expert asks you to supply two pieces of information during configuration:

- an estimate of the number of messaging transactions per second
- an estimate of the number of Global Data subscriptions received per second

Using this information, the software is able to display the load percentage of the module for:

- I/O Scanner *(see page 82)*
- Global Data *(see page 82)*
- messaging services *(see page 104)*
- remaining services *(see page 81)*

NOTE: During the configuration phase, these pieces of information are only estimates. The actual distribution is displayed in online mode.

Static and Dynamic Bandwidth Monitoring

Introduction

For the BMX NOE 01x0 and BMX P34 20x0 CPUs, the static bandwidth monitoring feature shows the approximate load as a percentage of the maximum available. Dynamic monitoring tracks certain counters to display the value of the load at the time of monitoring.

Static Bandwidth Monitoring

Configuration choices have an impact on the module load. The static bandwidth monitoring feature reports the approximate load as a percentage of the maximum available. For example, each I/O Scanner entry and subscription variable adds to the load, as do Modbus messaging function blocks. However, it is not possible to predict how often the Modbus messaging will run.

Estimating Traffic

To estimate network traffic, enter:

- the number of Global Data publications within the group (per second)
- the number of incoming Modbus/TCP requests from the network (per second)
- the number of outgoing Modbus/TCP requests per second (from the EFBs in the application)

Network Environment Choice

For the BMX P34 20x0 CPUs, choose among three network environments for network polling:

- **isolated**: lessens the impact of Ethernet communication on the scan by limiting the number of polled Ethernet messages to 700 per second
- **mastered**: increases the maximum number of polled messages to 1400 per second
- **open**: increases the maximum number of polled messages to 2100 per second

Be aware that an increase in the number of polled messages increases the load on the CPU module, which may be evident in a slower scan time.

Dynamic Bandwidth Monitoring

For the BMX NOE 01x0 and BMX P34 20x0 CPUs, the Bandwidth Monitoring service allows dynamic monitoring by tracking the total number of:

- packets processed per second
- *broadcast* packets processed per second
- *multicast* packets processed per second
- *unicast* packets processed per second
- *useless* packets (dropped by software) processed per second

As part of Ethernet Statistics, the service tracks the total number of dropped packets:

- total packets dropped by the hardware, due to buffer

Elsewhere in this guide is an illustration of the bandwidth debugging window *(see page 220)*.

The Bandwidth Monitoring service tracks these counters to allow for dynamic monitoring:

Packets	Message Type	Comment
Total	All packets	
Broadcast	All ARP, etc.	
Multicast	All non-filtered multicast	Includes non-subscribed Global Data
Unicast	Modbus 502, FTP, HTTP, SNMP	
Useless	Software dropped	
Dropped	Lost	Dropped by hardware

Section 7.7
TCP/IP Messaging

About this Section

This section describes the functions and characteristics of the TCP/IP profile.

NOTE: Elsewhere in this guide is reference material for the detailed technical characteristics of TCP/IP communication *(see page 333)*.

What Is in This Section?

This section contains the following topics:

Topic	Page
Data Exchange	105
Transparent Device Access	106

Data Exchange

Exchanges

Data exchanges take place in one of two modes:

- **server mode**: All Modbus-over-TCP requests from the PLC are supported by the Ethernet module.
- **client mode**: This type of exchange enables Modbus-over-TCP requests to be sent using the functions:
  - READ_VAR *(see page 339)*
  - WRITE_VAR *(see page 339)*
  - DATA_EXCH (see the *Communication Block Library (see EcoStruxure™ Control Expert, Communication, Block Library)* for Control Expert)

Transparent Device Access

About TDA

Transparent Device Access (TDA) functionality means that clients that run Control Expert (and that are connected to a USB, Ethernet, or Modbus terminal port of a communications module) can access or download applications to devices on distributed control networks. The reverse, however, is not true. In other words, a Control Expert PC connected to the CPU's Modbus port can access devices on other core networks, but those remote devices can not access other devices on different networks through the Modicon M340 station.

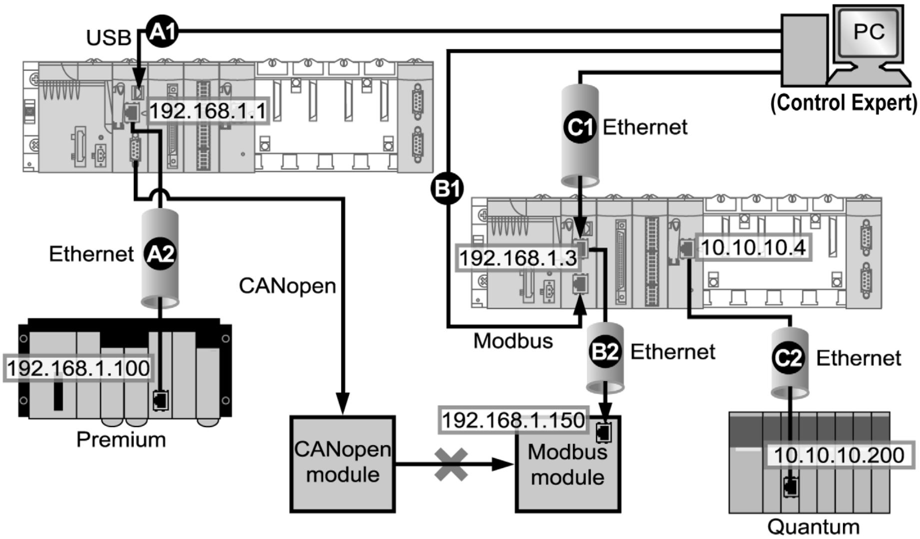
Scenarios

The following communication scenarios are possible:
- Ethernet port on the BMX P34 20x0x CPUs to the BMX NOE 01x0 Ethernet modules
- BMX NOE 01x0 Ethernet modules to the Ethernet port on the BMX P34 20x0x CPUs
- USB port on the BMX P34 20x0x CPUs to the Ethernet port on the BMX P34 20x0x CPUs
- USB port on the BMX P34 20x0x CPUs to the BMX NOE 01x0 Ethernet modules
- Modbus port on the BMX P34 2020 CPU to the Ethernet port on the BMX P34 20x0x CPUs
- Modbus port on the BMX P34 2020 CPU to the BMX NOE 01x0 Ethernet modules

Examples

The following figure shows three different scenarios for downloading information from a PC running Control Expert to an end device:
- **scenario A**: TDA through the USB port of a BMX P34 2030 CPU
- **scenario B**: TDA through the Modbus port of a BMX P34 2020 CPU
- **scenario C**: TDA through the Ethernet port of a BMX NOE 01x0 module

This figure shows the ports (identified by IP address, where applicable):



Legend:

A1: Data is sent over USB source port on a BMX P34 2030 CPU in slot position 0 of a Modicon M340 station.

A2: The program is downloaded to an Ethernet module (`192.168.1.100`) in a Premium rack.

(For scenario A, the Control Expert address syntax is `SYS\\CPU Ethernet port rack.slot.channel{destination IP address}`. Therefore, the address in the figure is `SYS\\0.0.3{192.168.1.100}`.)

B1: Data is sent over Modbus source port on a BMX P34 2020 CPU (Modbus address: 2) at slot position 0 of a Modicon M340 station.

B2: The program is downloaded via Ethernet to a Modbus end device (**192.168.1.150**).

(For scenario B, the Control Expert address syntax is `CPU Modbus address\\CPU Ethernet port rack.slot.channel{destination IP address}`. Therefore, the address in the figure is `2\\0.0.3{192.168.1.150}`.)

C1: Data is sent over the Ethernet source port on a BMX P34 0100 CPU in slot position 0 of a Modicon M340 station.

C2: The BMX NOE 01•0 transmits over Ethernet to the Ethernet module (**10.10.10.200**) in a Quantum rack assembly in another network.

(For scenario C, the Control Expert address syntax is `source IP address\\source port rack.slot.channel{destination IP address}`. Therefore, the address in the figure is `192.168.1.3\\0.3.0{10.10.10.200}`.)

Section 7.8
Time Synchronization Service

About this Section

This section describes the time synchronization service, which establishes an accurate local clock by referencing a network time protocol (NTP) server via the network transfer protocol (NTP).

What Is in This Section?

This section contains the following topics:

Topic	Page
Introducing the Time Synchronization Service	109
Using the Time Synchronization Service	112
Using the R_NTPC Block for Time Synchronization	113

Introducing the Time Synchronization Service

Overview

The time synchronization service establishes accuracy among computer clocks on an Ethernet system. For example, the time of one client may be synchronized either with another server, a referenced time source such as a radio or satellite receiver, or a GPS time server.

Typical time service configurations use redundant servers and diverse network paths to establish high accuracy and reliability. Time service accuracy can be within a millisecond on LANs and within tens of milliseconds on WANs.

Use the time synchronization service for:
- event recording (for example, tracking a sequence of events)
- event synchronization (for example, triggering simultaneous events)
- alarm and I/O synchronization (for example, time stamping alarms)

Features of the Service

The time synchronization service offers:
- periodic time corrections obtained from the reference standard, for example, the NTP server
- automatic switchover to a backup time server if a problem occurs with the normal server system
- local time zone configurable and customizable (including daylight saving time adjustments)

Controller projects use a function block to read the clock, a feature that allows events or variables in the project to be time stamped. Time stamping is accurate to:
- 5 ms typical
- 10 ms worst case

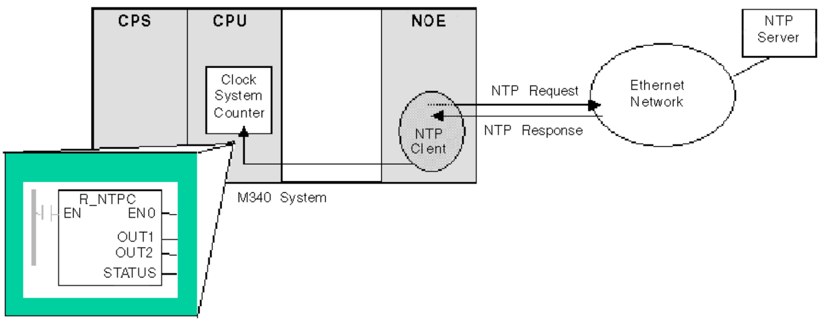
Time Synchronization and Time Stamps

The BMX NOE 01x0 Ethernet communications modules send a source time-synchronization signal to a CPU. The NOE's firmware includes an NTP client, which provides time synchronization. The synchronization process occurs as follows:

The NTP Client ...		Result
1	... requests a time synchronization signal from the NTP server over an Ethernet network.	The NTP server sends a signal.
2	... stores the time.	
3	... sends a message to the clock system counter in the CPU.	The CPU updates its internal clock. The CPU's clock is now typically within 5 ms of the NTP server, with a worst case of 10 ms. Before the next time synchronization signal, the CPU's clock is updated each ms by an internal timer.

Use the R_NTPC function block *(see page 113)* in either MAST, FAST, or Interrupt sections to read the clock from the PLC application.

All the CPUs on an Ethernet network should be synchronized with the same NTP server.



Time Synchronization Terms

Term	Description of Service
local clock offset	Accurate local time adjustments are made via a local clock offset. The local clock offset is calculated as: `((T2 - T1) + (T3 - T4))2` where: • T1 = time when NTP request is transmitted from the module • T2 = time when NTP server receives the request (provided by the module in response) • T3 = time when the NTP server transmits the response (provided to the module in the response) • T4 = time when NTP response is received by the module
time accuracy	The local time error is < 10 ms compared to the referenced NTP server's time. • typical: 5 ms • worst case: <10 ms
settling time	Maximum accuracy is obtained after 2 updates from the NTP server.
polling period dependency	Accuracy depends on the polling period. Less than 10 ms of error is guaranteed for polling periods of 120 s or less. To obtain the best possible accuracy (when your network bandwidth allows), reduce the polling period to a small value—e.g., a polling time of 5 s provides better accuracy than a time of 30 s.
time zone	The default format is universal time, coordinated (UTC). Optionally you may configure the service to use a local time zone—e.g., GMT+1 for Barcelona or Paris
daylight saving time	The module automatically adjusts the time change in the spring and fall.
leap second	To compensate for the deceleration of the earth's rotation, the module automatically inserts a leap second in the UTC time every 18 months via an international earth rotation service (IERS). Leap seconds are inserted automatically as needed. When needed, they are inserted at the end of the last minute in June or December, as commanded by the NTP server.

Using the Time Synchronization Service

Establishing Accuracy at Power Up

Before starting a system, configure the Ethernet network for a predefined interval, within which the accuracy is established. Accuracy is established at power-up, when the Ethernet module boots and then obtains the time from the NTP server.

Several updates may be required to achieve peak accuracy. Once an accurate time is obtained, the time synchronization service sets the STATUS *(see page 114)* in the associated time service register.

Obtaining and Maintaining Accuracy

The time service clock starts at 0 and increments until the Ethernet network time is fully updated from the module.

Model	Starting Date
M340 with Control Expert	January 1, 1980 00:00:00.00

Clock characteristics:
- Clock accuracy is not affected by issuing stop/run commands on the PLC
- Clock updates are not affected by issuing stop/run commands on the PLC
- Mode transitions do not affect the accuracy of the Ethernet network

Reinitializing the Time Service Register

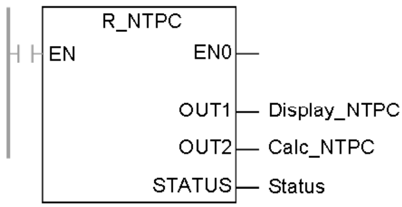
After a download or an NTP server swap, the status clock value associated with the time service register in the CPU is reinitialized.

Two polling periods elapse before an accurate time is reestablished.

Using the R_NTPC Block for Time Synchronization

R_NTPC Representation

The block reads the Ethernet network system time and transfers it into specified parameters. The additional parameter EN should be configured.



R_NTPC block has a 16-bit status word.

R_NTPC Parameter Description

Description of parameters:

<table>
<tr><th>Parameter</th><th>Data Type</th><th colspan="3">Description</th></tr>
<tr><td>Display_NTPC
(OUT1)</td><td>DT + INT</td><td colspan="3">NTP clock value displayed in:
● year, month, day, hours, minutes, and seconds using the DT format
● milliseconds as an INT</td></tr>
<tr><td>Calc_NTPC
(OUT2)</td><td>UDINT+INT</td><td colspan="3">NTP clock value displayed in:
● seconds as an UDINT
● fractions of a second as an INT</td></tr>
<tr><td rowspan="7">Status</td><td rowspan="7">INT</td><td>Low Byte</td><td>High Byte</td><td>Description</td></tr>
<tr><td>0</td><td>0</td><td>un-initialized state</td></tr>
<tr><td>1</td><td>0</td><td>illegal</td></tr>
<tr><td>0</td><td>1</td><td>The CPU is out of synchronization with the NTP server, but the clock has been updated at least once by an external server.</td></tr>
<tr><td>1</td><td>1</td><td>normal operation</td></tr>
<tr><td colspan="3">The low byte is managed by the controller
● Set = 0
  ❍ The clock value is NOT available.
  ❍ The date/time is NOT updated within last two minutes.
● Set = 1
  ❍ The date/time is updated within the last two minutes.
  ❍ The date/time is acceptable.
The high byte is managed by the NOE.
● Set = 0
  ❍ The NTP server clock value is not available.
● Set = 1
The updated date/time is received from server and sent to the module (at least once).
  ❍ within two-minute time interval
  ❍ acceptable (10 ms or less error)</td></tr>
<tr><td colspan="3">For a valid time in the CPU, the low and high bytes of the STATUS parameter must be set to 1.</td></tr>
</table>

Section 7.9
Electronic Mail Notification Service

About this Section

This section describes the electronic mail notification service, which uses SMTP to send e-mail messages.

What Is in This Section?

This section contains the following topics:

Topic	Page
Introducing the Electronic Mail Notification Service	116
Using the Electronic Mail Notification Service	117
Using the SEND_EMAIL Block for Electronic Mail Notification	118
Electronic Mail Notification Service Error Codes	120

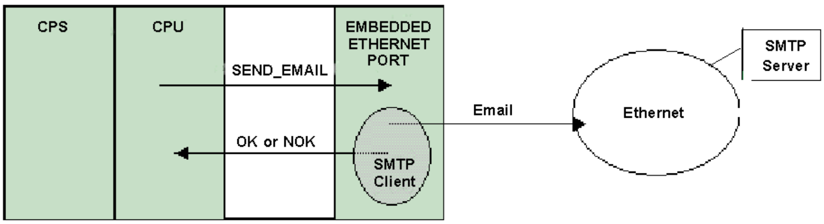
Introducing the Electronic Mail Notification Service

Introduction

The electronic mail notification service allows controller-based projects to report alarms or events. The controller monitors the system and dynamically creates an electronic mail message to alert local or remote users.

Mail Service Client

The BMX P34 20x0 processors include an SMTP client. When the module receives a specific request from the project, the module sends an email message to the mail server on the network.



Mail System Types

SMTP provides two mechanisms for the transmission of email messages—direct connection and a relay system:

Mechanism	Condition	Result
Direct connection	Sender and receiver are connected to the same transport service.	Email messages are sent to host.
Relay system	Sender and receiver are connected to different transport services.	Email messages are relayed from one server to another. The SMTP server must be supplied with the address of both the destination host and the destination mailbox.

Operating Modes and Sending Requests

Because the application program sends the email request, a controller cannot send an email message either while in the stopped mode or while downloading a project. As soon as the controller is in run mode, the function block sends a request during the first project scan.

Diagnostic counters are reset to 0 after either a power-up, a project download, or a reconfiguration of the electronic mail notification service.

Using the Electronic Mail Notification Service

Configuring the Service

An authorized administrator may use the SMTP configuration screen to:

- configure the electronic mail notification service
- set the IP address of the mail server

NOTE: The default TCP port number for SMTP is 25. Ensure that you configure the port specified by your local mail server.

Message Creation and Delivery

A user-defined event or condition triggers the SEND_EMAIL block to create a message. Each message uses one of three user-defined headers. Each message sent from the controller can contain text and variable information (up to a maximum of 1 022 bytes).

The project selects the appropriate header. Each header contains:

- sender name
- list of recipients
- subject

Header Examples

An authorized administrator can define and update the text and variable information via the SMTP configuration screen. You should define mail headers to indicate different levels of importance. For example:

- header 1 could be *Urgent problem reported by PLC 10*
- header 2 could be *Notification from substation 10*
- header 3 could be *Info message from water system*

Listing different recipients in each of the three headers assures that information flows quickly to the right recipients. The project adds pertinent information such as the specific device, process, or location. This information is added to the body of the mail message. Then the complete message is sent to an electronic mail server for distribution to recipients.

Recipients may be engineers, managers, or process owners.

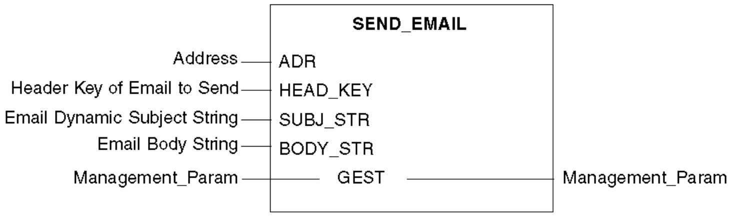
Security (Authentication)

An optional login (system ID) and password can be used to authenticate the connection to the SMTP mail server. The SMTP-supported authentication method is LOGIN.

Using the SEND_EMAIL Block for Electronic Mail Notification

SEND_EMAIL Representation

To send an email message from the application (configured through the SMTP Control Expert screen *(see page 200)*), use a SEND_EMAIL function block. The addressing to use to send a message to the email service is: *rack, slot, channel=3*. The address entry of the SEND_EMAIL function block requires the NetLink where the email should go out from. The sendBuffer takes the user payload (ASCII text string). A dedicated email errCode output is defined.



Parameter Description

The following table describes the input parameters:

Parameters	Data Type	Description
ADR	ARRAY [0...7] OF INT	Use the ADDM function block to build this field. Note that this email service is limited to the embedded Ethernet port on the CPU. If the application sends an email to an NOE module, you will get a **destination address incorrect** *(see page 119)* error code.
HEAD_KEY	INT	Corresponds to the email's addresses handled with Control Expert software (only 1, 2, and 3 are accepted).
SUBJ_STR	STRING	Represents the dynamic part of the email subject that is appended to the static subject string.
BODY_STR	STRING	Represents the body of the email.

The following table describes the input/output parameters:

Parameters	Data Type	Description
Management_ Param	ARRAY [0...3] OF INT	Exchange management table consisting of four words (activity, report, timeout, length) used to control parameters of the execution. The length parameter (4th word of the management table) is an output parameter. This word is written by the system with the total length of the email (dynamic subject + body). The maximum size of the dynamic subject + body is 1 024 characters. If the size of the dynamic subject + body exceeds 1 024 characters, an **incorrect send length** *(see page 119)* error code is generated.

SEND_EMAIL Rules

After the launch of a SEND_EMAIL function block, the activity bit is set until the email is sent. There is no confirmation that the email has arrived to the destination address. If a timeout is programmed (third management word <> 0), the email is cancelled if it has not been sent within this time. In this case, the second management word receives an **exchange stop on timeout (0x01)** *(see page 119)* error code.

You may send four simultaneous emails, using four function blocks. A fifth function block trying to send an email will receive a **no processor system resources (0x0B)** *(see page 119)* error code until one of the resources is free.

SEND_EMAIL Example

```
IF (default_id = 0) THEN
    (* PUMP IS OK *)
      SEND_EMAIL(ADDM('0.0.3'),1,'Pump n°3 is OK', '',  Mng_send_email);
ELSE
    (* PUMP IS FAULTY *)
     str_default          := INT_TO_STRING(default_id);
      str_email_body := CONCAT_STR(' Default = ', str_default);
      SEND_EMAIL(ADDM('0.0.3'),1,'Pump n°3 is faulty', str_email_body,
                                        Mng_send_email);
END_IF;
```

SEND_EMAIL EF Local Error Codes

Error Code (hex)	Description
16#00	No error
16#01	Time out
16#02	User cancelled
16#03	Bad address format
16#04	Destination address incorrect
16#06	Com Fb parameters incorrect (e.g., HEAD_KEY parameter not equal to 1, 2, or 3)
16#07	Generic transmission problem
16#09	Receive buffer too small
16#0B	No system resources: the number of simultaneous communication EFs exceeds the maximum that can be managed by the processor
16#0E	Incorrect send length

NOTE: The SEND_EMAIL EF local error codes provide more information than the Ethernet channel diagnostic information (in the Control Expert Debug screen). For example, when the message body is greater than allowed, the email is truncated and sent. Whereas, with the Ethernet diagnostic, there is no error, while the EF displays the code 16#0E.

Electronic Mail Notification Service Error Codes

Error Codes

The following codes are available only on the diagnostic screen for the electronic mail notification service:

Error Code (hex)	Description
5100	Internal error detected
5101	SMTP component not operational
5102	Mail header not configured
5104	Cannot connect to SMTP server
5105	Error detected during transmitting content of email body to SMTP server
5106	Closing SMTP connection with the server returned an error message
5107	SMTP HELO request unsuccessful
5108	SMTP MAIL request unsuccessful — SMTP server may require authentication
5109	SMTP RCPT request unsuccessful
510A	No recipient accepted by the SMTP server
510B	SMTP DATA request unsuccessful
510C	Send email request contains an invalid length
510D	Authentication unsuccessful
510E	A reset component request was received while the connection was open

Part III
Modicon M340 Module Descriptions and Specifications

About this Part

This part describes physical characteristics and technical specifications for the Modicon M340 modules that support Ethernet communications:

- BMX NOE 0100 module
- BMX NOE 0110 module
- BMX P34 2020 CPU
- BMX P34 2030 CPU

NOTE: Elsewhere in this book is a guide for selecting the appropriate hardware for your system and application requirements *(see page 39)*. For Modicon M340 system installation and specifications, see the book *Modicon X80 Racks and Power Supplies, Hardware, Reference Manual*.

What Is in This Part?

This part contains the following chapters:

Chapter	Chapter Name	Page
8	The BMX NOE 01x0 Communication Module	123
9	The BMX P34 2020 and BMX P34 2030 CPUs	129

Chapter 8
The BMX NOE 01x0 Communication Module

About this Chapter

The BMX NOE 01x0 is the network option module for communications on Ethernet systems with Modicon M340 range I/O rack-based modular platforms.

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
BMX NOE 01x0 Ethernet Services	124
BMX NOE 01x0 Electrical Characteristics	127
Standards and Certifications	128

BMX NOE 01x0 Ethernet Services

Introduction

Described below are the Ethernet services that are available for the BMX NOE 01x0 modules.

NOTE: Elsewhere in this guide are more detailed descriptions of the individual services *(see page 81)*.

Security

You can enhance security for your project by disabling the FTP/TFTP and HTTP services at times when you do not need to use them. The module uses the HTTP service to provide access to the embedded Web pages. The module uses the FTP and TFTP services to support various features including firmware upgrades, FDR services, and Ethernet remote IO.

Modbus TCP Messaging

This service allows the exchange of data between devices supporting Modbus over TCP/IP.

Maximum Capacity

The maximum Ethernet frame size depends on the type of transaction. The maximum frame size is 256 bytes for messaging.

The BMX NOE 01x0 modules are used to:

- manage these TCP connections using port 502 messaging:
  - servers (32 connections)
  - clients (16 connections)
  - Transparent Device Access (2 connections)
- scan up to a maximum of 64 devices using the I/O scanner
- serve as the address server for a maximum of 64 devices

I/O Scanning

The module can scan up to 64 Modbus TCP devices. It can send:

- Modbus read (FC03) requests (a maximum of 125 registers)
- Modbus write (FC16) requests (a maximum of 100 registers)
- Modbus read/write (FC23) requests (a maximum of 125 read registers and 100 write registers)

There are 2 048 available registers each for storing read and write data.

The following table indicates I/O Scanner performance limits:

Parameter	Value Range for Option Module
Number of I/O devices	0 ... 64
Modbus function code available	FC3, FC16, FC23
Size of the Read	0 ... 125 words for each device

Parameter	Value Range for Option Module
Size of the Write	0 ... 100 words for each device
Total size of the Read	2 K words
Total size of the Write	2 K words
Repetitive rate	0 ... 60,000 msec
Application storage area	contiguous %IW, %MW
Health status	1 bit for each variable in the IODDT

DHCP

⚠ CAUTION

UNINTENDED EQUIPMENT OPERATION

Do not use a write-protected memory card with the module. The DHCP server service writes to the card during operations, so a write-protected card prevents the service from starting.

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

DHCP (dynamic host configuration protocol) is a TCP/IP protocol that allows a server to assign an IP address with a device name that corresponds to a network node. The DHCP server supplies other parameters to clients on the network.

SNMP

SNMP (simple network management protocol) is a UDP/IP standard protocol used to monitor and manage nodes on an IP network. The SNMP agent supports both the MIB II and the Transparent Ready Private MIB *(see page 359)*.

Global Data

Global data provides the automatic exchange of data variables for the coordination of PLC applications. The BMX NOE 01x0 modules support Global Data.

The table describes the parameters for the BMX NOE 01x0 modules in the Transparent Ready service:

Parameter	Value Range for Option Module
Number of variables in the distribution group	0 ... 64 variables
Number of publications per device	0 ... 1 variable
Size of published variables	Up to 512 words each, total size of 512 words per module
Publication rate	10 ms ... 15000 ms (in 10 ms increments)
Number of subscriptions per device	0 ... 64 variables

Parameter	Value Range for Option Module
Size of subscribed variables	Up to 512 words each, mapped to several non-contiguous application areas, total size of 2048 words
Application storage area	non-contiguous %IW, %MW and unlocated
Health status	1 bit for each variable in the IODDT

Every Ethernet port can manage only one distribution group, identified with an IP multicast address. The range of IP multicast addresses for global data is between 224.0.0.0 and 239.255.255.255.

Device Management

The FDR (fast device replacement) service on the BMX NOE 01x0 offers a method of handling device replacement without disrupting the system nor interrupting service. The FDR server parameter files are stored on a memory card *(see page 50)*. The FDR server uses the device path to initialize the rest of the FDR system on bootup, as shown in the table below.

Parameters	BMX NOE 01x0
Number of I/O devices	64
memory size (max)	256K bytes

NOTE: Memory card considerations:
- The FDR server does not run on a write-protected memory card.
- FDR server operations require 256 Kbytes of available space on the memory card.

Bandwidth Monitoring

The bandwidth monitoring service indicates how a communication module's CPU is shared between services (such as gobal data, I/O scanner, messaging, etc.).

NTP

The NOE’s firmware includes an NTP client, which provides time synchronization. The time synchronization service establishes accuracy among computer clocks on an Ethernet system. For example, the time of one client may be synchronized either with another server, a referenced time source such as a radio or satellite receiver, or a GPS time server.

Use the time synchronization service for:
- event recording (for example, tracking a sequence of events)
- event synchronization (for example, triggering simultaneous events)
- alarm and I/O synchronization (for example, time stamping alarms)

BMX NOE 01x0 Electrical Characteristics

Consumed Current

The BMX NOE 01x0 modules can be inserted into any rack slot on the station assembly *(see page 37)*.

The table shows the current that the BMX NOE 01x0 and the BMX NOE 01x0H *(see page 128)* consume from the 24 VDC rack power and the residual dissipated power:

	Ethernet Module
	BMX NOE 01x0
Consumed Current	90 mA
Dissipated Power	2.2 W

NOTE: At the temperature extremes (-25... 0 ºC and 60... 70 ºC) (-13...32 ºF and 140...158 ºF), the BMX NOE 01x0H operating characteristics are the same as the BMX NOE 01x0 characteristics within its (0…60 ºC) (32...140 ºF) temperature range.

Standards and Certifications

Online Help

From the Control Expert online help, you can access the standards and certifications that apply to the modules in this product line by referring to the *Modicon M580, M340, and X80 I/O Platforms, Standards and Certifications* guide.

Download

Click the link that corresponds to your preferred language to download the standards and certifications (PDF format) that apply to the modules in this product line:

Language	
English	*Modicon M580, M340, and X80 I/O Platforms, Standards and Certifications*
French	*Modicon M580, M340, and X80 I/O Platforms, Standards and Certifications*
German	*Modicon M580, M340, and X80 I/O Platforms, Standards and Certifications*
Italian	*Modicon M580, M340, and X80 I/O Platforms, Standards and Certifications*
Spanish	*Modicon M580, M340, and X80 I/O Platforms, Standards and Certifications*
Chinese	*Modicon M580, M340, and X80 I/O Platforms, Standards and Certifications*

Chapter 9
The BMX P34 2020 and BMX P34 2030 CPUs

About this Chapter

This chapter describes the physical characteristics of the embedded Ethernet ports on the BMX P34 2020 and BMX P34 2030 CPUs.

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
BMX P34 20x0 Ethernet Services	130
BMX P34 20x0x Electrical Characteristics	132

BMX P34 20x0 Ethernet Services

Introduction

Described below are the Ethernet services that are available for the BMX P34 20x0 CPUs.

NOTE: Elsewhere in this guide are more detailed descriptions of the individual services *(see page 81)*.

Security

You can enhance security for your project by disabling the FTP/TFTP and HTTP services at times when you do not need to use them. The CPU uses the HTTP service to provide access to the embedded Web pages. The CPU uses the FTP and TFTP services to support various features including firmware upgrades, FDR services, and Ethernet remote IO.

Modbus TCP Messaging

The Modbus TCP messaging service allows the exchange of data between devices supporting Modbus over TCP/IP.

Maximum Capacity

The maximum frame size depends on the type of transaction:
- For messaging, the maximum frame size is 256 bytes.

The BMX P34 20x0 CPUs allow you to:
- manage these TCP connections using port 502 messaging:
  - servers (32 connections)
  - clients (16 connections)
  - Transparent Device Access (2 connections)

NOTE: Elsewhere in this guide is detailed information for opening and closing connections *(see page 333)*.

SNMP

SNMP (simple network management protocol) is a UDP/IP standard protocol used to monitor and manage nodes on an IP network. The SNMP agent supports both the MIB II and the Transparent Ready Private MIB *(see page 359)*.

FDR Client

In the event of a device malfunction, the Fast Device Replacement service *(see page 99)* automatically reconfigures the replacement CPU in accordance with its device name. The new device retrieves its IP addresses, network parameters, and FDR file path from a DHCP server.

Embedded Web Pages

The BMX P34 20x0 CPUs support the diagnostics Web pages that you can access through the Modicon M340 Diagnostics screen.

Bandwidth Monitoring

The Bandwidth Monitoring service *(see page 100)* indicates how a communication module's CPU is shared between services (such as global data, I/O scanner, messaging, etc.).

SMTP

The BMX P34 20x0 processors include an SMTP client, which is an electronic mail notification service that allows controller-based projects to report alarms or events. The controller monitors the system and dynamically creates an electronic mail message to alert local or remote users. The PLC sends the mail message to a mail server on the network for distribution.

BMX P34 20x0x Electrical Characteristics

Consumed Current

The BMX P34 20x0x CPUs are inserted into in the rack assembly *(see page 37)*.

The table shows the current that the BMX P34 20x0x CPUs consume from the 24 VDC rack power and the residual dissipated power:

	Embedded Ethernet Port	
	BMX P34 2020	BMX P34 2030/20302
Consumed Current	95 mA	135 mA
Dissipated Power	2.3 W	3.2 W

These values do not include the consumption of devices connected on the communication port 5 V supply.

Part IV
Ethernet Configuration with Control Expert

About this Part

This part describes the Control Expert configuration of the BMX NOE 01x0 communication modules and BMX P34 20x0 CPUs.

What Is in This Part?

This part contains the following chapters:

Chapter	Chapter Name	Page
10	Software Configuration Parameters	135
11	Configuring an Ethernet Network	209
12	Debugging with Control Expert	215
13	Ethernet Language Objects	225
14	M340 Ethernet Communications Quick Start	249

Chapter 10
Software Configuration Parameters

About this Chapter

This chapter introduces the configuration parameters for the different facilities used by the BMX NOE 01x0 modules and the BMX P34 20x0 CPUs.

What Is in This Chapter?

This chapter contains the following sections:

Section	Topic	Page
10.1	The Configuration Screen	136
10.2	Security	137
10.3	IP Configuration Parameters	139
10.4	Messaging Configuration Parameters	144
10.5	I/O Scanner Configuration Parameters	148
10.6	Global Data Configuration Parameters	185
10.7	SNMP Configuration Parameters	191
10.8	Address Server Configuration Parameters	196
10.9	Bandwidth Checking	198
10.10	Electronic Mail Notification Service Configuration Parameters	200
10.11	Time Synchronization Service Configuration Parameters	203

Section 10.1
The Configuration Screen

The Module Configuration Screen

Configuring an Ethernet Network

Use the multi-zone Ethernet network configuration screen to declare the communication channel and to configure the necessary parameters for an Ethernet link.

Use the configuration screen to declare the communication channel and to configure the necessary parameters for an Ethernet port. This screen can include the following zones and functions, depending on the module type (CPU or communication module):

<table>
<tr><th>Zone</th><th colspan="2">Function</th></tr>
<tr><td>Module Family</td><td colspan="2">Use this to choose the network family to be configured.</td></tr>
<tr><td>Module Address</td><td colspan="2">After you associate the network with a module, the module address displays.</td></tr>
<tr><td>Module IP Address</td><td colspan="2">Displays the IP address settings.</td></tr>
<tr><td>Module Utilities</td><td colspan="2">Select the services that are enabled for the network.</td></tr>
<tr><td rowspan="11">Service tabs</td><td>Tab</td><td>Description</td></tr>
<tr><td>Security tab</td><td>Allows enabling and disabling of FTP, TFTP, and HTTP (see page 137).</td></tr>
<tr><td>IP Configuration tab</td><td>Enables the configuration of TCP/IP services (see page 139)</td></tr>
<tr><td>Messaging tab</td><td>Allows the configuration of Access Control (see page 144) for the module.</td></tr>
<tr><td>IO Scanning tab</td><td>Allows configuration of I/O Scanning (see page 148)</td></tr>
<tr><td>Global Data tab</td><td>Allows configuration of Global Data (see page 185)</td></tr>
<tr><td>SNMP tab</td><td>Allows configuration of SNMP (see page 191)</td></tr>
<tr><td>Address Server tab</td><td>Allows configuration of the address server (see page 196)</td></tr>
<tr><td>Bandwidth tab</td><td>Allows you to check that the services configured are compatible with the processing capacity of the Ethernet channel (see page 198)</td></tr>
<tr><td>Mail Service web page</td><td>Allows configuration of Mail Service (see page 200)</td></tr>
<tr><td>NTP tab</td><td>Allows configuration of Time Service (see page 203)</td></tr>
</table>

NOTE: The SMTP utility is only available on the CPU modules, and the NTP utility is only available on the NOE modules.

NOTE: Instructions on configuring an Ethernet network *(see page 254)* from the communication module in the project browser are found in a later chapter.

Section 10.2
Security

Security Features

Security and HTTP, FTP, and TFTP Services

You can enhance security for your project by disabling the FTP/TFTP and HTTP services at times when you do not need to use them. The module uses the HTTP service to provide access to the embedded webpages. The module uses the FTP and TFTP services to support various features including firmware upgrades, and FDR services.

The module's HTTP, FTP, and TFTP services can be disabled or enabled using the **Security** screen of the Ethernet network configuration window.

HTTP, FTP, and TFTP services are disabled by default in DTM instances created using Unity Pro 8.1 or later, with respect to the following modules and firmware versions:
- BMX NOE 0100 firmware version 2.90 or later
- BMX NOE 0110 firmware version 6.00 or later
- BMX P34 20•0 firmware version 2.60 or later

NOTE: Unity Pro is the former name of Control Expert for version 13.1 or earlier.

HTTP, FTP, and TFTP services are enabled by default in instances created using previous versions of Control Expert.

You can use Control Expert to enable or disable HTTP, FTP, and TFTP services as described in the following procedure.

If the HTTP, FTP, or TFTP services have been enabled with Control Expert, they can also be enabled or disabled at run time using a DATA_EXCH function block. (See the *Communication Block Library* for Control Expert.)

Using Control Expert to Enable and Disable Firmware Upgrade & FDR and Web Access Services

Perform the following steps to enable or disable FTP/TFTP or HTTP services on the module.

Step	Action
1	In the Control Expert main menu, select **Tools** → **Project Browser** to open the **Project Browser**.
2	In the **Project Browser**, navigate to **Communication** → **Networks**, then double-click on an Ethernet network. The **Ethernet** network configuration window opens.
3	Click the **Security** tab to open the **Security** screen.
4	On the **Security** screen, choose the appropriate setting: (**Enabled** or **Disabled**) for the service or services.
5	In the Control Expert toolbar, click **Validate**, then **Save** your edits

The edits do not take effect until they are successfully downloaded from your PC to the CPU and from the CPU to the communication modules and network devices.

Section 10.3
IP Configuration Parameters

About this Section

This section introduces the configuration parameters on the **IP Configuration** tab.

What Is in This Section?

This section contains the following topics:

Topic	Page
The IP Configuration Tab	140
Configuration Parameters for IP Addresses	142
Ethernet Frame Format	143

The IP Configuration Tab

Introduction

To communicate on Ethernet networks through the BMX NOE 01x0 or BMX P34 20x0 modules, it is necessary to set the configuration parameters linked to TCP/IP. On the **IP Configuration** tab you can:

- declare the communication channel
- configure the necessary parameters for an Ethernet port

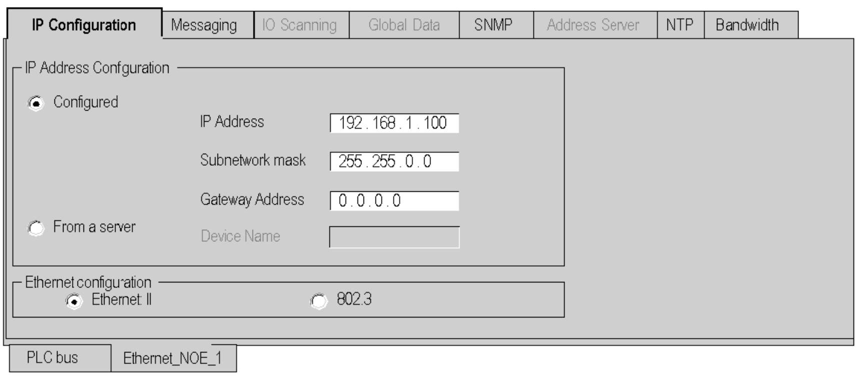
IP Configuration Tab

The following procedure shows how to access the **IP Configuration** tab from the index page:

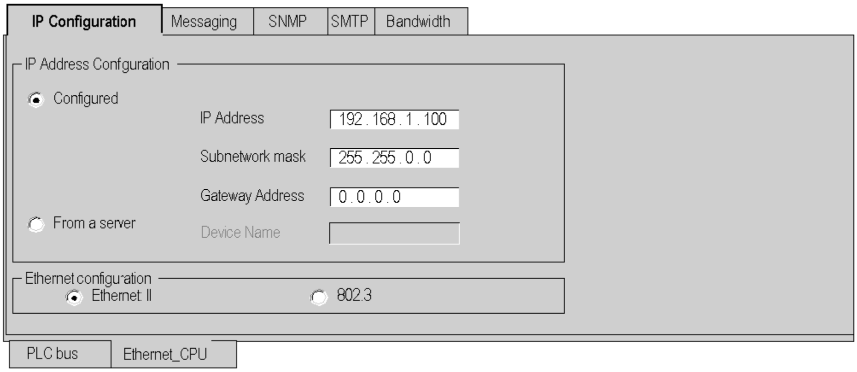
Step	Action
1	Access the module configuration screen.
2	Select the **IP Configuration** tab (see illustration below).

The following figure shows the **IP Configuration** tab:

NOE configuration screen:



CPU configuration screen:



The IP configuration parameter zones are discussed in detail elsewhere in this guide:
- IP Address configuration *(see page 142)*
- Ethernet configuration *(see page 143)*

Configuration Parameters for IP Addresses

Introduction

On the IP Configuration tab *(see page 140)*, you can define the IP address of a module in the **IP Address Configuration** zone. The options are:

- **Configured:** Manually enter the IP address, subnetwork mask, and gateway address.
- **From a server:** The configuration is supplied by a server device.

NOTE: To configure IP addresses, obtain the appropriate network address and subnetwork mask from your system administrator. Elsewhere in this guide is detailed information for IP addressing *(see page 61)*.

Configured IP Address

Selecting the **Configured** field allows manual configuration according to your own requirements:

- **IP Address:** The IP address of the module
- **Subnetwork mask:** The mask defines the part allocated to the subnetwork identifier in the IP address.
- **Gateway Address:** The gateway address is the IP address of the default gateway to which messages for other networks are transmitted.

NOTE: If the module is connected to an existing TCP/IP network, the IP addresses are administered globally, therefore the IP parameters must be configured. Otherwise there is a risk of disturbance on the existing network caused by possible double allocation of the IP addresses.

From a Server

Selecting the **From a server** field allows the module's IP address to be configured from a remote device acting as a DHCP/BOOTP server *(see page 87)*. When the **From a server** button is active:

- Leaving the **Device Name** field empty facilitates communications that are compatible with any setting on the rotary switches *(see page 64)*.
- To use a device name in the **Device Name** field, set the lower rotary switch to its STORED position. If you set the switch any other position, the result depends on the setting of the lower rotary switch, as described in the Ethernet Port Status table *(see page 70)*.
- The configured IP parameters have no effect and are grayed out.

NOTE: The M340 Ethernet modules will not receive an IP address from a BOOTP/DHCP server on application download if the IP configuration has not changed.

NOTE: The maximum length for the device name is 16 characters. Valid characters include alphanumerics (0 to 9, A to Z) and underscores.

Ethernet Frame Format

Introduction

The **Ethernet configuration** field on the IP Configuration tab *(see page 140)* is used to define the frame format for TCP/IP communications in accordance with those formats required by end devices (valid for configured IP addresses only). Options are:

- **Ethernet II:** The Ethernet II format complies with the RFC 894 standard (the most common standard).
- **802.3:** The 802.3 format complies with the RFC 1042 standard.

Section 10.4
Messaging Configuration Parameters

About this Section

The section discusses the configuration of IP messaging parameters.

What Is in This Section?

This section contains the following topics:

Topic	Page
The Messaging Configuration Tab	145
Messaging Configuration Parameters	147

The Messaging Configuration Tab

Introduction

To limit access to the BMX NOE 01x0 and BMX P34 20x0 CPUs, set the access control parameters on the **Messaging** tab.

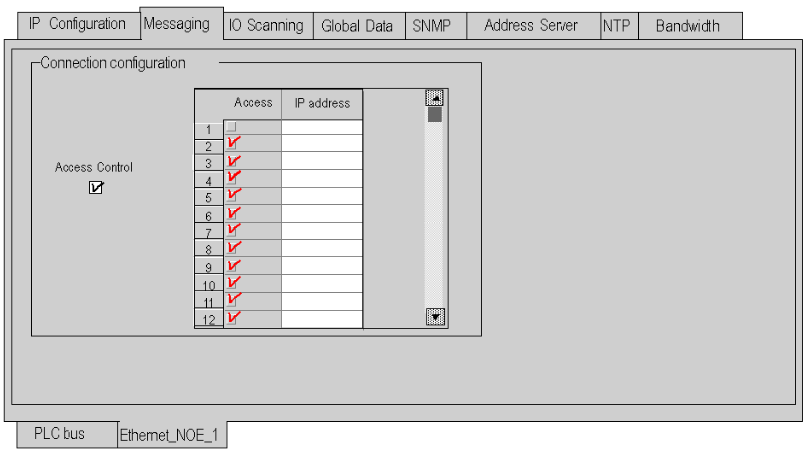
Messaging Tab

The following procedure shows how to access the **Messaging** page from the index page:

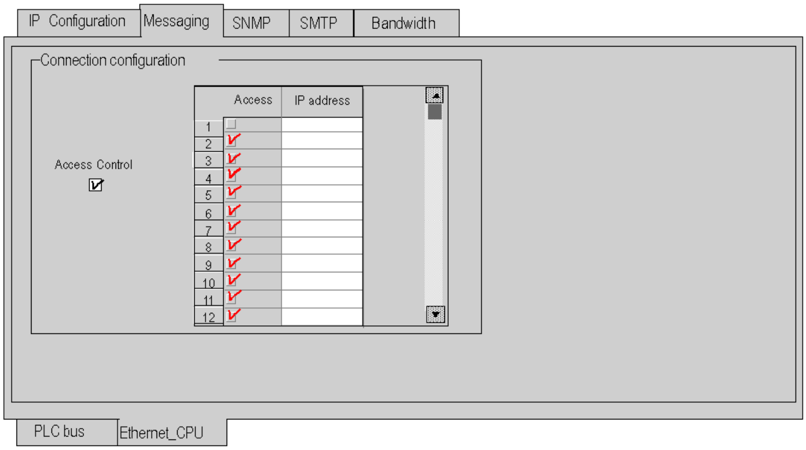
Step	Action
1	Access the module configuration screen.
2	Select the **Messaging** tab (see illustration below).

The **Messaging** tab is shown below:

NOE configuration screen:



CPU configuration screen:



The messaging configuration parameters are discussed in detail on the following pages.

Messaging Configuration Parameters

Accessing Messaging Configuration Parameters

Configuration parameters can be accessed in two areas on the Messaging tab screen:

- the **Connection Configuration** area
- the **Access Control** area

Connection Configuration Area

The **Connection Configuration** area is used to:

- activate an access control utility
- list the remote devices that can connect to the module according to a communication protocol

Access Control

The **Access Control** box is used to activate or deactivate control of remote devices that are attempting to open a TCP connection to the module. The functionality depends on whether the box is checked or not:

- **checked:** Access control management is activated and the **Access** column of the table is active (no longer grayed out).
  - The module can only communicate to the addresses entered in the 128 available spaces in the **IP address** column.
  - With the module in client mode it can only connect to remote devices selected by the **Access** column in the **Connection Configuration** table.
- **unchecked:** Access control management is inoperative and the **Access** column of the table is not active (grayed out).
  - With the module in server mode, remote third-party devices can connect as clients (before communicating with the module) without being declared in the table.

NOTE: Access control is only effective on the TCP/IP profile and assists module operations in server and client mode.

NOTE: If you select the **Access Control** check box but do not enter addresses in the **IP address** column, messaging will stop working.

Section 10.5
I/O Scanner Configuration Parameters

About this Section

This section introduces the configuration parameters linked to the I/O Scanner.

An I/O Scanner in the BMX NOE 01x0 communication modules transfer data between network devices and allows a CPU to regularly read data from and write data to scanned devices. The I/O Scanner is configured with Control Expert.

What Is in This Section?

This section contains the following topics:

Topic	Page
The I/O Scanning Tab	149
I/O Scanning Contextual Menu for Copy/Cut/Paste	154
I/O Scanning with Multiple Lines	156
Introduction to Configuring Advantys from Control Expert	158
Introduction to Configuring the PRM Master DTM	162
Introduction to Configuring a BMX PRA 0100 from Control Expert	164
Property Box	168
Saving an Advantys Configuration in an Control Application	174
Managed Variables	175
I/O Scanner Concepts	177
Scanning Period	178
Configuration of Parameters Linked to the I/O Scanner Utility	179
Configuration of General Parameters for the I/O Scanner	180
I/O Scanner: Health Block Bits	181
I/O Scanner: Device Control Block	183

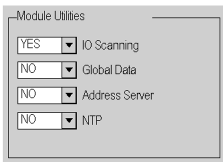
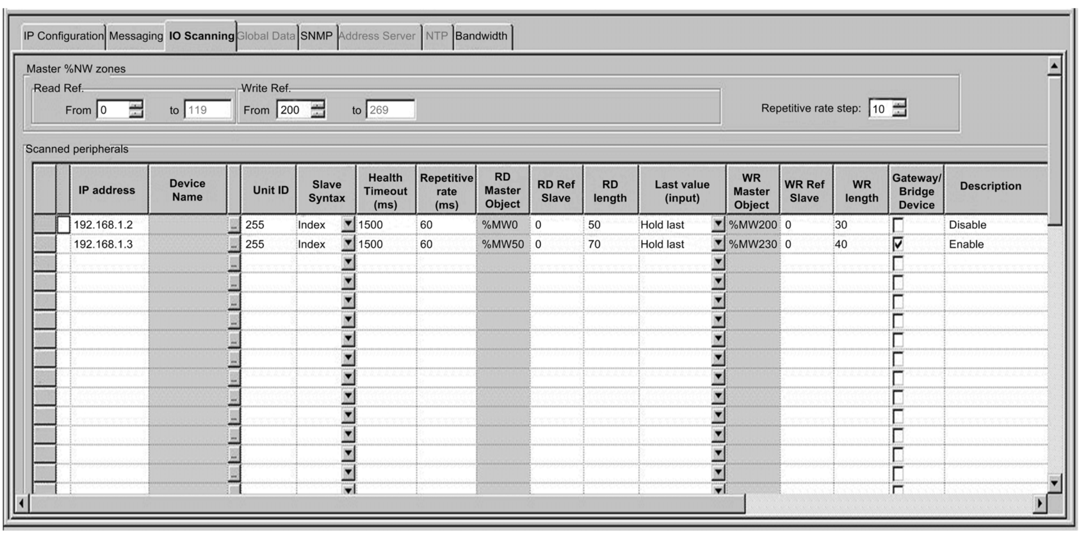
The I/O Scanning Tab

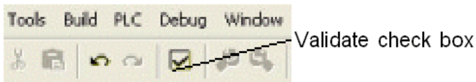
Introduction

Use the I/O Scanner to scan I/O modules to get inputs and set outputs.

I/O Scanning Tab

The following procedure shows how to access the BMX NOE **I/O Scanning** tab from the index page:

Step	Action
1	Open your application using the **BMX NOE 01x0** in Control Expert.
2	In the **Project Browser**, locate the **Communication** → **Networks** sub-directory.
3	Click the Ethernet module (**BMX NOE 01x0** in our example) to open the Ethernet module configuration screen.
4	In the **Module Utilities** field, select **Yes** in the **IO Scanning** menu. 
5	Select the **I/O Scanning** tab to open the I/O scanning configuration screen:
	

Step	Action
6	Enter the parameter settings under each of the column headings for one line of the I/O Scanner Configuration. Refer to I/O Scanning Parameters (below) to see the settings used for this example.
7	Click the validate check box in the upper tool bar to confirm the I/O scanning parameter settings. 

I/O Scanning Parameters above Table

Master %MW zone parameters:

Parameter	Field	Description
Read Ref.	A pair of data boxes — **From** and **to**	The values in these boxes define the range of destination address values in the CPU for the data read from each device. The addresses you enter here are displayed in the **RD Master Object** column of the dialog. In the example above, the **Read Ref.** values range from 0 to 599. Notice that these values are displayed as %MW0, %MW599, etc. in the **Master Object** column.
Write Ref.	A pair of data boxes — **From** and **to**	The values in these boxes define the range of source address values in the CPU. The address you enter here is displayed in the **WR Master Object** column. In the example above, values starting at %MW2000 are shown in the WR Master Object column.

The **Repetitive rate step** parameter:

Parameter	Field	Description
Repetitive rate step	data box	The **Repetitive rate step** is set in multiples of 5 ms (the minimum) through 200 ms (the maximum). The **Repetitive rate (ms)** column is where you enter a rate of time for how often you want the I/O scanner to send a query to the device after the rate has timed out. **NOTE:** The **Repetitive rate (ms)** of the I/O scanner table is a multiple of the rate displayed in the **Repetitive rate step** box. The real repetitive rate being executed by the I/O scanner service is shown in the **Repetitive rate (ms)** column. **NOTE:** An entry in the **Repetitive rate (ms)** column is rounded up to the next multiple that was entered in the **Repetitive rate step** if the entry is not a multiple of the **Repetitive rate step**. For example, if the entry in the **Repetitive rate step** is 5 and you enter a 7 in the **Repetitive rate (ms)** column, the 7 is rounded up to 10. If you change the **Repetitive rate step** to 6 and enter a 7 in the **Repetitive rate (ms)** column, the 7 is rounded up to 12.

I/O Scanning Table Parameters

The **I/O Scanning** table configuration parameters are:

Parameter	Description	Example
Entry #	This is the first column; it has no name. Valid range: 1 ... 64 Each entry represents an I/O Scanning exchange on the network.	
IP address	This is the IP address of the scanned Ethernet slave device.	192.168.1.100
Device Name	To configure a device (Advantys island, DTM or PRA), click the ... button to open the **Property** box *(see page 168)* to start the device configuration software. For an introduction to this procedure for Advantys, go to Advantys configuration *(see page 158)*. For an introduction to this procedure for DTMs, go to PRM Master DTM configuration *(see page 162)*. For an introduction to this procedure for PRA, go to BMX PRA 0100 configuration *(see page 164)*. **NOTE:** While the **Property** box is open, **I/O scanning** cannot be edited.	MySTB1, Master_PRM_DTM_10, PRA1
Unit ID	This field associates the slave address of the device connected to an Ethernet/Modbus gateway with the IP address of that gateway: ● value range: 1 to 255 ● default value: 255 When using a bridge, enter the bridge index (1 to 255) in this field.	255
Slave Syntax	Use this drop-down menu to pick the way **RD Ref Slave** and **WR Ref Slave** values are displayed. The 4 choices are (with an example): ● **Index**: 100 ● **Modbus**: 400101 ● **IEC 0**: %MW100 ● **IEC 1**: %MW101	**Index** (default value)
Health Timeout (ms)	This field sets the maximum interval between the responses from a remote device: ● range: 0ms to 50 seconds ● interval: 1ms After this time period expires, the received data is invalid. The health timeout must longer than the repetitive rate. For an NOE Ethernet module, it also must be longer than the CPU scan time.	1500ms

Parameter	Description	Example
Repetitive rate (ms)	The rate at which data will be scanned, from 0...60000 in multiples of the **Repetitive rate step**.	60ms
RD Master Object*	Destination address in the master PLC where, from each device, newly read information is stored. This parameter cannot be accessed. It is calculated automatically as the sum of: ● The **From** address (beginning) of **Read ref.** (in the zone above the table) ● The **RD length** value (in the table below)	%mw10
RD Ref Slave**	Source address index in the slave/remote device	The format of this value depends on the **Slave Syntax**: ● **Index**: 5 ● **Modbus**: 400006 ● **IEC 0**: %MW5 ● **IEC 1**: %MW6
RD length	Number of words to read	10
Last value (input)	This field configures the behavior of inputs in the event of an access error in relation to the remote device (for example: inoperative network or device power supply, etc.): ● **Set to 0**: fall back to 0 ● **Hold last**: maintain last value	**Hold last**
WR Master Object*	Source address of the master PLC whose data is being written into the slave/remote device. This parameter cannot be accessed. It is calculated automatically as the sum of: ● The **From** address (beginning) of **Write ref.** (in the zone above the table) ● The **WR length** value (in the table below) Write operations are always performed at the word level.	%mw20
WR Ref Slave**	The address of the first word written into the slave/remote device.	The format of this value depends on the **Slave Syntax**: ● **Index**: 1 ● **Modbus**: 400002 ● **IEC 0**: %MW1 ● **IEC 1**: %MW2
WR length	Number of words to be written	10

Parameter	Description	Example
Gateway/Bridge Device	To allow slower TCP/IP network devices (ex: gateways and bridge) to be compatible with the I/O Scanner: ● Select the check box to enable this feature. Defines a new bit, and sets it to high (1). ● Deselect the check box to disable this feature (default). Defines a new bit, and sets it to zero (0).	Values: ● **Disable** = deselected check box ● **Enable** = selected check box
Description	Additional information	
*Master refers to the client PLC that makes the request. **Slave refers to the server from which data is read or to which data is written.		

NOTE: For more information about the I/O Scanning table, refer to the Contextual Menu for Copy/Cut/Paste topic *(see page 154)*.

NOTE: For more information about the I/O Scanning table, refer to the I/O Scanning with Multiple Lines topic *(see page 156)*.

I/O Scanning Contextual Menu for Copy/Cut/Paste

At a Glance

A right-click on a line in the **I/O Scanning** table opens the **I/O Scanning Contextual Menu**. Use this menu to perform common operations on the lines of the **I/O Scanning** table, such as, delete a device, copy & paste, cut & paste, insert a new line, etc.

Contextual Menu

The following illustration is the **I/O Scanning** contextual menu:

The following table describes the menu functions:

Menu Item	Description
Delete Device	For an ACS or PRA configuration, **Delete Device** permanently deletes the **Device Name** and all its data (and associated ACS symbols). For a **PRM Master DTM**, its link to the **I/O Scanning** table is deleted, **NOTE:** Deleting a **PRM Master DTM** link from the **I/O Scanning** table does not delete the corresponding DTM from the connectivity tree in the DTM Browser.
Cut line(s)	**Cut line(s)** copies and deletes the selected I/O Scanning lines. The lines are copied without the **Device Name** information. For an ACS or PRA configuration, it permanently deletes the **Device Name** and all its data (and associated ACS symbols). For a **PRM Master DTM**, the link between the DTM and the **I/O Scanning** line is removed.
Copy line(s)	**Copy line(s)** copies the selected lines, but without the **Device Name**.
Paste line(s)	**Paste line(s)** has 2 actions depending on its target line: ● If the line is empty, it fills the line with the copied line (without a **Device Name)** ● If the line is not empty, it replaces the line with the copied line (without a **Device Name**). Be careful, it also permanently deletes the **Device Name** link to the **I/O Scanning** table and, for an ACS or PRA configuration, all its data (and associated ACS symbols) of the old line before replacing it with the copied line.

Menu Item	Description
Insert copied line(s)	**Insert copied line(s**) inserts the copied line between the selected line and the line just above it. Be careful with ACS or DTM configurations, all the lines below the inserted line become desynchronized. To synchronize these lines, open and close the device configuration tool, then do an **Update** from the Property box *(see page 168)*.
Insert empty line	**Insert empty line** inserts an empty line above the line selected line. Inserting an empty line does not desynchronize the devices below the line, but using this line for a new device can, depending the number of words needed, desynchronize the devices below the line.
Pack all lines	**Pack all lines** removes any empty lines between the top of the **I/O Scanning** table and the last non-empty of the table.

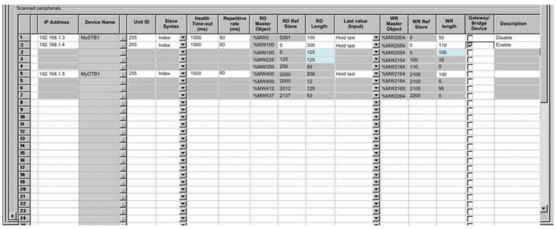
I/O Scanning with Multiple Lines

At a Glance

Modbus exchanges are limited to a maximum of 125 input words and 100 output words. If an application needs to exchange more than these limits for a device, more than one **I/O Scanning** line can be used: multiple lines for one device.

When the length is higher than the authorized limit for one Modbus exchange, the length is divided into 2 or more Modbus exchanges. New lines are created for each Modbus exchanges with the PLC.

The following **I/O Scanning** table is used for the multiple device lines example:



Scanned peripherals

	IP Address	Device Name	Unit ID	Slave Syntax	Health Time-out (ms)	Repetitive rate (ms)	RD Master Object	RD Ref Slave	RD Length	Last value (Input)	WR Master Object	WR Ref Slave	WR length	Gateway/ Bridge Device	Description
1	192.168.1.3	MySTB1	255	Index	1500	60	%MW0	5391	100	Hold last	%MW2004	0	50	☐	Disable
2	192.168.1.4		255	Index	1500	60	%MW100	0	300	Hold last	%MW2054	0	110	☑	Enable
2							%MW100	0	125		%MW2054	0	100	☐	
3							%MW225	125	125		%MW2154	100	10	☐	
4							%MW350	250	50		%MW2164	110	0	☐	
5	192.168.1.5	MyOTB1	255	Index	1500	60	%MW400	2000	200	Hold last	%MW2164	2100	100	☐	
5							%MW400	2000	12		%MW2164	2100	5	☐	
6							%MW412	2012	125		%MW2169	2105	95	☐	
7							%MW537	2137	63		%MW2264	2200	0	☐	

NOTE: This example shows an Advantys island, but DTM and PRA devices work the same way.

Multiple Line Length Configuration Example

In this example, the *first* (the main) **I/O Scanning** line 2 contains all the information for the exchanges with the device including the totals for the **RD length** and **WR length**.

The *second* line 2 contains the specific word lengths (125 and 100) needed so that it can also be used for part of the exchanges.

Line 2 needs a **RD length** of 300 word and a **WR length** of 110 words. How many extra lines are needed?

- **RD length** = 300/125 = 2.72 = 3 lines needed
- **WR length** = 110/100 = 1.10 = 2 lines needed

The larger of the 2 numbers is used:

- Three lines are needed to accommodate the **RD length**: 125 words, 125 words, 50 words for a total of 300 words
- The 3 lines for the **WR length** are: 100 words, 10 words, 0 words for a total of 110 words

The *second* line 2, line 3, and line 4 correspond to the Modbus exchange queries.

When multiple lines are used, only the **RD length** and **WR length** columns of these new lines can be edited. In the case of Advantys or DTM, the software supplies the **RD length** and the **WR length**, and they cannot be changed in the **I/O Scanning** table.

NOTE: It is not necessary to have a **Device Name** defined to use multiple lines.

The total number of words allowed in an **I/O Scanning** table is:
- 4 KW for Premium extended and Quantum networks
- 2 KW for Premium ETY and M340 NOE modules

Line Length for Multiple Word Variables

When using variables with 2 or more words, adjust the **RD** and **WR lengths** so that a variable is not partly on one **I/O Scanning** line and partly on the next. Because the 2 newly created lines result in 2 independent Modbus exchanges that can be sent non-synchronized to the device. The variables can receive the wrong values (if the 2 parts are received at different times). It may be necessary to use a **RD length** < 125 and a **WR length** < 100 for some of the scanned lines, in order to get each variable on only one exchange line.

WARNING

UNEXPECTED SYSTEM BEHAVIOR

Verify that multiple word variables are completely on the same **I/O Scanning** line to avoid sending parts of a variable data in 2 non-synchronized **I/O Scanning** Modbus exchanges.

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

Introduction to Configuring Advantys from Control Expert

At a Glance

The Advantys Configuration Software (ACS) is integrated in Control Expert. This allows you to configure Advantys STB and OTB islands from the Control Expert **Ethernet I/O scanning** tab.

Configuring an Advantys Island

⚠ WARNING
UNEXPECTED SYSTEM BEHAVIOR
Always launch ACS from Control Expert in order to synchronize variables and data between Control Expert and ACS.
Failure to follow these instructions can result in death, serious injury, or equipment damage.

⚠ WARNING
UNEXPECTED SYSTEM BEHAVIOR
Stop the PLC before transferring an ACS configuration and/or I/O scanning modifications.
Failure to follow these instructions can result in death, serious injury, or equipment damage.

The following procedure configures an Advantys STB or OTB island:

Step	Action	Results
1	Open the Ethernet network configuration screen.	
2	In the **Module Utilities** zone select **YES** for **IO Scanning**.	
3	Select the **I/O Scanning** tab.	**I/O Scanning** screen opens.
4	Enter, on a free line, the IP address for the connection you want to use to communicate with the Advantys island.	
5	Enter **RD length** and **WR length** on the same line. The lengths must be long enough for the expected Advantys configuration.	
6	Validate the **I/O Scanning** screen.	
7	Click on the **...** button (that is next to **Device Name** cell on the same line).	The **Property** box *(see page 168)* opens.
8	Select **STB** or **OTB** in the **Device Type** drop-down menu.	
9	Enter a **Device Name** (following the naming rules *(see page 171)*).	

Step	Action	Results
10	You have 2 choices: 1. If you want to go to ACS **now** to configure an island, click on the **Launch Advantys Configuration Software** button. Click on **Yes** in the “*Confirm device name and type*” Message Box and go to Step 11. 2. If you want to configure the Advantys island **later**, click on the **OK** button. Click on **Yes** in the “*Confirm device name and type*” Message Box. To open the ACS later: ❍ Carry out Step 7. ❍ Click on the **Launch Advantys Configuration Software** button.	Results for both 1. and 2. are: ● A Control Expert message box opens: “*The device name and device type won’t be modifiable. Do you want to confirm the device name and device type?*” ● The **Device Type** and **Device Name** are verified and saved. ● The **Property** box closes.
11	After ACS opens, configure your Advantys island. **NOTE:** While the ACS is open Ethernet screen is locked and cannot be edited, but the other Control Expert services can be edited. **NOTE:** The `User Defined Label` must be filled in the `IO image`. If not, the Advantys varialble will no be added in the Control Expert Data Editor.	
12	When your Advantys island has been built and validated, close ACS.	A Control Expert message box opens “*Do you want to update your symbols now?*”
13	You have 2 choices: 1. Click on **Yes** in the “*update*” Message Box and go to Step 14. 2. Click on **No** in the “*update*” Message Box. You are returned to the **I/O Scanning** screen without carrying out the **Yes** results. Later, when you want to update the Advantys symbols into Control Expert: ❍ Carry out Step 7 ❍ In the **Property** box, click on the **Update** button and go to Step 14.	If you clicked on **No**: ● You are returned to **I/O Scanning** without carrying out the results in Step 14. ● The **Device Name** is displayed in the **I/O Scanning** in red. This indicates that the island configuration has not been synchronized with Control Expert.

Step	Action	Results
14	Your Advantys island configuration is being **synchronized** with Control Expert. After the synchronization is finished, you are returned to **I/O Scanning**. Verify that the **Device Name** is now displayed in black.	The results are: ● The Advantys island modifications are synchronized with the Control Expert application. ● The Advantys island symbols are imported into the Control Expert Data Editor. ● The Advantys **Device Name** is displayed in the **I/O Scanning** in black. This indicates that the island configuration is synchronized.
15	Build your Control Expert application.	
16	STOP the PLC.	
17	Transfer: ● Control Expert application to the PLC ● STB or OTB configuration to the Advantys island using ACS	
18	RUN your application in the PLC.	

Copy an Existing Island

This following procedure copies an existing Advantys island file (*.isl) into a new Advantys island configuration:

Step	Action
1	From Control Expert, open a new Advantys island in ACS.
2	In ACS, select **File menu** → **Copy Island Contents**.
3	In the **Open island** window, select the island file (*.isl) to copy.
4	Click on **Yes** in the “Do you want to proceed?” message box.
5	The message “Island file has been saved.” in the **Log Window** verifies that the operation was successful.

Copy an Island File to a New Location

The following procedure copies an Advantys island file (*.isl) to a new directory:

Step	Action
1	In ACS, open an island configuration, for example, STB1.
2	Select **File** menu => **Copy STB1 Contents**
3	In the **Copy STB1.isl to** window, select the target directory.
4	The message “A copy of the island file has been saved with another name.” includes in the **Log Window** verifies that the operation was successful. The name is new because its path has changed.

Introduction to Configuring the PRM Master DTM

At a Glance

The **PRM Bus Master** uses the Control Expert **I/O Scanner** to communicate with the CPU through an Ethernet port. This requires configuring the **PRM Master DTM** in the Control Expert Ethernet **I/O Scanning** tab.

Configuring a PRM Master DTM

The following procedure configures a **PRM Master DTM** in the **I/O Scanner**:

Step	Action
1	Install the **PRM Master DTM** on the Host PC. **NOTE:** After installing new DTMs, the **Hardware Catalog** must be updated.
2	Add a **PRM Master DTM** to the connectivity tree in the DTM Browser using the contextual **Device menu** service.
3	In the DTM Browser, select the **PRM Master** and use the contextual **Device menu** function to open the DTM **PRM Offline Parameter** screen.
4	In the **General Setting** part of this screen set the IP address of the **PRM device.**
5	Open the **I/O Scanning** configuration editor (tab).
6	In the **Module Utilities** zone select **YES** for **IO Scanning**.
7	Select the **I/O Scanning** tab. **Results: I/O Scanning** configuration editor opens.
8	Enter, on a free line, the IP address for the connection to be used to communicate with the **PRM Bus Master**.
9	Set correct values for the **Read Ref.** and **Write Ref.** parameters.
10	Enter **RD length** and **WR length** for the IP address line (within the **Read Ref.** and **Write Ref.** constraints). **NOTE:** The lengths must be long enough for the expected configuration **PRM Master DTM** and its subnode DTMs.
11	Validate the **I/O Scanning** screen.
12	Click on the **...** button (next to **Device Name** cell). Results: The **Property** box *(see page 168)* opens.
13	Select **DTM** in the **Device Type** drop-down menu.
14	Select the protocol in the **DTM Protocol** drop-down menu.
15	Select a **PRM Master DTM** in the **DTM Name** drop-down menu.
16	Click on **OK** to validate the choices you made. **Results:** ● The **Device Type**, **Device Protocol** and **Device Name** are verified and saved. ● The **Property** box closes.

Step	Action
17	Update the **I/O Scanning** line, refer to Updating I/O Scanning for a PRM Master DTM *(see page 163)*.
18	Build the Control Expert application.
19	Stop the PLC.
20	Transfer the Control Expert application to the PLC.
21	In the DTM Browser, right click on **PRM Master** and select the **Connect** function.
22	In the DTM Browser, right click on **PRM Master** and select the **Store data to device** function.
23	**Run** the application in the PLC.

Updating I/O Scanning for a PRM Master DTM

The following procedure updates the **I/O Scanning** information for a **PRM Master DTM**:

Step	Action
1	Configure and validate the **PRM Bus Masters** in the DTM Browser using the contextual **Device menu** function.
2	Open the **I/O Scanning** configuration editor (tab).
3	Click on the **...** button (that is next to the **Device Name** of the **PRM Master DTM** to update).
4	In the open **Property** box *(see page 168)*, click on the **Update** button. **Results:** ● The **PRM Master DTM** modifications are synchronized with the Control Expert application. ● The **PRM Master DTM** symbols are imported into the Control Expert Data editor. ● The **DTM Name** is displayed in the **I/O Scanning** configuration tab in black. This indicates that the PRM configuration is synchronized. ● The **Property** box closes.

Introduction to Configuring a BMX PRA 0100 from Control Expert

At a Glance

Control Expert allows configuration of BMX PRA 0100 modules through the Ethernet **I/O scanning** tab. The PRA device configuration is done in a **second** instance of Control Expert.

Configuring a PRA

The following procedure configures a PRA device:

Step	Action	Results
1	Open the Ethernet network configuration screen.	
2	In the **Module Utilities** zone select **YES** for **IO Scanning**.	
3	Select the **I/O Scanning** tab.	**I/O Scanning** screen opens.
4	Enter, on a free line, the IP address for the connection you want to use to communicate with the **PRA**. **NOTE:** The IP address in the **I/O Scanning** table must be the same as the IP address of the **PRA** device.	
5	Enter **RD length** and **WR length** on the same line.	
6	Validate the **I/O Scanning** screen.	
7	Click on the **...** button (that is next to **Device Name** cell on the same line).	The **Property** box *(see page 168)* opens.
8	Select **PRA** in the **Device Type** drop-down menu.	
9	Enter a **Device Name** (following the naming rules *(see page 171)*).	

Step	Action	Results
10	You have 2 choices: 1. If you want to **now** configure a **PRA**, click on the **Launch PRA** button. Click on **Yes** in the "*Confirm device name and type*" Message Box and go to Step 11. 2. If you want to configure a **PRA later**, click on the **OK** button. Click on **Yes** in the "*Confirm device name and type*" Message Box. **NOTE:** The Device Name becomes red in the **I/O Scanning** table. This indicates that a PRA has not been configured for the table line that contains the **Device Name** To configure a **PRA** later: ❍ Carry out Step 7. ❍ Click on the **Launch PRA** button. **NOTE:** While the second **PRA** instance of Control Expert is running no changes can be made to the Ethernet Editor in the first (master) instance of Control Expert.	Results for both 1. and 2. are: ● A Control Expert message box opens: "*The device name and device type won't be modifiable. Do you want to confirm the device name and device type?*" ● The **Device Type** and **Device Name** are verified and saved. ● The **Property** box closes.
11	After the second instance of Control Expert opens: ● **File menu → Open** ● Change the file type to **.XEF** ● Open the **PRA** application template, **PRA_Template.XEF**	

Step	Action	Results
12	When your PRA application is configured: • If desired, you can build the **PRA** application now. • Save the application. **NOTE:** The **Save As** function is not available. To copy your PRA application use the Export or Save Archive function. • Close this instance of Control Expert. **NOTE:** You are asked if you want to save the PRA application in the master application *.stu file. If you select **No**, all changes are lost. **NOTE:** Later you can build your **PRA** application by carrying out Step 7. Because the **PRA** application is saved (embedded) in the master application *.stu file, it is opened. You can then build the **PRA** application. **NOTE:** If there is no **PRA** application in the master *.stu, an empty application is opened (as happens the first time the **Launch PRA** button is used in the **Property** box)	
13	Build your Control Expert application.	
14	STOP the PLC.	
15	Transfer: • Control Expert application to the PLC • **PRA** configuration to the PRA device **NOTE:** There are no imported variables, the user must ensure the synchronization of the data exchange.	
16	RUN your application in the PLC.	

NOTE: When the second (**PRA**) instance of Control Expert is closed, there is no indication if the **PRA** application has been built or not.

Copy an Existing PRA Application

This following procedure copies an existing **PRA** application:

Step	Action
1	From the Control Expert **I/O Scanning** table using the **...** button, open an existing **PRA** application.
2	In the second Control Expert instance, save the existing **PRA** application with a new name as a *.sta or .xef file.
3	Close this second Control Expert instance.
4	In the Control Expert **I/O Scanning** table create a new **PRA** application on a new line.
5	Import or Open the *.xef or *.sta file previously saved.
6	If desired, build the new **PRA** application and transfer it to the **PRA** device.
7	Close the second Control Expert instance.

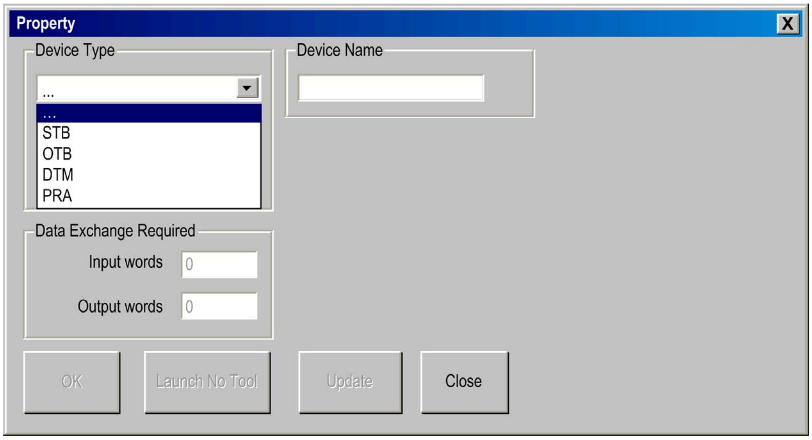
Property Box

At Glance

The **Property** box is the link between Control Expert and a device configuration tool. It is used to select and name a device and to launch the configuration tool for the device.

Property Box

The following illustration is the **Property** box before selecting the **Device Type**.

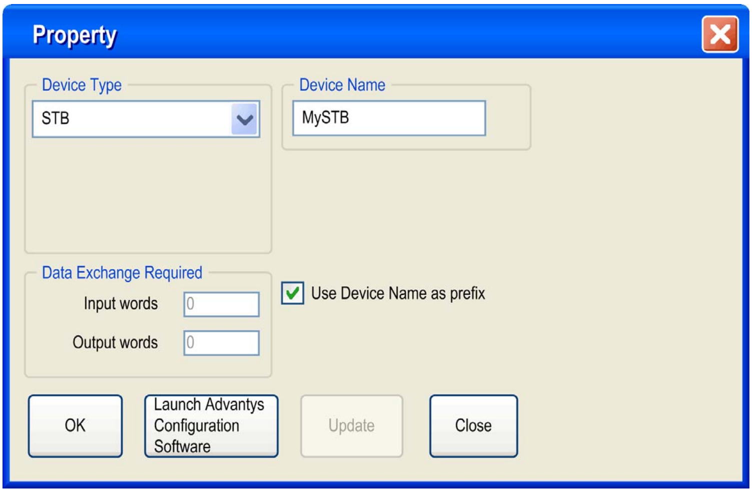


For details on how top use the **Property** box, refer to:
- Advantys *(see page 169)*
- DTM *(see page 172)*
- BMX PRA 0100 *(see page 170)*

Property Box for Advantys

This **Property** box allows you to choose the name and type of Advantys island to be configured using the Advantys Configuration Software (ACS).

The following illustration is the **Property** box for Advantys *after* **Device Type** and **Device Name** entered:



Property Box for Advantys Elements

The elements of the Advantys **Property** box are:

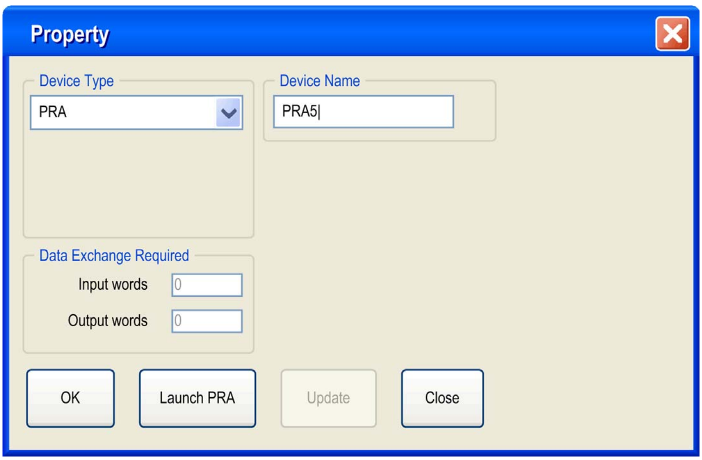
Element	Description
Device Type	Selection of a STB or OTB device is made from this drop-down list. After the first validation (using the **OK** or **Launch Advantys** button) the **Device Type** cannot be changed.
Device Name	The **Device Name** *(see page 171)* is used as a prefix to all variables created for an Advantys island in ACS. This allows unique variables for duplicated islands. After validation (using the **OK** or **Launch Advantys** button) the **Device Name** cannot be changed.
Data Exchange Required	These are the minimum number of words necessary for communication between the Control Expert module and the Advantys island. These values cannot be changed via the **Property** box. **NOTE:** Increasing the RD/WR lengths via the **I/O Scanning** tab leaves enough data exchange words for the future expansion of your Advantys island. Expanding an island that does not use the last line in the **I/O Scanning** table requires changing the values for all the lines below the line that needs the additional exchange words.

Element	Description
Use Device Name as prefix	If this checkbox is unchecked, the user is in charge of giving unique names to the variables and symbols in **all** Advantys islands. This checkbox is only available for ASC V5.5 or higher. For versions less than 5.5 the Device Name is automatically added to all variables and symbols in all Advantys islands.
OK	This button is only available after entering the **Device Type** and **Device Name**. When clicked, the **Device Type** and **Device Name** are checked to see if they are valid. If there is a problem, a message box opens explaining the why they were not valid. **OK** is only available during the first use of the **Property** box for a new island.
Launch Advantys Configuration Software	This button is only available if both: ● The **Device Type** and **Device Name** have been entered ● ACS is installed This button does two things: ● It carries out the action of the **OK** button ● If there is no problem during validation, it launches ACS
Update	When clicked, the ACS modifications are synchronized with your Control Expert application (after these modifications have been validated in ACS). It also imports and updates all ACS symbols and variables into the Control Expert variable manager. **NOTE:** All variables modified in ACS are deleted and rewritten in the Control Expert Data Editor. But they are not updated in the program.
Close	This button closes the **Property** box without saving anything.

Property Box for BMX PRA 0100

This **Property** box allows you to choose the name for the PRA module to be configured.

The following illustration is the **Property** box for the PRA *after* **Device Name** validation:



Property Box for PRA Elements

The elements of the PRA **Property** box are:

Element	Description
Device Type	Selection of the PRA device is made from this drop-down list. After the first validation (using the **OK** or **Launch PRA** button) the **Device Type** cannot be changed.
Device Name	The **Device Name** *(see page 171)* is the name of PRA application.
Data Exchange Required	This is not used when configuring a PRA device.
OK	This button is only available after entering the **Device Type** and **Device Name**. When clicked, the **Device Type** and **Device Name** are checked to see if they are valid. If there is a problem, a message box opens explaining why they are not valid. **OK** is only available during the first use of the **Property** box for a new PRA configuration.
Launch PRA	This button is only available if the **Device Type** and **Device Name** has been entered. This button does two things: ● It carries out the action of the **OK** button ● If there is no problem during validation, it launches another instance of Control Expert, which is used to do the actual configuration of the PRA.
Close	This button closes the **Property** box without saving anything.

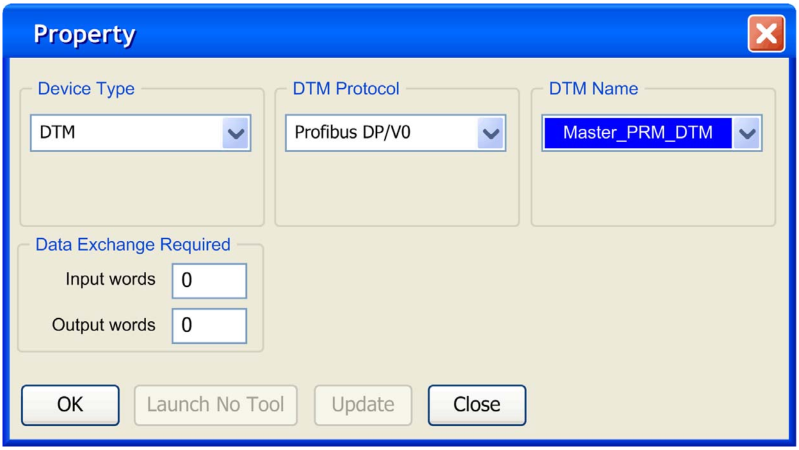
Valid Name

A valid **Device Name** for a configuration:
- Does not already exist in the application
- Is not a empty name
- Starts with a letter
- Has a maximum of 8 characters
- Only ASCII characters, not Unicode characters
- Has no spaces
- Follows the Windows file naming conventions: no slashes, question marks, etc.
- Follows Control Expert variable naming conventions

Property Box for a PRM Master DTM

This **Property** box allows you to choose the type and protocol for a **PRM Master DTM**:

The following illustration is the **Property** box *after* selecting the **Device Type**, **DTM Protocol** and **DTM Name**:



Property Box PRM Master DTM Elements

The elements of the DTM **Property** box are:

Element	Description
Device Type	Selection of DTM device type is made from this drop-down list.
DTM Protocol	Select the protocol to be used from this drop-down list. This list contains the DTM protocols of all the DTMs in the DTM Browser that can be linked with I/O Scanning.
Device Name	Select a **PRM Master DTM** from this drop-down list.This list uses the DTM Browser Alias names. This list contains all the **PRM Master DTMs** in the DTM Browser that support the selected DTM Protocol. To validate the choices, click on the **OK** button.
Data Exchange Required	These are the minimum number of words necessary for communication between Control Expert and the **PRM Master DTMs**. These values cannot be changed via the **Property** box. **NOTE:** Increasing the RD/WR lengths via the **I/O Scanning** tab leaves enough data exchange words for the future expansion of your DTM topology tree. Expanding a tree that does not use the last line in the **I/O Scanning** table requires changing the values for all the lines below the line that needs the additional exchange words.

Element	Description
OK	The **OK** button is only available after selecting the **Device Type**, **DTM Protocol** and **DTM Name**. When clicked, the **DTM Protocol** and **DTM Name** are checked to see if they are valid. If there is a problem, a message box opens explaining the why they were not valid. The **OK** button is only available during the first use of the **Property** box for a new **PRM Master DTM**.
Launch No Tool	This button is never available for **PRM Master DTMs**.
Update	Use the **Update** button after validating or changing the configuration of the linked **PRM Master DTM**. Refer to Update I/O Scanning for a PRM Master DTM *(see page 163)*.
Close	The **Close** button closes the **Property** box without saving anything.

Saving an Advantys Configuration in an Control Application

At a Glance

ACS saves an island configuration in an *.isl file. To add the island to an application, it is necessary for Control Expert to know the location of the island configuration information.

Saving the Configuration

The recommended way to save your island configuration information is to save your Control Expert application as a *.stu or *.sta file. The *.isl file is automatically included in these files.

Uploading or Importing

There are 2 situations where the information contained in the *.isl file is not available:

1. Uploading the application running in the PLC
2. Importing an *.xef file

In these 2 cases, if ACS is launched from the **Property** box *(see page 168)*, it automatically tries to open the latest **Device Name**.isl file the Control Expert **General Path => Project Path** directory:

- If the same PC is used for the import (upload) and export (download) and the Control Expert **Project Path** has not changed, the island configuration is synchronized with ACS.
- If the same PC is not used for the import (upload) and export (download) or if the Control Expert Project Path has changed, either:
  - Create a new island
  - Use the **File** menu **=> Copy Island Contents** function

    NOTE: The new **Device Name**.isl file is copied to the **Project Path** directory.

Managed Variables

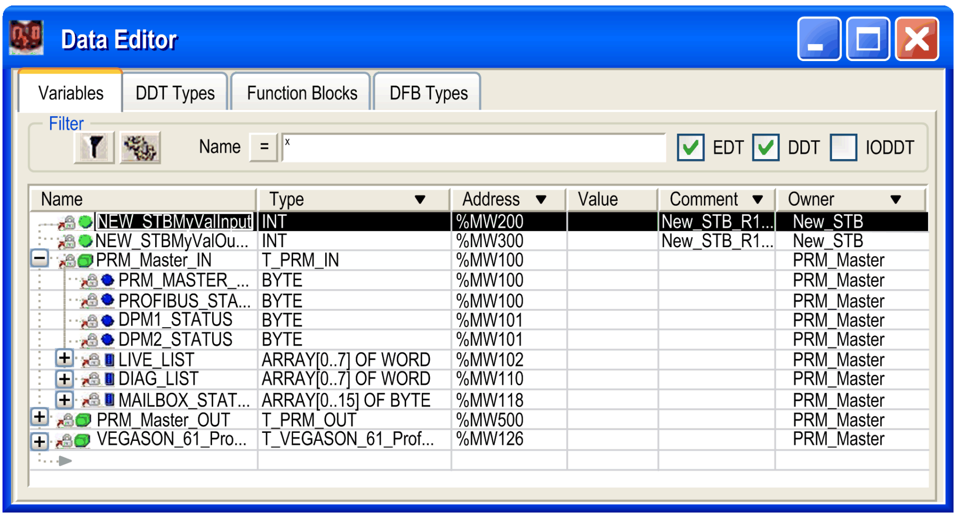
At a Glance

Variables of devices that are linked to Control Expert through **I/O Scanning** or **DTM** are *managed variables*. They are created by the device configuration tool or by the DTM and are imported into Control Expert. They are named as a concatenation of the Property box *(see page 168)* **Device Name** + device symbol name.

Managed Variables in the Data Editor

Advantys symbols become managed variables when imported into Control Expert. An Advantys managed variable name is a concatenation of:
the Advantys island name + Advantys symbol name.

This **Data Editor** illustration shows managed variables with their device name **prefixes** and their **Owner** attributes:



Managed variables follow the usual rules for Control Expert and ACS or DTM naming.

The optional **Owner** column lists the owner attribute of the managed variables. This allows you to filter the variables according to their **Device Name**.

The Control Expert managed variables are locked and cannot be modified through the **Data Editor**. You have to use the device configuration tool (ACS or the DTM) to modify these variables.

Importing Managed Variables from a Device (Advantys, DTM)

Using the **Update** button on the Property box *(see page 168)* imports the device Symbols into the Data Editor as Control Expert managed variables.

In the case of a conflict between an device Symbol and an existing variable in the Data Editor:
- If the Control Expert variable *is not* managed, a message box allows you to replace this variable with the managed variable coming from the ACS- or DTM-controlled device).
- If the Control Expert variable *is already* managed, the update is cancelled.

For an already managed variable, there are 2 options, either:
1. Use the device configuration tool (ACS or the DTM) to rename the variable.
2. Delete the old managed variable using the tool that manages the variable, then use the tool to perform an Update.

After performing one of these options, use the **Update** button again on the device being updated to complete the import without a conflict.

Permanent Deletion of a Managed Variable

Managed variables cannot be deleted directly from the Data Editor.

Removing a managed variable from a configuration must be done from the tool (ACS or the DTM) that manages the device (either delete the device using the DTM or delete the Symbol using the ACS).

NOTE: During an **Update**, all managed variables are deleted and recreated during synchronization between Control Expert and the device.

Partial Import of a Managed Variable

Starting with Unity Pro V5.0, the managed variables become *unmanaged* during a partial import from an .XSY file. This allows deletion of the variables if the linked device is not also imported.

NOTE: Unity Pro is the former name of Control Expert for version 13.1 or earlier.

After importing variables from an .XSY file, an **Update** is needed to resynchronize the managed variables linked to a device. During this **Update**, a conflict box appears to allow validation of the replacement managed variables.

I/O Scanner Concepts

I/O Scan List

An I/O scan list is a configuration table that identifies the targets with which repetitive communication is authorized. While the CPU is running, the Ethernet module transfers data to and from the CPU's registers in accordance with I/O scan list.

Connections

The I/O Scanner opens one connection for each entry in the I/O scanner table. If several table entries have the same IP address, multiple connections are opened.

I/O Scanner Limits

The I/O Scanner on the BMX NOE 01x0 modules are limited to:
- maximum number of devices: 64
- maximum number of input words: 2048
- maximum number of output words: 2048

Using the I/O Scanner across a Network Router

The I/O Scanner can scan devices through an IP router with a TTL (time to live) of 32.

Scanning Period

Remote input/outputs are scanned periodically depending on the application requirements. A scanning period is defined for each device through configuration, according to the update speed.

NOTE: Keep in mind:
- The lower the scanning period, the faster the input/outputs are updated. However, this speed increases the network load.
- %SW8 and %SW9 do not stop remote station scanning, but inhibit the copying of I/Os to and from the application variables.

Scanning Period

At a Glance

Remote input/outputs are scanned periodically depending on the application requirements.

A scanning period is defined for each device through configuration, according to the update speed.

NOTE: The lower the scanning period, the faster the input/outputs are updated. However, this speed increases the network load.

NOTE: %SW8 and %SW9 do not stop remote station scanning, but inhibit the copying of I/Os to and from the application memory.

NOTE: If you configure a scanning period of 0, the request is sent immediately after the response to the previous request is received.

NOTE: The entry in the **Repetitive rate step** field should be a multiple of 10. Any other number will not work correctly.

Configuration of Parameters Linked to the I/O Scanner Utility

Parameter Table

The BMX NOE 01x0 modules have configuration parameters linked to the I/O Scanner:

Parameters		BMX NOE 01x0
master %MW zones	read ref.	X
	write ref.	X
Repetitive rate (ms)		in ms
RD Master Object		automatic
RD Slave Index		X
RD length		X
Last value (input)		Hold last/Set to 0 (fallback)
WR Master Object		automatic
WR Slave Index		X
WR length		X
Health timeout		X
Legend:		
X: available		

Configuration of General Parameters for the I/O Scanner

Introduction

Configure the general parameters (**Master %MW zones**) on the I/O Scanner tab *(see page 149)* to periodically read or write remote inputs/outputs on the Ethernet network without specific programming.

Master %MW Zones

In the **Master %MW zones** zone you can define the ranges of internal words of the application memory (%MW) specific to the read and write zones. To do this, you must complete:

- **Read Ref.**: This read zone lists the starting address in the table of internal words for reading inputs.
- **Write Ref.**: This write zone lists the starting address in the table of internal words for writing outputs.

For the BMX NOE 01x0, the length of tables has a maximum exchange capacity of:

- **read zone**: 2048 words
- **write zone**: 2048 words

NOTE: The tables must not overlap and an overrun check is made on global validation.

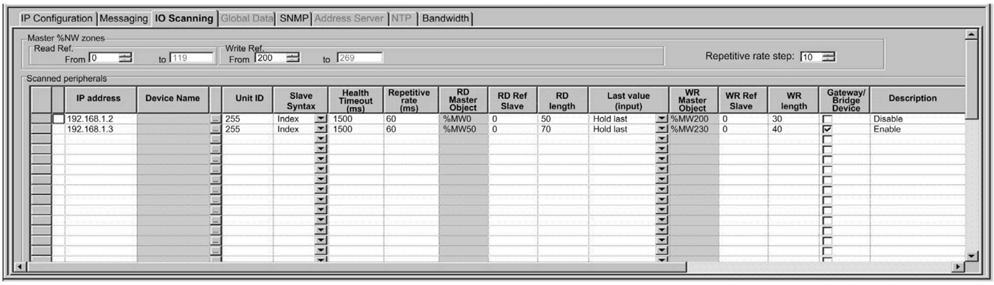
I/O Scanner: Health Block Bits

Health Block Bits

The health block is the block of 4 words (%IW) topological objects that give the health status of the I/O scan.

Each health block bit corresponds to an entry in the I/O Scanner table. Each entry represents one logical device. If a health bit is switched to 0, the health status on the corresponding device is bad.

The following table shows the corresponding health bit for each device based on its table entry:



The bits in the health block (mapped to %IW topological objects) correspond to the different data types:

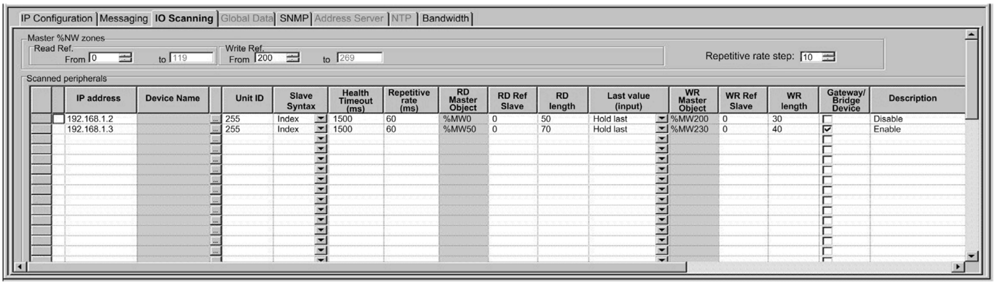
Table Entry	Health Bits	Comment
1	%IWr.m.c.1.0	r: rack number m: slot number located by the module c: module channel number (always 0 for BMX NOE 01x0) **Example:** If a BMX NOE 01x0 module is configured in rack 0, slot 3, then the health bit for table entry 2 is stored in $IW0.3.0.1.1.
2	%IWr.m.c.1.1	
3	%IWr.m.c.1.2	
. . .		
17	%IWr.m.c.2.0	
18	%IWr.m.c.2.1	
19	%IWr.m.c.2.2	
. . .		
33	%IWr.m.c.3.0	
34	%IWr.m.c.3.1	
35	%IWr.m.c.3.2	
. . .		
49	%IWr.m.c.4.0	
50	%IWr.m.c.4.1	
51	%IWr.m.c.4.2	
. . .		
64	%IWr.m.c.4.15	

I/O Scanner: Device Control Block

Device Control Block

The device control block is a block of 4 word (%QW) topological objects that enable and disable the I/O scanner for each table entry.

Each device control block bit corresponds to an entry in the I/O Scanner table. Each entry represents one logical device:



To disable an individual scanner device:

Step	Action
1	Create element variables with predefined IODDT type T_COM_ETH_BMX.
2	Set the bit to 1 to disable it from within the application or from an animation table.

WARNING

UNEXPECTED EQUIPMENT BEHAVIOR

Do not create I/O Scanner entries with both read and write lengths set to0.

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

NOTE: A value of 1 in the device control block bit disables the I/O Scanner table entry. A value of 0 in the device control block bit enables the entry. Each entry corresponds to a particular logical device in the IO Scanning configuration table.

Each I/O Scanner table entry can be disabled by setting the corresponding device control bit to 1. The following table shows the mapping between the I/O Scanner table entries and the device control bits stored in %QW topological objects:

Table Entry	Device Control Bits	Comment
1	%QWr.m.c.0.0	r: rack number m: slot number located by the module c: module channel number (always 0 for the BMX NOE 01x0) The device control block bits are mapped to I/O Scanner entries (%QWrack.slot.channel.word(0-3). **Example:** Table entry 2 can be disabled by setting %QWr.m.0.0.1 to 1. (%QWr.m.0.0.1 is associated with DISABLE_IO_2 in the predefined IODDT type T_COM_ETH_BMX for the BMX NOE 01x0 modules.)
2	%QWr.m.c.0.1	
3	%QWr.m.c.0.2	
. . .		
17	%QWr.m.c.1.0	
18	%QWr.m.c.1.1	
19	%QWr.m.c.1.2	
. . .		
33	%QWr.m.c.2.0	
34	%QWr.m.c.2.1	
35	%QWr.m.c.2.2	
. . .		
49	%QWr.m.c.3.0	
50	%QWr.m.c.3.1	
51	%QWr.m.c.3.2	
. . .		
64	%QWr.m.c.3.15	

NOTE: M340 devices use topological addresses to represent device control block bits. Premium and Quantum PLCs have data structures that differ from this.

Section 10.6
Global Data Configuration Parameters

About this Section

This section introduces the configuration parameters linked to Global Data.

What Is in This Section?

This section contains the following topics:

Topic	Page
The Global Data Configuration Tab	186
Configuration of Global Data Variables	187
Configuration of General Parameters for Global Data	189
Configuration of Parameters Linked to Global Data	190

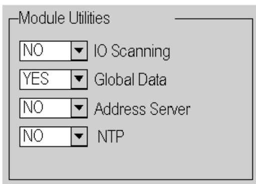
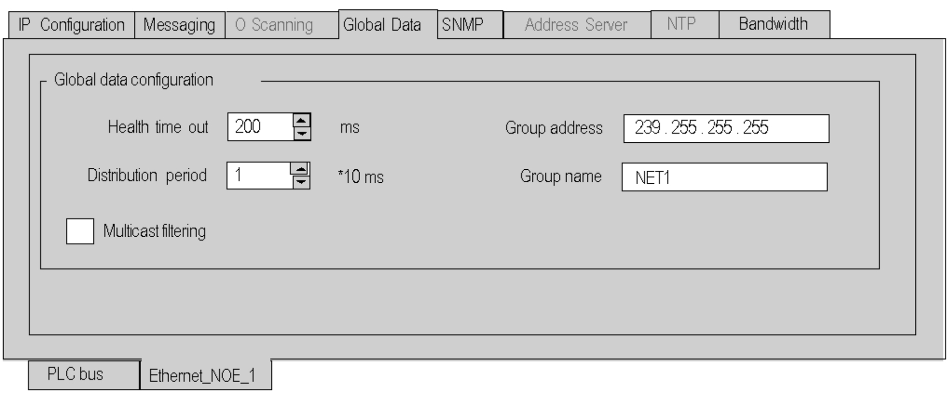
The Global Data Configuration Tab

Introduction

In order to use the BMX NOE 01x0 with global data, it is necessary to set the configuration parameters.

Global Data Tab

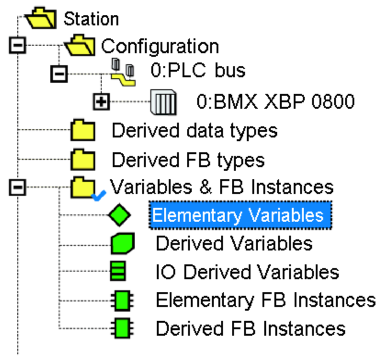
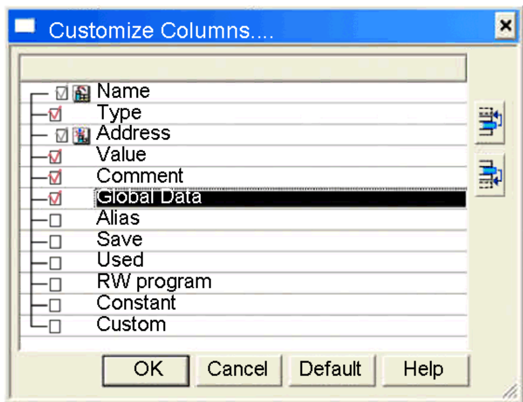
To access the configuration parameters on the global data tab:

Step	Action
1	Go to the module configuration screen *(see page 136)*.
2	In the **Module Utilities** field, select **Yes** in the **Global Data** menu. 
3	Select the **Global Data** tab. 
4	In accordance with your system and application needs, elect and assign the global data configuration parameters *(see page 189)*.
5	Configure the global data variables *(see page 187)*.

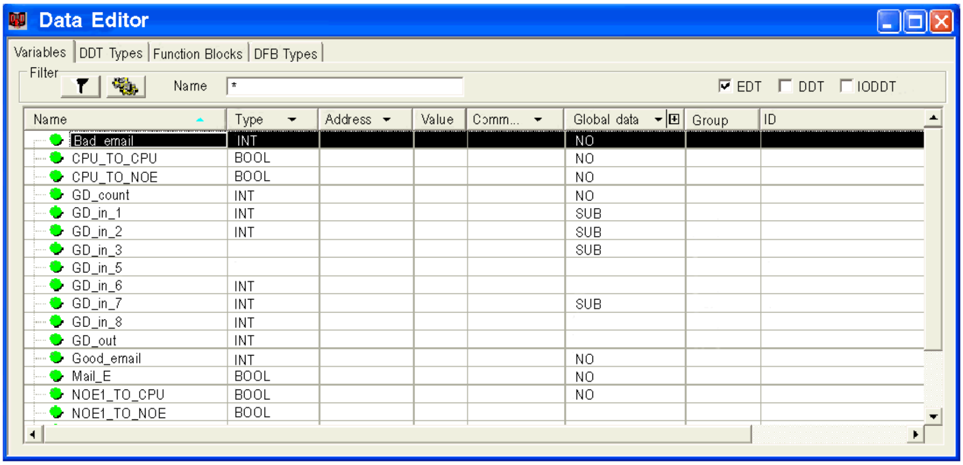
Configuration of Global Data Variables

Configure Variables

Before you configure Global Data variables, you must go to the Control Expert variable editor:

Step	Action	Illustration and Comments
1	In the **Project Browser**, go to the **Structural view** and double-click **Variables and FB Instances** (see figure at right). **Result**: The **Data Editor** table (in step 3) appears. By default, the **Global data** column does not yet appear.	
2	To make the **Global data** column visible, right-click in the first row on the **Data Editor** screen, and scroll down to **Customize Columns**. The **Customize Columns** window appears (see figure at right). Select the **Global Data** check box, and press **OK**.	

<table>
<tr><th>Step</th><th>Action</th><th>Illustration and Comments</th></tr>
<tr><td>3</td><td colspan="2">The Data Editor screen appears:



</td></tr>
<tr><td>4</td><td>In the Global data column, there are three choices for variable configuration: NO, PUB, and SUB.</td><td>Definition of variable types:
● NO: neither published nor subscribed
● PUB: published
● SUB: subscribed</td></tr>
<tr><td>5</td><td colspan="2">Give the Global Data group a name in the Group field.</td></tr>
<tr><td>6</td><td colspan="2">Fill in the DataID field: identifier of a remote stations in a distribution group.</td></tr>
</table>

NOTE: IODDT data structures for M340 devices are different from the data structures for Premium and Quantum PLCs.

Configuration of General Parameters for Global Data

General Parameters

The following parameters can be set in the **Global Data configuration** zone on the Global Data configuration screen *(see page 186)*:

Parameter	Description
Health time out	The **Health time out** zone is used to adjust the health "time-out" value. The value is from 50 to 15 000 ms in increments of 50 ms. An associated status bit (%IW topological objects between %IWr.m.c.5 and %IWr.m.c.8) is linked to each Global Data item and is used to monitor whether the data has been published and received by the end of the time indicated in this window. If yes, the value is 1, otherwise the bit is set to 0.
Group address	The **Group address** indicates the multicast IP address (class D) of the distribution group to which the station belongs: ● minimum value: 224.0.0.0 ● maximum value (default): 239.255.255.255
Distribution period	The **Distribution period** zone is used to select the distribution period of the publication. The publication is time-based and is not synchronized with the PLC master task. The distribution value is multiplied by 10, so the available parameter range of 1 to 1500 represents distribution periods between 10 and 15 000 ms (in increments of 10). **NOTE:** With a small distribution period, it is needed to check if the switch connected to the BMX NOE module is able to manage a such amount of frames. Otherwise the switch will send back broadcast frames and the BMX NOE will go in error to cut the data flow.
Group name	The **Group name** is defined in the Control Expert data editor. The name associates a variable from the variable editor with a particular module.
Multicast filtering	Filtering can reduce data flow on large networks. Multicast filtering requires the use of switches that support this function (GMRP IEEE 802.1D protocol). The status of the check box indicates: ● checked: on ● unchecked: off

Configuration of Parameters Linked to Global Data

Introduction

In addition to the general Global Data parameters *(see page 189)*, you must also configure the Global Data variables in the data editor.

Publish/Subscribe Variables

Association between Global Data variables (network variables) and application variables is carried out in the Control Expert variable editor.

Each application variable published or subscribed (**Global Data** field) in a Distribution Group (**Group** field) is linked to a Global Data item (network variable).

Each Global Data item has a unique identification (**Data ID**) within a Distribution Group. The rank of the status bit in the Health Bit zone of Global Data corresponds to the identifier (**Data ID**) of the Global Data.

Global Data Properties

Type	Value
max. number of publications	1
size of a variable at publication	1 to 512 words
size of a variable at subscription	1 to 512 words
max. number of subscriptions	64 (see note)
max. variable size at subscription	total of 2K words
Note: The combined total number of variables is 64. Therefore, when 1 publication variable is configured only 63 subscription variables are available.	

Section 10.7
SNMP Configuration Parameters

About this Section

This section introduces the configuration parameters linked to SNMP.

What Is in This Section?

This section contains the following topics:

Topic	Page
Configuring SNMP as an Agent	192
SNMP Configuration Parameters	194

Configuring SNMP as an Agent

Introduction

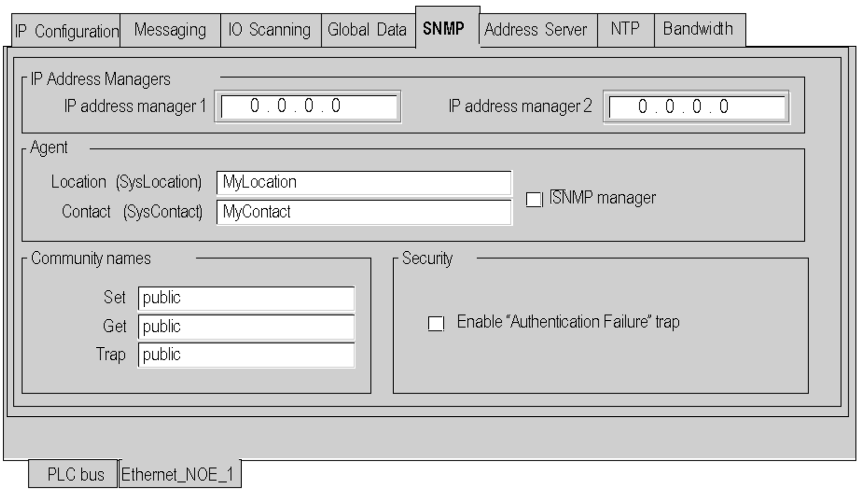
To use the BMX NOE 01x0 or BMX P34 20x0 module as an SNMP agent, it is necessary to adjust the SNMP configuration parameters.

The SNMP Tab

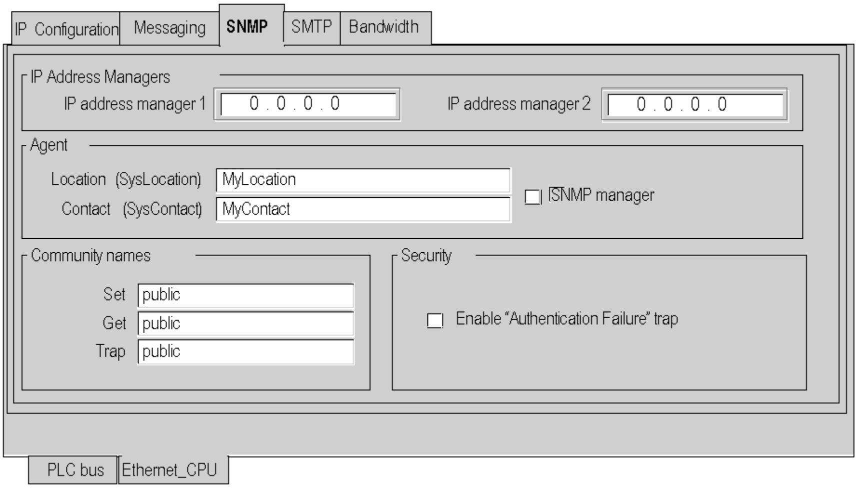
The following procedure shows how to access the SNMP configuration page from the index page:

Step	Action
1	Access the module configuration screen *(see page 136)*.
2	Select the SNMP tab (see illustration below).

NOE configuration screen:



CPU configuration screen:



Configuring SNMP

The following procedure gives the configuration principle for SNMP:

Step	Action
1	Enter the **IP Address Managers** addresses: ● IP address manager 1 ● IP address manager 2
2	Fill in the **Agent** fields: ● Location (SysLocation) ● Contact (SysLocation) Or alternatively check the **SNMP manager** box to indicate that the information will be completed by the SNMP manager.
3	If you want to set access rights, fill in the **Community names**: ● Set ● Get ● Trap
Note: Elsewhere in this guide are discussions of: ● SNMP configuration parameters *(see page 194)* ● SNMP message types *(see page 94)*	

SNMP Configuration Parameters

Introduction

Parameters on the SNMP configuration tab *(see page 192)* are divided into four categories:

- the IP addresses of SNMP manager devices
- SNMP agents
- the community names
- security

NOTE: Only 7-bit ASCII characters can be used in the character string entry fields.

IP Address Managers

This zone allows you to complete the IP addresses of the SNMP managers. The modules authorize a maximum of two managers.

These addresses are used during possible transmission of events (TRAP). The transmission of supervised data is detailed at the topic SNMP *(see page 91)*.

Agent

This zone allows the localization and identification of an agent from the SNMP manager.

It comprises two fields:

- The **Location (SysLocation)** field: indicates the physical location of the device (32 characters maximum).
- The **Contact (SysLocation)** field: indicates the person to contact for device management and the method of contact (strings of 32 characters maximum).
- If you prefer to have this information assigned by an SNMP Manager tool for network management, check the **SNMP Manager** box.

Community Name

This zone is used to define community names for the Set, Get and Trap utilities. It comprises three fields:

- The **Set** field defines the community name for the Set utility (strings of 16 characters maximum). The default value of the field is *Public*.
- The **Get** field defines the community name for the Get utility (strings of 16 characters maximum). The default value of the field is *Public*.
- The **Trap** field defines the community name for the Trap utility (strings of 16 characters maximum). The default value of the field is *Public*.

The purpose of these fields is to define the access rights for the MIB objects of the SNMP agent (local module) in relation to requests sent by the manager.

Example: If the manager sends a SetRequest request with the community name *Test* and the module has the community name *Public*, the request is not executed.

Security

CAUTION
UNEXPECTED NETWORK BEHAVIOR - SNMP PARAMETERS RESET
The SNMP manager is able to modify the value of certain configurable parameters(Enabling "Authentication failure," Location, Contact. etc.).
the **SNMP Manager** box is not checked and there is a cold start, warm restart, or application download, the initially configured values are restored.
Failure to follow these instructions can result in injury or equipment damage.

This zone contains the **Enable "Authentication Failure" trap** check box.

Checking this box allows you to validate the transmission of an authentication failure event (TRAP) from the SNMP agent to the configured manager.

In this way, the agent warns the manager that the request has been refused following an identification error (community name configured in the manager is different from the one configured in the agent).

Section 10.8
Address Server Configuration Parameters

Address Server

Introduction

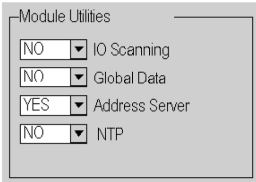
The BMX NOE 01x0 modules use the **Address Server** tab to configure the correspondence table between the MAC addresses or the Name (device name) and the IP addresses of the module if the latter is configured as a DHCP/BOOTP server.

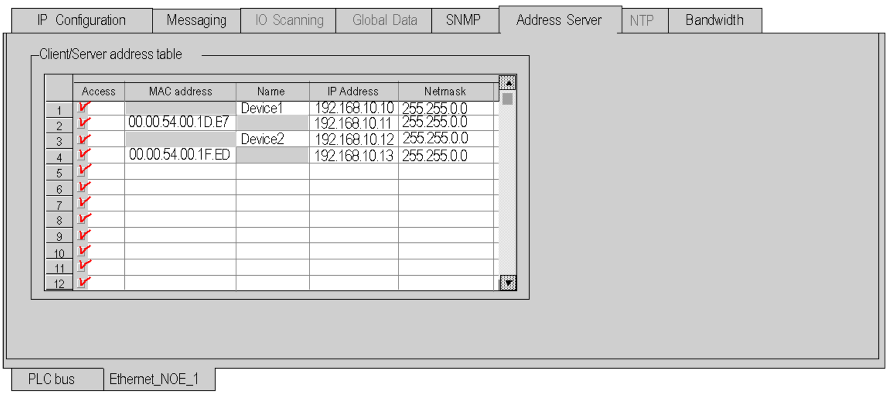
Address management is performed in a dedicated IP address server. To avoid the task of managing each device’s IP address individually, BOOTP/DHCP automatically provides devices with IP parameters. The address server employs DHCP (an extension of BOOTP), to automatically assign IP configuration parameters to devices.

This function is useful when replacing a failed remote device (for example, a faulty Momentum module).

Address Server Tab

The following procedure describes access the **Address Server** page from the index page:

Step	Action
1	Access the module configuration screen *(see page 136)*.
2	In the **Module Utilities** field, select **Yes** in the **Address Server** menu. 
3	Select the **Address Server** tab. (See illustration below.)



Address Server Configuration

The **Address Server** tab has only one zone, the **Client/Server address table**. This table:

- lists by MAC address or name (16 ASCII characters) the remote stations that need the DHCP server to start up
- provides a correspondence between the MAC address or name and the IP address of the remote station, subnetwork mask, and gateway

To configure the address server, fill in the fields of the table for each device that requires the module as DHCP server:

- MAC Address or Name
- IP Address
- Netmask

Section 10.9
Bandwidth Checking

Bandwidth Checking Configuration

Introduction

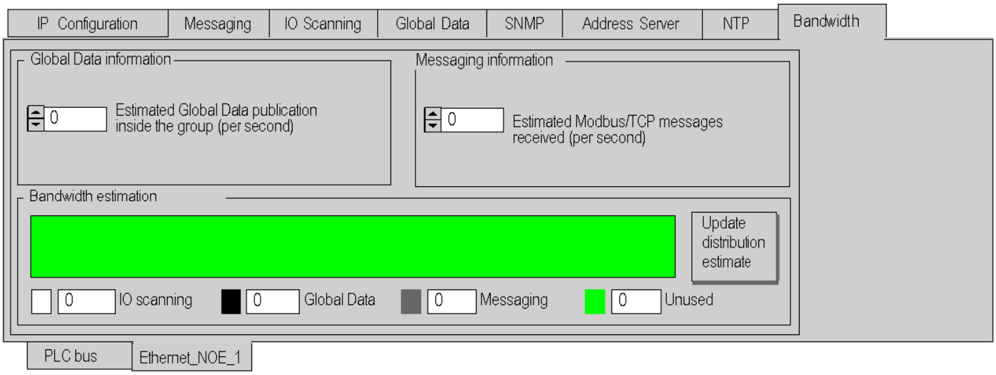
The BMX NOE 01x0 communication modules and BMX P34 20x0 CPUs provide a service for checking the implemented bandwidth.

Bandwidth Tab

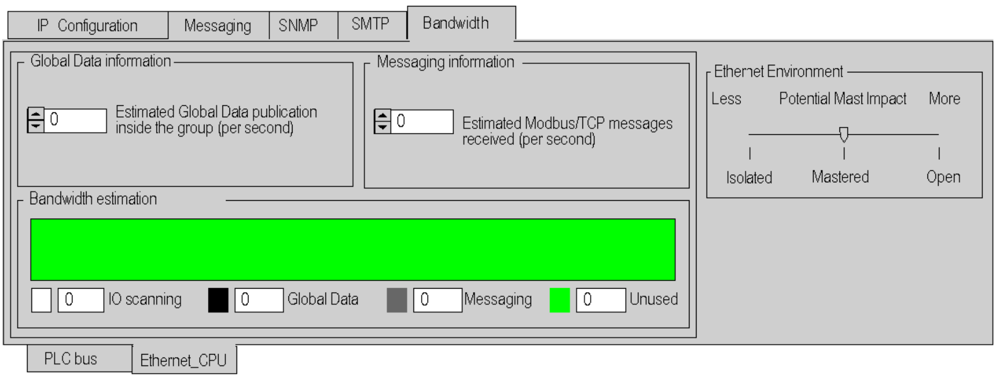
To access the **Bandwidth** page from the index page:

Step	Action	Comment
1	Configure the I/O scanning tab *(see page 149)*.	This step is required for the BMX NOE 01x0 modules.
2	Access the module configuration screen *(see page 136)*.	
3	Select the **Bandwidth** tab.	See the illustration below.
4	Select the appropriate **Ethernet Environment** zone for the BMX P34 20x0 CPUs.	These parameters are not available (or visible) for the BMX NOE 01x0 modules.
5	Enter the estimate for the **Global Data information** zone.	This is an estimate of the number of publication periods. The value entered must be the estimated average publication frequency of the distribution (local and remote) group stations.
6	Enter the estimate for the **Messaging information** zone.	This is an estimate of the number of transactions per second.
7	If the module overflows, the message "The maximum number of messages for the Extended Ethernet network must not exceed 2000 messages" appears.	

NOE configuration screen:



CPU configuration screen:



NOTE: An explanation of **Isolated**, **Mastered**, and **Open** *(see page 102)* are discussed earlier in this manual.

Section 10.10
Electronic Mail Notification Service Configuration Parameters

The SMTP Tab

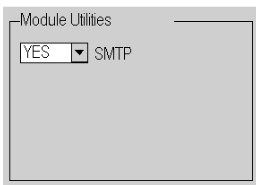
Introduction

In order to use the embedded Ethernet feature on the BMX P34 20x0 processors with SMTP, you must set the configuration parameters.

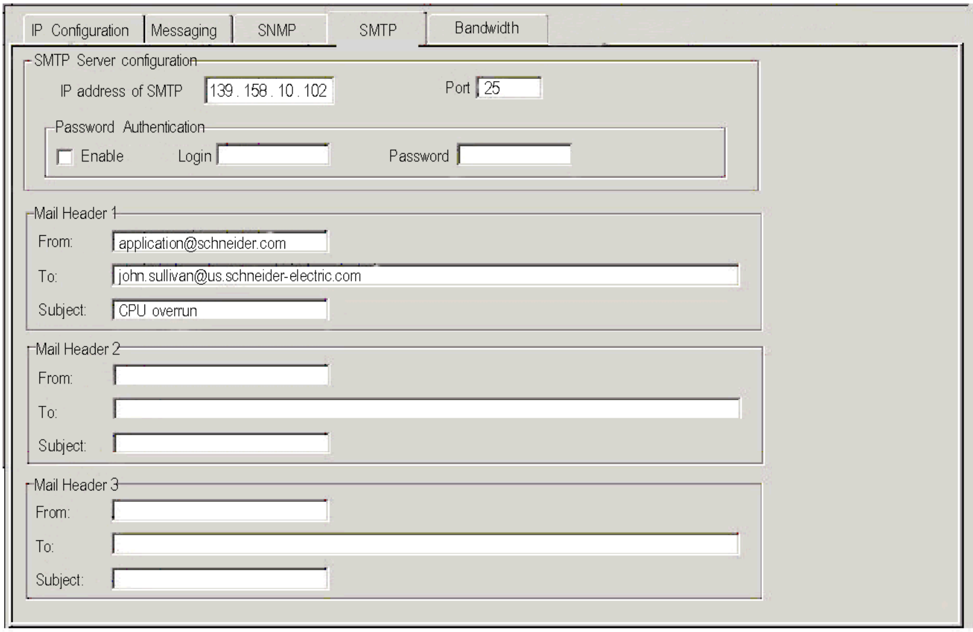
NOTE: The SMTP feature is available only on V2 modules or later.

SMTP Tab

The following procedure shows how to access the **SMTP** tab from the index page:

Step	Action
1	Access the module configuration screen.
2	In the **Module Utilities** field, select **Yes** in the **SMTP** menu. 
3	Select the **SMTP** tab. (See illustration below.)

The figure shows the SMTP dialog box for BMX P34 20x0 processors:



SMTP Configuration

Configure the following parameters:

Parameter	Description
IP Address of SMTP	Enter a valid IP address. This parameter identifies SMTP server.
Port	Default = 25 If needed, enter a new value to match the SMTP server's port.
Password Authentication	If security is needed, select the **Enable** check box. Enter values for: ● **Login** ❍ Any printable character allowed ❍ 12-character maximum ● **Password** ❍ Any printable character allowed ❍ 12-character maximum
3 Mail Headers	Each header must contain: 1. Sender's ID in the **From** field ❍ 32-character maximum (no spaces) 2. List of recipients in the **To** field ❍ Separate each email address with a comma. ❍ 128-character maximum 3. Fixed part of message in the **Subject** field[1] ❍ 32-character maximum
Subject field *(see page 118)* consists of two parts (1 024-character maximum): 1. **dynamic subject** 2. **body**	

Section 10.11
Time Synchronization Service Configuration Parameters

The NTP Tab

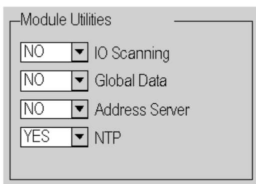
Introduction

In order to use the BMX NOE 01x0 modules with NTP, you must set the configuration parameters.

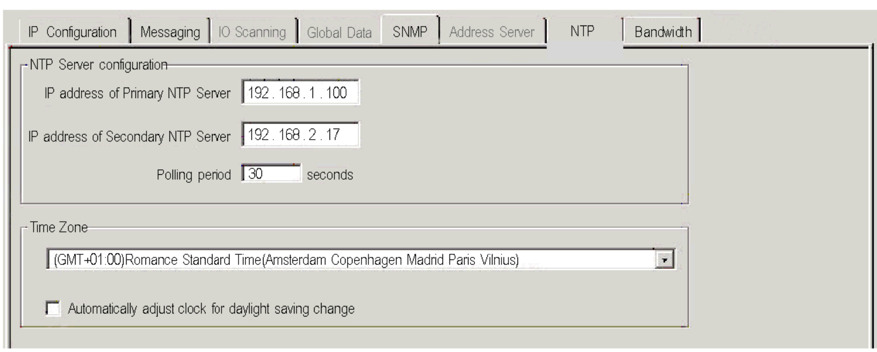
NOTE: The NTP feature is available only on V2 modules or later.

NTP Tab

The following procedure shows how to access the **NTP** tab from the index page:

Step	Action
1	Access the module configuration screen.
2	In the **Module Utilities** field, select **Yes** in the **NTP** menu. 
3	Select the **NTP** tab. (See illustration below.)

The figure shows the NTP dialog box for BMX NOE 01x0 modules:



Configuring NTP

Configure or change the following parameters on the NTP configuration page:

1. **IP address of Primary NTP Server**
   - Enter a valid IP address.
2. **IP address of Secondary NTP Server**
   - Enter a valid IP address.
3. **Polling period (in seconds)** (the time between time updates from the NTP server)
   Enter a value:
   - min = 1 sec
   - max = 120 sec
   - default = 5 sec
4. **Time Zone** (see the following table for available times zones)
   - Select from drop-down menu.
     Universal Time, Coordinated (GMT) = default
   - Custom time zone
5. **Automatically adjust clock for daylight saving change**
   - Parameter is selected by default (check mark appears) if daylight saving time is chosen.

Time Zones Available

Select a time zone the from drop-down menu.

Time Zone	Description	DST Available
Custom		Yes
(GMT-12:00)	Dateline Standard Time [Eniwetok Kwajalein]	No
(GMT-11:00)	Samoa Standard Time [Midway Is Samoa]	No
(GMT-10:00)	Hawaiian Standard Time [Hawaii Honolulu]	No
(GMT-09:00)	Alaskan Standard Time [Anchorage]	Yes
(GMT-08:00)	Pacific Standard Time [Los Angeles Tijuana]	Yes
(GMT-07:00)	Mexican Standard Time [Chihuahua La Paz Mazatlan]	Yes
(GMT-07:00)	Mountain Standard Time [Arizona Phoenix]	No
(GMT-07:00)	Mountain Standard Time [Denver]	Yes
(GMT-06:00)	Central Standard Time [Chicago]	Yes
(GMT-06:00)	Mexico Standard Time [Tegucigalpa]	No
(GMT-06:00)	Canada Central Standard Time [Saskatchewan Regina]	No
(GMT-06:00)	Central America Standard Time [Mexico_city]	Yes
(GMT-05:00)	SA Pacific Standard Time [Bogota Lima Quito]	No
(GMT-05:00)	Eastern Standard Time [New York]	Yes
(GMT-05:00)	Eastern Standard Time [Indiana (East)] [Indianapolis]	No

Time Zone	Description	DST Available
(GMT-04:00)	SA Western Standard Time [Caracas La Paz]	No
(GMT-04:00)	Pacific SA Standard Time [Santiago]	Yes
(GMT-03:30)	Newfoundland Standard Time [Newfoundland St Johns]	Yes
(GMT-03:00)	E. South America Standard Time [Brasilia Sao_Paulo]	Yes
(GMT-03:00)	SA Eastern Standard Time [Buenos Aires Georgetown]	No
(GMT-02:00)	Mid-Atlantic Standard Time [South_Georgia]	No
(GMT-01:00)	Azores Standard Time [Azores Cape Verde Island]	Yes
(GMT)	Universal Coordinated Time [Casablanca, Monrovia]	No
(GMT0)	Greenwich Mean Time [Dublin Edinburgh Lisbon London]	Yes
(GMT+01:00)	Romance Standard Time [Amsterdam CopenHagen Madrid Paris Vilnius]	Yes
(GMT+01:00)	Central European Standard Time [Belgrade Sarajevo Skopje Sofija Zagreb]	Yes
(GMT+01:00)	Central Europe Standard Time [Bratislava Budapest Ljubljana Prague Warsaw]	Yes
(GMT+01:00)	W. Europe Standard Time [Brussels Berlin Bern Rome Stockholm Vienna]	Yes
(GMT+02:00)	GTB Standard Time [Athens Istanbul Minsk]	Yes
(GMT+02:00)	E. Europe Standard Time [Bucharest]	Yes
(GMT+02:00)	Egypt Standard Time [Cairo]	Yes
(GMT+02:00)	South Africa Standard Time [Johannesburg Harare Pretoria]	No
(GMT+02:00)	FLE Standard Time [Helsinki Riga Tallinn]	Yes
(GMT+02:00)	Israel Standard Time [Israel Jerusalem]	Yes
(GMT+03:00)	Arabic Standard Time [Baghdad]	Yes
(GMT+03:00)	Arab Standard Time [Kuwait Riyadh]	No
(GMT+03:00)	Russian Standard Time [Moscow St. Petersburg Volgograd]	Yes
(GMT+03:00)	E. Africa Standard Time [Nairobi]	No
(GMT+03:30)	Iran Standard Time [Tehran]	Yes
(GMT+04:00)	Arabian Standard Time [Abu Dhabi Muscat]	No
(GMT+04:00)	Caucasus Standard Time [Baku Tbilisi]	Yes
(GMT+04:00)	Afghanistan Standard Time [Kabul]	No
(GMT+05:00)	Ekaterinburg Standard Time [Ekaterinburg]	Yes
(GMT+05:00)	West Asia Standard Time [Islamabad Karachi Tashkent]	No
(GMT+05:30)	India Standard Time [Bombay Calcutta Madras New Delhi]	No
(GMT+06:00)	Central Asia Standard Time [Almaty Dhaka]	Yes
(GMT+06:00)	Sri Lanka Standard Time [Columbo]	No
(GMT+07:00)	SE Asia Standard Time [Bangkok Hanoi Jakarta]	No

Time Zone	Description	DST Available
(GMT+08:00)	China Standard Time [Beijing Chongqing Hong Kong Urumqi]	No
(GMT+08:00)	W. Australia Standard Time [Perth]	No
(GMT+08:00)	Singapore Standard Time [Singapore]	No
(GMT+08:00)	Taipei Standard Time [Taipei]	No
(GMT+09:00)	Tokyo Standard Time [Osako Sapporo Tokyo]	No
(GMT+09:00)	Korea Standard Time [Seoul]	No
(GMT+09:00)	Yakutsk Standard Time [Yakutsk]	Yes
(GMT+09:30)	Cen. Australia Standard Time [Adelaide]	Yes
(GMT+09:30)	AUS Central Standard Time [Darwin]	No
(GMT+10:00)	E. Australia Standard Time [Brisbane]	No
(GMT+10:00)	AUS Eastern Standard Time [Canberra Melbourne Sydney]	Yes
(GMT+10:00)	West Pacific Standard Time [Guam Port Moresby]	No
(GMT+10:00)	Tasmania Standard Time [Hobart]	Yes
(GMT+10:00)	Vladivostok Standard Time [Vladivostok]	Yes
(GMT+11:00)	Central Pacific Standard Time [Magadan Solomon Is New Caledonia]	Yes
(GMT+12:00)	New Zealand Standard Time [Auckland Wellington]	Yes
(GMT+12:00)	Fiji Standard Time [Fiji Kamchatka Marshall Is]	No

Important Information about the Time Service

NOTE: Without an SD card in an NOE module, time zones will not function correctly when changing from daylight saving time.

Customizing Time Zone Parameters

If you want a time zone not listed in the time zone table:

Step	Action	Comment
1	Write the text rules for the custom time zone.	If you want more information, the syntax to write those rules along with a few examples are found in the module in: /FLASH0/wwwroot/conf/NTP/instructions.txt
2	Using an FTP client, store your rules in the file: /FLASH0/wwwroot/conf/NTP/customrules user ID: ntpupdate password: ntpupdate	Root directory to store 'customrules' is set by the FTP server as /FLASH0/wwwroot/conf/NTP
3	When the rules are written, choose the drop down menu on the NTP configuration screen, and configure (or reboot) the module by selecting: **Time Zone = Custom**	The NTP component looks for customrules, calls the tz compiler, and generates a new file called 'tz_custom'. This file is binary file and should not be edited. If the tz compiler detects a syntax error in customrules, the syntax error is logged in the file: /FLASH0/wwwroot/conf/NTP/error.log 1. NTP component is not launched. 2. NTP Status field in the diagnostic screen displays **NOT OK**.

Chapter 11
Configuring an Ethernet Network

About this Chapter

This chapter describes the creation and configuration of an Ethernet network for BMX P34 20x0 CPUs through these major stages:

<table>
<tr><th>Stage</th><th>Description</th><th>Comment</th></tr>
<tr><td>1</td><td>creation of an Ethernet logic network</td><td rowspan="3">run from the project browser</td></tr>
<tr><td>2</td><td>configuration of an Ethernet logic network</td></tr>
<tr><td>3</td><td>choose a logical network family</td></tr>
<tr><td>3</td><td>declaration of the module</td><td rowspan="2">run from the hardware configuration editor</td></tr>
<tr><td>4</td><td>association of the module with the logic network</td></tr>
<tr><td colspan="3">Note: The benefit of this method is that from the second stage onwards, you can design your communication application and use the simulator to test its operation. (You do not have to have the hardware to start working.)</td></tr>
</table>

What Is in This Chapter?

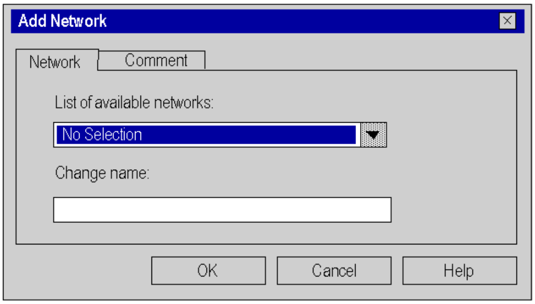
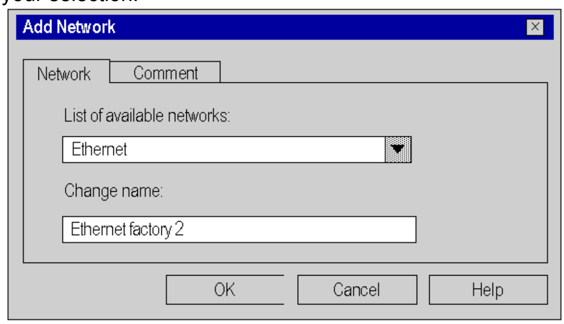
This chapter contains the following topics:

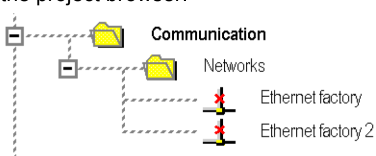
Topic	Page
Configuration Methodology for an Ethernet Network	210
Adding the Module to an Ethernet Network	212

Configuration Methodology for an Ethernet Network

Creating a Network

To create a logical Ethernet network:

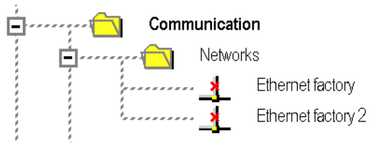
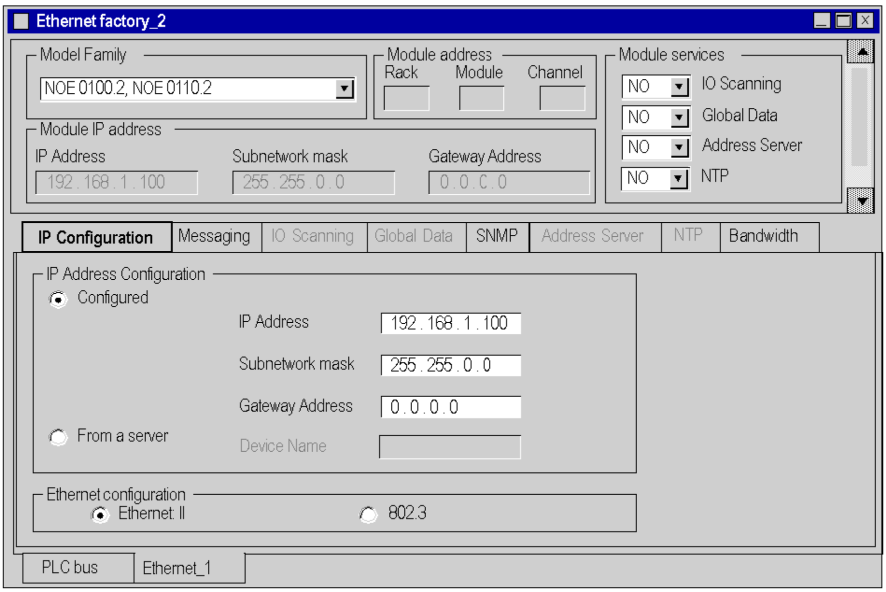
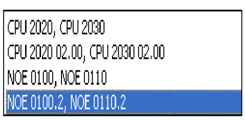
<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>In the Project Browser, right-click the Network subdirectory in the Communication directory and select the New Network option. The Add Network screen appears:

</td></tr>
<tr><td>2</td><td>Choose Ethernet in the List of available networks and choose a meaningful name for your selection:


Note: If desired, a comment may be added by clicking on the Comment tab.</td></tr>
<tr><td>3</td><td>Click OK, and a new logic network is created. The new Ethernet network appears in the project browser:


Note: As you can see, a small icon indicates that the logical network is not associated with a PLC device.</td></tr>
</table>

Accessing Network Configuration

To access the logical Ethernet network configuration:

Step	Action
1	Open the project browser in order to see the logic networks of your application: 
2	Right-click the Ethernet logic network to be configured and select **Open**. The Ethernet configuration screen is displayed. 
3	Scroll to choose the **Model Family** of your network: 

Adding the Module to an Ethernet Network

Declare the Module

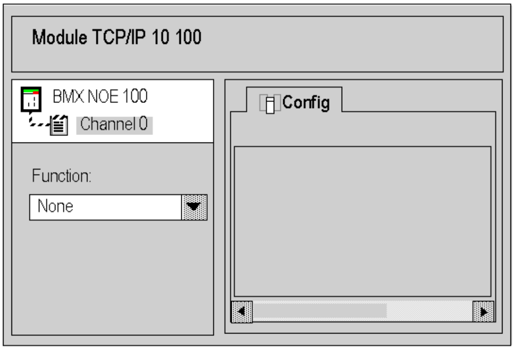
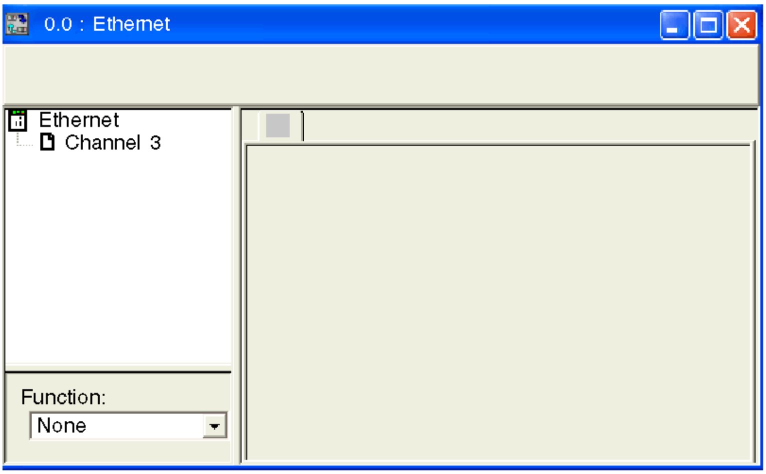
To declare an Ethernet module:

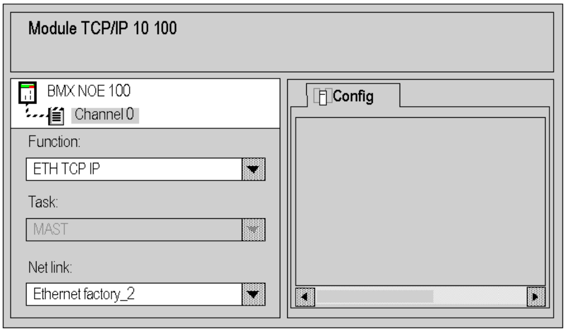
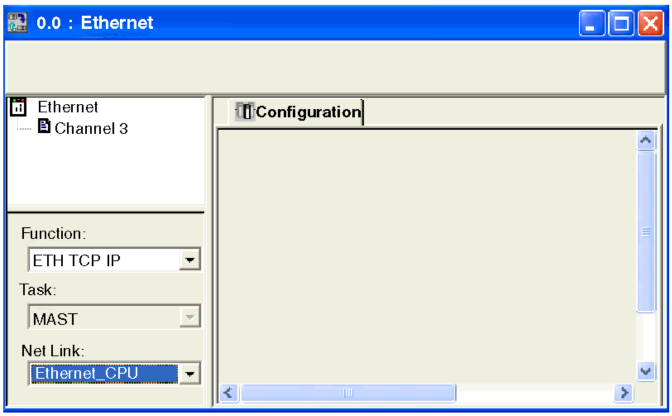
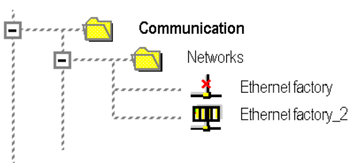
Step	Action	Result
1	Open the hardware configuration editor.	
2	Double-click the empty slot in which you wish to place the module.	The **New Device** window appears.
3	Expand (+) the **Communication** family.	
4	Left-click to select the desired Ethernet module from the list of **Communication** family modules.	
5	Press **OK**.	The module appears in the rack (see note).
Note: In the case of Ethernet solutions integrated in the processors, the Ethernet communication channel is automatically declared when a processor is chosen: ● BMX NOE 01x0 modules: Channel 0 ● BMX P34 20x0 Ethernet port: Channel 3		

NOTE: You can also use the drag and drop method to add a module *(see page 251)* to an Ethernet network.

Associate the Module with the Network

To associate the logical Ethernet network with the module you have just declared:

Step	Action
1	Open the hardware configuration editor.
2	Double-click the module. **NOE:**  **CPU:** 

Step	Action
3	In the **Function** menu, scroll to a network to associate with the module. Then, in the **Net link** menu, scroll to the logic network you wish to associate with the Ethernet channel of the module. **NOE:**  **CPU:** 
4	Confirm your choice and close the window. The **Ethernet factory_2** logic network is associated with the Ethernet BMX NOE 0100 module. The module address is written in the logic network's configuration window. The icon associated with this logic network changes and indicates the links with a PLC. 

Chapter 12
Debugging with Control Expert

About this Chapter

This chapter describes procedures for debugging the configuration of the BMX NOE 01•0 modules with Control Expert.

What Is in This Chapter?

This chapter contains the following topics:

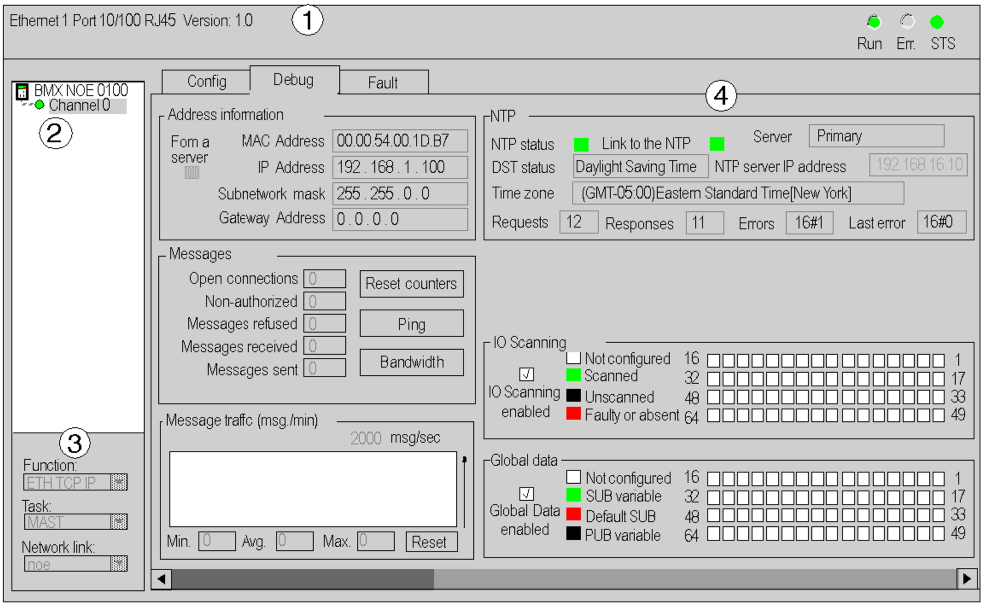
Topic	Page
Module Debugging Screen	216
General Debugging Parameters	218
Debugging Parameters for TCP/IP Utilities	221
Debugging Parameters for I/O Scanning	222
Debugging Parameters for Global Data	223

Module Debugging Screen

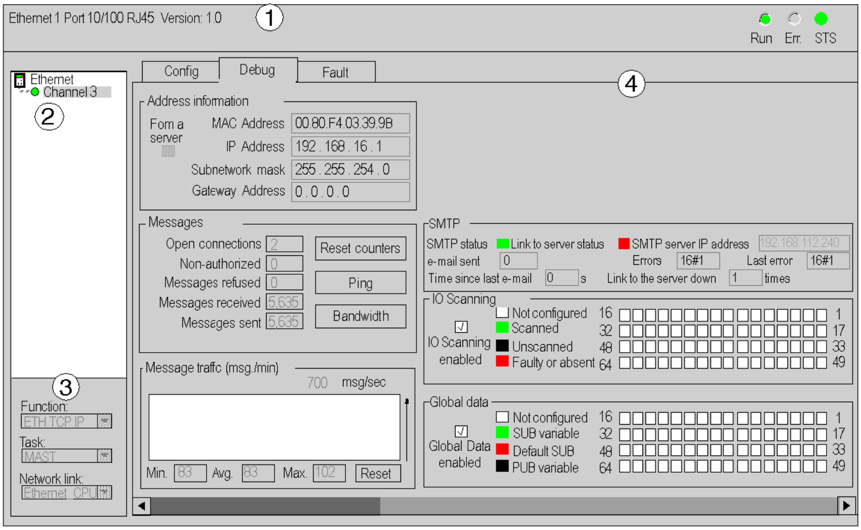
The Screen

This four-zone Control Expert **Debug** tab provides options to debug an Ethernet port.

NOE screen:



CPU screen:



This table describes the zones in the configuration screen:

Zone	Function		
1: **Module**	module description zone (For details refer to LED Indicators *(see page 28)*.)	**Run**	● on: module is operating ● off: PLC not configured
		Err.	● on: configuration or system error ● off: operation is normal (no error)
		STS	● on: communication is OK ● flashing: communication error
2: **Channel**	channel selection zone		
3: **Parameters**	general parameters zone		
4: **Debug** tab	**Address information**	● displays TCP/IP utility configuration ● tests communication of the TCP/IP profile	
	Messages	displays the number of open connections and the number of messages that are unauthorized, refused, received, and sent.	
	Message traffic	displays the number of messages processed by the module per minute	
	IO Scanning	displays the status for each remote input/output module	
	Global Data	displays the status of global data variables	
	NTP	displays the status of the NTP server (NOE modules only)	
	SMTP	displays the status of the SMTP server (CPU modules only)	

General Debugging Parameters

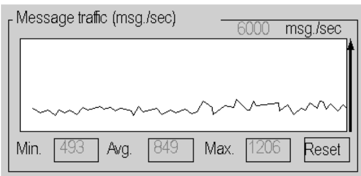
Introduction

Double-click the module to open the debug tab *(see page 263)*.

The general debugging parameters on the module debugging screen *(see page 216)* are grouped into two windows:
- the **Message traffic** window
- the **Messages** window

Message Traffic

The **Message traffic** window looks like this:

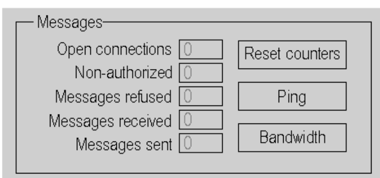


It graphically shows the number of Ethernet packets per second handled by the module (sent and received).

The **Reset** button resets the **Min.**, **Av.**, and **Max** counters to 0.

Messages

The **Messages** window looks like this:

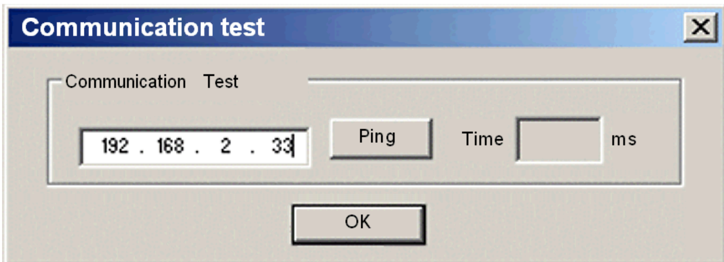
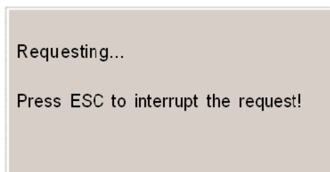


This window reports the number of:
- open TCP/IP connections (the combined number of server, client, and Transparent Device Access connections that are open)
- non-authorized TCP/IP connections
- refused TCP/IP messages
- received TCP/IP messages
- sent TCP/IP messages

This window includes three buttons:
- **Reset counters**: Press this button to reset the counters to 0.
- **Ping** (see below)
- **Bandwidth** (see below)

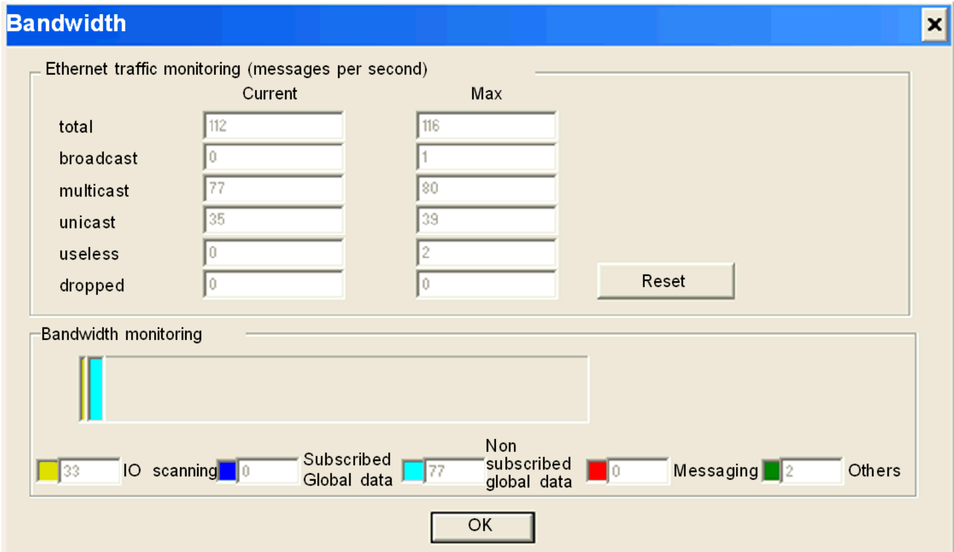
Ping

You can test the routing between your module and another device through a PING request:

Step	Action	Comment
1	Enter the IP address of the device for which you want to test communications and press Ping. 	
2	Wait for the request to be processed	This window appears: 
3	The **COMMUNICATION** window informs you that the exchange was successful.	The **COMMUNICATION** window: 
4	Press **OK**.	With the successful PING request, a value appears in the **ms** field.

Bandwidth

Press the **Bandwidth** button to see the number of messages received per second in the **Bandwidth** window. Available bandwidth and network load are reported in terms of dynamic bandwidth monitoring *(see page 103)*.

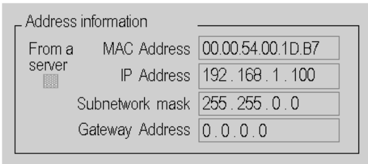


NOTE: Press **Reset** to clear the **Max** column fields.

Debugging Parameters for TCP/IP Utilities

Address Information

The debugging parameters for TCP/IP utilities on the module debugging screen *(see page 216)* are grouped together in the **Address information** window:

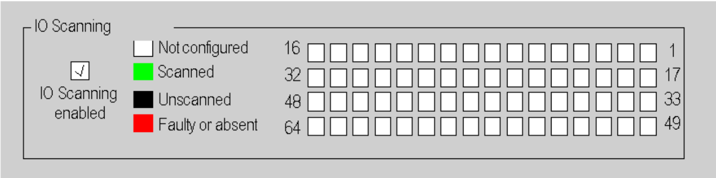


This window displays the configuration of:
- MAC Address
- IP Address
- Subnetwork mask
- Gateway Address

Debugging Parameters for I/O Scanning

IO Scanning Dialog

When the I/O Scanner is activated, the status of each configured device is displayed on the module debugging screen *(see page 216)*:



A device referenced in the I/O Scanning configuration tab can have the following states:
- **Not configured** (white)
- **Scanned** (green)
- **Unscanned** (black): caused by the I/O Scanner not being activated (nothing selected in configuration)
- **Faulty or Absent** (red)

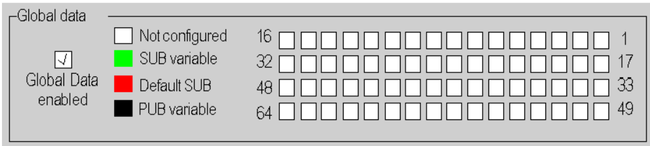
NOTE: In the cases of **Not configured** or **Unscanned**, no Modbus request is sent to the device.

NOTE: This is the same information as seen in the IODDT *(see page 225)* for the module.

Debugging Parameters for Global Data

Global Data Dialog

When Global Data is activated, the status of a Global Data variable is displayed on the module debugging screen *(see page 216)*:



A Global Data variable can have the following status:

- **Not configured** (white)
- **SUB variable** (green): subscribed variable received in the health timeout limit
- **Default SUB** (red): subscribed variable not received in the health time out limit
- **PUB variable** (black): published variable

Chapter 13
Ethernet Language Objects

About this Chapter

This chapter describes the language objects associated with the Ethernet communication modules.

There is also a discussion of IODDTs. The IODDT (Input/Output Derived Data Type) is a data type associated with a PLC channel or module. Expert modules are associated with specific IODDTs.

What Is in This Chapter?

This chapter contains the following sections:

Section	Topic	Page
13.1	Language Objects and IODDTs of Ethernet Communication	226
13.2	Exchange Objects of Type T_COM_ETH_BMX	233
13.3	Language Objects Associated with BMX NOE 01x0 and BMX P34 20x0 CPU Configuration	239
13.4	Language Objects and Generic IODDT Applicable to Communication Protocols	246

Section 13.1
Language Objects and IODDTs of Ethernet Communication

About this Section

This section provides a general overview of language objects and IODDTs of Ethernet communication.

What Is in This Section?

This section contains the following topics:

Topic	Page
Language Objects and IODDTs of Ethernet Communication	227
Details of T_COM_EIP IODDT	228
Implicit Exchange Language Objects Associated with the Application-Specific Function	230
Explicit Exchange Language Objects Associated with the Application-Specific Function	231

Language Objects and IODDTs of Ethernet Communication

General

Ethernet communication has the following IODDT:

- `T_COM_ETH_BMX`: specific to modules with Ethernet communication

IODDTs are predefined by the manufacturer and contain input/output language objects belonging to the channel of an application-specific module.

NOTE:
IODDT variables can be created with:

- the I/O objects tab
- the Data Editor

Types of Language Objects

Each IODDT has a set of language objects that is used to control and check the operation of the IODDT. There are two types of language objects:

- **implicit**: Implicit exchange objects are exchanged automatically on each cycle turn of the task associated with the module. These exchanges concern the states of modules, communication signals, slaves, etc.
- **explicit**: Explicit exchange objects are exchanged at the request of the application, using explicit exchange instructions. These exchanges set parameters and diagnose the module.

Elsewhere in this guide are detailed descriptions for the IODDT types *(see page 239)*.

Details of T_COM_EIP IODDT

Exchange Objects of the IODDT

The T_COM_EIP IODDT supports both implicit and explicit exchange objects:
- Implicit exchange objects are automatically exchanged at each cycle of a task associated with the channel.
- Explicit exchange objects can be reached via explicit messaging controlled either by program logic, or by operator commands.

Exchange objects are %I, %IW, %M and %MW.

The tables below presents the various implicit and explicit exchange objects of IODDT T_COM_EIP, as supported by the BMX NOC 0401 communication module.

The parameters r, m, and c shown in the following-tables represent the topological addressing of the module. Each parameter has the following signification:
- **r** represents the rack (or station) number
- **m** represents the module (or slot) number
- **c** represents the channel number

Implicit and Explicit Exchange Objects of the T_COM_EIP IODDT

The T_COM_EIP IODDT presents the following implicit communication objects:

Standard symbol	Type	Access	Description	Address
CH_ERROR	BOOL	R	Channel detected error bit	%Ir.m.c.ERR
STS_ETH_SERVICES	INT	R	Status of Ethernet services:	%IWr.m.c.0
			Bit 0: EIP Scanner (0 = OK, 1 = NOK)	
			Bit 1: EIP Adapter (0 = OK, 1 = NOK)	
			Bit 2: EIP Client (0 = OK, 1 = NOK)	
			Bit 3: EIP Server (0 = OK, 1 = NOK)	
			Bit 4: Modbus scanner (0 = OK, 1 = NOK)	
			Bit 5: Modbus TCP Client (0 = OK, 1 = NOK)	
			Bit 6: Modbus TCP Server (0 = OK, 1 = NOK)	
			Bit 7: FDR Server (0 = OK, 1 = NOK)	
			Bit 8: RSTP (0 = OK, 1 = NOK)	
			Bit 9–Bit 15: (Reserved)	

Explicit Exchange Objects of the T_COM_EIP IODDT

The T_COM_EIP IODDT presents the following explicit communication objects:

Standard symbol	Type	Access	Description	Address
EXCH_STS	INT	R	Exchange Status:	%MWr.m.c.0
STS_IN_PROGR	BOOL	R	Bit 0: Status parameter read in progress	%MWr.m.c.0.0
CMD_IN_PROGR	BOOL	R	Bit 1: Command parameter write in progress	%MWr.m.c.0.1
ADJ_IN_PROGR	BOOL	R	Bit 2: (Reserved)	%MWr.m.c.0.2
EXCH_RPT	INT	R	Channel report	%MWr.m.c.1
STS_ERR	BOOL	R	Bit 0: Error detected while reading channel status	%MWr.m.c.1.0
CMD_ERR	BOOL	R	Bit 1: Error detected while sending a command on the channel	%MWr.m.c.1.1
ADJ_ERR	BOOL	R	Bit 2: (Reserved)	%MWr.m.c.1.2
CH_FLT	INT	R	Channel faults detected	%MWr.m.c.2
NO_DEVICE	BOOL	R	Bit 0: (Reserved)	%MWr.m.c.2.0
ONE_DEVICE_FLT	BOOL	R	Bit 1: (Reserved)	%MWr.m.c.2.1
BLK	BOOL	R	Bit 2: (Reserved)	%MWr.m.c.2.2
TO_ERR	BOOL	R	Bit 3: (Reserved)	%MWr.m.c.2.3
INTERNAL_FLT	BOOL	R	Bit 4: Internal error detected: channel inoperative	%MWr.m.c.2.4
CONF_FLT	BOOL	R	Bit 5: (Reserved)	%MWr.m.c.2.5
COM_FLT	BOOL	R	Bit 6: X-bus communication error detected	%MWr.m.c.2.6
APPLI_FLT	BOOL	R	Bit 7: Application configuration error detected	%MWr.m.c.2.7
ETH_GLOBAL_STS	INT	R	Ethernet global status	%MWr.m.c.3
	BOOL	R	Bit 0: detected configuration error	
	BOOL	R	Bit 1: (Reserved)	
	BOOL	R	Bit 2: duplicate IP address detected	
	BOOL	R	Bit 3: Configuration mismatch	
	BOOL	R	Bit 4: All 4 external Ethernet ports are disconnected	
	BOOL	R	Bit 5: The module is in the process of obtaining an IP address (BOOTP or in duplicate IP address)	
IP_ADDR	DINT	R	IP address	%MWr.m.c.4

Implicit Exchange Language Objects Associated with the Application-Specific Function

At a Glance

An integrated application-specific interface or the addition of a module automatically enhances the language objects application used to program this interface or module.

These objects correspond to the input/output images and software data of the module or integrated application-specific interface.

Reminders

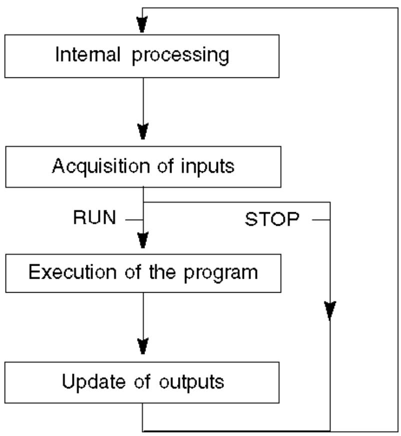
The module inputs (`%I` and `%IW`) are updated in the PLC memory at the start of the task, the PLC being in RUN or STOP mode.

The outputs (`%Q` and `%QW`) are updated at the end of the task, only when the PLC is in RUN mode.

NOTE: When the task occurs in STOP mode, either of the following are possible, depending on the configuration selected:
- outputs are set to fallback position (fallback mode)
- outputs are maintained at their last value (maintain mode)

Figure

The following diagram shows the operating cycle of a PLC task (cyclical execution).



Explicit Exchange Language Objects Associated with the Application-Specific Function

Introduction

Explicit exchanges are performed at the user program's request using these instructions:
- READ_STS (read status words)
- WRITE_CMD (write command words)
- WRITE_PARAM (write adjustment parameters)
- READ_PARAM (read adjustment parameters)
- SAVE_PARAM (save adjustment parameters)
- RESTORE_PARAM (restore adjustment parameters)

For more details about instructions, refer to *EcoStruxure™ Control Expert, I/O Management, Block Library*.

These exchanges apply to a set of %MW objects of the same type (status, commands or parameters) that belong to a channel.

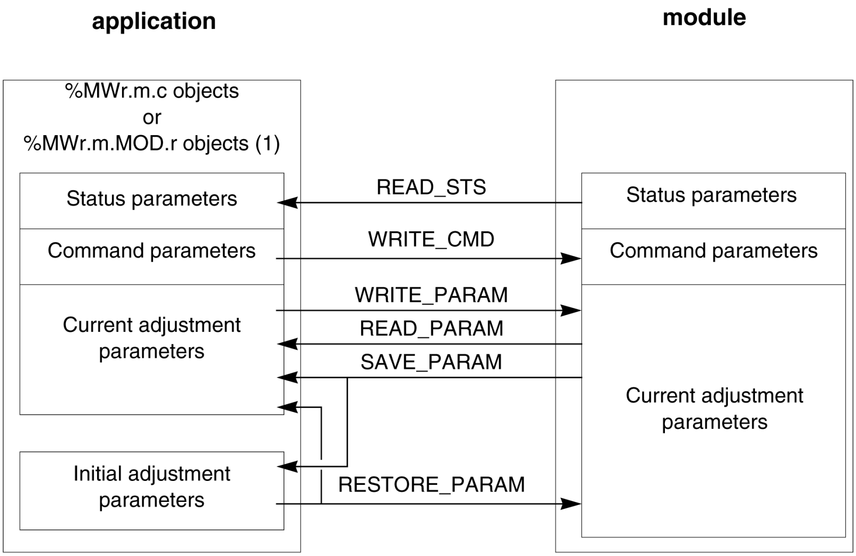
These objects can:
- provide information about the module (for example, type of error detected in a channel)
- have command control of the module (for example, switch command)
- define the module's operating modes (save and restore adjustment parameters in the process of application)

NOTE: To avoid several simultaneous explicit exchanges for the same channel, it is necessary to test the value of the word EXCH_STS (`%MWr.m.c.0`) of the IODDT associated to the channel before calling any EF addressing this channel.

NOTE: Explicit exchanges are not supported when X80 analog and digital I/O modules are configured through an eX80 adapter module (BMECRA31210) in a Quantum EIO configuration. You cannot set up a module's parameters from the PLC application during operation.

General Principle for Using Explicit Instructions

The diagram below shows the different types of explicit exchanges that can be made between the application and module.



(1) Only with READ_STS and WRITE_CMD instructions.

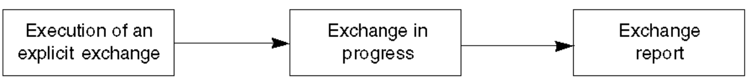
Managing Exchanges

During an explicit exchange, check performance to see that the data is only taken into account when the exchange has been correctly executed.

To do this, two types of information is available:

- information concerning the exchange in progress *(see EcoStruxure™ Control Expert, I/O Management, Block Library)*
- the exchange report *(see EcoStruxure™ Control Expert, I/O Management, Block Library)*

The following diagram describes the management principle for an exchange.



NOTE: In order to avoid several simultaneous explicit exchanges for the same channel, it is necessary to test the value of the word EXCH_STS (`%MWr.m.c.0`) of the IODDT associated to the channel before calling any EF addressing this channel.

Section 13.2
Exchange Objects of Type T_COM_ETH_BMX

About this Section

The section describes the implicit and explicit exchange objects of type T_COM_ETH_BMX.

What Is in This Section?

This section contains the following topics:

Topic	Page
Details of Implicit Exchange Objects of the IODDT Type T_COM_ETH_BMX	234
Details of Explicit Exchange Objects of the IODDT Type T_COM_ETH_BMX	236
Details of Explicit Exchange Objects for an Ethernet Function	238

Details of Implicit Exchange Objects of the IODDT Type T_COM_ETH_BMX

Objects

The IODDT of type T_COM_ETH_BMX has implicit exchange objects, which are described below. This type of IODDT applies to the BMX P34 20x0 and BMX NOE 01x0 modules.

Standard Symbol		Type	Access	Meaning	Address
CH_ERROR		BOOL	R	line error bit	%Ir.m.c.ERR
SERVICES_STS		INT	R	status of the different services	%IWr.m.c.0
	P502_STATUS_BIT	BOOL	R	Port 502 messaging service status (0=OK, 1=NOK)	%IWr.m.c.0.0
	IOS_STATUS_BIT	BOOL	R	IO Scanner service status (0=OK, 1=NOK)	%IWr.m.c.0.1
	GLBD_STATUS_BIT	BOOL	R	Global Data service status (0=OK, 1=NOK)	%IWr.m.c.0.2
	EMAIL_STATUS_BIT	BOOL	R	e-mail service status (0=OK, 1=NOK)	%IWr.m.c.0.3
	FDRS_STATUS_BIT	BOOL	R	FDR server service status (0=OK, 1=NOK)	%IWr.m.c.0.4
	NTPC_STATUS_BIT	BOOL	R	NTP Client service status (0=OK, 1=NOK)	%IWr.m.c.0.5
	TCPOPEN_STATUS_BIT	BOOL	R	Reserved for L2 (for future use)	%IWr.m.c.0.6
REFRESH_IO_1 to REFRESH_IO_16		BOOL	R	indicates that the inputs/outputs of the I/O Scanner from stations 1 to 16 are refreshed	%IWr.m.c.1.0 to %IWr.m.c.1.15
REFRESH_IO_17 to REFRESH_IO_32		BOOL	R	indicates that the inputs/outputs of the I/O Scanner from stations 17 to 32 are refreshed	%IWr.m.c.2.0 to %IWr.m.c.2.15
REFRESH_IO_33 to REFRESH_IO_48		BOOL	R	indicates that the inputs/outputs of the I/O Scanner from stations 33 to 48 are refreshed	%IWr.m.c.3.0 to %IWr.m.c.3.15
REFRESH_IO_49 to REFRESH_IO_64		BOOL	R	indicates that the inputs/outputs of the I/O Scanner from stations 49 to 64 are refreshed	%IWr.m.c.4.0 to %IWr.m.c.4.15
VALID_GD_1 to VALID_GD_16		BOOL	R	indicates that Global Data from stations 1 to 16 are refreshed	%IWr.m.c.5.0 to %IWr.m.c.5.15
VALID_GD_17 to VALID_GD_32		BOOL	R	indicates that Global Data from stations 17 to 32 are refreshed	%IWr.m.c.6.0 to %IWr.m.c.6.15
VALID_GD_33 to VALID_GD_48		BOOL	R	indicates that Global Data from stations 33 to 48 are refreshed	%IWr.m.c.7.0 to %IWr.m.c.7.15

Standard Symbol	Type	Access	Meaning	Address
VALID_GD_49 to VALID_GD_64	BOOL	R	indicates that Global Data from stations 49 to 64 are refreshed	%IWr.m.c.8.0 to %IWr.m.c.8.15
DISABLE_IO_1 to DISABLE_IO_16	BOOL	R/W	enables/disables the refreshing of inputs/outputs of the I/O Scanner for stations 1 to 16	%QWr.m.c.0.0 to %QWr.m.c.0.15
DISABLE_IO_17 to DISABLE_IO_32	BOOL	R/W	enables/disables the refreshing of inputs/outputs of the I/O Scanner for stations 17 to 32	%QWr.m.c.1.0 to %QWr.m.c.1.15
DISABLE_IO_33 to DISABLE_IO_48	BOOL	R/W	enables/disables the refreshing of inputs/outputs of the I/O Scanner for stations 33 to 48	%QWr.m.c.2.0 to %QWr.m.c.2.15
DISABLE_IO_49 to DISABLE_IO_64	BOOL	R/W	enables/disables the refreshing of inputs/outputs of the I/O Scanner for stations 49 to 64	%QWr.m.c.3.0 to %QWr.m.c.3.15
R = read only R/W = read/write				

Details of Explicit Exchange Objects of the IODDT Type T_COM_ETH_BMX

System Words

The table below shows the meaning of the system word bits:

Standard Symbol	Type	Access	Meaning	Address
EXCH_STS	INT	R	exchange status	%MWr.m.c.0
STS_IN_PROGR	BOOL	R	reading of status words of the channel in progress	%MWr.m.c.0.0
CMD_IN_PROGR	BOOL	R	command parameter write in progress	%MWr.m.c.0.1
ADJ_IN_PROGR	BOOL	R	adjust parameter exchange in progress	%MWr.m.c.0.2
RECONF_IN_PROGR	BOOL	R	reconfiguration in progress	%MWr.m.c.0.15
EXCH_RPT	INT	R	channel report	%MWr.m.c.1
STS_ERR	BOOL	R	channel status cannot be read	%MWr.m.c.1.0
CMD_ERR	BOOL	R	a command cannot be sent on the channel	%MWr.m.c.1.1
ADJ_ERR	BOOL	R	the channel cannot be adjusted	%MWr.m.c.1.2
RECONF_ERR	BOOL	R	the channel cannot be reconfigured	%MWr.m.c.1.15
R = read only				

Status Words

The table below gives the meanings of the status word bits CH_FLT (%MWr.m.c.2). The reading is taken by a READ_STS:

Standard Symbol	Type	Access	Meaning	Address
INTERNAL_FLT	BOOL	R	an internal error has been detected or the self-test cannot be completed	%MWr.m.c.2.4
APPLI_FLT	BOOL	R	an adjustment or configuration error has been detected	%MWr.m.c.2.7
R = read only				

The table below shows the result of a READ_STS call:

Standard Symbol	Type	Access	Meaning	Address
ETH_STATUS	INT	R	Ethernet port global status	%MWr.m.c.3
IP_ADDR	DINT	R	IP address	%MDr.m.c.4
P502_NB_CONN_DENIED	INT	R	number of denied Port 502 connections	%MWr.m.c.6
BW_MAX_MSG_IN	INT	R	maximum number of received messages on the Ethernet port per second	%MWr.m.c.10
BW_MAX_MSG_BC	INT	R	maximum number of broadcast messages received per second	%MWr.m.c.14
reserved	INT	R	reserved for future use	%MWr.m.c.15
R = read only				

Command Words

This table shows the available command words:

Standard Symbol	Type	Access	Meaning	Address
ETH_RESET	BOOL	W	Ethernet component reset	%MWr.m.c.16.0
BW_CNT_RESET	BOOL	W	max. message counters reset	%MWr.m.c.16.1
P502_CNT_RESET	BOOL	W	messaging counters reset	%MWr.m.c.16.2
W = write only				

The command is carried out with the `WRITE_CMD (IODDT_VAR1)` function.

Details of Explicit Exchange Objects for an Ethernet Function

Overview

The following table objects are not integrated into the IODDTs. Reading is performed by a `READ_STS` call.

Explicit Exchange Objects

Address	Type	Access	Meaning
%MWr.m.c.7	INT	R	number of received messages on the Ethernet port per second (BW_NB_MSG_IN)
%MWr.m.c.8	INT	R	number of useless messages filtered by the Ethernet port per second (BW_NB_MSG_FILTER)
%MWr.m.c.9	INT	R	number of messages dropped by the Ethernet port per second (BW_NB_MSG_DROP)
%MWr.m.c.11	INT	R	maximum number of useless messages filtered by the Ethernet port per second (BW_MAX_MSG_FILTER)
%MWr.m.c.12	INT	R	maximum number of messages dropped by the Ethernet port per second (BW_MAX_MSG_DROP)
%MWr.m.c.13	INT	R	maximum number of multicast messages received per second (BW_MAX_MSG_MC)
R = read only			

Section 13.3
Language Objects Associated with BMX NOE 01x0 and BMX P34 20x0 CPU Configuration

About this Section

This section describes the configuration language objects associated with the Ethernet communication modules on BMX NOE 01x0 communication modules and BMX P34 20x0 CPUs.

What Is in This Section?

This section contains the following topics:

Topic	Page
Language Objects for Configuration	240
Language Objects for Implicit Exchange	241
Language Objects for Explicit Exchange	243

Language Objects for Configuration

Introduction

This topic describes the language objects for the configuration of the BMX NOE 01x0 modules and BMX P34 20x0 CPUs.

Configuration Words

The following table describes the configuration word (%KW) language objects:

Object		Description
BMX NOE 01x0	BMX P34 20x0	
%KWr.m.c.0	%KWr.m.c.0	reserved for future use
%KWr.m.c.1	%KWr.m.c.1	reserved for future use
%KWr.m.c.2	%KWr.m.c.2	general services configured: ● bit 0 = 1: I/O scanning configured ● bit 1 = 1: address server configured ● bit 2 = 1: global data configured ● bit 3 . . . 15: reserved

Language Objects for Implicit Exchange

Introduction

This topic describes the implicit exchange language objects for the BMX NOE 01x0 module and BMX P34 20x0 CPUs.

Input Bits

The following table describes the input bit (%I) language objects:

Object		Description
BMX NOE 01x0	BMX P34 20x0	
%Ir.m.0.ERR	%Ir.m.3.ERR	line error bit (CH_ERROR)

Input Words

The following table describes the input word (%IW) language objects:

Object		Description
BMX NOE 01x0	BMX P34 20x0	
%IWr.m.0.0	%IWr.m.3.0	status of Ethernet services: ● bit 0: port 502 messaging service status (0=OK, 1=NOK) ● bit 1: IO Scanner service status (0=OK, 1=NOK) ● bit 2: Global Data service status (0=OK, 1=NOK) ● bit 3: e-mail service status (0=OK, 1=NOK) ● bit 4: FDR Server service status (0=OK, 1=NOK) ● bit 5: ❍ BMX NOE 01x0: reserved for future use ❍ BMX P34 20x0: reserved for compatibility with the BMX NOE 01x0 ● bit 6: reserved for future use ● bit 7: reserved
%IWr.m.0.1 ... %IWr.m.0.4	%IWr.m.3.1 ... %IWr.m.3.4	IOS health (or refresh) block: ● 64 devices (maximum) ● 1 bit per IOS device ● 1 = healthy; 0 = unhealthy
%IWr.m.0.5 ... %IWr.m.0.8	%IWr.m.3.5 ... %IWr.m.3.8	GD health (or refresh) block: ● 64 GD stations (maximum) ● 1 bit per GD station ● 1 = healthy; 0 = unhealthy

Output Words

The following table describes the output word (%QW) language objects:

Object		Description
BMX NOE 01x0	BMX P34 20x0	
%QWr.m.0.0 ... %QWr.m.0.3	%QWr.m.3.0 ... %QWr.m.3.3	IOS device control (disable/enable) block: • 64 devices (maximum) • 1 bit per IOS device • 1 = disable; 0 = enable

Language Objects for Explicit Exchange

Introduction

This topic describes the explicit exchange language objects for the BMX NOE 01x0 module and BMX P34 20x0 CPUs.

System Words

The following table describes the system word (%MW, READ) language objects:

Object		Description
BMX NOE 01x0	BMX P34 20x0	
%MWr.m.0.0	%MWr.m.3.0	exchange status (EXCH_STS): ● bit 0 = 1: reading of status words of the channel in progress (STS_IN_PROGR) ● bit 1 = 1: command write in progress (CMD_IN_PROGR)
%MWr.m.0.1	%MWr.m.3.1	exchange report (EXCH_RPT): ● bit 0 = 1: error while reading channel status (STS_ERR) ● bit 1 = 1: error while writing a command to the channel (CMD_ERR) **Note:** Always 0 for the BMX P34 20x0

Status Words

The following table describes the status word (%MW or %MD, READ) language objects:

Object		Description
BMX NOE 01x0	BMX P34 20x0	
%MWr.m.0.2	%MWr.m.3.2	Standard channel faults (CH_FLT): ● bit 4 (%MWr.m.0.2.4) = 1: Internal or channel self-test fault (INTERNAL_FLT) ● bit 7 (%MWr.m.0.2.7) = 1: Application fault (APPLI_FLT)
%MWr.m.0.3	%MWr.m.3.3	Ethernet Port Global status (ETH_PORT_STATUS)
%MDr.m.0.4	%MDr.m.3.4	IP address (IP_ADDR)
%MWr.m.0.6	%MWr.m.3.6	number of denied Port 502 connections (P502_NB_CONN_DENIED)
%MWr.m.0.7	%MWr.m.3.7	number of received messages on the Ethernet port per second (BW_NB_MSG_IN)
%MWr.m.0.8	%MWr.m.3.8	number of useless messages filtered by the Ethernet port per second (BW_NB_MSG_FILTER)

Object		Description
BMX NOE 01x0	BMX P34 20x0	
%MWr.m.0.9	%MWr.m.3.9	number of messages dropped by the Ethernet port per second (BW_NB_MSG_DROP)
%MWr.m.0.10	%MWr.m.3.10	maximum number of received messages on the Ethernet port per second (BW_MAX_MSG_IN)
%MWr.m.0.11	%MWr.m.3.11	maximum number of useless messages filtered by the Ethernet port per second (BW_MAX_MSG_FILTER)
%MWr.m.0.12	%MWr.m.3.12	maximum number of messages dropped by the Ethernet port per second (BW_MAX_MSG_DROP)
%MWr.m.0.13	%MWr.m.3.13	maximum number of Multicast messages received per second (BW_MAX_MSG_MC)
%MWr.m.0.14	%MWr.m.3.14	maximum number of broadcast messages received per second (BW_MAX_MSG_BC)
%MWr.m.0.15	%MWr.m.3.15	reserved for future use

Command Words

The following table describes the command word (%MW, WRITE) language objects:

Object		Description
BMX NOE 01x0	BMX P34 20x0	
%MWr.m.0.16	%MWr.m.3.16	Ethernet command word (ETH_CMD): • bit 0 = 1 for Ethernet Component Reset (ETH_RESET) • bit 1 = 1 for Max Message Counters reset (BW_CNT_RESET) • bit 2 = 1 for Messaging counters reset (P502_CNT_RESET)
%MWr.m.0.17	%MWr.m.3.17	reserved for modulo 4 address alignment

Parameter Words

The following table describes the parameter word (%MW or %MD, R/W) language objects, please note that parameters are accessible in read using the `READ_STATUS` function:

Object		Description
BMX NOE 01x0	BMX P34 20x0	
%MWr.m.0.18	%MWr.m.3.18	PARAM_NET_CONF: ● 0 = disabled (not supported) ● 1 = from screen configuration (default) ● 2 = from a server (DHCP server)
%MWr.m.0.19	%MWr.m.3.19	PARAM_DRIVER_CONF: (TBD) Bit 0: FRAME_TYPE
%MDr.m.0.20	%MDr.m.3.20	IP address (PARAM_IP_ADDR)
%MWr.m.0.22	%MWr.m.3.22	subnetwork mask (PARAM_IP_NETMASK)
%MWr.m.0.24	%MWr.m.3.24	default gateway (PARAM_IP_GATEWAY)
%MWr.m.0.26	%MWr.m.3.26	device name (character 1 and 2) – PARAM_IP_DEVICE_NAME1
%MWr.m.0.27	%MWr.m.3.27	device name (character 3 and 4) – PARAM_IP_DEVICE_NAME2
%MWr.m.0.28	%MWr.m.3.28	device name (character 5 and 6) – PARAM_IP_DEVICE_NAME3
%MWr.m.0.29	%MWr.m.3.29	device name (character 7 and 8) – PARAM_IP_DEVICE_NAME4
%MWr.m.0.30	%MWr.m.3.30	device name (character 9 and 10) – PARAM_IP_DEVICE_NAME5
%MWr.m.0.31	%MWr.m.3.31	device name (character 11 and 12) – PARAM_IP_DEVICE_NAME6
%MWr.m.0.32	%MWr.m.3.32	device name (character 13 and 14) – PARAM_IP_DEVICE_NAME7
%MWr.m.0.33	%MWr.m.3.33	device name (character 15 and 16) – PARAM_IP_DEVICE_NAME8

NOTE: For explicit exchanges, the `READ_PARAM` function is not available for the following modules:
- BMX P34 20x0
- BMX NOE 0100

Section 13.4
Language Objects and Generic IODDT Applicable to Communication Protocols

Details of the Language Objects of the IODDT of Type T_GEN_MOD

Introduction

The Modicon X80 modules have an associated IODDT of type T_GEN_MOD.

Observations

In general, the meaning of the bits is given for bit status 1. In specific cases an explanation is given for each status of the bit.

Some bits are not used.

List of Objects

The table below presents the objects of the IODDT.

Standard Symbol	Type	Access	Meaning	Address
MOD_ERROR	BOOL	R	Module detected error bit	%Ir.m.MOD.ERR
EXCH_STS	INT	R	Module exchange control word	%MWr.m.MOD.0
STS_IN_PROGR	BOOL	R	Reading of status words of the module in progress	%MWr.m.MOD.0.0
EXCH_RPT	INT	R	Exchange report word	%MWr.m.MOD.1
STS_ERR	BOOL	R	Event when reading module status words	%MWr.m.MOD.1.0
MOD_FLT	INT	R	Internal detected errors word of the module	%MWr.m.MOD.2
MOD_FAIL	BOOL	R	module inoperable	%MWr.m.MOD.2.0
CH_FLT	BOOL	R	Inoperative channel(s)	%MWr.m.MOD.2.1
BLK	BOOL	R	Terminal block incorrectly wired	%MWr.m.MOD.2.2
CONF_FLT	BOOL	R	Hardware or software configuration anomaly	%MWr.m.MOD.2.5
NO_MOD	BOOL	R	Module missing or inoperative	%MWr.m.MOD.2.6
EXT_MOD_FLT	BOOL	R	Internal detected errors word of the module (Fipio extension only)	%MWr.m.MOD.2.7
MOD_FAIL_EXT	BOOL	R	Internal detected error, module unserviceable (Fipio extension only)	%MWr.m.MOD.2.8
CH_FLT_EXT	BOOL	R	Inoperative channel(s) (Fipio extension only)	%MWr.m.MOD.2.9

Standard Symbol	Type	Access	Meaning	Address
BLK_EXT	BOOL	R	Terminal block incorrectly wired (Fipio extension only)	%MWr.m.MOD.2.10
CONF_FLT_EXT	BOOL	R	Hardware or software configuration anomaly (Fipio extension only)	%MWr.m.MOD.2.13
NO_MOD_EXT	BOOL	R	Module missing or inoperative (Fipio extension only)	%MWr.m.MOD.2.14

Chapter 14
M340 Ethernet Communications Quick Start

Overview

This quick start procedure helps you to quickly configure Modicon M340 Ethernet communication modules (BMX NOE 01x0, BMX P34 20x0) and set up basic communication services such as I/O scanning.

NOTE:
The performance of your module depends on the specific configuration of services and features on the module. To maximize module performance, refer to these Modicon M340 automation platform catalogs:

- Ethernet TCP/IP network, Transparent Ready: Performances (43425)
- Communication, integrated ports and modules (0504Q)
- Ethernet TCP/IP network, Transparent Ready: Processor solutions with integrated port or module (43417)

What Is in This Chapter?

This chapter contains the following topics:

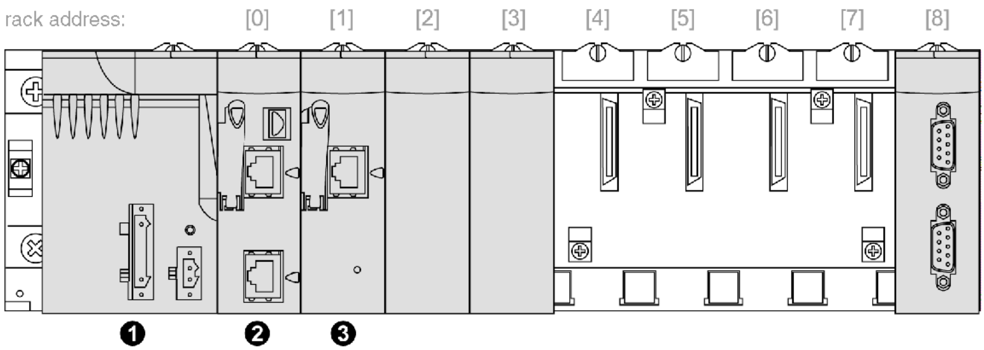
Topic	Page
Hardware Installation	250
Modicon M340 Configuration with Control Expert	251
Ethernet Network Configuration with Control Expert	254
Assign the BMX NOE 0100 IP Address	255
Configure Ethernet Communication Service (I/O Scanning)	256
Associate the Network with the Module	260
Build a Program	261
Connect the System and Download the Configuration	262
Debugging the Module	263

Hardware Installation

Rack Assembly

You can select the appropriate power supply, processor, Ethernet communication module(s), and other M340 modules from the Modicon M340 automation platform catalog (part number 43423).

The following figure shows the rack assembly used for this Quick Start example:



1 power supply
2 BMX P34 2020 in rack slot 0
3 BMX NOE 0100 in rack slot 1

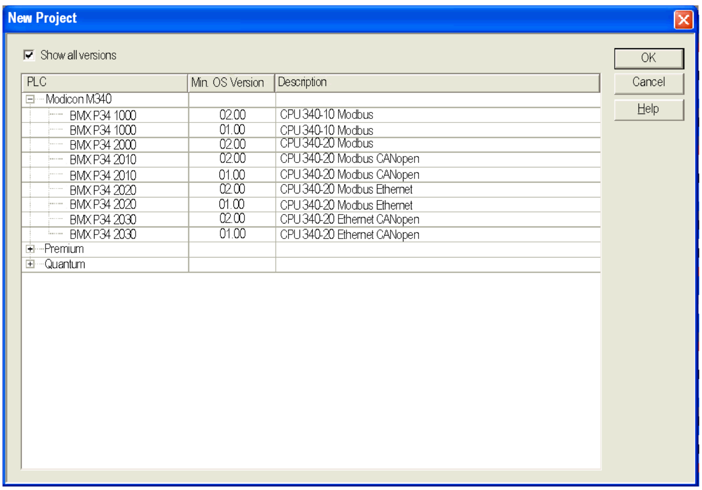
To assemble the rack:

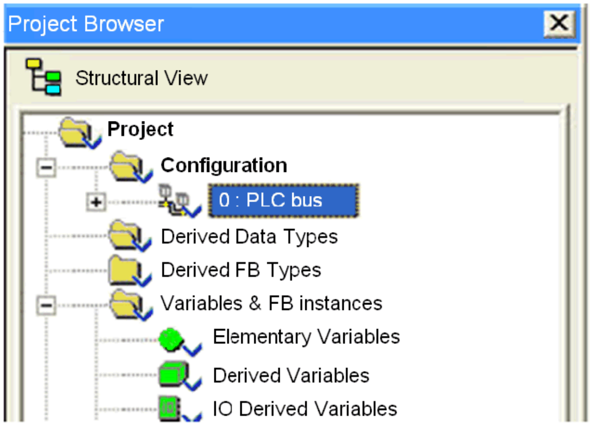
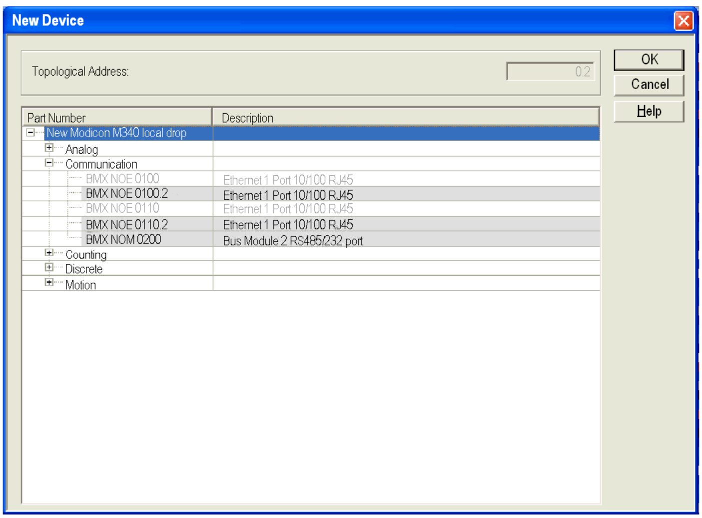
Step	Action	Result
1	Plug the power supply into the leftmost position on the rack.	See the above figure.
2	Add the CPU in the next rack slot.	The example shows the BMX P34 2020 at rack address 0.
3	Place other modules into the remaining available rack slots.	The example shows the BMX NOE 0100 at rack address 1.

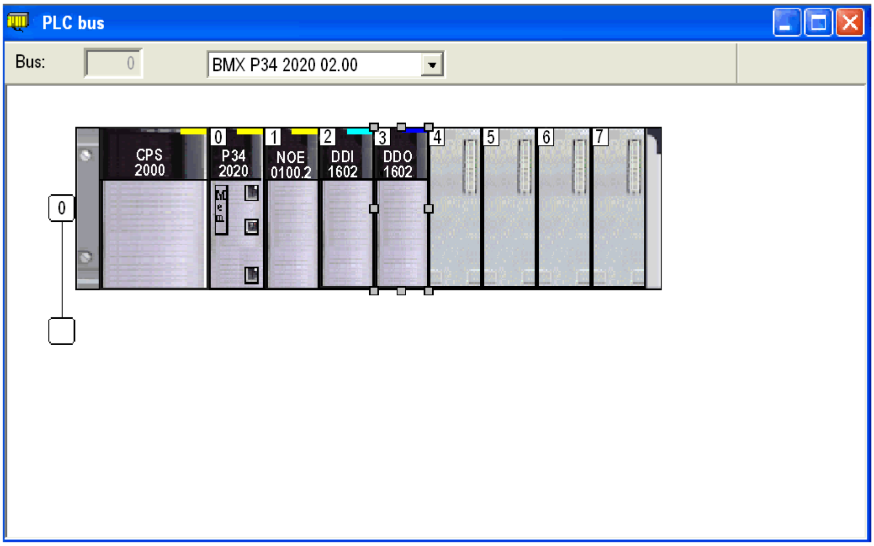
Modicon M340 Configuration with Control Expert

Instructions

Use these instructions to configure an M340 Ethernet module with Control Expert:

Step	Action
1	Open the Control Expert software from the **Start** menu. **Note:** The name of your Control Expert package (*Control Expert M*, *Control Expert L*, *Control Expert XL*, etc.) may vary.
2	In the **File** menu, select **New...** to create a new project.
3	In the **New Project** screen, expand the **Modicon M340** family to select the installed processor: 

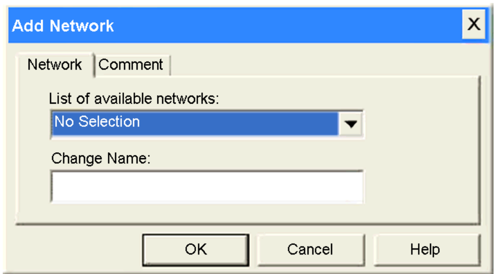
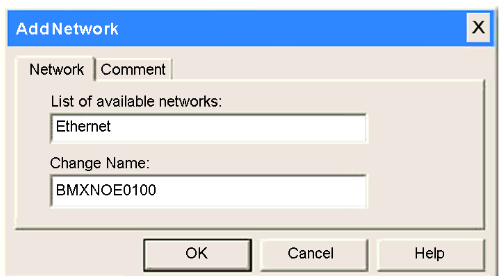
Step	Action
4	In the **Project Browser** screen, double-click **Project** → **Configuration** → **PLC bus** to access the configuration of the local rack: 
5	Double-click each slot to show the hardware catalog. Choose the appropriate module part numbers. Drag or double-click the part numbers from the catalog list to insert modules into the local rack: 

Step	Action
6	A sample view of a finished rack assembly shows the BMX P34 2020 at rack address 0 and the BMX NOE 0100 at address 1: 

Ethernet Network Configuration with Control Expert

Instructions

Add a new Ethernet network:

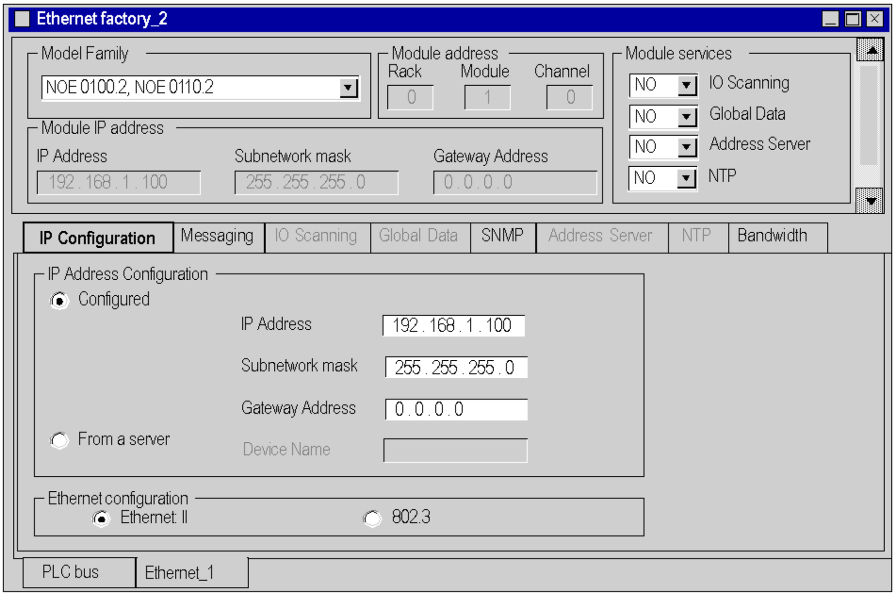
Step	Action
1	In the **Project Browser**, right-click the **Network** subdirectory in the **Communication** directory and select the **New Network** option. The **Add Network** screen appears: 
2	In the **List of available networks**, scroll to **Ethernet** and enter a meaningful network name in the **Change Name** field. (We used the name *BMXNOE0100* in our example). 
3	Click **OK**.

Assign the BMX NOE 0100 IP Address

Assign IP Parameters

Assign IP parameters to the M340 Ethernet communications module:

<table>
<tr><th>Step</th><th>Action</th><th>Comment</th></tr>
<tr><td>1</td><td colspan="2">From the Project Browser, open the new logical network (BMXNOE0100 in this example) under Communication → Networks. The BMXNOE0100 network configuration screen appears:



Note: The example uses the BMX NOE 0100 module. Use these same steps to configure IP parameters for M340 CPUs with Ethernet ports (BMX P34 2020 and BMX P34 2030/20302).</td></tr>
<tr><td>2</td><td>In the Model Family list, slect the appropriate module family.</td><td></td></tr>
<tr><td>3</td><td>On the IP Configuration tab, select Configured.</td><td>You can now manually configure IP parameters.</td></tr>
<tr><td>4</td><td>Enter the appropriate values in the IP address, Subnetwork mask, and Gateway address fields. (For safety reasons, consult your network administrator who can assign network parameters.)</td><td>Example parameters:
● IP address: 192.168.1.100
● subnetwork mask: 255.255.255.0
● gateway address: 0.0.0.0</td></tr>
<tr><td>5</td><td>Click the validate toolbar icon to confirm the IP configuration setting.</td><td></td></tr>
</table>

Configure Ethernet Communication Service (I/O Scanning)

Introduction

The Modicon BMX NOE 01x0 modules support Ethernet communication services (such as I/O scanning, global data, Modbus messaging, SNMP, etc.).

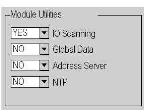
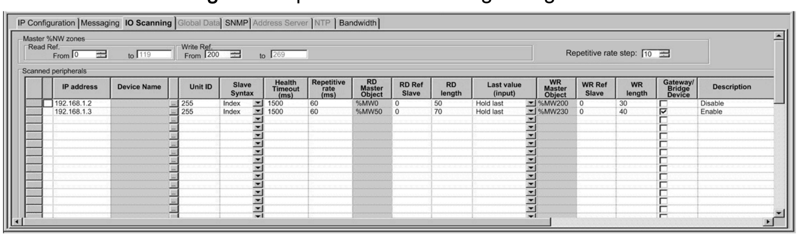
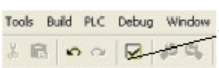
I/O Scanning Service

This example shows you how to configure one service, I/O scanning. Use this service to:

- Transfer data between network devices
- Allow a CPU to regularly read data from and write data to scanned devices

Accessing I/O Scanning

Configure the I/O scanning service with the Control Expert software:

Step	Action
1	Open your application using the **BMX NOE 01x0** in Control Expert.
2	In the Project browser, locate the **Communication\Networks** sub directory.
3	Click the Ethernet module (**BMX NOE 01x0** in our example) to open the Ethernet module configuration screen.
4	Select **Yes** in the **Module Utilities** menu: 
5	Select the **IO Scanning** tab to open the I/O scanning configuration screen. 
6	Enter the parameter settings under each of the column headings for one line of the I/O Scanner Configuration. Refer to I/O Scanning Parameters (below) to see the settings used for this example.
7	Click the validate check box in the upper tool bar to confirm the I/O scanning parameter settings: 

I/O Scanner Parameters above I/O Scanner Table

A description of the parameters above the I/O scanning table used in the example are listed in the following table:

Parameter	Field	Description
Read Ref.	*From* and *to* data boxes	The values in these boxes define the range of destination address values in the CPU for the data read from each device. The addresses you enter here are displayed in the **RD Master Object** column of the dialog. In the example above, the **Read Ref.** values range from 0 to 599; notice that these values are displayed as %MW0, %MW599, etc. in the **Master Object** column.
Write Ref.	*From* and *to* data boxes	The values in these boxes define the range of source address values in the CPU. The address you enter here is displayed in the **WR Master Object** column. In the example above, values starting at %MW2000 are shown in the **WR Master Object** column.
Repetitive Rate Step	data box	The **Repetitive Rate Step** is set in multiples of 5 ms (the minimum) through 200 ms (the maximum). The **Repetitive Rate** column is where you enter a rate of time for how often you want the I/O scanner to send a query to the device after the rate has timed out. **NOTE:** The **Repetitive Rate** of the I/O scanner table is a multiple of the rate displayed in the **Repetitive Rate Step**. The real repetitive rate being executed by the I/O scanner service is shown in the **Repetitive Rate** column. **Note:** An entry in the **Repetitive Rate** column is rounded up to the next multiple that was entered in the **Repetitive Rate Step** box if the entry is not a multiple of the **Repetitive Rate Step**. For example, if the entry in the **Repetitive Rate Step** is 5 and you enter a 7 in the **Repetitive Rate** column, the 7 is rounded up to 10; if you change the **Repetitive Rate Step** to 6 and enter a 7 in the **Repetitive Rate** column, the 7 is rounded up to 12.

I/O Scanner Table Parameters

A description of the parameters in the I/O scanning table used in the example are listed in the following table:

Parameter	Description	Example
Entry #	This is the first column; it has no name. Valid range: 1 ... 64 Each entry represents an I/O Scanning exchange on the network.	
IP Address	This is the IP address of the scanned Ethernet slave device.	192.168.1.100
Device Name	To configure a device (Advantys island or DTM), click the **...** button to open the **Property** box *(see page 168)* to start the device configuration software. For an introduction to this procedure for Advantys, go here *(see page 158)*. For an introduction to this procedure for DTMs, go FDT Container *(see EcoStruxure™ Control Expert, Operating Modes)*. **NOTE:** While the **Property** box is open, **I/O scanning** cannot be edited.	MySTB1 or Master_PRM_DTM_10
Unit ID	This field associates the slave address of the device connected to an Ethernet/Modbus gateway with the IP address of that gateway: • Value range: 1 to 255 • Default value: 255 When using a bridge, enter the bridge index (1 to 255) in this field.	255

Slave Syntax	Use this drop-down menu to pick the way **RD Ref Slave** and **WR Ref Slave** values are displayed. There are 4 types of display available:
	• Index: 100
	• Modbus: 400101
	  ○ (Modbus register)
	• IEC 0: %MW100
	  ○ M340 and Premium PLC slaves
	• IEC 1: %MW101
	  ○ Ouantum PLC slaves
		Index (default value)

Health Timeout (ms)	This field sets the maximum interval between the responses from a remote device. After this time period expires, the received data is invalid. The **Health Timeout** must be longer than the **Repetitive Rate** time (ms). For a BMX NOE Ethernet module, it also should be longer than the CPU scan time. • Range: 1ms to 50 s • Interval: 1ms	1500ms
Repetitive rate (ms)	The rate at which data is scanned, from 0...60000 in multiples of the **Repetitive Rate Step**.	60 ms

Parameter	Description	Example
RD Master Object*	Destination address in the master PLC where, from each device, newly read information is stored. This parameter cannot be accessed. It is calculated automatically as the sum of: ● The **From** address (beginning) of **Read ref.** (in the zone above the table) ● The **RD length** value (in the table below)	%mw10
RD Slave Ref.**	Source address index in the slave/remote device	The format of this value depends on the **Slave Syntax**: ● Index: 5 ● Modbus: 400006 ● IEC 0: %MW5 ● IEC 1: %MW6
RD length	Number of words to read	10
Last value (Input)	This field configures the behavior of inputs in the event of an access error in relation to the remote device (for example: inoperative network or device power supply, etc.): ● Set to 0: fall back to 0 ● Hold last: maintain last value	Hold last
WR Master Object*	Source address of the master PLC whose data is being written into the slave/remote device. This parameter cannot be accessed. It id calculated automatically as the sum of: ● The **From** address (beginning) of **Write ref.** (in the zone above the table) ● The **WR length** value (in the table below) Write operations are always performed at the word level.	%mw20
WR Slave Ref.**	The address of the first word written into the slave/remote device.	The format of this value depends on the **Slave Syntax**: ● Index: 1 ● Modbus: 400002 ● IEC 0: %MW1 ● IEC 1: %MW2
WR length	Number of words to be written	10
Description	Additional information	
*Master refers to the client PLC that makes the request. **Slave refers to the server from which data is read or to which data is written.		

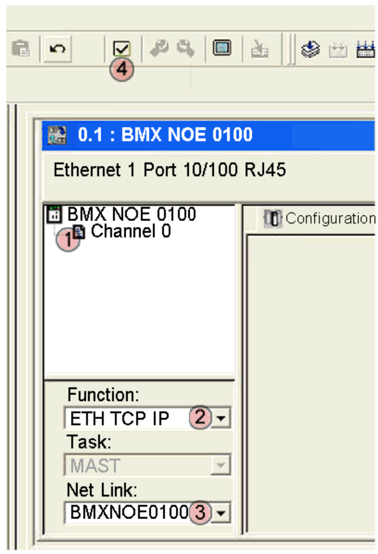
NOTE: Refer to Information about **I/O Scanning** with Multiple Lines *(see page 156)*.

NOTE: Refer to Information about the **I/O Scanning** table Contextual Menu for Copy/Cut/Paste *(see page 154)*.

Associate the Network with the Module

Instructions

Associate the new logical network with the BMX NOE 0100 module:

Step	Action	Comment
1	In the **Project Browser**, double-click **PLC Bus** to show the rack configuration.	
2	Double-click the BMX NOE 0100 module.	The network link screen appears.
3	Under **BMX NOE 0100**, click **Channel 0** (item 1 in the figure) to display the function.	
4	At **Function** (item 2), scroll to **ETH TCP IP**.	
5	Choose the name of your logical network (in this case, **BMXNOE0100**) from **Net Link** (item 3).	
6	Click the validate toolbar icon (item 4) to confirm the network link (Net Link) configuration.	

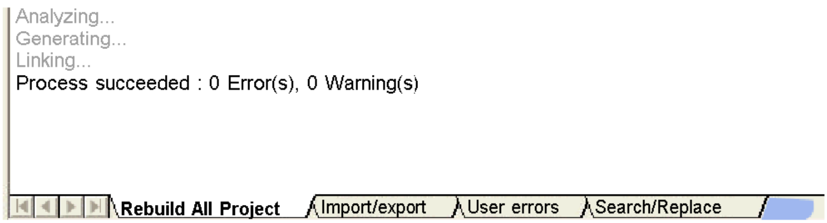
Build a Program

The Build Command

Before building the program, make sure you select the standard mode, not the simulation mode, on the toolbar.

Build the entire program before downloading it to your PLC.

Choose **Build → Rebuild All Project** to build the program:



The program should build without errors.

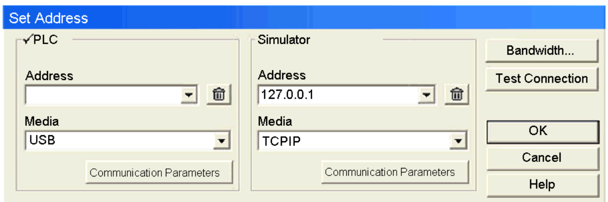
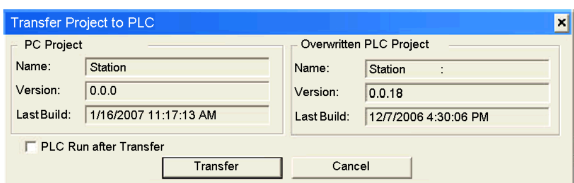
Connect the System and Download the Configuration

Introduction

This topic tells you how to connect the M340 system to the Control Expert software and download the configuration program.

Connect and Download

After building the program you can download it to the PLC. To do this, link the PLC to the Control Expert software through USB (cable), Ethernet, or Modbus. This example uses a USB cable to connect Control Expert to the M340 system as an example:

Step	Action
1	Make sure the M340 system is powered up.
2	Open the **Set Address** screen by choosing the Control Expert tab. Select **PLC → Set Address**. The **Set Address** screen appears: 
3	In the **Media** menu, select **USB**, as shown in the above figure.
4	Press **OK**.
5	On the Control Expert tab, select **PLC → Connect** to link to the M340 system.
6	Open the **Transfer Project to PLC** screen by selecting **PLC → Transfer Project to PLC** on the Control Expert tab: 
7	Click the **Transfer** button to transfer the product.
8	Press **OK** on the confirmation screen.
9	At the Control Expert screen, press **Run** to run the program.

Debugging the Module

Introduction

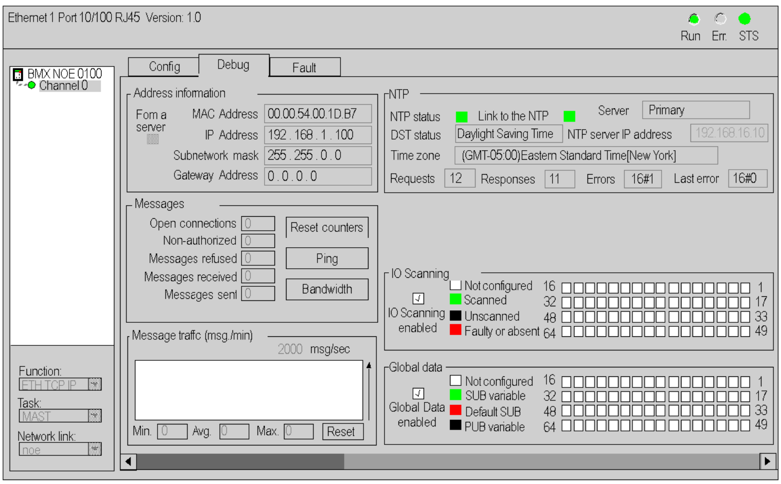
You can debug the Ethernet module by examining the physical LEDs on the front of the module or the debug screen in the Control Expert software. This topic describes the latter, the debug screen.

Access the Debug Screen

To access Control Expert’s debug screen:

Step	Action
1	On the Control Expert tab, select **PLC** → **Connect** to link to the M340 system.
2	In the **Project Browser**, double-click BMXNOE0100 at **Station** → **Configuration** → **PLC bus**.
3	Select the **Debug** tab to display the debug screen (below).

The Control Expert debug screen:



NOTE: The debug screen dynamically updates the Ethernet module’s communication services (I/O scanning, global data, etc.).

Part V
Embedded Web Pages

Chapter 15
Embedded Web Pages

Introduction

This chapter discusses the embedded web pages for modules that can communicate on Ethernet networks.

The installed HyperText Transfer Protocol (HTTP) server transmits Web pages between a server and a browser, providing Ethernet communications modules with easy access to devices anywhere in the world through standard browsers such as Internet Explorer or Netscape Navigator.

What Is in This Chapter?

This chapter contains the following sections:

Section	Topic	Page
15.1	M340 Web Pages	268
15.2	Ethernet Service Diagnostics Pages	282
15.3	Class C Services	288

Section 15.1
M340 Web Pages

Introduction

This section describes the Web pages associated with the BMX NOE 0100 and BMX P34 2020/2030 CPUs in terms of appearance and access rights.

What Is in This Section?

This section contains the following topics:

Topic	Page
Introduction to Web Services	269
Embedded HTTP Server	271
BMX NOE 0100 Home Page	272
BMX NOE 0100 Monitoring Page	273
BMX NOE 0100 Diagnostics Page	274
Server Rack Display Page	275
BMX NOE 0100 Setup Page	276
FTP Security Page	277
HTTP and Data Editor (Write) Page	278
Upload MIB File	280
Properties	281

Introduction to Web Services

Enabling Web Services

Depending on the type of memory card *(see page 50)* the slot, the BMX NOE 0100 can access either:
- the basic pages on the Web site
- the user-customizable pages on the Web site

Embedded Ethernet

Schneider Electric communication modules (and CPUs with embedded Ethernet ports) have integrated Web services that communicate on Ethernet TCP/IP using:
- real-time communication functions based on Ethernet TCP/IP
- predefined Web pages for advanced installation diagnostics

When the memory card with class C services is inserted into a communications module, the module provides transparent access to system and application diagnostic information in real time, using Web technologies.

Communication modules integrate TCP/IP services (Modbus messaging, SNMP functions, etc.) and offer:
- standard Web services
- the capacity to host dynamic user-defined Web pages or any document (doc, pdf, etc.) designed to assist maintenance

NOTE: The sample Web screens shown in this chapter are for the BMX NOE 0100 module with class C services *(see page 288)*. They may not represent the exact functionality of other modules.

Creating Web Pages

Web modules have an 16-Mbyte memory (accessed like a hard drive) that you can use to host Web pages and user-defined documents (maintenance manuals, diagrams, etc.) created in standard applications such as Word or Acrobat Reader. These pages can be created using any standard tool that enables creation and editing in HTML format (such as MicroSoft FrontPage).

You may want to create Web pages *(see page 289)*:
- to display and modify all PLC variables in real time
- to create hyperlinks to other external Web servers (documentation, suppliers, etc.)

This function is particularly suited to creating graphic screens for:
- display, monitoring, diagnostics
- generation of real-time production reports
- maintenance help
- operator guides

Diagnostics from Web Pages

The embedded Web server provides Web pages to diagnose the following Transparent Factory / Real Time services:

- Global Data diagnostics
  - status of all Global Data services
  - status of all subscribed and published variables
  - publication/subscription rate
- I/O scanning diagnostics
  - status of all I/O Scanning services
  - status of individual scanned devices
  - actual I/O scanning rate
- messaging diagnostics
  - diagnostic information for Port 502 messaging
- bandwidth monitoring
  - throughput measurement of NOE by service

NOTE: All these pages are protected by the general HTTP password.

Embedded HTTP Server

Introduction

Some Ethernet modules include an embedded Web server, which allows:
- access to PLC data
- diagnostics to be carried out on the entire configuration

All the processor or module data is presented as standard Web pages in HTML format. Access Web pages with Internet Explorer 4.0 or higher running JRE 1.4.1_04 or higher.

None of the functions supplied by the Web site require any prior configuration or programming within the module.

The summary table below shows the various selections possible. According to the type of module, the availability of these functions changes:

Function	BMX NOE 0100	BMX P34 2020	BMX P34 2030/20302
server	X	X	X
predefined pages	X	X	X
user-customized pages*	X	—	—
client site size*	16 MB	—	—
Legend: X: available —: not available *Requires the BMXRWSC016M memory card *(see page 50)*.			

Embedded Server Functions

The functions available on an embedded server are generally as follows:
- Ethernet services *(see page 282)*: These pages report the status of Ethernet network services.
- security *(see page 278)*: This page is used to modify the user name and the password to access the site.
- rack display *(see page 275)*: This page enables you to display the configuration of the PLC which is controlling the module.
- data editor *(see page 187)*: This page allows you to display PLC data.
- diagnostics *(see page 274)*: This pages allow network diagnostics.

BMX NOE 0100 Home Page

Home Page

Access the BMX NOE 0100 home page by entering the IP address of the module in a web browser. (No password is required to display the home page.)

Links

From the BMX NOE 0100 home page, you can access the following pages:

- Monitoring *(see page 273)*
- Diagnostics *(see page 274)*
- Setup *(see page 276)*

BMX NOE 0100 Monitoring Page

Monitoring Page

From the BMX NOE 0100 home page *(see page 272)*, click the **Monitoring** link to display this page:

Links

From the BMX NOE 0100 Monitoring page, you can access the following Data Editor pages:

- **Lite**: This smaller version of the Data Editor loads faster, but can access most of the same Modicon M340 PLC data.
- **Standard**: Allows access to Modicon M340 PLC data.

BMX NOE 0100 Diagnostics Page

Diagnostics Page

From the BMX NOE 0100 home page *(see page 272)*, click the **Diagnostics** link to display this page:

Links

From the BMX NOE 0100 Diagnostics page, you can access the following pages:

- **Rack viewer** (**Lite**): See the description for the Rack Viewer page *(see page 275)*.
- **Ethernet**: You can diagnose the status of Ethernet services through these links:
  - Global Data *(see page 283)*
  - IO Scanning *(see page 284)*
  - Messaging *(see page 285)*
  - Statistics *(see page 286)*
  - Bandwidth Monitoring *(see page 287)*
  - Upload MIB file *(see page 280)*
- Properties *(see page 281)*

Server Rack Display Page

Introduction

This page allows you to carry out diagnostics on the modules in the local rack configuration that includes the Ethernet module.

By clicking on a module in the configuration, you obtain a set of diagnostic information on this module:

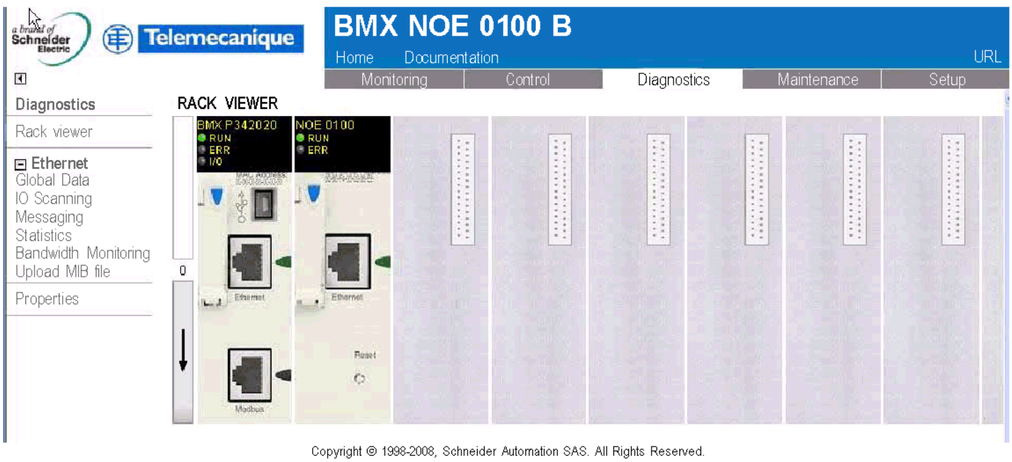
- LED status display
- the type and the version of the module as well as its position in the rack
- information specific to the functions of the module

Accessing the Server Rack Display Page

Follow this procedure to access the rack display page from the index page:

Step	Action
1	Click the **Diagnostics** link.
2	Click the **Rack Viewer** link.

The NOE 0100 rack display page appears. An example is shown below:



BMX NOE 0100 Setup Page

Setup Page

From the BMX NOE 0100 home page *(see page 272)*, click the **Setup** link to display this page:

Links

From the BMX NOE 0100 **Setup** page, you can access the following password pages:

- Security *(see page 278)*

FTP Security Page

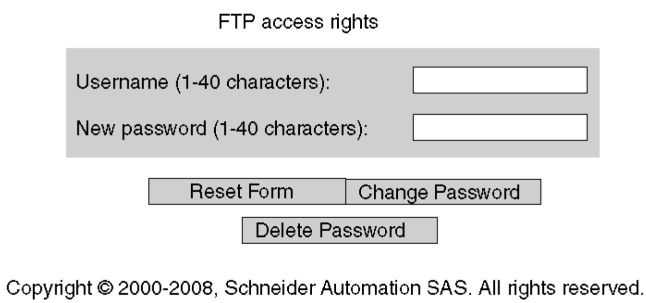
Introduction

You can modify the username and password for FTP access rights on this page.

NOTE: You can download Web pages to the C type memory card over FTP.

FTP Page

The Setup page *(see page 276)* has a link to the FTP password page:



To modify the FTP username and password:

Step	Action
1	Enter the new username. (The default is USER.)
2	Enter the new password. (The default is USER.)
3	Confirm the new password by entering it again.
4	Confirm the modification using the **Change Password** button.

HTTP and Data Editor (Write) Page

Introduction

Access this page with the **Security** link on the Setup page *(see page 276)*. Use the Security page to:

- modify the user name and the password for accessing the index page
- modify the password for writing variables in the data editor (You can read the data editor values without a password.)

The maximum size of the user name or passwords is 16 characters (non-extended ASCII).

The Security Page

The security page appears:

Modify the HTTP access rights:

Step	Action
1	Enter the new username (default is USER).
2	Enter the new password (default is USER).
3	Confirm the new password by entering it again.
4	Confirm the modification using the **Change Password** button. **Result**: An **Ethernet Configuration** page appears.
5	Click the **Reboot Device** button to recognize the modification in the module.

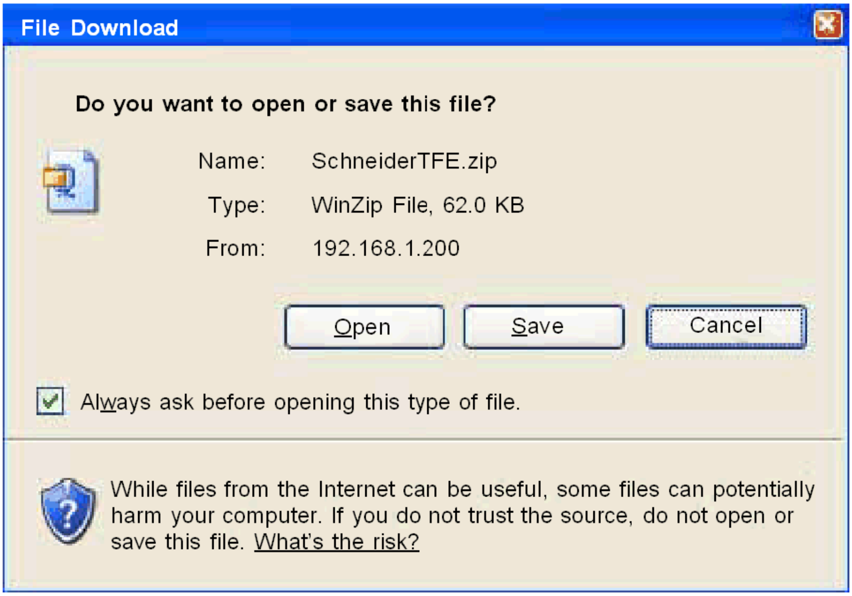
Modify the Data Editor Write Password:

Step	Action
1	Enter the current password (case sensitive). (The default value of this field is: USER.)
2	Enter the new password (default is USER).
3	Confirm the new password by entering it again.
4	Confirm the modification with the **Change Write Password** button. **Result**: An **Ethernet Configuration** page appears, indicating that the password has been modified.

Upload MIB File

File Download Dialog

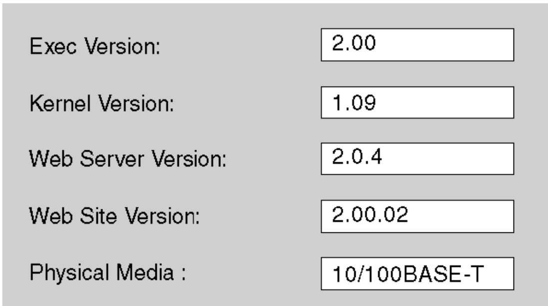
When you select **Upload MIB File**, the **File Download** dialog box appears. You are asked if you want to save the MIB file or open it:



Properties

Dialog Box

The **Properties** dialog box is accessible through a link from several embedded Web pages. The **Properties** link reports the properties of the Web pages:



Section 15.2
Ethernet Service Diagnostics Pages

Introduction

You can link to the screens in this section to diagnose the performance of Ethernet services. Access these screens through the Ethernet menu on the Web pages associated with your module.

What Is in This Section?

This section contains the following topics:

Topic	Page
Global Data	283
I/O Scanning	284
Messaging	285
Statistics	286
Bandwidth Monitoring	287

Global Data

Diagnostics Page

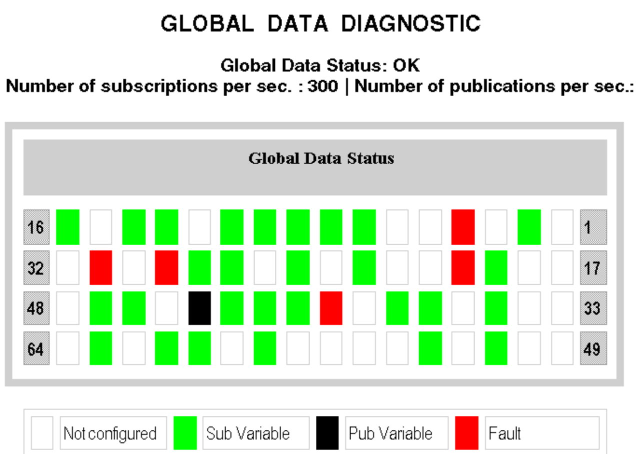
Click this link to see these Global Data diagnostics:

- status
- number of publications per second
- number of subscriptions per second

This page also shows a table that regroups all published and subscribed variables in the same distribution group. The nature of each variable is identified by its color code:

- green
  subscribed variables
- black
  published variables
- white
  unconfigured variables
- red
  variables with communication faults

View of the Global Data diagnostics page:



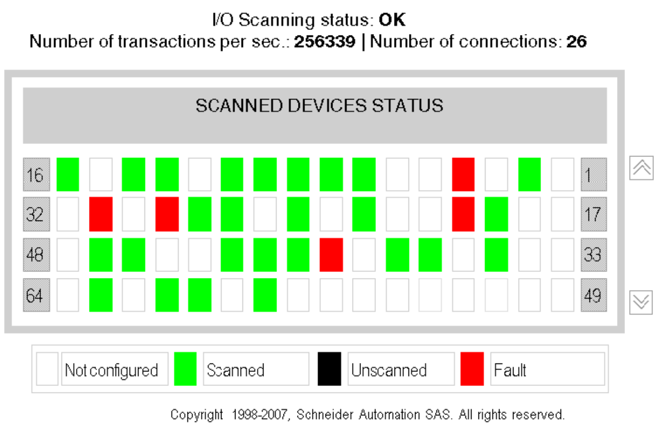
I/O Scanning

Diagnostics Page

Click this link to see these diagnostics of the I/O Scanner:
- status
- number of transactions per second
- number of connections per second

This page also displays a summary of the status of all modules:
- **green**: scanned modules
- **black**: unscanned modules
- **white**: unconfigured modules
- **red**: faulty modules

View of the I/O scanning diagnostics page:



Messaging

Diagnostics Page

Click this link to see the current information on the open TCP connection on port 502.

MESSAGING DIAGNOSTIC

Number of Messages sent: 38 | Number of Messages received: 183

Conn.#	Remote address	Remote Port	Local Port	Mess. Sent	Mess. Received	Error Sent
1	192.160.10.20	1920	502	20	12	0
2	139.160.235.90	2020	502	0	30	02
3	192.160.10.21	502	3000	3	60	0
4	139.160.234.20	1050	502	15	42	0
5	139.160.234.18	5120	502	0	39	1

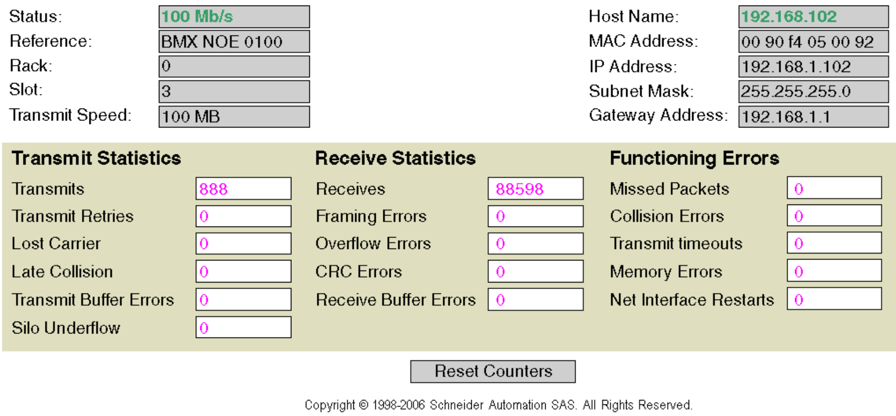
The number of sent/received messages on the port can be found at the top of this page. A table provides, for each connection (numbered from 1 to 64):

- Remote address
  remote IP Address
- Remote Port
  remote TCP port
- Local Port
  local TCP port
- Mess. Sent
  number of messages sent from this connection
- Mess. Received
  number of messages received from this connection
- Error Sent
  error number on this connection

Statistics

Diagnostics Page

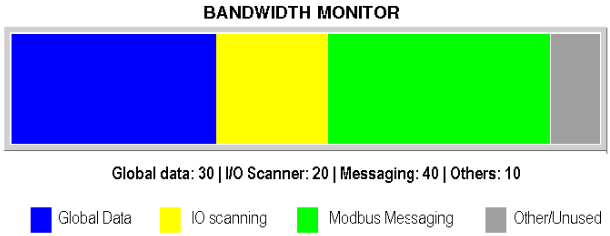
This page shows the Ethernet module statistics that are used to diagnose network activity:



Bandwidth Monitoring

Diagnostics Page

Click this link to see the load distribution of the module among services (Global Data, I/O Scanning, Messaging, and others.) The distribution of the load among services is represented as a percentage:



Section 15.3
Class C Services

Introduction

This section describes the Class C Web services that are applicable to the BMX NOE 0100 module.

What Is in This Section?

This section contains the following topics:

Topic	Page
Introduction to Class C Services	289
Graphic Editor Overview	290
Top Window User Functions	292
Display Window User Functions	297
Property Sheet	300
Security	302
Graphic Editor Applet Parameters	303
Graphic Objects	304
Extended Graphic Objects	321

Introduction to Class C Services

Overview

Features of Class C services for the BMX NOE 0100 module:

- Using a simple Internet browser, class C services allow you to control, configure, and monitor plant data locally and remotely. Monitoring and control can be enhanced with user-customized Web pages.
- Class C services provide the functions and features of the Graphic Editor, a Java applet that enables you to create dynamic graphic displays with a Web browser, using a set of pre-defined graphic objects. The Graphic Editor is only used as an editor for creating and modifying displays. The Graphic Viewer is the run-time environment for viewing displays while they are dynamically animated with run-time data from the PLC. The viewer is lighter than the editor, resulting in faster loading times and operation.

Download Web Pages to Memory Card

The BMX NOE 0100 has an embedded FTP server. Classs C services allow you to use any FTP client (like Windows Explorer) to download Web pages or user-defined documentation to the memory card through the FTP security page *(see page 277)*.

You can also change the FTP password.

NOTE: Downloading write-protected files to the memory card can prevent the Unity loader from properly upgrading the module. Some FTP clients (for example, the Windows Explorer client) can not remove write-protected files from the card. You can delete write-protected files from the card with some FTP clients.

Graphic Editor Overview

The Interface

The Graphic Editor applet is separated into three windows:
- Top Window: This window provides the area for presenting all the user controls and dialogs for creating, saving, reading, and editing a graphic display.
- Display Window: This window provides an area for presenting the current graphic display. When you create a new graphic display, this window becomes an "empty canvas" on which you can add the graphic objects that will compose the desired graphic display.
- Message Window: This window presents any messages generated by the Graphic Editor.

Illustration

The figure below shows the Graphic Editor applet with its initial top window, and empty display and message windows.

Graphic Objects

All graphic objects provided with the Graphic Editor are capable of communicating with the PLC from which the Graphic Editor applet was downloaded. There is no additional "wiring" of the graphic objects with "communication objects." All graphic objects are standalone, meaning there are no connections required between them and each is capable of operating independently.

Viewing a Graphic Display

After the Graphic Editor applet has been uploaded to a Web browser, you will usually want to either view a graphic display (for monitoring/controlling the PLC application) or create or modify a graphic display. A user who only wants to view and interact with existing graphic displays (e.g., an operator) can select the Graphic Viewer link instead of Graphic Editor. They will see a window with the widgets that does not include the Edit menu. This viewer loads faster than the standard Graphic Editor because it is lighter. You need only to enter a password to write data to the PLC.

Create and Modify Graphic Displays

To create and modify graphic displays, click the **Edit...** button to see the standard graphic editor functions. These include selecting objects from a palette, dropping them onto a canvas, moving and resizing them with a mouse, and setting object properties. You can immediately test the modified graphic display with run-time data from the PLC by clicking the **Done** button to exit edit mode. When satisfied with your creation, the graphic display can be saved to the PLC for re-use by clicking the **Save...** button, assuming you entered the correct password.

User Functions

Most of the Graphic Editor's user functions are available as Top Window User Functions *(see page 292)*. From the display window, you can directly manipulate a graphic object's size and location. All properties of a graphic object (e.g., scaling values, labels, colors, PLC addresses of the run-time data) are set in the Property Sheet *(see page 300)*.

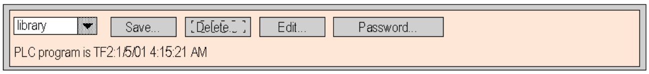
Top Window User Functions

Overview

The Graphic Editor applet's top window consists of several "dialog panels," only one of which is visible at any time. Switching from one dialog to another is done by clicking buttons on the current dialog. This topic describes the dialog panels that compose the top window.

Top Dialog

The **Top Dialog** is the dialog panel that is initially shown in the top window when the Graphic Editor applet is started. Access to other dialog panels of the top window is from this dialog.

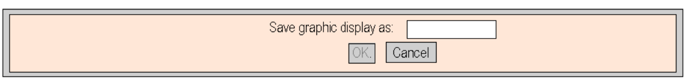


The controls of the **Top Dialog** provide the following functions:

- **Drop-down List**. The drop-down list box shows graphic display files that have been saved to the Web server module and are available for retrieval. When you select a graphic display from this list, the graphic display currently in the window is replaced with the selected one. If the current graphic display has been modified since it was last saved, you are asked for confirmation that the changes are to be discarded. If the special entry <new> is chosen from the list, the display window is cleared and a new graphic display can be created.
- **Save**. The **Save** button makes the **Save Dialog** visible. This button is disabled until you have entered a correct write-enabled password.
- **Delete**. The **Delete...** button makes the **Delete Dialog** visible. This button is disabled until you have entered a correct password, or if the current graphic display has not yet been saved.
- **Edit**. The **Edit...** button makes the **Edit Dialog** visible.
- **Password**. The **Password...** button makes the **Password Dialog** visible.
- **Information display area**. The information display area shows the name and version of the Concept, PL7, or Control Expert program that is running in the connected PLC.

Save Dialog

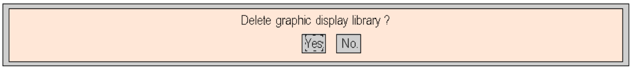
The **Save Dialog** allows you to save the current graphic display.



When the **Save Dialog** is presented, the name of the current graphic display is shown in the dialog's text field. If the current graphic display has not been saved (i.e., a "new" graphic display), then the text field is blank. Once you have either accepted the current name (with a "save" operation) or provided a new name (with a "save as" operation), then you can click the **OK** button to save the contents of the current graphic display to the Web server module. The **Cancel** button will cause the **Top Dialog** to be shown again, with no action being taken.

Delete Dialog

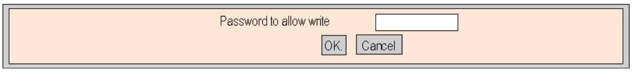
The **Delete Dialog** allows you to delete the current graphic display.



If you click **Yes**, the existing graphic display window is cleared and the graphics file on the Web server module is deleted. Clicking **No** will cause the **Top Dialog** to be shown again, with no action being taken.

Password Dialog

The **Password Dialog** allows you to enter the password that enables those user functions that modify graphic display files or PLC run-time data values.

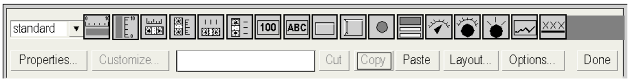


If you enter the correct password and click **OK**, then you will be allowed to save and delete the current graphic display. Correct entry of the password also permits you to write new values to the PLC (via those graphic objects that support writing values to a PLC, if any). Clicking **OK** when the text field is empty clears the current password permissions (if there are any). The **Cancel** button redisplays the **Top Dialog** without changing the current password permissions.

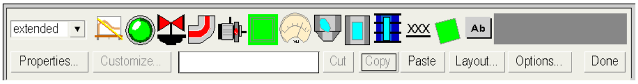
Edit Dialog

The **Edit Dialog** allows you to select a graphic object for placement in the display window, and provides access to graphic editing functions. The available graphic objects are presented in a set of palettes, with one palette visible at a time. There are two palettes.

The standard palette:



The extended palette:

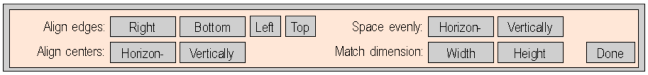


The controls of the **Edit Dialog** provide the following functions:

- The **Drop-down List Box** shows the set of available palettes. When you select the name of a palette from the list, the graphic objects in that palette appear in the palette display area of the dialog.
- The **Palette** shows the graphic objects in the current palette. An icon depicts each graphic object's type (meter, button, etc.). When you click any icon in the palette, a graphic object of the corresponding type is selected for insertion. If you click in an open area of the display window while the Graphic Editor is in "insert mode," an instance of the selected graphic object is inserted into the graphic display.
- The **Information Area** shows the name and size of the graphic object that is currently selected.
- The **Cut** button causes the currently selected graphic object(s) to be removed from the graphic display and saved to a buffer (i.e., an internal clipboard), replacing the existing contents of the buffer.
- The **Copy** button causes the currently selected graphic object(s) to be copied to the buffer, replacing the existing contents of the buffer.
- The **Paste** button causes the content of the clipboard to be inserted into the upper left corner of the graphic display. The pasted graphic objects can then be moved to the desired location in the display.
- The **Properties** button displays the currently selected graphic object's Property Sheet.
- The **Customize** button displays the currently selected object's Customizer *(see page 299)* (if the graphic object has one).
- The **Layout** button makes the **Layout Dialog** visible.
- The **Options** button makes the **Options Dialog** visible.
- The **Done** button makes the **Top Dialog** visible again.

Layout Dialog

The **Layout Dialog** allows you to change the position and size of a group of graphic objects.



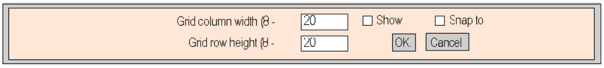
The controls of the **Layout Dialog** provide the following functions:

- For aligning the edges of graphic objects, the **Right**, **Bottom**, **Left**, and **Top** buttons move the selected graphic objects so that their specified sides are at the same position. Select at least two graphic objects to enable these buttons.
- For aligning the center lines of graphic objects, the **Horizontally** and **Vertically** buttons move the selected graphic objects so that their horizontal or vertical center lines, respectively, are at the same position. Select at least two graphic objects to enable these buttons.
- For positioning graphic objects so that they are evenly spaced, the **Horizontally** and **Vertically** buttons move the selected graphic objects so that either the horizontal or vertical spacing between the objects is the same. Select at least three graphic objects to enable these buttons.
- To automatically size graphic objects, use the **Width** and **Height** buttons to re-size the currently selected graphic objects so that either the widths or heights, respectively, of the objects match. Select at least two graphic objects to enable these buttons.
- The **Done** button makes the **Edit Dialog** visible again.

NOTE: For most layout operations (except **Space evenly**) one of the selected objects is considered the "reference object" to which the other selected objects adjust for their new position or dimension. For example, when the "Width" button is pressed, the selected objects have their widths changed to match that of the reference object. The reference object is differentiated from the other selected objects by making its selection box a different color than the others.

Options Dialog

The **Options Dialog** allows you to change the settings related to a grid that can be drawn in the display window. The grid is solely for assistance in editing a graphic display and is shown only when the Graphic Editor is in "edit mode." Edit mode starts when you switch to the **Edit Dialog** and ends when you return to the **Top Dialog**.



The **Options Dialog** controls provide the following functions:

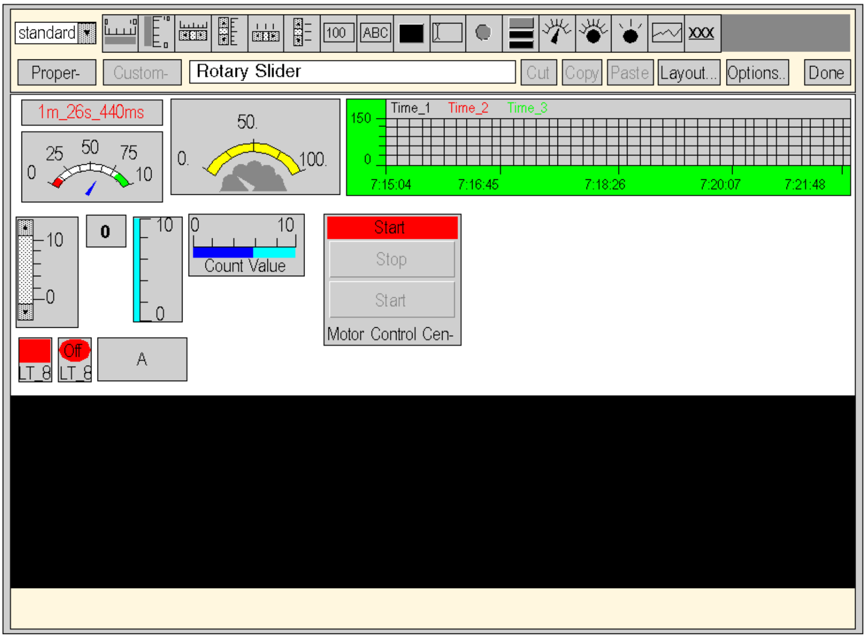
- The cell size of the grid can be changed by the entering the grid's column width and row height in the dialog's text fields.
- If the **Show grid** check-box is checked, the grid is drawn; otherwise, no grid is shown.
- If the **Snap to grid** check-box is checked, then, when you change the size or position of a graphic object, the changed coordinate(s) or dimension(s) is automatically adjusted to coincide with a grid point.
- The **OK** button causes the current option settings to become active, and the **Edit Dialog** to be shown again.
- The **Cancel** button causes the **Edit Dialog** to be shown again, with no option settings being changed.

Display Window User Functions

Overview

The user functions available from the **Graphic Editor** display window allow you to select, move, and size objects. Moving and sizing operations require that your first select those graphic object(s) that you want to modify. A selected object is indicated by its surrounding selection box; an unselected or deselected object has no surrounding selection box.

The figure below shows the **Graphic Editor** display.



Selecting Graphic Objects

A graphic object’s selection state (selected/deselected) can be set via the following user actions:

- A single graphic object can be selected by simply clicking on it with a mouse. If any other objects are currently selected, they will be deselected.
- Multiple graphic objects can be selected with a selection box in the display window. If you press a mouse button in an open area of the display window (i.e., not on a graphic object) and drag the mouse without releasing it, you will see a dotted outline box. One corner of the box is fixed where the mouse button was initially pressed while the opposite corner tracks the current mouse position. When the mouse button is released, every object that intersects the selection box is selected. Objects outside the box are deselected.
- A graphic object's selection state can be toggled between selected and deselected without affecting the selection state of other objects by pressing the CTRL key when clicking on the object. With this action, graphic objects can be individually added or removed from the group of selected objects.
- A graphic object can be selected without affecting the selection state of any other objects by pressing the SHIFT key when clicking on the object. When an object is selected this way, it becomes the *reference object* (see Layout Dialog *Top Window User Functions,* *page 292*) for the group of selected objects. The primary purpose of this action is to change the reference object in a group of selected objects prior to invoking one of the **Layout** operations.
- Previously selected graphic objects can be deselected by clicking the mouse in an open area of the display window, that is, not on a graphic object.

Sizing Graphic Objects

A graphic object's size can be changed by first selecting it, then using the mouse to change the size of the object's selection box. As you move the mouse over an object's selection box, the mouse pointer changes to reflect the type of sizing operation to be performed. If you press a mouse button while the mouse is over an object's selection box and drag the mouse without releasing it, a dotted outline box appears. When the mouse button is released, the object's size is changed to match the size of the outline. There are eight possible sizing actions depending on which part of an object's selection box is dragged. Each corner of the box will allow only its adjacent sides to move; each side of the box will allow only that side to move.

Moving Graphic Objects

A graphic object can be moved in the display window with the mouse. If you press a mouse button while the cursor is over an object and drag without releasing the button, then a selection box will be shown. When the mouse button is released, the object moves to the location of the selection box.

Multiple graphic objects can be moved by first selecting the objects to be moved, and then dragging the entire group of objects in the same way a single object is moved. While a group of objects is moved, a selection box is shown for each object in the group.

Setting Graphic Object Properties

You can set a graphic object's properties via the Property Sheet. If the Property Sheet is visible, the properties of the selected graphic object are presented for editing. Display the Property Sheet by pressing the **Properties...** button or by double-clicking the mouse anywhere on the selected object in the display window.

Customizing Complex Graphic Objects

Some complex graphic objects have a very large number of properties. Configuring such an object with the Property Sheet can be cumbersome. An available Customizer can ease the configuration of complex graphic objects. A Customizer is a dialog window designed specifically to configure its associated graphic object. When the Graphic Editor detects that a selected graphic object has a Customizer, it will enable the **Customizer...** button, which brings up the graphic object's Customizer. When you double-click on a graphic object that is associated with a Customizer, the Customizer comes up (instead of the Property Sheet). If a graphic object is associated with a Customizer, the only item in the Property Sheet is its name.

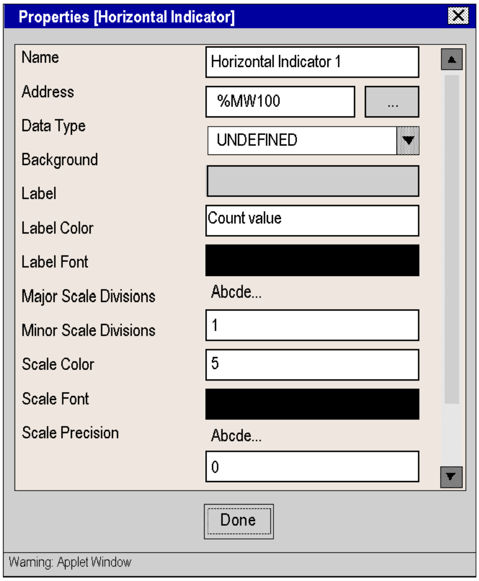
Display Background Image

A Graphic Editor display has a **Background Image** property that can be used to designate an image as the display's background. The image can be a GIF or JPEG file. File locations are relative to the Embedded Server's /wwwroot directory. For example, if the image "cool.gif" was put in the Embedded Server's /wwwroot/images directory, then the Background Image property should be set to /images/cool.gif.

Property Sheet

Overview

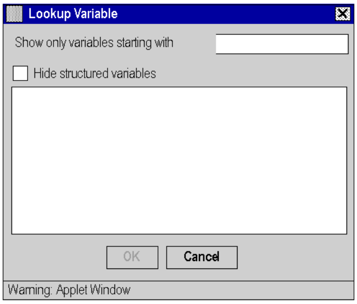
The Property Sheet is a "floating" (non-modal) dialog that shows the configurable properties of the currently selected graphic object:



The properties of a graphic object are specific to an object's type. The properties are presented in a scrollable list, with the name and the value of each property listed. The Graphic Editor comes with a description of graphic objects *(see page 304)*.

Lookup Dialog

For each of the graphic objects provided with the Graphic Editor, a property editor is provided for its **Address** property. This editor not only allows you to directly enter the address of a Quantum/Premium/Micro register (or Concept/PL7/Control Expert variable name), but also provides access to the **Lookup Dialog**. This dialog allows you to pick a Concept/PL7/Control Expert symbol (variable) name from a list of symbolized variables that have been "Web enabled" by the FactoryCast Configurator:



NOTE: The variables window is empty because it is not possible to access variables in this manner at this time.

Security

Three security elements are provided to help you protect your data:

- The HTML page containing the Graphic Editor applet has been placed in the *secure* directory on the Web module, then the Web browser user is asked for a password before being allowed to download the HTML page.
- The **Password** dialog box enables you to save/delete files or to transfer data values. This dialog is password protected. When you transfer data values, the Graphic Editor reinforces the read-only mode by deactivating the user commands related to graphic objects.
- Web Designer for FactoryCast allows you to specify that an item is read-only. The **Graphic Editor** will enforce the read-only attribute of a symbol (variable) or address by rejecting any request to set a new value for the data item, and informing the user in the **Graphic Editor** message window.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

- Do not use graphic objects in a situation where loss of communication to the FactoryCast module can affect human or material integrity.
- Do not use graphic objects in safety critical machine functions.

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

For example, say you have programmed a pushbutton object to jog a motor when the button is depressed and to stop jogging when the button is released. If communications are lost while the button is depressed, the motor will continue to jog even when the button is released. Graphic objects should not be used to control situations such as this unless other interlock methods are installed in the system.

Graphic Editor Applet Parameters

Overview

Three applet parameters can customize the behavior of the **Graphic Editor**. Applet parameters are specified with <PARAM> tags within the <APPLET> tag in the Graphic Editor's HTML page. The parameters recognized by the **Graphic Editor** applet are:

- **LOAD**: This parameter tells the **Graphic Editor** to auto-load a specific graphics file when it starts. If the file does not exist, a message appears. If this parameter is not provided in the <APPLET> tag, then a file is not auto-loaded at startup and you have to select an initial graphics file from the list provided by the **Graphic Editor**.
- **MODE**: This parameter tells the **Graphic Editor** whether to startup in its normal "Edit Mode" or in a special "View Mode." When started in view mode, the **Graphic Editor** shows only its display window. When this parameter is used with the LOAD parameter, a Web site can be designed using HTML pages that are dedicated to specific graphic displays. No explicit selection of graphic files is required by a user, providing more typical HMI screen behavior. The possible values for this parameter are:
  - **EDIT** (default value): The **Graphic Editor** starts up in its normal Edit Mode.
  - **VIEW_RO**: The **Graphic Editor** starts up in read-only view mode. The Web browser user will not be allowed to send data values to the PLC.
  - **VIEW_RW**: The **Graphic Editor** starts up in read/write view mode. The Web browser user will be allowed to send data values to the PLC after entering the write-access password.
- **AUTO_LOGIN**: This parameter tells the **Graphic Editor** to automatically enter the password that is required to permit writing to the PLC. If the **MODE** parameter is set to **VIEW_RW** or **EDIT**, then setting this parameter to **TRUE** will cause the **Graphic Editor** to allow writing to the PLC without requiring the user to enter the password. The possible values for this parameter are **FALSE** (default) and **TRUE**.

Example

The following is an example of an applet tag for the **Graphic Editor** that causes it to start up in view mode while automatically loading a graphics file named **UNIT_1**. In this case, the Web browser allows you to send values to the PLC via any graphic objects that support sending values (assuming you have entered the write-access password).

```
<APPLET codebase="/classes"
archive="SAComm.jar,GDE.jar,Widgets.jar"
code="com.schneiderautomation.gde.GdeApplet"
width="700" height="514">
<PARAM name="LOAD" value="UNIT_1">
<PARAM name="MODE" value="VIEW_RW">
<PARAM name="AUTO_LOGIN" value="FALSE">
</APPLET>
```

Graphic Objects

Overview

The set of graphic objects provided with the **Graphic Editor** supports the construction of graphic displays that mimic conventional instrument panels. The data monitoring and control objects have built-in communication capabilities and are designed as standalone graphic objects.

Be aware, however, that if communication to the device linked to the graphic object is lost, the object becomes inoperative without the end device's knowledge.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

- Do not use graphic objects in a situation where loss of communication to the FactoryCast module can affect human or material integrity.
- Do not use graphic objects in safety critical machine functions.

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

For example, say you have programmed a pushbutton object to jog a motor when the button is depressed and to stop jogging when the button is released. If communications are lost while the button is depressed, the motor will continue to jog even when the button is released. Graphic objects should not be used to control situations such as this unless other interlock methods are installed in the system.

Additionally, each object in the **Graphic Editor** set is available in an applet version to support customers that want to put several simple applets on a single HTML page. When used in conjunction with `LiveBeanApplet`, the **Graphic Editor** graphic objects can be used in the same way as the `LiveLabelApplet`.

This topic describes standard graphic objects and their properties.

Horizontal Indicator

A Horizontal Indicator provides an analog representation of the value of a symbol (variable) or direct address in a PLC. It is a horizontal bar that represents the value as a percentage of its range in engineering units. Optionally, a digital indication of the value can be shown in the center of the bar area.

The following table describes the properties for the Horizontal Indicator:

Property	Description	Limits
Name	The name of the graphic object	
Address	The direct address or the name of a symbol (variable) to monitor	See Note 1, *Notes*, *page 320*
Data Type	The data type of the direct address or symbol (variable)	See Note 2, *Notes*, *page 320*
Background	The background color of the graphic object	
Label	The label to be displayed as part of the graphic object	
Label Color	The color of the label	
Label Font	The font used on the label	
Major Scale Divisions	The number of major (labeled) scale divisions	0 to 100
Minor Scale Divisions	The number of minor (unlabeled) scale divisions	0 to 100
Scale Color	The color of the scale and its labels	
Scale Font	The font used on scale labels	
Scale Precision	The number of fractional digits to be shown for scale labels (set to -1 to use a general exponential format)	-1 to 6
Maximum EU Value	The maximum value, in engineering units, of the direct address or symbol (variable)	
Minimum EU Value	The minimum value, in engineering units, of the direct address or symbol (variable)	
Maximum PLC Value	The maximum raw (unscaled) value of the direct address or symbol (variable) in the PLC	See Note 3, *Notes*, *page 320*
Minimum PLC Value	The minimum raw (unscaled) value of the direct address or symbol (variable) in the PLC	See Note 3, *Notes*, *page 320*
Value Visible	Indicates whether a digital display of the scaled value is to be shown	
Value Font	The font for the digital display of the value (if shown)	
Bar Background	The background color of the bar indicator area	
Bar Color	The color of the indicator bar (when the scaled value within High/Low limits)	
High High Limit Value	The value in engineering units for the ‘High High’ limit	
High High Limit Color	The color of the indicator bar when the scaled value is greater than the ‘High High’ limit	

Property	Description	Limits
High Limit Value	The value in engineering units for the ‘High’ limit	
High Limit Color	The color of the indicator bar when the scaled value is greater than the ‘High’ limit	
Low Limit Value	The value in engineering units for the ‘Low’ limit	
Low Limit Color	The color of the indicator bar when the scaled value is less than the ‘Low’ limit	
Low Low Limit Value	The value in engineering units for the ‘Low Low’ limit	
Low Low Limit Color	The color of the indicator bar when the scaled value is less than the ‘Low Low’ limit	
Limit Deadband	The deadband (as percentage of EU range) to apply to High/Low limit checking	0 to 10
Border Width	The width (in pixels) of the graphic object’s border	0 to 32
Border Color	The color of the graphic object’s border	
PLC Value	A simulated, raw (unscaled) input value for testing the graphic object	See Note 3, *Notes, page 320*

Vertical Indicator

A Vertical Indicator provides an analog representation of the value of a symbol (variable) or direct address in a PLC. It is a vertical bar that represents the value as a percentage of its range in engineering units.

The following table describes the properties of the Vertical Indicator:

Property	Description	Limits
Name	The name of the graphic object	
Address	The direct address or the name of a symbol (variable) to monitor	See Note 1, *Notes, page 320*
Data Type	The data type of the direct address or symbol (variable)	See Note 2, *Notes, page 320*
Background	The background color of the graphic object	
Label	The label to be displayed as part of the graphic object	
Label Color	The color of the label	
Label Font	The font used for the label	
Major Scale Divisions	The number of major (labeled) scale divisions	0 to 100
Minor Scale Divisions	The number of minor (unlabeled) scale divisions	0 to 100
Scale Color	The color of the scale and its labels	
Scale Font	The font used for scale labels	
Scale Precision	The number of fractional digits to be shown for scale labels (set to -1 to use a general exponential format)	-1 to 6

Property	Description	Limits
Maximum EU Value	The maximum value, in engineering units, of the direct address or symbol (variable)	
Minimum EU Value	The minimum value, in engineering units, of the direct address or symbol (variable)	
Maximum PLC Value	The maximum raw (unscaled) value of the direct address or symbol (variable) in the PLC	See Note 3, *Notes,* *page 320*
Minimum PLC Value	The minimum raw (unscaled) value of the direct address or symbol (variable) in the PLC	See Note 3, *Notes,* *page 320*
Bar Background	The background color of the bar indicator area	
Bar Color	The color of the indicator bar (when the scaled value within High/Low limits)	
High High Limit Value	The value in engineering units for the ‘High High’ limit	
High High Limit Color	The color of the indicator bar when the scaled value is greater than the ‘High High’ limit	
High Limit Value	The value of the ‘High’ limit in engineering units	
High Limit Color	The color of the indicator bar when scaled value is greater than the ‘High’ limit	
Low Limit Value	The value of the ‘Low’ limit in engineering units	
Low Limit Color	The color of the indicator bar when the scaled value is less than the ‘Low’ limit	
Low Low Limit Value	The value of the ‘Low Low’ limit in engineering units	
Low Low Limit Color	The color of the indicator bar when the scaled value is less than the ‘Low Low’ limit	
Limit Deadband	The deadband (as percentage of EU range) to apply to High/Low limit checking	0 to 10
Border Width	The width (in pixels) of the graphic object’s border	0 to 32
Border Color	The color of the graphic object’s border	
PLC Value	A simulated, raw (unscaled) input value for testing the graphic object	See Note 3, *Notes,* *page 320*

Horizontal or Vertical Slider

A Horizontal or Vertical Slider provides an analog representation of the value of a symbol (variable) or direct address in a PLC. It is a scroll bar with a "thumb" position that represents the value as a percentage of its range in engineering units. With the mouse, you can change the value of the scroll bar, sending a new value to the PLC.

The following table describes the properties for the Horizontal or Vertical Slider:

Property	Description	Limits
Name	The name of the graphic object	
Address	The direct address or the name of a symbol (variable) to monitor	See Note 1, *Notes, page 320*
Data Type	The data type of the direct address or symbol (variable)	See Note 2, *Notes, page 320*
Background	The background color of the graphic object	
Label	The label to be displayed as part of the graphic object	
Label Color	The color of the label	
Label Font	The font used for the label	
Major Scale Divisions	The number of major (labeled) scale divisions	0 to 100
Minor Scale Divisions	The number of minor (unlabeled) scale divisions	0 to 100
Scale Color	The color of the scale and its labels	
Scale Font	The font used for scale labels	
Scale Precision	The number of fractional digits to be shown for scale labels (set to -1 to use a general exponential format)	-1 to 6
Maximum EU Value	The maximum value, in engineering units, of the direct address or symbol (variable)	
Minimum EU Value	The minimum value, in engineering units, of the direct address or symbol (variable)	
Maximum PLC Value	The maximum raw (unscaled) value of the direct address or symbol (variable) in the PLC	See Note 3, *Notes, page 320*
Minimum PLC Value	The minimum raw (unscaled) value of the direct address or symbol (variable) in the PLC	See Note 3, *Notes, page 320*
Block Increment	The amount that the scaled value should change when the scroll bar’s scroll area is clicked	
Unit Increment	The amount that the scaled value should change when the scroll bar’s arrow buttons are clicked	
Border Width	The width (in pixels) of the graphic object’s border	0 to 32
Border Color	The color of the graphic object’s border	

Horizontal or Vertical Selector

A Horizontal or Vertical Selector allows you select from a set of choices. When a selection is made, the value corresponding to the choice is sent to the PLC. The choices are shown as labels of a "scale," with the current selection indicated by the position of the "thumb" of a scroll bar.

The following table describes the properties of the Horizontal or Vertical Selector:

Property	Description	Limits
Name	The name of the graphic object	
Address	The direct address (or the name of a symbol (variable)) to monitor	See Note 1, *Notes, page 320*
Data Type	The data type of the direct address or symbol (variable)	See Note 2, *Notes, page 320*
Background	The background color for the graphic object	
Choices	The choices for the selector. Each choice is given as a ‘label=value’ entry (when you select ‘label,’ ‘value’ is sent to PLC).	Minimum of two choices required
Label	The label to be displayed as part of the graphic object	
Label Color	The color of the label	
Label Font	The font used for the label	
Scale Visible	Indicates whether a "scale," labeled with the choices, is to be shown	
Scale Color	The color of the scale and its labels	
Scale Font	The font used for scale labels	
Border Width	The width (in pixels) of the graphic object’s border	0 to 32
Border Color	The color of the graphic object’s border	

Digital Indicator

A Digital Indicator provides a numeric representation of the value of a symbol (variable) or direct address in a PLC. The value can be shown in various formats, and can be made to change color when a preset high or low limit is exceeded.

The following table describes the properties of the Digital Indicator:

Property	Description	Limits
Name	The name of the graphic object	
Address	The direct address or the name of a symbol (variable) to monitor	See Note 1, *Notes, page 320*
Data Type	The data type of the direct address or symbol (variable)	See Note 2, *Notes, page 320*
Background	The background color of the graphic object	

Property	Description	Limits
Label	The label to be displayed as part of the graphic object	
Label Color	The color of the label	
Label Font	The font used for the label	
Value Format	The format (decimal, hex, etc.) to use in displaying the scaled value	
Value Precision	The number of fractional digits to be shown for the scaled value (set to -1 to use a general exponential format)	-1 to 6
Value Background	The background color of the value display area	
Value Color	The text color for the digital display of the value	
Value Font	The font used for the digital display of the value	
Units	The label for the value's engineering units (appended to the numerical display of the value)	
Maximum EU Value	The maximum value, in engineering units, of the direct address or symbol (variable)	
Minimum EU Value	The minimum value, in engineering units, of the direct address or symbol (variable)	
Maximum PLC Value	The maximum raw (unscaled) value of the direct address or symbol (variable) in the PLC	See Note 3, *Notes, page 320*
Minimum PLC Value	The minimum raw (unscaled) value of the direct address or symbol (variable) in the PLC	See Note 3, *Notes, page 320*
High High Limit Value	The value of the 'High High' limit in engineering units	
High High Limit Color	The color of the indicator bar when the scaled value is greater than the 'High High' limit	
High Limit Value	The value of the 'High' limit in engineering units	
High Limit Color	The color of the indicator bar when the scaled value is greater than the 'High' limit	
Low Limit Value	The value of the 'Low' limit in engineering units	
Low Limit Color	The color for the indicator bar when scaled value is less than the 'Low' limit	
Low Low Limit Value	The value of the 'Low Low' limit in engineering units	
Low Low Limit Color	The color of the indicator bar when the scaled value is less than the 'Low Low' limit	
Limit Deadband	The deadband (as percentage of EU range) to apply to High/Low limit checking	0 to 10
Border Width	The width (in pixels) of the graphic object's border	0 to 32
Border Color	The color of the graphic object's border	
PLC Value	A simulated, raw (unscaled) input value for testing the graphic object	See Note 3

Message Display

A Message Display shows a text message based on the value of a symbol (variable) or direct address in a PLC. For each specified message, a specified value triggers its display.

The following table describes the properties of the Message Display:

Property	Description	Limits
Name	The name of the graphic object	
Address	The direct address or the name of a symbol (variable) to monitor	See Note 1, *Notes, page 320*
Data Type	The data type of the direct address or symbol (variable)	See Note 2, *Notes, page 320*
Background	The background color of the graphic object	
Messages	The set of messages to display. Each message is given as a 'value=text' entry (when the PLC value equals 'value,' 'text' is displayed as the message).	Minimum of one message required
Message Background	The background color of the message display area	
Message Color	The color of the message text	
Message Font	The font used for the message text	
Label	The label to be displayed as part of the graphic object	
Label Color	The color of the label	
Label Font	The font used for the label	
Border Width	The width (in pixels) of the graphic object's border	0 to 32
Border Color	The color of the graphic object's border	
PLC Value	A simulated input value for testing the graphic object	See Note 3, *Notes, page 320*

Push Button

A Push Button allows you to send preset value(s) to a PLC when clicked with the mouse.

The following table describes the properties of the Push Button:

Property	Description	Limits
Name	The name of the graphic object	
Address	The direct address or the name of a symbol (variable) to monitor	See Note 1, *Notes, page 320*
Data Type	The data type of the direct address or symbol (variable)	See Note 2, *Notes, page 320*
Background	The background color of the graphic object	

Property	Description	Limits
Values	The value(s) to send to the PLC	See Note 4, *Notes, page 320*
Reset Values	The value(s) to send to the PLC after the reset delay time has expired. If no reset values are provided, no reset action will occur.	
Reset Delay	The delay time (in milliseconds) that the Push Button should wait after sending the value(s) to the PLC before sending the reset value(s).	0-2000
Label	The label to be displayed as part of the graphic object	
Label Color	The color of the label	
Label Font	The font used for the label	
Button Label	The text label for the button	
Button Background	The color of the button	0 to 100
Button Label Color	The color of the button label	
Button Label Font	The font used for the button label	
Border Width	The width (in pixels) of the graphic object's border	0 to 32
Border Color	The color of the graphic object's border	

Direct Output Station

The Direct Output Station allows you to enter a numeric value in a text field directly with the keyboard. When the entered value is within preset high and low limits, a **Set** button is enabled. In this case, the entered value will be sent to the PLC when you press either the **Set** button or the ENTER key (if the input field has keyboard input focus).

The following table describes the properties of the Direct Output Station:

Property	Description	Limits
Name	The name of the graphic object	
Address	The direct address or the name of a symbol (variable) to monitor	See Note 1, *Notes, page 320*
Data Type	The data type of the direct address or symbol (variable)	See Note 2, *Notes, page 320*
Background	The background color of the graphic object	
Label	The label to be displayed as part of the graphic object	
Label Color	The color of the label	
Label Font	The font used for the label	

Property	Description	Limits
Maximum EU Value	The maximum value, in engineering units, of the direct address or symbol (variable)	
Minimum EU Value	The minimum value, in engineering units, of the direct address or symbol (variable)	
Maximum PLC Value	The maximum raw (unscaled) value of the direct address or symbol (variable) in the PLC	See Note 3, *Notes, page 320*
Minimum PLC Value	The minimum raw (unscaled) value of the direct address or symbol (variable) in the PLC	See Note 3, *Notes, page 320*
Maximum Input	The maximum value, in engineering units, that is valid for the entered input value	
Minimum Input	The minimum value, in engineering units, that is valid for the entered input value	
Border Width	The width (in pixels) of the graphic object's border	0 to 32
Border Color	The color of the graphic object's border	

Indicator Light

The Indicator Light provides a dual-state indication of the value of a symbol (variable) or direct address in a PLC. Unless the **Input Inverted** property is set to **TRUE**, an input value of zero is deemed **OFF** and a non-zero value is deemed **ON**. If the **Flash Interval** property is set to greater than zero, the light will flash while the input value is on.

The following table describes the properties of the Indicator Light:

Property	Description	Limits
Name	The name of the graphic object	
Address	The direct address (or the name of a symbol (variable)) to monitor	See Note 1, *Notes, page 320*
Data Type	The data type of the direct address or symbol (variable)	See Note 2, *Notes, page 320*
Background	The background color of the graphic object	
Label	The label to be displayed as part of the graphic object	
Label Color	The color of the label	
Label Font	The font used for the label	
Off Word	The text to show when the input value is off	
Off Word Background	The background color of the light when the **Off Word** is shown	
Off Word Color	The color of the **Off Word** text	
Off Word Font	The font used for the **Off Word** text	
On Word	The text to show when the input value is on	

Property	Description	Limits
On Word Background	The background color of the light when the **On Word** is shown	
On Word Color	The color of the **On Word** font	
On Word Font	The font used for the **On Word** text	
Flash Interval	The flashing time period (in milliseconds) of the light when the input value is on. Set to zero for no flashing.	200 to 2000
Shape	The shape (circle, rectangle, etc.) of the light	
Input Inverted	If **TRUE**, inverts the input value. (Light will show the **Off Word** when input value is on.)	
Border Width	The width (in pixels) of the graphic object's border	0 to 32
Border Color	The color of the graphic object's border	
PLC Value	A simulated input value for testing the graphic object	See Note 3, *Notes, page 320*

Motor Control Station

The Motor Control Station is designed to mimic the typical start/stop push button station that is often used to control motors. This graphic object is essentially a composite of two push buttons and an indicator light. A Customizer is provided to make it easier to set the object's many properties. Most properties (except Name) are set with its Customizer, not with the **Graphic Editor's** Property Sheet.

The following table describes the properties of the Motor Control Station:

Property	Description	Limits
Name	The name of the graphic object	
Background	The background color of the graphic object	
Label	The label to be displayed as part of the graphic object	
Label Color	The color of the label	
Label Font	The font used for the label	
Border Width	The width (in pixels) of the graphic object's border	0 to 32
Border Color	The color of the graphic object's border	
Indicator Light	Same properties as the Indicator Light graphic object, excluding the shared properties listed above	
Top Push Button	Same properties as the Push Button graphic object, excluding the shared properties listed above	
Bottom Push Button	Same properties as the Push Button graphic object, excluding the shared properties listed above	

Analog Meter

An Analog Meter provides an analog representation of the value of a symbol (variable) or direct address in a PLC. It is represented as a pointer on a circular dial; its position indicates the value as a percentage of its range in engineering units. You can set the size of the meter's circular dial (degrees sweep of a circle), the colors of the dial, and the style of the pointer.

The following table describes the properties of the Analog Meter:

Property	Description	Limits
Name	The name of the graphic object	
Address	The direct address or the name of a symbol (variable) to monitor	See Note 1, *Notes, page 320*
Data Type	The data type of the direct address or symbol (variable)	See Note 2, *Notes, page 320*
Background	The background color of the graphic object	
Label	The label to be displayed as part of the graphic object	
Label Color	The color of the label	
Label Font	The font used for the label	
Major Scale Divisions	The number of major (labeled) scale divisions	0 to 100
Minor Scale Divisions	The number of minor (unlabeled) scale divisions	0 to 100
Scale Color	The color of the scale and its labels	
Scale Font	The font used for scale labels	
Scale Precision	The number of fractional digits to be shown for scale labels (set to -1 to use a general exponential format)	-1 to 6
Maximum EU Value	The maximum value, in engineering units, of the direct address or symbol (variable)	
Minimum EU Value	The minimum value, in engineering units, of the direct address or symbol (variable)	
Maximum PLC Value	The maximum raw (unscaled) value of the direct address or symbol (variable) in the PLC	See Note 3, *Notes, page 320*
Minimum PLC Value	The minimum raw (unscaled) value of the direct address or symbol (variable) in the PLC	See Note 3, *Notes, page 320*
Dial Degrees Sweep	The amount of a circular arc to use for drawing the dial	60 to 300
Pointer Type	The type (needle, arrow head, etc.) of pointer to use	
Pointer Color	The color for the pointer	
Dial Color	The color of the dial (that part that is within the High/Low limits)	
High High Limit Value	The value of the 'High High' limit in engineering units	
High High Limit Color	The color of the indicator bar when the scaled value is greater than the 'High High' limit	
High Limit Value	The value of the 'High' limit in engineering units	

Property	Description	Limits
High Limit Color	The color of the indicator bar when the scaled value is greater than the 'High' limit	
Low Limit Value	The value of the 'Low' limit in engineering units	
Low Limit Color	The color of the indicator bar when the scaled value is less than the 'Low' limit	
Low Low Limit Value	The value of the 'Low Low' limit in engineering units	
Low Low Limit Color	The color of the indicator bar when the scaled value is less than the 'Low Low' limit	
Border Width	The width (in pixels) of the graphic object's border	0 to 32
Border Color	The color of the graphic object's border	
PLC Value	A simulated, raw (unscaled) input value for testing the graphic object	See Note 3, *Notes, page 320*

Rotary Slider

A Rotary Slider provides an analog representation of the value of a symbol (variable) or direct address in a PLC. It is represented as a knob on a circular dial; its position indicates the value as a percentage of its range in engineering units. The dial size and knob color can be set. With the mouse, you can change the position of the knob, sending a new value to the PLC.

The following table describes the properties of the Rotary Slider:

Property	Description	Limits
Name	The name of the graphic object	
Address	The direct address or the name of a symbol (variable) to monitor	See Note 1, *Notes, page 320*
Data Type	The data type of the direct address or symbol (variable)	See Note 2, *Notes, page 320*
Background	The background color of the graphic object	
Label	The label to be displayed as part of the graphic object	
Label Color	The color of the label	
Label Font	The font used for the label	
Major Scale Divisions	The number of major (labeled) scale divisions	0 to 100
Minor Scale Divisions	The number of minor (unlabeled) scale divisions	0 to 100
Scale Color	The color of the scale and its labels	
Scale Font	The font used for scale labels	
Scale Precision	The number of fractional digits to be shown for scale labels (set to -1 to use a general exponential format)	-1 to 6
Dial Degrees Sweep	The amount of a circular arc to use for drawing the dial	60 to 300

Property	Description	Limits
Dial Color	The color of the dial	
Knob Color	The color used for the knob	
Maximum EU Value	The maximum value, in engineering units, of the direct address or symbol (variable)	
Minimum EU Value	The minimum value, in engineering units, of the direct address or symbol (variable)	
Maximum PLC Value	The maximum raw (unscaled) value of the direct address or symbol (variable) in the PLC	See Note 3, *Notes, page 320*
Minimum PLC Value	The minimum raw (unscaled) value of the direct address or symbol (variable) in the PLC	See Note 3, *Notes, page 320*
Border Width	The width (in pixels) of the graphic object's border	0 to 32
Border Color	The color of the graphic object's border	

Rotary Selector

A Rotary Selector allows you to select from a set of choices. When a selection is made, the value corresponding to the choice is sent to the PLC. The choices are shown as labels of a "scale," with the current selection indicated by the position of the knob. The size of the circular dial (degrees sweep of a circle) and knob color can be set.

The following table describes the properties of the Rotary Selector:

Property	Description	Limits
Name	The name of the graphic object	
Address	The direct address or the name of a symbol (variable) to monitor	See Note 1, *Notes, page 320*
Data Type	The data type of the direct address or symbol (variable)	See Note 2, *Notes, page 320*
Background	The background color of the graphic object	
Choices	The choices for the selector. Each choice is given as a 'label=value' entry. (When you select 'label,' 'value' is sent to PLC.)	Minimum of two choices required
Label	The label to be displayed as part of the graphic object	
Label Color	The color of the label	
Label Font	The font used for the label	
Scale Visible	Indicates whether a "scale," labeled with the choices, is to be shown	
Scale Color	The color of the scale and its labels	
Scale Font	The font used for scale labels	

Property	Description	Limits
Dial Degrees Sweep	The amount of a circular arc to use for drawing the dial	60 to 300
Knob Color	The color of the knob	
Border Width	The width (in pixels) of the graphic object's border	0 to 32
Border Color	The color of the graphic object's border	

Trend Recorder

A Trend Recorder provides a continuous, time-based charting of the value of up to six symbol(s) (variables) or direct addresses in a PLC. A Trend Recorder emulates a strip-chart recorder, with the pens on the right, and the "paper" moving from right to left. A vertical scale can be shown on the left side of the chart for showing the range of the values being recorded, and a horizontal scale can be shown below the chart for showing the time span of the chart. You can set the rate at which the chart is updated, and the appearance of the chart.

A Customizer is provided to make it easier to set this object's many properties. Most properties (except Name) are set with its Customizer, not with the **Graphic Editor's** Property Sheet.

The following table describes properties of the Trend Recorder. Properties available for each pen are described in the next table:

Property	Description	Limits
Name	The name of the graphic object	
Background	The background color of the graphic object	
Label	The label to be displayed as part of the graphic object	
Label Color	The color of the label	
Label Font	The font used for the label	
Major Scale Divisions	The number of major (labeled) scale divisions	0 to 100
Minor Scale Divisions	The number of minor (unlabeled) scale divisions	0 to 100
Scale Color	The color of the scale and its labels	
Scale Font	The font used for scale labels	
Scale Precision	The number of fractional digits to be shown for scale labels (set to -1 to use a general exponential format)	-1 to 6
Maximum EU Value	The maximum value, in engineering units, of the direct address or symbol (variable)	
Minimum EU Value	The minimum value, in engineering units, of the direct address or symbol (variable)	
Update Period	The update interval (in seconds) for the chart	0.5 to 120
Time Scale Divisions	The number of horizontal scale divisions	0 to 6
Chart Background	The color of the chart area	

Property	Description	Limits
Grid Color	The color of the grid drawn in the chart area	
Vertical Grid Divisions	The number of vertical divisions for the grid	0 to 100
Border Width	The width (in pixels) of the graphic object's border	0 to 32
Border Color	The color of the graphic object's border	

These Trend Recorder properties are available for each pen:

Property	Description	Limits
Address	The direct address (or the name of a symbol (variable)) to monitor	See Note 1, *Notes, page 320*
Data Type	The data type of the direct address or symbol (variable)	See Note 2, *Notes, page 320*
Maximum PLC Value	The maximum raw (unscaled) value of the direct address or symbol (variable) in the PLC	See Note 3, *Notes, page 320*
Minimum PLC Value	The minimum raw (unscaled) value of the direct address or symbol (variable) in the PLC	See Note 3, *Notes, page 320*
Pen Color	The color of the "pen" used to record the scaled value	
Pen Label	The label used to identify the pen	

Display Link

A Display Link is a special graphic object that allows you to switch to another graphic display by clicking on it with a mouse. To indicate that the object is a link to another display, the text label for the link is underlined and the mouse cursor changes to a hand icon when the mouse is moved over it. This object is especially useful when the **Graphic Editor** is used in **view mode**, where no drop-down list of graphic displays is available for selecting a display.

A Display Link can also be used as a hyperlink to an HTML file. If a URL is entered as the **Link Display Name**, the URL can be opened in a new browser window if you press the SHIFT key while clicking the link; otherwise, the existing browser window is replaced with the URL when the link is clicked.

If the **Link Display Name** is blank, then the **Label** will be shown as not underlined, and the displayed object becomes a simple text label.

The following table describes the properties of the Display Link:

Property	Description	Limits
Label	The label of the link	
Link Display Name	The name of the graphic display to be loaded when the link is clicked, or a URL of a Web page	
Label Color	The color of the label	
Label Font	The font used for the label	

Notes

These are the notes for this topic:

<table>
<tr><td>1.</td><td>If the Address property of a graphic object is a direct address, the Data Type property is set to UNDEFINED, a default Data Type (BOOL, INT, DINT, or REAL, based on the implied size of the data value) is used. If the Address property is a symbol (variable) name, the Data Type property does not have to be specified and can be set to UNDEFINED. If, however, the Data Type property is specified for a symbol (variable), it must exactly match the symbol’s (variable’s) actual data type.
If the Address property is a direct address for a discrete PLC reference (Quantum 0x/1x reference), set the Data Type property to BOOL. The Data Type property may be set to BOOL only for a discreet PLC reference.</td></tr>
<tr><td>2.</td><td>The meanings of the possible values of the Data Type property are:
<table>
<tr><th>Data Type</th><th>Meaning</th></tr>
<tr><td><code>UNDEFINED</code></td><td><code>no data type specified</code></td></tr>
<tr><td><code>BOOL</code></td><td><code>1-bit discreet (Boolean)</code></td></tr>
<tr><td><code>SHORT</code></td><td><code>8-bit signed integer</code></td></tr>
<tr><td><code>USHORT</code></td><td><code>8-bit unsigned integer</code></td></tr>
<tr><td><code>INT</code></td><td><code>16-bit signed integer</code></td></tr>
<tr><td><code>UINT</code></td><td><code>16-bit unsigned integer</code></td></tr>
<tr><td><code>DINT</code></td><td><code>32-bit signed integer</code></td></tr>
<tr><td><code>UDINT</code></td><td><code>32-bit unsigned integer</code></td></tr>
<tr><td><code>REAL</code></td><td><code>32-bit IEEE floating point</code></td></tr>
<tr><td><code>TIME</code></td><td><code>32-bit unsigned integer (in milliseconds)</code></td></tr>
<tr><td><code>DATE</code></td><td><code>Date (32-bit BCD)</code></td></tr>
<tr><td><code>TOD</code></td><td><code>Time of Day (32-bit BCD)</code></td></tr>
<tr><td><code>DT</code></td><td><code>Date and Time (64-bit BCD)</code></td></tr>
</table></td></tr>
<tr><td>3.</td><td>The limits for the Maximum PLC Value and Minimum PLC Value properties are the natural limits of the Data Type property that is set. A Data Type setting of UNDEFINED is treated as REAL with respect to its limit values.</td></tr>
<tr><td>4.</td><td>For a Push Button, provide at least one value. If the Address property is a symbol (variable) name, then only one value will ever be sent to the PLC, and any additional values are ignored. If the Address property is a direct address, then all of the values provided are sent to the PLC as an array of values starting at the specified direct address.</td></tr>
</table>

Extended Graphic Objects

Overview

The extended graphic objects provided in the Graphic Editor are used to build graphic displays that mimic advanced graphic panels. All of the data monitoring and control objects have built-in communication capabilities and are designed as standalone graphic objects.

Be aware, however, that if communication to the device linked to the extended graphic object is lost, the object becomes inoperative without the end device's knowledge.

WARNING

UNINTENDED EQUIPMENT OPERATION

- Do not use graphic objects in situations where loss of communication to the FactoryCast module can affect human or material integrity.
- Do not use graphic objects in safety critical machine functions.

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

For example, say you have programmed a pushbutton object to jog a motor when the button is depressed and to stop jogging when the button is released. If communications are lost while the button is depressed, the motor will continue to jog even when the button is released. Graphic objects should not be used to control situations such as this unless other interlock methods are installed in the system.

Additionally, to support customers that want to put several simple applets on a single HTML page, each object in the Graphic Editor set is provided in an applet version. When used in conjunction with the `LiveBeanApplet`, Graphic Editor graphic objects can be used in the same way as the `LiveLabelApplet`.

ASCII Text Writer

The ASCII text writer is based on the message display widget. It allows you to input new text.

The properties of the ASCII text writer are:

Property	Description	Limits
Name	The name of the graphic object	
Address	The direct address or the name of a symbol (variable) to monitor	See Note 1, *Notes, page 330*
Max. Text Length	The maximum length of the text	
Text Color	The color of the text	
Text Font	The font of the text	
Swap Bytes	False if target order of bytes is same as PC one.	
Value	The text itself	

Bar Graph

A bar graph provides an analog representation of the value of a symbol (variable) or direct address in a PLC. It draws a vertical bar whose length is proportional to the value as a percentage of its range in engineering units.

The properties of the bar graph are:

Property	Description	Limits
Name	The name of the graphic object	
Address	The direct address or the name of a symbol (variable) to monitor	See Note 1, *Notes, page 330*
Data Type	The data type of the direct address or symbol (variable)	See Note 2, *Notes, page 330*
Background	The background color of the graphic object	
Label	The label to be displayed as part of the graphic object	
Label Color	The color of the label	
Label Font	The font used on the label	
Maximum EU Value	The maximum value, in engineering units, of the direct address or symbol (variable)	
Minimum EU Value	The minimum value, in engineering units, of the direct address or symbol (variable)	
Maximum PLC Value	The maximum raw (unscaled) value of the direct address or symbol (variable) in the PLC	See Note 3, *Notes, page 330*
Minimum PLC Value	The minimum raw (unscaled) value of the direct address or symbol (variable) in the PLC	See Note 3, *Notes, page 330*
Bar Background	The background color of the bar indicator area	

Property	Description	Limits
Bar Color	The color of the indicator bar (when scaled value within High/Low limits)	
High High Limit Value	The value of the High High limit in engineering units	
High High Limit Color	The color of the indicator bar when the scaled value is greater than the High High limit	
High Limit Value	The value of the High limit in engineering units	
High Limit Color	The color of the indicator bar when scaled value is greater than the High limit	
Low Limit Value	The value of the Low limit in engineering units	
Low Limit Color	The color of the indicator bar when the scaled value is less than the Low limit	
Low Low Limit Value	The value of the Low Low limit in engineering units	
Low Low Limit Color	The color of the indicator bar when the scaled value is less than the Low Low limit	
Limit Deadband	The deadband (as percentage of EU range) to apply to High/Low limit checking	0 to 10
Border Width	The width (in pixels) of the graphic object's border	0 to 32
Border Color	The color of the graphic object's border	
PLC Value	A simulated, raw (unscaled) input value for testing the graphic object	See Note 3, *Notes, page 330*

Bitmap

The bitmap widget displays a static bitmap on the screen.

The properties of the bitmap widget are:

Property	Description	Limits
Name	The name of the graphic object	
Background	The background color of the graphic object	See Note 1, *Notes, page 330*
Label	The label to be displayed as part of the graphic object	
Label Color	The color of the label	
Label Font	The font used for the label	
Border Width	The width (in pixels) of the graphic object's border	
Border Color	The color of the graphic object's border	
Bitmap Choices	Represents the filenames of the custom bitmaps to display. The default root path of the file location directory is **/FLASH1/wwwroot;**; **":images/** thus refers to **/FLASH1/wwwroot/images/** .	

Generic Bitmap

The generic bitmap widget lets you display one static bitmap for each distinct value of a PLC variable. It can be used to display dynamic animations, for instance the changing level in a tank.

The properties of the Generic Bitmap widget are:

Property	Description	Limits
Name	The name of the graphic object	
Address	The direct address or the name of a symbol (variable) to monitor	See Note 1, *Notes, page 330*
Data Type	The data type of the direct address or symbol (variable)	See Note 2, *Notes, page 330*
Background	The background color of the graphic object	See Note 1, *Notes, page 330*
Label	The label to be displayed as part of the graphic object	
Label Color	The color of the label	
Label Font	The font used for the label	
Bitmap Choices	Represents the filenames of the custom bitmaps to display. Clicking on this property opens a text editor that makes it possible to type the PLC value conditions and related bitmaps to display, such as "0:key.gif:images/" where 0 is the PLC value, "key.gif" the bitmap file related to the value, "images" the directory in which the file is located. The default root path of the file location directory is **/FLASH1/wwwroot**; **images/** thus refers to **/FLASH1/wwwroot/images/**.	
Border Width	The width (in pixels) of the graphic object's border	
Border Color	The color of the graphic object's border	
PLC Value	A simulated input value for testing the graphic object behavior.	

Graphic Link

A graphic link is a special graphic object that lets you switch to another graphic display by clicking on it. Graphic links can also be recognized by their underlined labels, and the mouse cursor changes to a hand icon when the mouse moves over them. This object is especially useful when the Graphic Editor is used in view mode, where no drop-down list of graphic displays is available.

A graphic link can also be used as a hyperlink to an HTML file. If a URL is entered as the **Link Display Name**, the URL can be opened in a new browser window by simultaneously pressing the SHIFT key and clicking the link. Otherwise, the URL opens in the existing browser window when the link is clicked.

If the **Link Display Name** is blank, then the label is not underlined, and the displayed object becomes a simple text label.

The properties of the display link are:

Property	Description	Limits
Label	The link label	
Link Display Name	The name of the graphic display to be loaded when the link is clicked, or the URL of a Web page	
Label Color	The color of the label	
Label Font	The font used on the label	
Bitmap Choices	The filename of the bitmap on which to click	

Indicator Light

The indicator light displays the value of a symbol (variable) or direct address in a PLC with two possible states. An input value of 0 is considered off and a non-zero value is considered on. If the **Flash Interval** property is set to a value greater than 0, the light flashes while the input value is on. There is a bitmap for the on-state and a different one for the off-state.

The properties of the indicator light are:

Property	Description	Limits
Name	The name of the graphic object	
Address	The direct address or the name of a symbol (variable) to monitor	See Note 1, *Notes, page 330*
Data Type	The data type of the direct address or symbol (variable)	See Note 2, *Notes, page 330*
Background	The background color of the graphic object	
Label	The label to be displayed as part of the graphic object	
Label Color	The color of the label	
Label Font	The font used for the label	
OFF Word	The text to display when the input value is off	
OFF Bitmap Choice	The light bitmap when the OFF word is displayed	
OFF Word Color	The color of the OFF word text	
OFF Word Font	The font used for the OFF word text	
ON Word	The text to display when the input value is on	
ON Bitmap Choice	The light bitmap when the ON word is displayed	
ON Word Color	The color of the ON word font	
ON Word Font	The font used for the ON word text	
Flash Interval	The flashing time period (in ms) of the light when the input value is on. Set to 0 for no flashing.	200 to 2000
Input Inverted	If true, inverts the input value. (Light will show the off-word when input value is on.)	

Property	Description	Limits
Border Width	The width (in pixels) of the graphic object's border	0 to 32
Border Color	The color of the graphic object's border	
PLC Value	A simulated input value for testing the graphic object	See Note 3, *Notes, page 330*

Motor

The motor widget displays the value of a symbol (variable) or direct address in a PLC with three possible states. An input value of 0 is considered off, a value of 1 is considered on and other values are considered default. The three states are represented by different bitmaps.

The properties of the motor widget are:

Property	Description	Limits
Name	The name of the graphic object	
Address	The direct address or the name of a symbol (variable) to monitor	See Note 1, *Notes, page 330*
Data Type	The data type of the direct address or symbol (variable)	See Note 2, *Notes, page 330*
Background	The background color of the graphic object	
Label	The label to be displayed as part of the graphic object	
Label Color	The color of the label	
Label Font	The font used for the label	
OFF Word	The text to display when the input value is off	
OFF Bitmap Choice	The motor bitmap when the OFF word is displayed	
OFFWord Color	The color of the OFF word text	
OFF Word Font	The font used for the OFF word text	
ON Word	The text to display when the input value is ON	
ON Bitmap Choice	The motor bitmap when the ON word is displayed	
ON Word Color	The color of the ON word font	
ON Word Font	The font used for the ON word text	
DEFAULTWord	The text to display when the input value is ON	
DEFAULT Bitmap Choice	The motor bitmap when the DEFAULT word is displayed	
DEFAULT Word Color	The color of the DEFAULT word font	
DEFAULT Word Font	The font used for the DEFAULT word text	
Border Width	The width (in pixels) of the graphic object's border	0 to 32

Property	Description	Limits
Border Color	The color of the graphic object's border	
PLC Value	A simulated input value for testing the graphic object	See Note 3, *Notes, page 330*

Pipe

The pipe displays the value of a symbol (variable) or direct address in a PLC with two possible states. An input value of 0 is considered off and a non-zero value is considered on. There is a bitmap for the on-state and a different one for the off-state.

The properties of the pipe are:

Property	Description	Limits
Name	The name of the graphic object	
Address	The direct address or the name of a symbol (variable) to monitor	See Note 1, *Notes, page 330*
Data Type	The data type of the direct address or symbol (variable)	See Note 2, *Notes, page 330*
Background	The background color of the graphic object	
Label	The label to be displayed as part of the graphic object	
Label Color	The color of the label	
Label Font	The font used for the label	
OFF Word	The text to display when the input value is off	
OFF Bitmap Choice	The pipe bitmap when the OFF word is displayed	
OFF Word Color	The color for the OFF word text	
OFF Word Font	The font used for the OFF word text	
ON Word	The text to display when the input value is on	
ON Bitmap Choice	The pipe bitmap when the ON word is displayed	
ON Word Color	The color of the ON word font	
ON Word Font	The font used for the ON word text	
Border Width	The width (in pixels) of the graphic object's border	0 to 32
Border Color	The color of the graphic object's border	
PLC Value	A simulated input value for testing the graphic object	See Note 3, *Notes, page 330*

Push Button

A push button sends preset value(s) to a PLC when the user clicks it with the mouse.

These are the properties of the Push Button.

Property	Description	Limits
Name	The name of the graphic object	
Address	The direct address or the name of a symbol (variable) to monitor	See Note 1, *Notes, page 330*
Data Type	The data type of the direct address or symbol (variable)	See Note 2, *Notes, page 330*
Background	The background color of the graphic object	
Values	The value(s) to send to the PLC	See Note 4, *Notes, page 330*
Reset Values	The value(s) to send to the PLC after the reset delay time has expired. If no reset values are provided, no reset action will occur.	
Reset Delay	The delay time (in milliseconds) that the Push Button should use after sending the value(s) to the PLC before sending the reset value(s).	0-2000
Label	The label to be displayed as part of the graphic object	
Label Color	The color of the label	
Label Font	The font used for the label	
Button Label	The text label for the button	
Button Label Color	The color of the button label	
Button Label Font	The font used for the button label	
OFF Bitmap Choice	The button bitmap when the OFF state is displayed	
ON Bitmap Choice	The button bitmap when the ON state is displayed	
Border Width	The width (in pixels) of the graphic object’s border	0 to 32
Border Color	The color of the graphic object’s border	

Valve

The valve displays the value of a symbol (variable) or direct address in a PLC with two possible states. An input value of 0 is considered off and a non-zero value is considered on. There is a bitmap for the on-state and a different one for the off-state.

The properties of the valve are:

Property	Description	Limits
Name	The name of the graphic object	
Address	The direct address or the name of a symbol (variable) to monitor	See Note 1, *Notes, page 330*
Data Type	The data type of the direct address or symbol (variable)	See Note 2, *Notes, page 330*
Background	The background color of the graphic object	
Label	The label to be displayed as part of the graphic object	
Label Color	The color of the label	
Label Font	The font used for the label	
OFF Word	The text to display when the input value is off	
OFF Bitmap Choice	The valve bitmap when the OFF word is displayed	
OFF Word Color	The color of the OFF word text	
OFF Word Font	The font used for the OFF word text	
ON Word	The text to display when the input value is ON	
ON Bitmap Choice	The valve bitmap when the ON word is displayed	
ON Word Color	The color of the ON word font	
ON Word Font	The font used for the ON word text	
Flash Interval	The flashing time period (in ms) of the light when the input value is on. Set to 0 for no flashing.	200 to 2000
Border Width	The width (in pixels) of the graphic object's border	0 to 32
Border Color	The color of the graphic object's border	
PLC Value	A simulated input value for testing the graphic object	See Note 3, *Notes, page 330*

Notes

These are the notes for the chapter.

<table>
<tr><td>1.</td><td colspan="2">If the address property of a graphic object is a direct address, the data type property is set to UNDEFINED, a default data type (BOOL, INT, DINT, or REAL based on the implied size of the data value) is used. If the address property is a symbol (variable) name, the data type property does not have to be specified and can be set to UNDEFINED. If, however, the data type property is specified for a symbol, it is valid only if it exactly matches the symbol’s actual data type.
If the address property is a direct address for a Quantum 0x/1x reference, the data type property must be set to BOOL. The data type property may be a BOOL only for a discrete PLC reference.</td></tr>
<tr><td rowspan="15">2.</td><td colspan="2">The meaning of the possible values of the Data Type property are:</td></tr>
<tr><td>Data Type</td><td>Meaning</td></tr>
<tr><td><code>UNDEFINED</code></td><td><code>no data type specified</code></td></tr>
<tr><td><code>BOOL</code></td><td><code>1-bit discreet (Boolean)</code></td></tr>
<tr><td><code>SHORT</code></td><td><code>8-bit signed integer</code></td></tr>
<tr><td><code>USHORT</code></td><td><code>8-bit unsigned integer</code></td></tr>
<tr><td><code>INT</code></td><td><code>16-bit signed integer</code></td></tr>
<tr><td><code>UINT</code></td><td><code>16-bit unsigned integer</code></td></tr>
<tr><td><code>DINT</code></td><td><code>32-bit signed integer</code></td></tr>
<tr><td><code>UDINT</code></td><td><code>32-bit unsigned integer</code></td></tr>
<tr><td><code>REAL</code></td><td><code>32-bit IEEE floating point</code></td></tr>
<tr><td><code>TIME</code></td><td><code>32-bit unsigned integer (in milliseconds)</code></td></tr>
<tr><td><code>DATE</code></td><td><code>Date (32-bit BCD)</code></td></tr>
<tr><td><code>TOD</code></td><td><code>Time of Day (32-bit BCD)</code></td></tr>
<tr><td><code>DT</code></td><td><code>Date and Time (64-bit BCD)</code></td></tr>
<tr><td>3.</td><td colspan="2">The limits for the Maximum PLC Value and Minimum PLC Value properties are the natural limits of the Data Type property that is set. A Data Type setting of UNDEFINED is treated as REAL with respect to its limit values.</td></tr>
<tr><td>4</td><td colspan="2">For a Push Button, enter at least one value. If the Address property is a symbol name, only one value is sent to the PLC, and any additional values are ignored. If the Address property is a direct address, all of the values are sent to the PLC as an array, starting at the specified direct address.</td></tr>
</table>

Appendices

Introduction

These technical appendices supplement the information in this guide.

What Is in This Appendix?

The appendix contains the following chapters:

Chapter	Chapter Name	Page
A	TCP/IP Technical Characteristics	333
B	IP Address Details	347
C	Transparent Ready Service Classes	355
D	Schneider Private MIB	359

Appendix A
TCP/IP Technical Characteristics

About this Chapter

This chapter contains some of the more technical network and communications characteristics of TCP/IP, as it relates to Modicon M340 devices.

What Is in This Chapter?

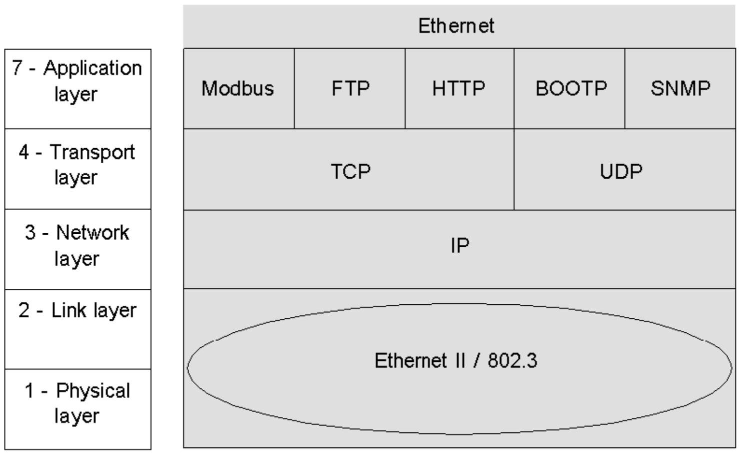
This chapter contains the following topics:

Topic	Page
Summary of TCP/IP Characteristics	334
Address Management for Ethernet Modules	336
Modbus Communication on the TCP/IP Profile	337
Managing TCP Connections for Modbus	340
Opening a TCP/IP Connection	341
Closing a TCP/IP Connection	343
Broken TCP/IP Connections	344

Summary of TCP/IP Characteristics

TCP/IP Communication Profile

The following figure shows the composition of a typical TCP/IP stack as it relates to the 7-layer OSI model:



Software Port 502

The port reserved for the BMX NOE 01x0 or the Ethernet port on the BMX P34 20x0 is TCP port 502. To access these modules' server, do so through this port.

Timeout on TCP Connection

If a TCP connection cannot be established (when the destination is absent for example), a timeout error occurs after 80 seconds.

Confirm that each communication function timeout is set to a value higher than 80 seconds if the first exchange was not successfully completed.

NOTE:
If you are using a derived function block (DFB), you can add a timer to verify the completion of a function block:

- To verify that the function block was **incomplete**, write an `On` timer to start when the function is triggered. If the time goes beyond the preset, the variable associated with the pin is set.
  Result: You receive a timeout error indicating the communication did **not** work.
- To verify that the function block was **complete**, watch the `active`, `error`, and `complete` outputs of the communication function block. (Depending on the blocks used, they may not be present.) Use the complete as an event. If the event completes within the time set, no alarm is recorded. (Essentially, it resets it self.)

Keep Alive Frame

The TCP layer sends a "keep alive" frame *(see page 344)* almost every two hours so that breaks in connection can be detected (for example, cable disconnection, detection of power outage from the client by a server, etc.).

Address Management for Ethernet Modules

Introduction

⚠ CAUTION
UNINTENDED EQUIPMENT OPERATION You must carefully manage the modules' IP addresses because each device on the network requires a unique address. If two devices have duplicate network addresses, you can not predict the operation of the equipment. **Failure to follow these instructions can result in injury or equipment damage.**

NOTE: Consult your system administrator to obtain the appropriate network address and subnetwork mask.

MAC Address

The MAC address is unique to each Ethernet module. It is defined in the factory by the module manufacturer.

NOTE: Given the risk of duplicate addresses, ensure that the address conforms to the manufacturer's addressing scheme.

IP Address

General case: Define this address when configuring the module. This address must be unique.

Exception: In the absence of configuration by Control Expert, the server, etc., the default IP address of the BMX NOE 01x0 and the Ethernet port of the BMX P34 20x0 CPUs is derived from the MAC address *(see page 66)*.

Modbus Communication on the TCP/IP Profile

Modbus Messaging and TCP Port 502

Modbus has been the standard for serial link protocols in industry since 1979. Millions of automation devices use Modbus for communications. For Ethernet, the TCP port 502 is reserved for Modbus.

Therefore, Modbus messaging can be used for exchanging automation data on both Ethernet TCP/IP and the Internet, as well as for all other applications (file exchange, Web pages, E-mail, etc.). The simple structure of Modbus allows you to download the specifications and source code for numerous devices that use the Modbus TCP/IP protocol. These items are available free of charge from the Modbus-IDA Web site (www.modbus-ida.org).

Port 502 messaging paths:

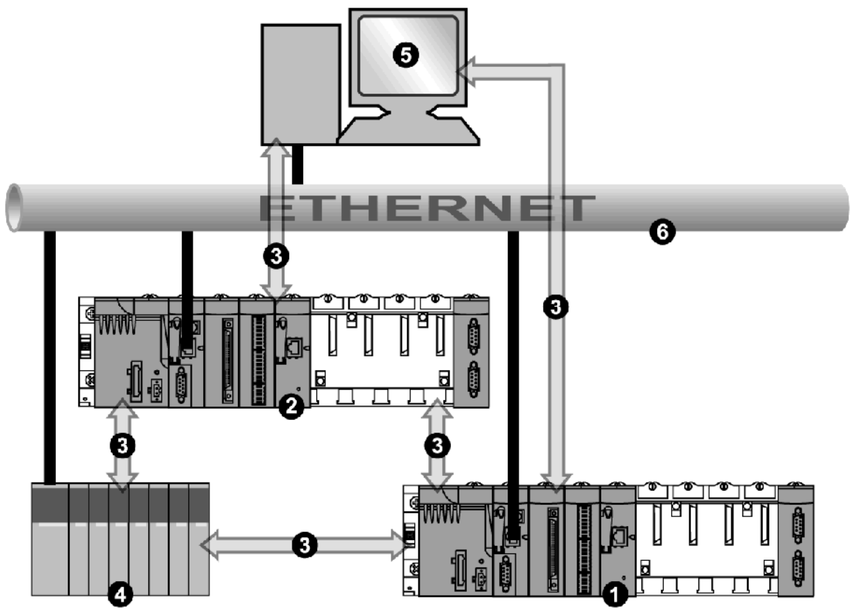
- server path:
  - Port 502 messaging can process up to 8 incoming requests from the network. Requests are received during the previous scan and sent to the Modbus server in the IN section.
  - Port 502 messaging can process up to 8 responses from the Modbus server in the IN section (including writing the data into the socket).
- client path:
  - Port 502 messaging can process up to 16 outgoing requests from the application in the OUT section (including writing the data into the socket).
  - Port 502 messaging can process up to 16 incoming responses from the network in the IN section. Responses are sent to the application.

Modbus Communications

This service enables communications through the Modbus protocol between a Modicon M340 PLC and:

- a Quantum PLC
- a Premium PLC
- a PC with supervisor software
- another device complying with the Modbus protocol

The following figure shows the Modbus communications over open TCP/IP connections:



1 Modicon M340 PLC (1)
2 Modicon M340 PLC (2)
3 Modbus protocol
4 Quantum server/client
5 client supervisor
6 TCP/IP Ethernet

The same module can communicate with a remote device in client mode (for example a Quantum PLC) and another remote device in server mode (for example a supervisor PC).

In the above figure, Modicon M340 PLC (1) is the client to the Quantum PLC. It opens the TCP/IP connection and sends Modbus messages to the Quantum. Modicon M340 PLC (2) is the server to the supervisor. The supervisor has opened a TCP/IP connection for sending Modbus messages to Modicon M340 PLC (2).

Data Exchange

The following requests are addressed to the device on which you want to perform variable read or write operations:

Modbus Requests	Function Code (Hexadecimal)	Communication Function
Read bits	16#01	`READ_VAR`
Read input bits	16#02	`READ_VAR`
Read words	16#03	`READ_VAR`
Write a bit or n bits	16#0F	`WRITE_VAR`
Write a word or n words	16#10	`WRITE_VAR`

NOTE: The timeout value for `READ_VAR` is user-configurable as follows:
- If you enter a 0 as the timeout value, the block will never timeout.
- If you enter a non-zero value, the block will timeout at the non-zero value you entered.

Correspondence of Object Types

This table describes object type correspondence between a Modicon M340 PLC and a Momentum, Quantum, or Premium PLC:

Modicon M340 Objects	Quantum or Momentum Objects
%MW: internal words	4x... memory area
%M: internal bits	0x... memory area
%IW: input words	3x... memory area
%I: input bits	1x... memory area

Managing TCP Connections for Modbus

Overview

The connection can be opened either by the local PLC or by a remote station that wants to communicate with the local PLC.

A connection is characterized by the pair:

`(local TCP port, local IP address; remote TCP port, remote IP address)`

NOTE: Connection management is transparent to the user.

Opening a TCP/IP Connection

Introduction

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

Do not exceed the maximum number of connections. Idle connections that are closed automatically when the limit is reached can affect system performance. See the discussion of closed connections *(see page 343)*.

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

A TCP/IP connection can be opened by a request from:
- a remote device
- the local PLC

With a Remote Device

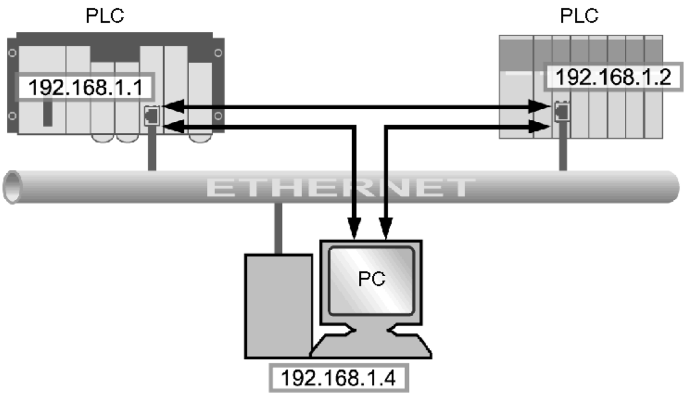
The module prepares for a connection coming from a remote device. When the connection is received, verification of the IP address of the remote machine is done only if an access control check *(see page 147)* is activated. The test checks to see if the address is on a list of remote machines authorized for connection:
- **positive result:** connection is opened
- **negative result:** connection is closed

With a Local PLC

When a message is sent out by a communication function while there is no connection to the remote device, the module automatically (internally) opens a connection to the remote 502 port.

Opening a Connection Example

The following figure shows an example of connections. (All communications are over Ethernet TCP/IP.)



In this example, three TCP connections are open for communication between the PC and the PLC stations or between two PLC stations.

Either of two PLCs can open the connection between them.

Closing a TCP/IP Connection

Summary

⚠ WARNING
UNINTENDED EQUIPMENT OPERATION Do not exceed the maximum number of connections. • Idle connections that are closed automatically when the limit is reached can affect system performance. • Control Expert configuration in TCP/IP mode is not a terminal connection; it can be closed. When the maximum number of connections is reached, the Control Expert connection may be closed. **Failure to follow these instructions can result in death, serious injury, or equipment damage.**

TCP/IP connections can be closed by the:

- **remote station**: The remote station ends communication by sending a TCP/IP connection closure.
- **local station**: When the maximum number of open connections is reached and a new connection is required, the connection that has been idle for the longest time is closed.

Elsewhere in this guide are details of the maximum number of connections for the BMX NOE 01x0 *(see page 124)* and the maximum number of connections for the BMX P34 20x0 CPUs *(see page 130)*.

Broken TCP/IP Connections

Introduction

There are two types of broken TCP/IP connections:
- a physical problem with the network cable (cut or disconnected)
- the disappearance of the remote device (break down, loss of power, etc.)

If the socket is active, the device can detect the failed connection quickly using the diagnostic bit, LED, health bit, etc. If the socket connection is not active, the failed connection is detected after 2 hours by the Keep Alive request. If the connection is reestablished during this interval, the method for restarting communications differs according to the type of break:
- cable disconnection
- lost connection to server
- lost connection to client

These situations are addressed below.

NOTE: Elsewhere in this guide are instructions for opening a connection *(see page 341)* and closing a connection *(see page 343)*.

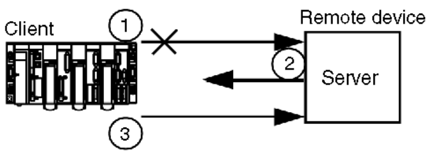
Cable Disconnection

In this case the break in connection is caused by a network cable but the two stations remain operational.

When the cable is reconnected, communication between the Modicon M340 module and the remote device will start again on the same TCP/IP connection that was previously open.

Lost Connection to Server

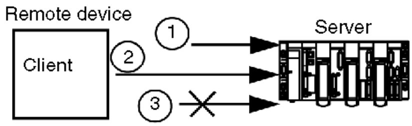
The remote device that disappeared was the server.



1 The client Modicon M340 module is still sending out data on the old connection (which remains half open).
2 The server receiving the information without associated connection sends out a Reset command and closes the old connection.
3 The client Modicon M340 module opens a new connection.

Lost Connection to Client

The remote device that disappeared was the client.



1 The client opens a new connection.
2 The server Modicon M340 module receives the request to open a new connection.
3 The server Modicon M340 module closes the old connection (if there is nothing in progress) and authorizes the new one.

Appendix B
IP Address Details

About this Chapter

This chapter discusses some important IP address details to consider when incorporating Transparent Ready capabilities into your network design.

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
IP Addresses and Classes	348
Multicasting Considerations	353

IP Addresses and Classes

Summary

An IP address allows a device to have a unique logical address to locate the device on the TCP/IP network and group it with others for network design and management purposes.

Dotted Decimal Notation

A computer sees an IP address in a binary form of 32 bits. For ease of use, the 32 bits have been divided into four 8-bit groups. Each group is converted into its decimal equivalent, which results in four decimal numbers separated by dots. As an example, an IP address in binary 10001011.00101101.00100100.00001100 can be written in a simpler format by converting each individual octet into a decimal value,139.45.36.12.

10001011	00101101	00100100	00001100
139	45	36	12

Network Address Defined

An IP address consists of two parts, the network address and the host or device address. The subnetwork mask is a filter that is applied to the IP address to determine which part of the IP address is the network address and which part is the host or device address. The network address is the part of an IP address that identifies the subnet that the address is a part of. The mask is a 32-bit value that uses one-bits for the network and subnet portions and zero-bits for the host portion. In classful addressing, the network address portion of the IP address consists of one, two or three octets, starting from the left.

IP Address	11000000	10100000	00010100	00110000	192.160.20.48
Subnetwork Mask	11111111	11111111	11111111	00000000	255.255.255.0
Network Portion of IP Address	11000000	10100000	00010100	00000000	192.160.20.0

Classful Addressing

In classful addressing, these are the possible classes of IP addresses to use, depending on the size of your enterprise:

- Class A = 0.0.0.0/8 through 127.0.0.0/8
- Class B = 128.0.0.0/16 through 191.255.0.0/16
- Class C = 192.0.0.0/24 through 223.255.255.0/24
- Class D = 224.0.0.0 through 239.255.255.255 is used for multicasting *(see page 353)*

The remaining addresses known as Class E are reserved for experimental use.

An address comprises 2 parts:

- the network information
- the host (node or end device) information

The IP address comprises four sets of decimal numbers called octets, each separated by a period, with a value from 0 to 255 that represents a converted binary-to-decimal number

Classless Addressing

Classless addressing (also known as CIDR or supernetting) was developed to improve current Internet problems regarding the efficient utilization of address space. It also is used to add to the routing scalability of networks. Allocating portions of the large but limited number of addresses to an enterprise all at one time often resulted in the waste of some reserved addresses. Including each network in a table resulted in overload. Also, medium-sized enterprises that fit the class B category have multiplied the fastest, using much of the space in that class. Classless addressing, by allowing the delineation point between network information and host information to be flexible, has expanded the number of addresses available to all sizes of enterprise and has reduced the size of routing tables.

Choosing an Address Range

Public addresses, for use on the Internet, are assigned by a governing organization called the Internet Assigned Numbers Authority (IANA). However, your company may already have been assigned a section of addresses and your IT person can allocate the quantity that you need. If you have not been given a predefined set of IP ranges, you should be aware that the following three blocks have been reserved by IANA for private Internets:

10.0.0.0 - 10.255.255.255 (10/8 prefix)

172.16.0.0 - 172.31.255.255 (172.16/12 prefix)

192.168.0.0 - 192.168.255.255 (192.168/16 prefix)

Special Addresses

There are three types of special addresses that should be mentioned:

- broadcast
- loopback
- network

A broadcast message, usually used for network management and diagnostic purposes, is addressed to all stations on the network. The destination address in a broadcast message is made up of all 1s (255.255.255.255).

A loopback address is used to test the implementation of the TCP/IP protocol on a host. The lower layers are bypassed by sending to a loopback address. This allows the higher layers (IP and above) to be tested without exposing problems at the lower layers. 127.0.0.1 is the address typically used for loopback testing.

As described in the previous section, network address refers to the network portion of an IP (Internet Protocol) address.

Sufficient Addresses

In planning for your network, you should anticipate the need for these addresses:

- for the gateway (one address)
- for broadcast
- for the number of services
- for future devices added to the network

Tools can be found on the Internet to help calculate the number of addresses your network requires.

Subnetting

Forming subnets divides a large network into more manageable segments; it can allow you to expand the number of networks, while using only the single IP address. You need not apply for more of the limited number of IP address numbers.

Network traffic is reduced by sending messages to only a limited segment of the network. Subnetting can be particularly helpful on a network that handles a lot of broadcast traffic. It can also be useful if you have a slow WAN link connecting your far-flung locations.

To subnet, the default subnetwork mask for a network is extended to cover bits of the address that would otherwise be part of the host field. Once these bits are masked, they become part of the network field and are used to identify subnets of the larger network.

Choose a subnet of a size (number of addresses) appropriate for the number of devices on it; a size that allows for growth, but is not wasteful of addresses. For example, if you have 50 devices, choose a subnet of 64 addresses, not 1024. The following table contains one column presenting the number of addresses and another with the corresponding mask.

subnetwork Mask	Number of Addresses
0.0.0.0	4,294,964,086
128.0.0.0	2,147,482,048
192.0.0.0	1,073,741,024
224.0.0.0	536,870,512
240.0.0.0	268,435,256
248.0.0.0	134,217,628
252.0.0.0	67,108,864
254.0.0.0	33,554,432
255.0.0.0	16,777,216
255.128.0.0	8,388,608
255.192.0.0	4,194,304
255.224.0.0	2,097,152
255.240.0.0	1,048,576
255.248.0.0	524,288
255.252.0.0	262,144
255.254.0.0	131,072
255.255.0.0	65,536
255.255.128.0	32,768
255.255.192.0	16,384
255.255.224.0	8,192
255.255.240.0	4,096
255.255.248.0	2,048
255.255.252.0	2048
255.255.254.0	1024
255.255.255.0	512
255.255.255.128	128
255.255.255.192	64
255.255.255.224	32
255.255.255.240	16
255.255.255.248	8
255.255.255.252	4

subnetwork Mask	Number of Addresses
255.255.255.254	2
255.255.255.255	1

For a subnet with 64 addresses, the subnetwork mask is 255.255.255.192. The IP address would therefore be 192.168.1.1, the network address would be 192.168.0 and the host range would be from 0.1 to .63.

Using Subnets in a Plant

By using subnets in your plant, you can divide the plant into sections to avoid traffic overload. Use a router to pass traffic between subnets. There should be no more than 200 to 300 devices per network. However, it is preferable to have a smaller network with 50 to 100 devices. Add networks if you must accommodate more devices than the preferred number.

Assigning Addresses

You may obtain addresses from the governing organization or use a group of those already assigned to your company. The next step is to assign a unique address to each end device by one of several methods. In static addressing, each user is assigned one fixed IP address to be used every time the user connects to the Internet. Dynamic addressing assigns the IP automatically, as needed. BootP (Bootstrap Protocol) as its name suggests, allows a workstation to configure itself without a hard drive or floppy disk. The workstation can discover its own IP address, the IP of a server and a file to be loaded into memory to boot the machine. DHCP assigns a different address to a device when it requests one. The software, rather than the administrator as in static addressing, keeps track of the IP addresses.

Multicasting Considerations

Summary

IP multicast, a method of selectively sending messages promoted by an industry consortium of prominent companies, is an up-and-coming technology that will be used increasingly for:

- *monitoring*: manufacturing and other types of real-time information, sensor equipment or security systems.
- *announcements*: network time, multicast session schedules, random numbers, keys, configuration updates, etc.
- *file distribution and caching*: Web site content, executable binaries
- *scheduled distribution* of audio and video
- *push media*: news headlines, weather updates, sports scores, etc.

On the Internet

You should make sure that your router and/or switches support multicast, your workstations are configured to join a multicast group and that you have installed any specific applications needed to receive the multicast.

IP Multicasting Transport

The UDP protocol is used for IP multicasting. The multicast address selected is important in allowing network managers to control the way hosts (end devices) join groups and how routers exchange multicast information.

IP Multicast Addresses

In IP multicasting, each group has a multicast group ID, a set of Class D IP addresses used to specify the destination of a message. The addresses range from 224.0.0.0 to 239.255.255.255. Each multicast IP address can have a number of hosts listening to it. Hosts can belong to a multicast group, and the IP addresses are associated with that group. Each configured device has a multicast IP address that is in addition to its own IP address.

Class D addresses can be classified as follows:

- *permanently assigned*: addresses in the range 224.0.0.0 to 224.0.0.225, permanently assigned by IANA for certain applications such as routing protocols; for example:
  - 224.0.0.0 for the base address
  - 224.0.0.1 for all systems on this subnet
  - 224.0.0.2 for all routers on this subnet
  - 224.0.0.4 for DVMRP routers
- *nonpermanent*: addresses in the range 224.0.1.0 to 238.255.255.255, used for assignment as needed on the Internet
- *administered nonpermanent*: addresses in the range 239.0.0.0 to 239.255.255.255, reserved for use in private Intranets

Appendix C
Transparent Ready Service Classes

Service Classes

Introduction

The Transparent Ready service classes make it possible to identify the services provided by each device, such as:

- diagnostic, display, and control services via Web technologies
- Ethernet communication services

Web Service Classes

The four Web service classes are defined by letter:

- class A: no Web services
- class B: standard Web services
- class C: configurable Web services
- class D: active Web services

Transparent Ready devices with an embedded Web server can provide four types of Web service:

- maintenance
- control
- diagnostic
- optional, such as documentation and configuration

NOTE: The availability of Web service classes depends on your choice from the standard and optional memory cards *(see page 50)*.

This table specifies the services provided by each Web service class (A, B, C, D):

Web Server Class		Web Services			
		Maintenance	Monitoring and IT Link	Diagnostics	Optional
A	none	• no web service			
B	standard	• remote device software update • remote auto-tests	• device description • data viewer	• device description • data diagnostic	• configuration of network parameters and Ethernet communication services • device documentation

Web Server Class		Web Services			
		Maintenance	Monitoring and IT Link	Diagnostics	Optional
C	configurable	• user Web site update	• PLC variables editor • remote commands • user Web pages • SOAP/XML (server)	• communication service diagnostics • state of internal device resources	• user documentation
D	active	• user Web site update	• autonomous execution of specific services (e.g., alarm notification by E-mail, exchange with databases, calculations, ...) • SOAP/XML (client/server)	• user-defined states	• user documentation

Ethernet Communication Service Classes

The Ethernet communication services provided by devices are (identified by number):
- class 10: standard Ethernet communication services
- class 20: Ethernet communication management services (network level and device level)
- class 30: advanced Ethernet communication services

Transparent Ready devices can provide these Ethernet communication services:
- Modbus TCP/IP messaging service *(see page 104)*
- I/O scanning service *(see page 82)*
- FDR (Fast Device Replacement) service *(see page 99)*
- SNMP network management service *(see page 91)*
- Global Data service *(see page 96)*
- Bandwidth management service *(see page 100)*

The following table specifies the services provided for each Ethernet communication service class:

Ethernet Communication Service Classes		Ethernet communication services		
		Modbus Messaging	I/O Scanning	FDR
30	advanced services	direct reading/writing of I/O	• periodic read/write of I/O • config. of the list of scanned devices	automatic control/update of device parameter config.
20	communication management services			• automatic assignment of IP address and network parameters • control/update config. and device parameters by the user
10	standard services	reading/writing of data words		Local assignment of the IP address Verification of duplicate IP addresses

Ethernet communication service classes (continued):

Ethernet Communication Service Classes		Ethernet communication services		
		Network Management SNMP	Global Data	Bandwidth Management
30	advanced services	use of the MIB library by SNMP manager	publish/subscribe network variables	monitor load level
20	communication management services	detection of devices by SNMP manager		
10	standard services			

Choosing Transparent Ready Devices

Transparent Ready devices are chosen from four main families:

- sensor and actuator type field devices (simple or intelligent)
- controllers and PLCs
- Human Machine Interface (HMI) applications
- dedicated gateways and servers

The services provided by a given Transparent Ready device are identified by the level of Web service (a letter) followed by the level of Ethernet communication service (a number). For example:

- a class A10 product is a device in with class A Web services (none) and class 10 Ethernet services (standard)
- a class C30 product is a device with class C Web services (configurable) and class 30 Ethernet services (advanced)

NOTE: Service classes are incremental subsets. That is, class D includes all the services in class C, and class C includes all the services in class B. (Class A has no services.)

The selection table on the following pages can be used for choosing Transparent Ready devices according to the required service classes:

Ethernet Communication Services		Class A	Class B	Class C	Class D
		no service	standard	configurable	active
Class 30	advanced service	A30	B30	C30	D30
Class 20	communication management services	A20	B20	C20	D20
Class 10	standard services	A10	B10	C10	D10

Appendix D
Schneider Private MIB

About this Chapter

This chapter provides the detailed tree structure of the Schneider private MIB and a description of its services.

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
The Schneider Private MIB	360
Schneider Private MIB Tree Structure	362
MIB Subtree Description	370
Switch Subtree Description	371
Port 502 Messaging Subtree Description	372
I/O Scanning Subtree Description	373
Global Data Subtree Description	374
Web Subtree Description	375
Address Server Subtree Description	376
Equipment Profile Subtree Description	377
Time Management Subtree Description	379
Email Subtree Description	380
Transparent Factory MIB Version	381
Private Traps and MIB Files	382

The Schneider Private MIB

Introduction

A MIB (Management Information Base) is an element used in network management. Network management services are based on the need to monitor and manage:

- performance
- fault occurrences
- security

NOTE: The Transparent Factory private MIB does not define specific management applications and policies.

Each MIB contains a finite number of objects. Use the SNMP manager's GET and SET to retrieve system information and to set system environment variables.

Schneider Private MIB

The Transparent Factory SNMP-embedded component controls the Schneider private MIB function. This private MIB, and its associated services, manages all system components. The private MIB provides the data to manage the main Transparent Factory communication services for all the communication components of the Transparent Factory architecture, including:

- Ethernet communication modules (NOE, ETY, M1E, etc.)
- CPUs with Ethernet communication ports

Elsewhere in this guide is the detailed tree structure of the transparentFactoryEthernet MIB *(see page 362)*.

Private MIB Identifier

Schneider Electric obtained a Private Enterprise Number (PEN) from the Internet Assigned Numbers Authority (IANA). That number represents a subtree in the SNMP MIB, a number that is a unique identifier used for Groupe Schneider.

The object identifier for the root of the Groupe Schneider subtree is `1.3.6.1.4.1.3833` and represents a path to the subtree as follows:

```
iso(1)
    org(3)
        dod(6)
            internet(1)
                private(4)
                    enterprises(1)
                        groupeschneider(3833)
                            transparentFactoryEthernet(1)
```

Under the Groupe Schneider private MIB is a TFE private MIB, transparentFactoryEthernet(1).

Schneider Private MIB Tree Structure

Introduction

This topic outlines the tree structure for the private Schneider MIB (Schneider TFE-V01-04.mib) for all Transparent Ready products.

The groupeschneider (3833) subtree is the root of Groupe Schneider's private MIB in the Structure of Management Information (SMI) used by SNMP and defined in RFC-1155, a specification that defines the structure and identification of management information for TCP/IP-based networks.

Tree Structure

```
groupeschneider (3833)
(1) transparentFactoryEthernet
|---(1) switch
|------(14) saConfiguration
|----------(1) saChassis
|----------(2) saAgent
|----------(3) saUserGroup
|----------(5) saRingRedundancy
|----------(7) saLLDP
|------(15) saPlatform4
|----------(1) saPlatform4BasicL2
|---(2) Port502Messaging
|------(1) port502Status
|------(2) port502SupportedProtocol
|------(3) port502IpSecurity
|------(4) port502MaxConn
|------(5) port502LocalConn
|------(6) port502RemConn
|------(7) port502IpSecurityTable
|----------(1) port502IpSecurityEntry
|--------------(1) attemptFails
|--------------(2) ipSourceAddress
|------(8) port502ConnTable
|----------(1) port502ConnEntry
|----------(1) port502ConnLocalPort
```

```
|----------(2) port502ConnRemAddress
|----------(3) port502ConnRemPort
|----------(4) port502ConnType
|----------(5) port502ConnMsgIn
|----------(6) port502ConnMsgOut
|----------(7) port502ConnMsgErr
|----------(8) port502XwayNet
|----------(9) port502XwayStation
|------(9) port502MsgIn
|------(10) port502MsgOut
|------(11) port502MsgOutErr
|------(12) port502AddStackStat
|------(13) port502AddStackStatTable
|------------(1) port502AddStackStatEntry
|----------------(1) port502AddStackStatIndex
|----------------(2) port502PeaKTcpRetransSegs
|---(3) ioScanning
|------(1) ioScanStatus
|------(2) ioScanMaxDevice
|------(3) ioScanPolledDevice
|------(4) ioScanTransSend
|------(5) ioScanGlbHealth
|------(6) ioScanningDeviceTable
|----------(1) ioScanDeviceEntry
|--------------(1) IoScanDeviceRemAddress
|--------------(2) IoScanDeviceHealth
|--------------(3) IoScanDeviceRate
|--------------(4) ioScanInputLocalAddress
|--------------(5) ioScanOutputLocalAddress
|---(4) globalData
|------(1) glbDataStatus
|------(2) glbDataMaxPub
|------(3) glbDataMaxSub
```

```
|------(4) glbDataPub
|------(5) glbDataSub
|------(6) glbDataPubErr
|------(7) glbDataSubErr
|------(8) glbDataGlbSubHealth
|------(9) glbDataPubTable
|----------(1) glbDataPubEntry
|--------------(1) glbDataPubSourceAddress
|--------------(2) glbDataPubHostId
|--------------(3) glbDataPubNetId
|--------------(4) glbDataPubGroupId
|--------------(5) glbDataPubCnt
|--------------(6) glbDataPubErrCnt
|--------------(7) glbDataPubDistribRate
|--------------(8) glbDataPubDuplicateErr
|------(10) glbDataSubTable
|--------------(1) glbDataSubEntry
|------------------(1) glbDataSubSourceAddress
|------------------(2) glbDataSubHostId
|------------------(3) glbDataSubNetId
|------------------(4) glbDataSubGroupId
|------------------(5) glbDataSubCnt
|------------------(6) glbDataSubErrCnt
|------------------(7) glbDataMinimumSeparation
|------------------(8) glbDataHealth
|------------------(9) glbDataHealthTimeOut
|------------------(10) glbDataLastRecErr
|---(5) Web
|------(1) webStatus
|------(2) webPassword
|------(3) webSuccessfullAccess
|------(4) webFailedAttempts
|---(6) addressServer
```

```
|------(1) addressServerStatus
|---(7) equipmentProfile
|------(1) profileProductName
|------(2) profileVersion
|------(3) profileCommunicationServices
|------(4) profileGlobalStatus
|------(5) profileConfigMode
|------(6) profileRoleName
|------(7) profileBandwidthMgt
|------(8) profileBandwidthDistTable
|---------(1) profileBandwidthDistEntry
|------------(1) bandwidthDistributionIndex
|------------(2) port502Bandwidth
|------------(3) ioScanningBandwidth
|------------(4) globalDataBandwidth
|------------(5) otherBandwidth
|------(9) profileLedDisplayTable
|---------(1) profileLedDisplayEntry
|------------(1) ledIndex
|------------(2) ledName
|------------(3) ledDescr
|------------(4) ledState
|------(10) profileSlot
|------(11) profileCPUType
|------(12) profileTrapTableEntriesMax
|------(13) profileTrapTable
|---------(1) profileTrapEntry
|------------(1) trapCommunityName
|------------(2) remoteIpAddress
|------------(3) authenticationTrap
|------------(4) port502Trap
|------------(5) ioScanningTrap
|------------(6) globalDataTrap
```

```
|------------(7) webTrap
|------------(8) addressServerTrap
|------------(9) profileTrap
|------------(10) timeManagementTrap
|------------(11) emailTrap
|------(14) profileSpecificId
|------(15) profileIpAddress
|------(16) profileIpNetMask
|------(17) profileIpGateway
|------(18) profileMacAddress
|------(19) profileImplementationClass
|------(100) premiumProfile
|------(101) quantumProfile
|------------(100) qnoe
|---------------(1) qNoeCommand
|------(102) microProfile
|------------(100) mEtz
|---------------(1) etzIpMgtStatus
|---------------(2) etzIpMgtDhcpTries
|---------------(3) etzIpMgtDhcpMode
|---------------(4) etzRepUserBkups
|---------------(5) etzRepAutoBkups
|---------------(6) etzRepStatus
|---------------(7) etzRepTFPcnxErrors
|---------------(8) etzRepTFPxferErrors
|------(103) momentumIoProfile
|------------(1) momentumIoBaseType
|------------(2) momentumIoBaseName
|------------(3) momentumIoMasterIPTable
|---------------(1) momentumIoMasterIPEntry
|------------------(1) momentumIoMasterIPValue
|------------(4) momentumIoModuleTimeOut
|------------(5) momentumIoASCIIModuleHeader
```

```
|------------(6) momentumIoReservationTime
|------------(7) momentumIoInputDataTable
|---------------(1) momentumIoInputDataEntry
|------------------(1) momentumIoInputDataIndex
|------------------(2) momentumIoInputDataValues
|------------------(3) momentumIoInputDataWords
|------------------(4) momentumIoInputDataPoints
|------------(8) momentumIoOutputDataTable
|---------------(1) momentumIoOutputDataEntry
|------------------(1) momentumIoOutputDataIndex
|------------------(2) momentumIoOutputDataValues
|------------------(3) momentumIoOutputDataWords
|------------------(4) momentumIoOutputDataPoints
|------(104) momentumM1eProfile
|------(105) advantysProfile
|------(106) gatewayProfile
|------(107) modiconM340Profile
|------(255) tfProducts
|------------(1) ety
|------------(2) noe
|------------(3) etz
|------------(4) momentumIo
|------------(5) momentumM1e
|------------(6) altivar
|------------(7) stbNip
|------------(8) tsxntp
|------------(9) nwm
|------------(10) wmy
|------------(11) quantumPLC
|------------(12) premiumPLC
|------------(13) etg
|------------(14) egx
|------------(15) ecc
```

```
|------------(16) cev
|------------(17) inducteIXGKS
|------------(18) ositrackTAP
|------------(19) twidoPLC
|------------(20) modiconM340PLC
|------------(21) modiconM340DPLC
|------------(22) modiconM340CPLC
|------------(23) modiconM340NOE
|---(8) timeManagement
|------(1) ntp
|--------(1) ntpStatus
|--------(2) ntpSrvAddr
|--------(3) ntpLnkSrvStatus
|--------(4) ntpReqCnt
|--------(5) ntpRespCnt
|--------(6) ntpErrCnt
|--------(7) ntpDate
|--------(8) ntpTime
|--------(9) ntpTimeZone
|--------(10) ntpDSTStatus
|--------(11) ntpLastErr
|---(9) email
|------(1) smtp
|---------(1) emailTable
|------------(1) emailEntry
|--------------(1) emailIndex
|--------------(2) smtpStatus
|--------------(3) smtpSrvAddr
|--------------(4) smtpMailSentCnt
|--------------(5) smtpErrCnt
|--------------(6) smtpLastErr
|--------------(7) smtpLastMailElapsedTime
|--------------(8) smtpLnkSrvStatus
```

```
|--------------(9) smtpSrvChkFailCnt
|---(255) tfeMibVersion
|------(1) tfeMibVersionNumber
|------(2) tfeMibVersionDate
```

MIB Subtree Description

Transparent Factory Ethernet Subtree

This topic details some of the objects in the Schneider private MIB tree. The **transparentFactoryEthernet(1)** subtree defines groups that support the TFE services and devices:

Service	Subtree Definition
switch(1) *(see page 371)*	the brand of switches labeled
port502Messaging(2) *(see page 372)*	objects for managing explicit client/server communications to support applications (for example, HMI, SCADA, or programming tools)
ioScanning(3) *(see page 373)*	objects for managing I/O device communications that use the I/O Scanner with the Modbus/TCP protocol
globalData(4) *(see page 374)*	objects for managing the application coordination service using a publish/subscribe protocol
web(5)	objects for managing the activity of the embedded Web servers
addressServer(6) *(see page 376)*	objects for managing the activity of the BOOTP or DHCP servers
equipmentProfile(7) *(see page 377)*	objects for each device type in Transparent Factory Ethernet product portfolio
timeManagement(8) (NTP) *(see page 379)*	objects for managing the UTC time stamp service
email(9) (SMTP) *(see page 380)*	objects for managing the email service
tfeMibVersion(255) *(see page 381)*	the version of the Schneider TFE MIB supported by the product

NOTE: All listed services are not available on all communications modules. Refer to the available services for your module.

When devices are added to the Schneider catalog, the private MIB is extended in the following manner:

- If needed, a Transparent Factory communication-service object is added for the new device in the subtree that corresponds to **equipmentProfile(7)** *(see page 377)*. This subtree can hold as many objects as are required.
- If needed, a new branch is added at the same level as **transparentFactoryEthernet(1)**. This subtree is created for product-specific objects.

When a new device is added to the catalog a corresponding object description is created in the ASN.1 format. The ASN.1 file(s) are then given to producers of SNMP manager software for inclusion in their products.

Switch Subtree Description

Switch Subtree

The switch (1) subtree, or group, indicates the brand of switches labeled. The following list describes the function of each object.

Service	Indicates . . .
saChassis(1)	configuration of the chassis
saAgent(2)	configuration of Agent
saRingRedundancy(3)	management of Ring Redundancy
saUserGroup(5)	management of user groups
saLLDP(7)	management of proprietary extensions of 802.1AB (station and Media access control Connectivity Discovery)

Port 502 Messaging Subtree Description

Port 502 Messaging Subtree

The port502Messaging (2) subtree, or group, provides connection management and data flow services. The following list describes the function of each object.

Service	Indicates . . .
port502Status(1)	status of the service (idle or operational)
port502SupportedProtocol(2)	supported protocols (MODBUS, X-way, etc.)
port502IpSecurity(3)	status of the Port 502 IP Security service (enabled or disabled)
port502MaxConn(4)	maximum number of TCP connections supported by the Port 502 entity
port502LocalConn(5)	number of TCP connections currently opened by the local Port 502 entity
port502RemConn(6)	number of TCP connections currently opened by the remote entity to the local Port 502 entity
port502IpSecurityTable(7)	a table containing the number of unsuccessful attempts to open a TCP connection from a remote TCP entity
port502ConnTable(8)	a table containing Port 502 TCP specific information (MsgIn, MsgOut)
port502MsgIn(9)	total number of Port 502 messages received from the network
port502MsgOut(10)	total number of Port 502 messages sent from the network
port502MsgOutErr(11)	total number of diagnostic messages built by the Port 502 messaging entity and sent to the network
port502AddStackStat(12)	the support of additional Port 502 stack statistics
port502AddStackStatTable(13)	additional stack statistics for Port 502 (optional)

I/O Scanning Subtree Description

I/O Scanning Subtree

The ioScanning (3) subtree, or group, contains the objects related to I/O scanning device management and associated Modbus communications on port 502.

Service	Indicates . . .
ioScanStatus(1)	global status of the I/O scanning service
ioScanMaxDevice(2)	maximum number of devices supported by the I/O scanning entity
ioScanPolledDevice(3)	number of devices currently polled by the I/O scanning entity
ioScanTransSend(4)	total number of transactions sent by the I/O scanning entity
ioScanGlbHealth(5)	global health status for the I/O scanning service
ioScanningDeviceTable(6)	a table containing information on each remote device polled by the I/O scanning entity

Global Data Subtree Description

Global Data Subtree

The globalData (4) subtree, or group, contains the objects related to Global Data.

Service	Indicates . . .
glbDataStatus(1)	global status of the Global Data service
glbDataMaxPub(2)	maximum number of published variables configured by the Global Data entity
glbDataMaxSub(3)	maximum number of subscribed variables configured by the Global Data entity
glbDataPub(4)	total number of publications sent to the network
glbDataSub(5)	total number of subscriptions received from the network
glbDataPubErr(6)	total number of publication errors detected by the local entity
glbDataSubErr(7)	total number of subscription errors detected by the local entity
glbDataGlbSubHealth(8)	global health status of the Global Data service
glbDataPubTable(9)	a table containing information on each published variable (the number of publications, the source IP address, the number of errors, etc.)
glbDataSubTable(10)	a table containing information on each subscribed variable (the number of subscriptions, the source IP address, the number of errors, health, etc.)

Web Subtree Description

Web Subtree

The web (5) subtree, or group, contains the objects related to the Web server service.

Service	Indicates . . .
webStatus(1)	global status of the Web service
webPassword(2)	enable or disable Web passwords
webSuccessfullAccess(3)	total number of successful attempts to access Web site
webFailedAttempts(4)	total number of failed attempts to access Web site

Address Server Subtree Description

Address Server Subtree

The addressServer (6) subtree, or group, contains the objects related to the Address Server. The address server can be either a BOOTP server or a DHCP server.

Service	Indicates . . .
addressServerStatus(1)	global status of the address server service

Equipment Profile Subtree Description

Equipment Profile Subtree

The equipmentProfile (7) subtree contains a set of common objects.

Service	Indicates . . .
profileProductName(1)	the commercial name of the communication product in string form (for example: 140 NOE 771 11, BMX NOE 0100, etc.)
profileVersion(2)	the software version of the communication product in string form (for example, Vx.y or V1.1)
profileCommunicationServices(3)	the communication services supported by the profile (Port502Messaging, I/O scanning Messaging, Global Data, Web, and Address Server)
profileGlobalStatus(4)	the global status of the communication module
profileConfigMode(5)	the IP configuration mode of the communication module
profileRoleName(6)	the role name for the IP address management if it exists (empty string if there is none)
profileBandwidthMgt(7)	the status of Bandwidth Management
profileBandwidthDistTable(8)	the CPU time distribution between Global Data, Port 502 Messaging, I/O scanning
profileLedDisplayTable(9)	a table giving the name and the state of each module's LEDs
profileSlot(10)	the position of the communication module inside the rack if there is one (if there is no rack, the profileSlot value is 0)
profileCPUType(11)	the host for which that communication module is a part when a CPU type exists (if there is no host, the string is empty)
profileTrapTableEntriesMax(12)	the maximum numbers of entries in the Trap Table (equal to the number of possible remote managers)
profileTrapTable(13)	a table allowing you to enable or disable the private traps for each of the communication services
profileSpecificId(14)	a unique Profile Specific Identification inside the equipmentProfile object of the Schneider Transparent Factory MIB (for example, the PLC Premium family is 100)
profileIpAddress(15)	the IP address of the SNMP agent
profileIpNetMask(16)	the subnetwork mask associated with the IP address of the SNMP agent (the value of the mask is an IP address with all the network bits set to 1 and all the host bits set to 0)
profileIpGateway(17)	the default Gateway IP address of the SNMP agent
profileMacAddress(18)	the Ethernet media-dependent address of the SNMP agent
profileImplementationClass(19)	a textual description of the implementation class supported by the product

Service	Indicates . . .
premiumProfile(100)	managed products (ETY, ETY port)
quantumProfile(101)	managed products (NOE)
microProfile(102)	managed products (ETZ)
momentumIoProfile(103)	managed products (ENT)
momentumM1eProfile(104)	managed products (M1E)
advantysProfile(105)	managed products (STB NIP)
gatewayProfile(106)	managed products (ETG)
modiconM340profile(107)	managed products (Modicon M340 PLC)
tfProducts(225)	Transparent Factory products

Time Management Subtree Description

Time Management Subtree

The timeManagement (8) subtree contains a set of common NTP objects.

Service	Indicates . . .
ntpStatus(1)	the status of the NTP service (not server)
ntpSrvAddr(2)	the IP address of the NTP server in dot notation format
ntpLnkSrvStatus(3)	the status of the link between the module and the NTP server
ntpReqCnt(4)	the number of requests sent to the NTP server
ntpRespCnt(5)	the number of responses received from the NTP server
ntpErrCnt(6)	the total number of communication errors
ntpDate(7)	date of the day
ntpTime(8)	time of the day
ntpTimeZone(9)	current time zone
ntpDSTStatus(10)	daylight saving time status
ntpLastErr(11)	last error code generated by system

Email Subtree Description

Email Subtree

The email(9) subtree contains a set of common SMTP objects.

Service	Indicates . . .
emailIndex(1)	the index value in the email service table
smtpStatus(2)	the status of SMTP service (not server)
smtpSrvAddr(3)	the IP address of SMTP server in dot notation format
smtpMailSentCnt(4)	the total number of emails sent to the network and successfully acknowledged by the server
smtpErrCnt(5)	the total number of email messages that could not be sent to the network or that have been sent but not acknowledged by the server
smtpLastErr(6)	the error code of the last error that occurred while trying to send an email message to the network
smtpLastMailElapsedTime(7)	the number of elapsed seconds since last successful email was sent to the server
smtpLnkSrvStatus(8)	the status of link with SMTP server
smtpSrvChkFailCnt(9)	the number of times the link to SMTP server is detected as ‘down.’

Transparent Factory MIB Version

tfeMibVersion Subtree

This group contains information about the version of the Schneider TFE MIB *(see page 360)* supported by the product.

Service	Indicates . . .
tfeMibVersionNumber(1)	the version of the SchneiderTFE Mib in Vxx.yy form (example V01.04)
tfeMibVersionDate(2)	the date of last update of the SchneiderTFE MIB in ‘ddMmmyy’ form (example: 09Jan06)

Private Traps and MIB Files

Private Traps and MIB Files

Traps are used to signal status changes to the manager while avoiding additional traffic:

- **LEDs** (`profileLED`): This trap is sent if the LED state changes.
- **communications ports** (`port502StatusChange`): This trap is sent if `port502Status` changes.
- **I/O scanning health value** (`ioScanStatusChange`): This trap is sent if `ioScanStatus` changes.
- **global data health value** (`glbDataStatusChange`): This trap is sent if `glbDataStatus` changes.
- **Web service** (`webStatusChange`): This trap is sent if `webStatus` changes.
- **address server** (`addressServerStatusChange`): This trap is sent if `addressServer-Status` changes.
- **NTP service** (see below)
- **SMTP service** (see below)

Private traps can:

- send messages to the two managers whose IP addresses are configured in the SNMP configuration
- use the community name given to this configuration
- enable or disable each of the Transparent Factory Ethernet Private MIB groups listed in the Transparent Factory Ethernet Subtree *(see page 370)*.

Private traps are described in the MIB ASN.1 description, which is contained in a `.mib` text file.

NTP Traps

- **NTP status** (`ntpStatusChange`): This trap is sent if `ntpStatus` changes.
- **server change** (`ntpServerChange`): This trap is sent if the NTP component switches from the Primary NTP server to the standby NTP server or vice versa.
- **link server status change** (`ntpLnkSrvStatusChange`): This trap is sent if the NTP link server status changes.
- **leap second** (`ntpLeapSecond`): This trap is sent when leap seconds are inserted.
- **DST change** (`ntpDSTChange`): This trap notifies the manager that the NTP server time has changed from either:
  - standard time to daylight savings time, or
  - daylight savings time to standard time

SMTP Traps

- **SMTP status change** (`smtpStatusChange`): This trap is sent if `smtpStatus` of the email service referenced by emailIndex changes.
- **SMTP link to server status** (`smtpLnkSrvChange`): This trap is sent when the `smtpLnkSrvStatus` of the email service referenced by `emailIndex` changes. The trap is sent when the service tries to send an email. Every 30 minutes a periodic test checks the connection to the SMTP server.

Glossary

!

%I

Represents an input bit.

%IW

Represents an input word register.

%M

Represents a memory bit.

%MW

Represents a memory word register.

%QW

Represents an output word register.

%S

Represents a system bit.

%SW

Represents a system word register.

10/100 Base-T

An adaptation of the IEEE 802.3 (Ethernet) standard, the 10/100 Base-T standard uses twisted-pair wiring with a maximum segment length of 100 m (328 ft) and terminates with an RJ-45 connector. A 10/100Base-T network is capable of transmitting data on normal Ethernet (10 Mbit/s) and Fast Ethernet (100 Mbits/s) networks.

802.3 frame

A frame format, specified in the IEEE 802.3 (Ethernet) standard, in which the header specifies the data packet length.

A

ASN.1

Abstract Syntax Notation One. ASN.1 is a method for encoding/decoding messages sent between systems of different types that use different languages. It is defined by ISO standards 8824/ITU X.208 and 8825/ITU X.209.

B

BOOTP

bootstrap protocol. A UDP/IP protocol that allows an Internet node to obtain its IP parameters based on its MAC address.

bps

bits per second.

bridge

A bridge device connects two or more physical networks that use the same protocol. Bridges read frames and decide whether to transmit or block them based on their destination address.

broadcast

Broadcast communications send packets from a one station to every network destination. Broadcast messages pertain to every network device or only one device for which the address is not known. (See *multicast* and *unicast*).

C

CAN

controller area network. The CAN protocol (ISO 11898) for serial bus networks is designed for the interconnection of smart devices (from multiple manufacturers) in smart systems for real-time industrial applications. CAN multi-master systems provide high data integrity through the implementation of broadcast messaging and advanced error detection mechanisms. Originally developed for use in automobiles, CAN is now used in a variety of industrial automation control environments.

CANopen

CANopen is higher level protocol that is used in automation networks. It is based on the CAN application layer (CAL) in accordance with CiA DS 301 (EN 50325-4).

channel

A logic RTU master or slave in an RTU module.

configuration

The arrangement and interconnection of hardware components within a system and the hardware and software selections that determine the operating characteristics of the system.

ConneXview

ConneXview is a set of configuration files to be used with HiVision 6.x network management software from Hirschmann Electronics GmbH & Co. KG. ConneXview makes it possible to manage Schneider Electric Transparent Factory devices using HiVision 6.0 or newer. ConneXview is built on the widely used simple network management protocol (SNMP).

D

default gateway

The IP address of the network or host to which all packets addressed to an unknown network or host are sent. The default gateway is typically a router or other device.

device name

A user defined, unique logical personal identifier for a network device. After the Ethernet communications module is configured with a valid device name, the DHCP server uses it to identify the rack at power up.

DHCP

dynamic host configuration protocol. DHCP is a TCP/IP protocol that allows network devices (DHCP clients) to obtain their IP addresses from a DHCP server through a request to the server.

E

EcoStruxure™ Control Expert

Control Expert is the programming software for all PACs. The software includes five IEC languages that comply with IEC 61131-3. Depending on requirements, the application may use a mixture of different languages.

EFB

elementary function block. EFBs are the elementary functions and function blocks (based on C language) that can be user-customized and stored in different block libraries.

embedded Web pages

Embedded Web pages (accessed by an installed HTTP server) provide Ethernet communications modules with easy access to devices anywhere in the world from standard browsers such as Internet Explorer or Netscape Navigator.

EMC

electromagnetic compatibility. Devices that meet EMC requirements can operate within a system's expected electromagnetic limits.

Ethernet

A LAN cabling and signaling specification used to connect devices within a defined area, e.g., a building. Ethernet uses a bus or a star topology to connect different nodes on a network.

Ethernet II

A frame format in which the header specifies the packet type, Ethernet II is the default frame format for STB NIP 2212 communications.

F

FactoryCast

FactoryCast is an open automation framework based on Internet technologies that is designed to provide seamless communication between plant floor and business systems. Its main capabilities include:

- Modbus TCP/IP for client-server messaging
- I/O scanner for handling I/O devices
- embedded web services for diagnostics and configuration
- a full set of Internet protocols

FAST

The fast (FAST) task is a periodic, high-priority task of a short duration that is run on a processor through its programming software. The fast speed of the task keeps it from interfering with the execution of lower priority master (MAST) tasks. A FAST task is useful when fast periodic changes in discrete inputs need to be monitored.

FDR

The *fast device replacement* service offers a method of handling device replacement without disrupting the system nor interrupting service.

Flash memory

Flash memory is nonvolatile memory that can be overwritten. It is stored on a special EEPROM that can be erased and reprogrammed.

frame

A frame is a group of bits that form a discrete block of information. Frames contain network control information or data. The size and composition of a frame is determined by the network technology being used.

framing type

Two common framing types for Ethernet are Ethernet II and IEEE 802.3.

FTP

File Transfer Protocol. FTP is the World Wide Web's file transfer protocol.

G

gateway

A device that connects networks with dissimilar network architectures and which operates at the Application Layer of the OSI model. This term may refer to a router.

Global Data

Global Data provides the automatic exchange of data variables for the coordination of PLC applications.

GMRP

GARP multicast registration protocol. GMRP is a GARP (Generic Attribute Registration Protocol) application that allows switches and bridges to dynamically manage the membership of multicast groups. GMRP is defined by IEEE 802.1D.

H

half duplex (HDX)

A method of data transmission capable of communication in either of two directions, but in only one direction at a time.

HMI

human-machine interface. An operator interface, usually graphical, for industrial equipment.

hot swapping

Replacing a component with a like component while the system remains operational. When the replacement component is installed, it begins to function automatically.

HTTP

HyperText Transfer Protocol. HTTP is the protocol for the formatting and transmission of files on the world wide web. HTTP runs on top of TCP/IP (Internet) protocols.

HTTP server

The installed HTTP server transmits Web pages between a server and a browser, providing Ethernet communications modules with easy access to devices anywhere in the world from standard browsers such as Internet Explorer or Netscape Navigator.

hub

A hub device connects a series of flexible and centralized modules to create a network.

I

I/O module

In a programmable controller system, an I/O module interfaces directly to the sensors and actuators of the machine/process. This module is the component that mounts in an I/O base and provides electrical connections between the controller and the field devices. Normal I/O module capacities are offered in a variety of signal levels and capacities.

I/O Scan List

A configuration table which identifies the targets with which repetitive communication is authorized.

I/O scanning

An I/O scan continuously polls I/O modules to collect data bits and status and diagnostics information. This process monitors inputs and control outputs.

ICMP

Internet Control Message Protocol. ICMP is a protocol within TCP/IP that reports detected errors in datagram transmissions.

IEEE

Institute of Electrical and Electronics Engineers, Inc. The international standards and conformity assessment body for all fields of electrotechnology, including electricity and electronics.

IOA

information object access. IOA is the exchange protocol carried out over the HTTP (HyperText Transfer Protocol) channel.

IODDT

input/output derived data type. IODDT is a structured data type representing a module or a channel of a PLC module. Each application expert module possesses its own IODDTs.

IP

Internet protocol. That part of the TCP/IP protocol family that tracks the Internet addresses of nodes, routes outgoing messages, and recognizes incoming messages.

IP address

Internet protocol address. This 32-bit address is assigned to hosts that use TCP/IP.

L

LAN

local area network. A short-distance data communications network.

layer

In the OSI model, a layer is a portion of the structure of a device that provides defined services for the transfer of information.

LED

light emitting diode. An indicator that lights up when electricity passes through it. It indicates the operation status of a communications module.

M

MAC address

media access control address. A 48-bit number, unique on a network, that is programmed into each network card or device when it is manufactured.

MAST

A master (MAST) task is a processor task that is run through its programming software. The MAST task has two sections:

- **IN:** Inputs are copied to the IN section before execution of the MAST task.
- **OUT:** Outputs are copied to the OUT section after execution of the MAST task.

MIB

management information base. The MIB is an object database that is monitored by a network management system like SNMP. SNMP monitors devices that are defined by their MIBs. Schneider has obtained a private MIB, groupeschneider (3833).

Modbus

Modbus is an application layer messaging protocol. Modbus provides client and server communications between devices connected on different types of buses or networks. Modbus offers many services specified by function codes. There are two types of Modbus transmission, based on information in the physical layer:
- MB/serial: the Modbus type that transmits data over serial RS-232 and RS-422/485
- MB/TCP: the Modbus type that transmits data over Ethernet

multicast

Multicast communications send packets from a single source to a predefined *multicast group* of network destinations, usually through a router or switch. Sending messages to just the group members relieves unnecessary traffic created by broadcast communications and does not require a separate unicast transmissions to each recipient. (See *broadcast*, *unicast*, *GMRP*.)

multicast filtering

Multicast filtering is a process for deciding that multicast messages are delivered only to the stations that are registered members of the appropriate *multicast group*.

N

NMT

network management. NMT protocols provide services for network initialization, diagnostics, and device status control.

NTP

network time protocol. NTP synchronizes the time of one client or server to the time of another server or referenced source (such as a satellite receiver).

O

OSI model

Open Systems Interconnection model. The OSI reference model is the abstract seven-layer model for establishing logical communications and protocol design. The model was developed by the International Standards Organization (ISO).

P

PAC

programmable automation controller The PAC is the brain of an industrial manufacturing automation process. It automates a process as opposed to relay control systems. PACs are computers suited to survive the harsh conditions of the industrial environment.

packet

The unit of data sent across a network.

PING

packet Internet groper. A PING program tests communications to another network destination.

PL7

PL7 software from Telemecanique is a programming language for TSX Micro and Modicon Premium PLCs.

PLC

programmable logic controller. The PLC is the brain of an industrial manufacturing process. It automates a process as opposed to relay control systems. PLCs are computers suited to survive the harsh conditions of the industrial environment.

port 502

TCP/IP reserves specific server ports for specific applications through IANA (Internet Assigned Numbers Authority). Modbus requests are sent to registered software port 502.

private MIB

Schneider has obtained a private MIB, groupeschneider (3833). Under the Groupe Schneider private MIB is a Transparent Factory Ethernet (TFE) private MIB. The Transparent Factory SNMP embedded component controls the Schneider private MIB function. This MIB includes a set of data that enables the network management system to supervise all the Transparent Ready services. The Transparent Ready private MIB can be downloaded from the Web server.

PUB

A Global Data variable that is published.

R

router

A router device connects two or more sections of a network and allows information to flow between them. A router examines every packet it receives and decides whether to block the packet from the rest of the network or transmit it. The router attempts to send the packet through the network on an efficient path.

RTU

Remote Terminal Unit.

S

service class

Transparent Ready service classes make it possible to identify the services provided by each device, such as:
- diagnostic, display, and control services via Web technologies
- Ethernet communication services

The Transparent Ready service classes thus simplify the choice of products and check their interoperability within an architecture.

SMTP

Simple Mail Transfer Protocol. SMTP is a transmission protocol for sending and receiving e-mail. SMTP messages are usually retrieved from a server with an e-mail client (such as POP or IMAP).

SNMP

simple network management protocol. The UDP/IP standard protocol used to monitor and manage devices on an IP network.

SNMP agent

The SNMP application that runs on a network device.

SUB

A Global Data variable that is defined as a subscription variable.

subnet

The subnet is that portion of the network that shares a network address with the other parts of the network. A subnet may be physically or logically independent from the rest of the network. A part of an Internet address called a subnet number, which is ignored in IP routing, distinguishes the subnet.

subnet mask

The subnet mask is a bit mask that identifies or determines which bits in an IP address correspond to the network address and which correspond to the subnet portions of the address. The subnet mask comprises the network address plus the bits reserved for identifying the subnetwork.

switch

A network switch connects two or more separate network segments and allows traffic to be passed between them. A switch determines whether a frame should be blocked or transmitted based on its destination address.

T

TCP/IP

Transmission Control Protocol/Internet Protocol. TCP/IP is the communication protocol of the Internet.

TFE

transparent factory Ethernet. Schneider Electric’s open automation framework based on TCP/IP.

TFTP

Trivial File Transfer Protocol. TFTP is a scaled-down version of FTP that uses UDP, often to initialize diskless workstations.

Transparent Device Access

Transparent Device Access (TDA) functionality means that clients that run Control Expert (and that are connected to a USB, Ethernet, or Modbus terminal port of a communications module) can access or download applications to devices on distributed control networks. The reverse, however, is not true. In other words, a Control Expert PC connected to the CPU's Modbus port can access devices on other core networks, but those remote devices can not access other devices on different networks through the PLC station.

Transparent Factory

See TFE.

Transparent Ready

Schneider Electric's Transparent Ready products (based on universal Ethernet TCP/IP and Web technologies) can be integrated into real-time, data sharing systems, with no need for interfaces.

U

UDP

user datagram protocol. UDP is an Internet communications protocol defined by IETF RFC 768. This protocol facilitates the direct transmission of datagrams on IP networks. UDP/IP messages do not expect a response, and are therefore ideal for applications in which dropped packets do not require retransmission (such as streaming video and networks that demand real-time performance).

unicast

Unicast communications send point-to-point packets from a single source to a specific network destination. It is an efficient means of communication between hosts that has a minimal impact on network traffic. (See *broadcast* and *multicast*.)

Unity Pro

Unity Pro is the programming software for all Unity PLCs. It includes 5 IEC languages that comply with IEC 61131-3. Depending on requirements, the application may use a mixture of different languages.

NOTE: Unity Pro is the former name of Control Expert for version 13.1 or earlier.

USB

universal serial bus. USB is a nearly universal hardware interface for connecting peripheral devices.

V

variable

A variable is a memory entity of the type BOOL, WORD, DWORD, etc., whose contents can be modified by the program during execution.

Index

0-9

10/100 BASE-T interface, *33*

A

addressing
  BOOTP, *86*
  DHCP, *86*
Advantys, *175*

B

bandwidth checking
  configuration parameters, *198*
bandwidth monitoring, *100*, *126*
block
  add timer to verify completion of block, *334*
BMXNOE01x0, *123*
  close memory card door, *23*
BMXP342020, *129*
BMXP342030, *129*
BMXRMS008MP, *50*
BMXRMS008MPF, *50*
BMXRMS128MPF, *50*
BMXRWSB000M, *50*
BMXRWSC016M, *50*

C

certifications, *128*
channel data structure for all modules
  T_GEN_MOD, *246*
configuration
  bandwidth checking, *198*
configuring for Ethernet, *133*
Control Expert
  Advantys, *168*
  DTM container, *168*

D

debugging communication, *215*
DFB
  add timer to verify completion of block, *334*
DHCP addressing, *125*
DTM container, *168*

E

electronic mail notification, *115*
error message
  without memory card, *56*
Ethernet modules, *21*
  communication requirements, *19*
  communication services, *59*
  device name, *64*, *71*
  hardware, *64*, *121*
  hardware requirements, *19*
  hardware selection, *39*
  IP parameters, *61*
  language objects, *225*
  memory card selection, *50*
  module overview, *35*
  port status, *70*
  Web pages, *265*
Ethernet network, *36*

F

fast device replacement, *99*, *126*
frame format, *69*
frame size
  Ethernet, *124*
function block
  add timer to verify completion of block, *334*

G

global data, *96*, *125*
grounding, *48*

I

I/O scanner, *82*, *124*
I/O scanning
  BMX NOE Ethernet, *149*
  multiple lines, *156*
IP address, *347*
IP parameters, *61*

M

managed variables, *175*
memory card
  features, *55*
memory card door
  close, BMXNOE01x0, *23*
memory cards, *50*, *50*
MIB, *92*, *359*
Modbus TCP messaging, *124*

N

NTP, *108*, *126*

P

parameter settings, *226*
PRA
  Control Expert, *168*

S

selecting Ethernet devices, *357*
SEND_EMAIL, *118*
service classes
  Transparent Ready, *355*
services
  addressing (BOOTP, DHCP), *86*
  bandwidth monitoring, *100*, *126*
  DHCP addressing, *125*
  electronic mail notification, *115*
  Ethernet, *81*
  fast device replacement, *99*, *126*
  global data, *96*, *125*
  I/O scanner, *82*, *124*
  Modbus TCP messaging, *124*
  NTP, *126*
  Security, *124*, *130*
  selecting, *42*
  SNMP, *91*, *125*
  TCP/IP messaging, *104*
  time synchronization, *108*
SMTP, *115*
SNMP, *91*, *125*
software configuration parameters, *135*
standards, *128*

T

T_COM_EIP
  M340, *228*
T_COM_ETH_BMX, *233*
T_GEN_MOD, *246*
TCP/IP
  messaging, *104*
  technical characteristics, *333*
TDA, *106*
time synchronization, *108*, *126*
timer
  add to block to verify completion of block, *334*

W

Web pages, *265*
website, *19*