Modicon Quantum Ethernet TCP/IP Module User Guide

840 USE 107 00 Version 4.0

043511452 76

Table of Contents

About the Book

At a Glance

Document Scope

This manual will acquaint you with the Ethernet TCP/IP modules (140-NOE-211-XX and 140-NOE-251-XX) and their parts, tell you how to install them, describe changes you may make in configuration, review the operation of the modules and provide maintenance procedures. It also describes how to obtain statistics about an Ethernet module and its controller from the embedded World Wide Web site.This manual is written for an Ethernet user and assumes familiarity with Ethernet networks. If you are not familiar with Ethernet, please consult your system administrator before connecting this module to your network.This manual also assumes that the user is acquainted with Quantum Automation Series control systems. For information about Quantum products, please refer to the Quantum Automation Series Hardware Reference Guide.

Validity Note

For the Ethernet module to work properly, you must have the proper version of other system components.Use the version specified in the table below or a later version Use the version specified in the table below or a later version.

Quantum Executive	**Modsoft**	**Concept**	**ModLink**
2.1	2.6	2.2 or higher	2.0

Related Documents

Title of Documentation	Reference Number
Modicon TSX Quantum Automation Series Hardware Reference Guide 840 USE 100 00 Ever. 6	Reference to related document
Modicon ModLink User Guide 890 USE 129 00	
Modsoft Programmer User Manual 890 USE 115 00	
Modbus Protocol Reference Guide PI-MBUS-300	

Product Related Warnings

Schneider Electric assumes no responsibility for any errors that may appear in this document. If you have any suggestions for improvements or amendments or have found errors in this publication, please notify us.

No part of this document may be reproduced in any form or by any means, electronic or mechanical, including photocopying, without express written permission of the publisher, Schneider Electric.

User Comments

We welcome your comments about this document. You can reach us by e-mail at TECHCOMM@modicon.com

Introduction

1

At a Glance

Overview

This chapter describes two of the four Modicon-designed Ethernet TCP/IP modules that allow a Quantum industrial control system to communicate with devices on an Ethernet network.

What's in this Chapter?

This chapter contains the following topics:

Ethernet TCP/IP Modules

At a Glance

The Quantum Ethernet TCP/IP modules make it possible for a Quantum industrial control system to communicate with devices on an Ethernet network. For example, the modules can be used to link a Quantum Automation Series controller to a PC. Each module contains a World Wide Web server, which allows users to obtain statistics about the NOE module and its controller from an embedded web site.

The Ethernet network is well supported worldwide, with a wide variety of third party products and services. TCP/IP is the de facto standard protocol.

Benefits of Quantum Design

Like all Quantum modules, the Ethernet modules are easy to install. They may be inserted into existing Quantum systems and connected to existing Ethernet networks. They do not require proprietary cabling.

The modules may be plugged into any slot in a local Quantum backplane and may be replaced while the system is running (hot swapped). They come fully configured and are recognized by the controller as soon as they connect with the backplane.

TCP/IP Module Models

Modicon has designed four Ethernet TCP/IP modules. Two are covered here: one for fiber optic networks and the other for networks using twisted pair cabling. Module part numbers are listed below.

Type of Cable Network	Part Number
Twisted Pair	140 NOE 211 00
Fiber Optic	140 NOE 251 00

Note: The Ethernet module must be routed through an Ethernet hub to function properly. Do not connect it directly to another device.

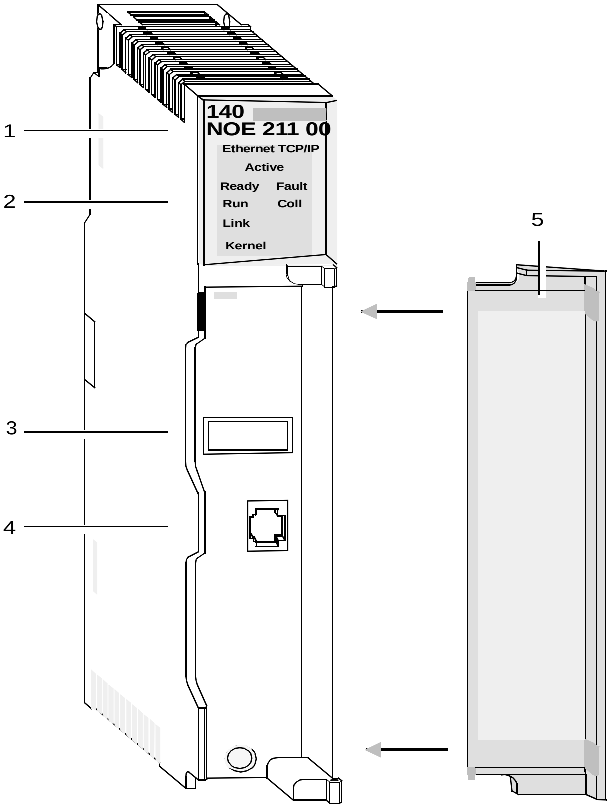
Front Panel Components

At a Glance

On the front panel of each Ethernet TCP/IP module, you will find an LED display, a global address label and a cable connector. Each component is described below.

140 NOE 211 00 Panel Layout

The front panel layout of the 140 NOE 211 00 module for twisted pair networks is shown below.



1 Model Number, Module Description, Color Code

2 LED Display

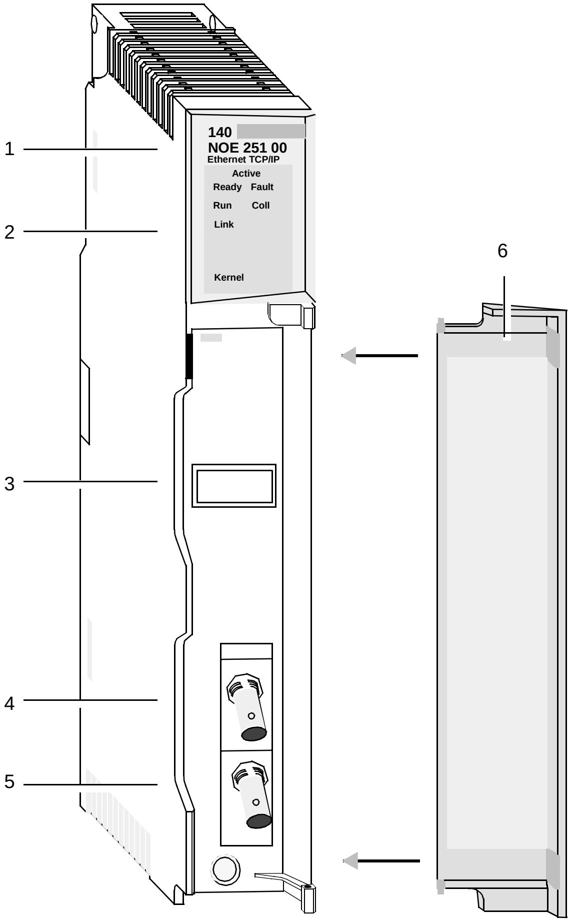
3 Global Address Label

4 Cable Connector

5 Removable Door

140 NOE 251 00 Panel Layout

The front panel layout of the 140 NOE 251 00 module for fiber optic networks is shown below.



1 Model Number, Module Description, Color Code

2 LED Display

3 Global Address Label

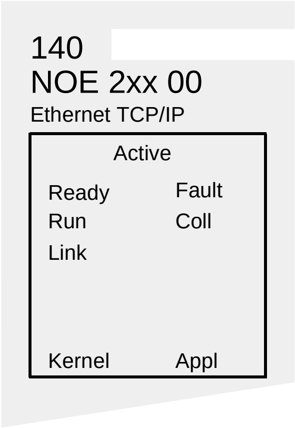
4 Transmit Cable Connector

5 Receive Cable Connector

6 Removable Door

LED Display

The LED Display is similar for both models as shown below.



LED	Color	Indication When ON
Active	Green	Module is communicating with backplane.
Ready	Green	Module has passed internal diagnostic tests.
Run	Green	Flashes during normal operation.
Link	Green	Ethernet link to hub is OK.
Kernel	Amber	If steady, module is operating in kernel mode. If flashing, module is waiting for download.
Fault	Red	An error has been detected, or a download has failed, or a reset is in process.
Coll	Red	If steady, cable is not connected. If flashing, Ethernet collisions are occurring.
Appl	Amber	Entry exists in crash log.

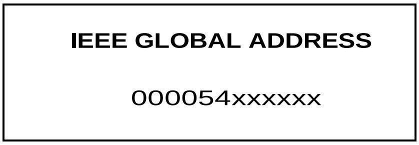
Address Labels

Each Quantum Ethernet web embedded server module has two address labels. One identifies the Ethernet or MAC address. The other label allows you to record the module’s Internet Protocol (IP) network address.

Ethernet Address Label

The Ethernet address or MAC address is assigned at the factory and is recorded on a label on the front panel, above the cable connector. This is a unique 48-bit global assigned address. It is set in PROM. The Ethernet address is recorded on the label in hexadecimal, in the form 00.00.54.xx.xx.xx

The Ethernet Address Label is similar to the illustration below.



IEEE GLOBAL ADDRESS

000054xxxxxx

Internet Protocol (IP) Network Address Label

You can use the derived address, which is calculated from the Ethernet address set by the factory. Or you can configure a unique address via Modsoft or Concept. Throughout this book, these alternatives will be referred to as the derived IP network address and a user-configured address.

The IP network address has the form xxx.xxx.xxx.xxx, where each group xxx is a decimal number from 0 to 255. A space is provided for recording this address on the label inside the front door panel of the module.

If you will be operating on an open network, you should opt for a user-configured address. Obtain a valid address from your network administrator.

If you will be operating on a local network, you may use the derived IP network address. However, you should check with your network administrator first to ensure that this address is not already in use.

To calculate the derived IP network address, convert the rightmost eight digits of the Ethernet address from hex to decimal. They will take the form 84.xxx.xxx.xxx, where each group xxx is a decimal number from 0 to 255.

Example: Calculating the Derived IP Network Address

The example below shows how to calculate the derived IP network address.

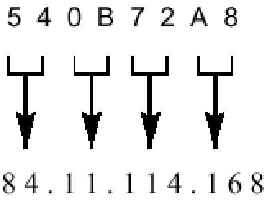
Locate the global address label on the front panel of the module.

Note the rightmost eight digits.

Convert them from hexadecimal to decimal. Each pair of hexadecimal numbers will result in a decimal number between 0 and 255. This is the derived IP address.

IEEE GLOBAL ADDRESS

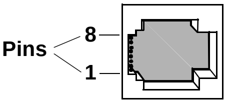
0000540B72A8



> **Note:** When you have determined which IP network address you will be using, register it with your system administrator to avoid duplication.

Twisted Pair Connector

The NOE 211 Module’s Twisted Pair connector is shown below.



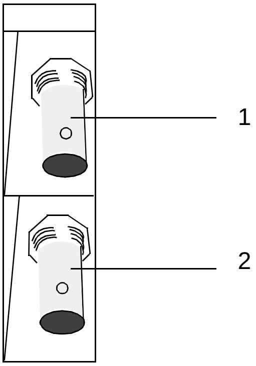
For the NOE 211, Schneider Automation recommends that you use Category 5 UTP cabling, which is rated to 100 Mbps, with an RJ-45 connector. You may also use Category 3 UTP cabling, which is rated to 16 Mbps.

The eight pins are arranged vertically and numbered in order from the bottom to the top. The RJ-45 pinout used by this module is:

- Receive Data (+): 3
- Receive Data (-): 6
- Transmit Data (+): 1
- Transmit Data (-): 2

Fiber Optic Connectors

The NOE 251 Module’s Fiber Optic Connectors are shown below.



1 Transmit Cable Connector

2 Receive Cable Connector

For the NOE 251, you need 62.5/125 micron fiber optic cable with ST-style connectors. Schneider Automation offers a 3 m cable with connectors (990 XCA 656 09).

This module comes with two fiber cable clasps and tubular plastic tools for installing the cable.

Utility Diskette

At a Glance

Included with this manual is a diskette containing two utilities for the Ethernet module: the Network Options Ethernet Tester utility and the ERRLOG utility.

Network Options Ethernet Tester

This utility will allow you to:

- establish a connection
- get and clear statistics
- read and write registers

The Network Options Ethernet Tester communicates with the module over the network from an IBM-compatible PC operating with Windows 3.1 or greater, and with WinSock. The source code for the Network Options Ethernet Tester is included on the diskette. For instructions on using the Network Options Ethernet, see *Ethernet Tester: Establishing a Connection with an Ethernet Module, p. 85*.

ERRLOG

This utility allows you to read and clear the crash log from an IBM-compatible PC communicating with the local Quantum controller via Modbus Plus.

The PC must be equipped with an SA85 Modbus Plus card and software driver. ERRLOG may be run in a native DOS environment or in a DOS box under Windows 3.1 or Windows 95. For instructions on using ERRLOG to read and clear the crash log, see *Responding to Errors, p. 96*.

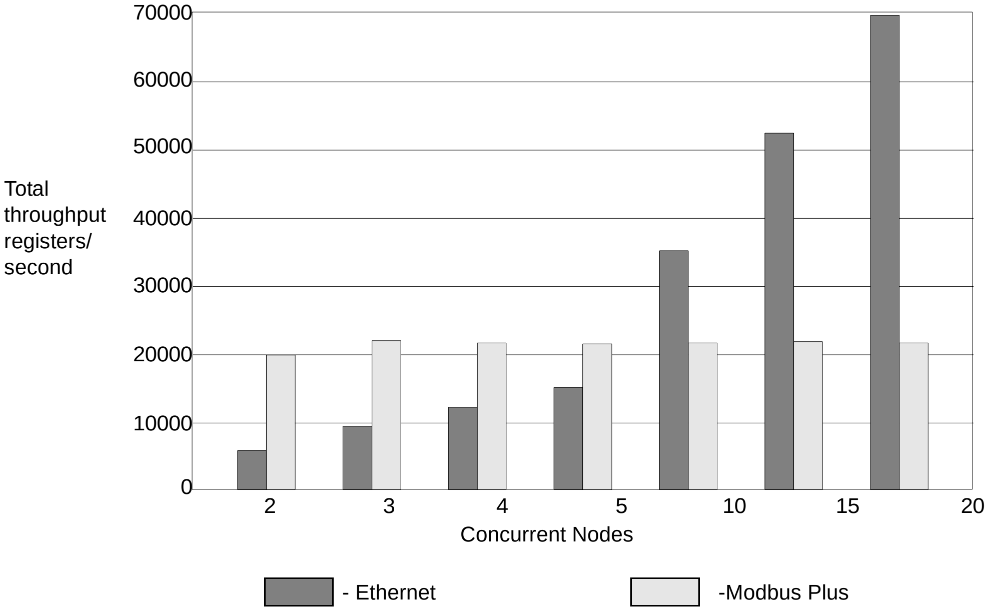
Ethernet and Your Application

At a Glance

Careful planning of your network can help you achieve optimum performance. You should consider whether Ethernet meets the demands of your application, which devices are compatible with your network and how to minimize congestion on the network.

Meeting the Demands of Your Application

The Quantum Ethernet modules provide connectivity to many different systems via an Ethernet network. However, Ethernet installations have characteristics which may not be suitable for all control applications.
The following graphic illustrates typical throughput versus concurrent nodes on a network.



Note: This data was measured between Quantum controllers on an otherwise empty LAN and as such reflects best case operation.

Ethernet network traffic, message length and routing are all variable and can be unpredictable. This can give rise to congestion and message collisions. When collisions occur, Ethernet uses a variable delay before retransmitting messages. Therefore, absolute determinism -- or totally predictable performance -- cannot be guaranteed on busy Ethernet networks.

Compatibility

Ethernet technology allows devices from different vendors to coexist on the same network. These devices include hubs, bridges, routers and gateways. However, for these devices to be compatible they must support the same set of protocols. Quantum Ethernet modules support Modbus protocol over TCP/IP. Systems that wish to communicate with Quantum Ethernet modules need to support this protocol stack.

Ethernet Developers Kit The Modbus protocol was chosen for its particular suitability for the real time control environment. It is a well-known and widely-adopted protocol and is fully described in the Ethernet Developers Kit. This kit (140 EDK 211 00) helps users develop Ethernet-based communications to their own host (PC-based) sockets applications. It contains a Quantum Ethernet module plus documentation and software tools which fully explain the protocols. The Ethernet Developers Kit is available from your distributor or local Schneider Electric Sales Office.

Note: The following refers to 140 NOE 211 and 140 NOE 251 only.

Ethernet and Quantum Hot Standby Systems Ethernet modules may be installed in a hot standby system, but they are not supported at switchover. When control shifts from the primary controller to the standby, the Ethernet network is not notified. The network continues to address the Ethernet module in the original primary rack, not the module in the new primary rack.

EMBP Gateway A Quantum Ethernet module can exist on the same Ethernet network as the EMBP Gateway, but it cannot communicate with the EMBP Gateway because of differences in formatting and network addressing.

Guidelines for Designing Your Network

A typical Ethernet installation carries many different types of traffic. Large data file transfers or World Wide Web graphics files can keep the network busy and cause network congestion and collisions. These collisions cause nodes to wait a variable amount of time before resending their messages. Because the size and frequency of non-control traffic is unpredictable, network performance may not be suitable for control applications. These problems can be greatly reduced by segregating the office and MIS traffic from control data.

Segregating Traffic The best method to protect Quantum Automation traffic from information systems traffic is to provide a completely separate physical network for automation control. Another method is to use readily available Ethernet devices such as bridges and routers to logically segment the network, isolating office traffic from control data.

Minimizing Delays Components such as repeaters, bridges, routers and hubs take a finite time to process each message. If messages pass through many of these devices, processing delays will accumulate. Delay times are available from device manufacturers. Check with your network administrator to quantify the effect on control messages and to determine whether it will be significant for your application.

Using Switches Ethernet switches can be used to ensure higher network performance. These devices allow each connection to have access to the full 10 Mbps bandwith instead of having to share the bandwith with all other nodes. They reduce the timing problems associated with Ethernet collisions and the resulting "back off" transmission delays. Check with your network administrator to see if your application would benefit from switching Ethernet devices.

Installing and Configuring the Module

2

At a Glance

Overview

This chapter describes how to install and configure the Ethernet TCP/IP module.

What's in this Chapter?

This chapter contains the following topics:

Before You Install the Module

At a Glance

Quantum Ethernet modules come fully configured. They are designed to go straight from the box to the backplane. But before you install your module, you must verify that the default configuration is appropriate for your network and that your Ethernet network is properly constructed.

⚠	**CAUTION**
	DUPLICATE ADDRESS HAZARD The default configuration includes the IP network address. Do not connect this module to your network until you have ensured that its IP address will be unique on the network. **Failure to observe this precaution can result in injury or equipment damage.**

Verifying the Default Configuration

You should change the default configuration before installing the module if any of the following is true.

- The module will be communicating on an open network
- The module's derived IP network address is already in use on your network
- The network uses IEEE 802.3 framing
- You need to specify the default Ethernet gateway and subnet mask

Consult your network administrator to see if any of these conditions apply. If they do, change the default configuration.

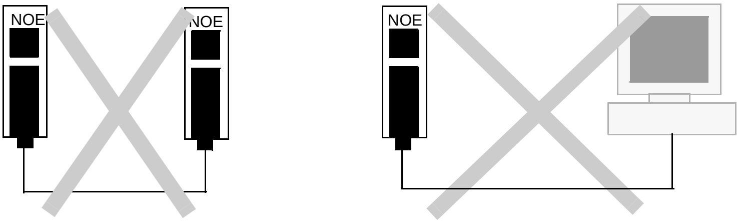
Note: If you will be changing the default configuration, you must stop the controller, then install the module, then change the configuration before starting the controller again.

The Ethernet module only reads its configuration data at power-up and when it is reset. Whenever the configuration data is changed, the module must be reset, either by hot swapping or through a reset command in the MSTR block. (See *MSTR: Reset Option Module, p. 60.*)Once the Ethernet module is installed, stopping and restarting the controller will not reset it.

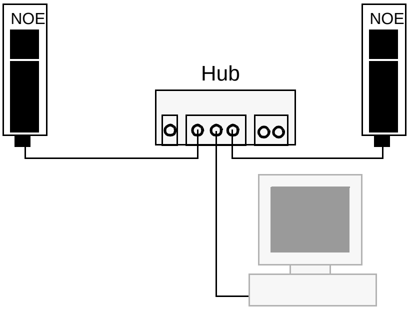
Verifying that the Network Has Been Constructed Properly

You should not connect an Ethernet module directly to another device with a length of cable. For the network to operate properly, you must route the cable for each device through an Ethernet hub. Hubs are widely available and can be purchased from many suppliers.

Examples of improper network topologies are shown below.



An example of the proper network topology is shown below.



Installing the Module

At a Glance

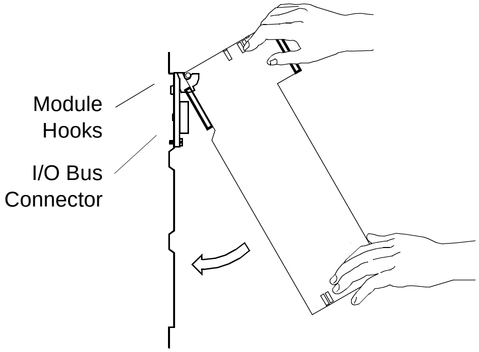
The Ethernet module comes fully ready to be installed. Installation consists of mounting the module on the backplane and connecting the cable.

Are You Ready to Install? Check!

Have you reviewed your configuration and network guidelines listed in *Before You Install the Module, p. 20*? You must meet those guidelines before installing the module. If you are planning to change the default configuration, stop the controller before installing the Ethernet module. Schneider Electric also recommends that you test to be sure your Ethernet cabling is working properly before connecting it to the Ethernet module.

Mounting the Module on the Backplane

Mount the module at an angle onto the two hooks located near the top of the backplane. Swing the module down to make an electrical connection with the backplane I/O bus connector.
The figure below shows how to mount the module on the backplane.



Tighten the screw at the bottom of the module to fasten it to the backplane. The maximum tightening torque for this screw is 2-4 in-lbs (.23 - .45 Nm).

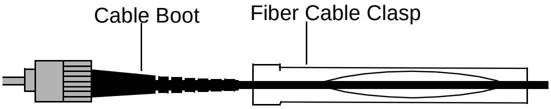
Connecting the Cable

Twisted Pair If you are using twisted pair cable, Schneider Electric recommends Category 5, which is rated to 100 Mbps. Use RJ-45 connectors. Slip the connector into the port. It should snap into place.

Fiber Optic Use 62.5/125 fiber optic cable with ST-style connectors. Schneider Electric sells a 3 m cable with connectors (990 XCA 656 09).

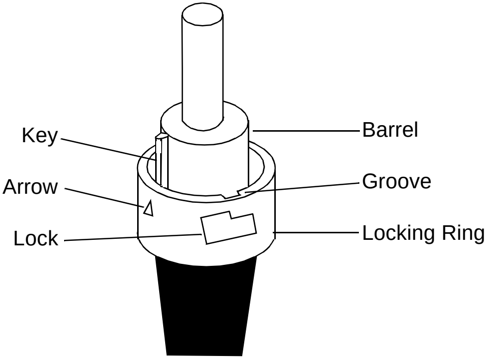
Remove the protective plastic coverings from the cable ports and the tips of the cable. Snap one of the fiber cable clasps onto the cable, carefully pressing the cable through the slot so that the wider end of the clasp is closest to the boot.

The figure below shows how to attach the fiber cable clasp to the cable.



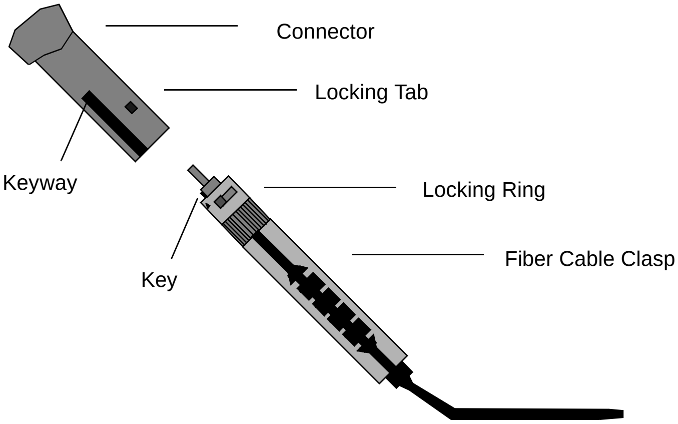
The key to installing the cable is to align the barrel, the locking ring and the connector.

The figure below shows how to align the key stem and locking ring.



Turn the locking ring to align an arrow with the key. Then align the key with the keyway. As a result, the locking tab, groove and lock should also be aligned.Slide the clasp up to the locking ring. Gripping the cable with the clasp, plug the cable into the lower (receive) cable connector. If it does not connect easily, realign the key with the arrow and try again.

The figure below shows how to attach the cable.



Turn the cable to the right, so that the tab locks securely. You may leave the fiber cable clasp on the cable for future use, but slide it off the boot of the cable to allow the module door to close.Repeat this process with the remaining strand of cable and the upper (transmit) cable connector.When connecting the cable to the hub, make sure that the strands are crossed. The transmit port of one device should be linked to the receive port of the other.

Changing the Default Configuration

At a Glance

If any of the following conditions apply, you should stop the controller, then install the module, then change the default configuration before starting the controller again:

- The module will be communicating on an open Ethernet network.
- The module’s IP address is already in use.
- The network uses IEEE 802.3 framing.
- You must specify a default Ethernet gateway and subnet mask.

Reset the Module Before Configuring It

If you change the configuration after installing the module, you must reset the module for your changes to take effect.

Software Tools for Configuring

You may configure the module using Modsoft or Concept.

Configuring the Module with Modsoft

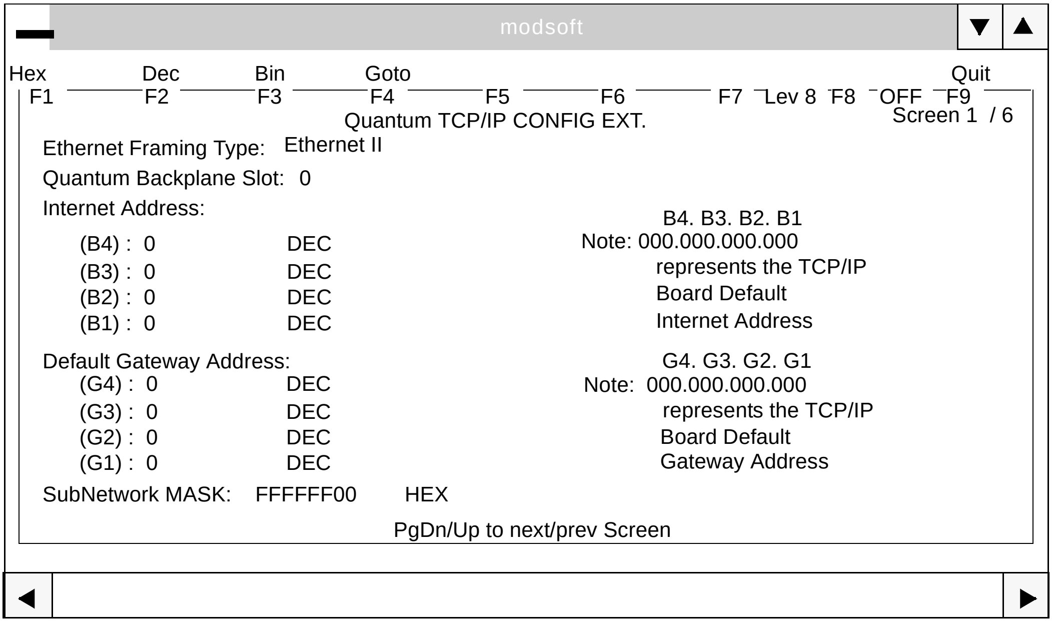
At a Glance

You can configure the module using Modsoft version 2.6 or a later version.

Starting in Modsoft

From the Modsoft Configuration Overview screen, select the Cfg Ext pulldown menu. Be sure that you have specified sufficient memory resources for the Ethernet configuration extension in the Cfg. Extension Size field. The first Ethernet module configured requires 20 words. Each additional module requires an additional 16 words.From the options, select TCP/IP Setup. You will reach the TCP/IP Configuration Extension screen.

The figure below shows a typical Configuration Extension screen.



```
                         Quantum TCP/IP CONFIG EXT.
Ethernet Framing Type:  Ethernet II
Quantum Backplane Slot:  0
Internet Address:
                                                     B4. B3. B2. B1
    (B4) :  0             DEC               Note: 000.000.000.000
    (B3) :  0             DEC                     represents the TCP/IP
    (B2) :  0             DEC                     Board Default
    (B1) :  0             DEC                     Internet Address

Default Gateway Address:                             G4. G3. G2. G1
    (G4) :  0             DEC               Note:  000.000.000.000
    (G3) :  0             DEC                      represents the TCP/IP
    (G2) :  0             DEC                      Board Default
    (G1) :  0             DEC                      Gateway Address
SubNetwork MASK:   FFFFFF00     HEX
                        PgDn/Up to next/prev Screen
```

Selecting the Ethernet Framing Type

You may choose between Ethernet II and IEEE 802.3, depending on your system. The default choice is Ethernet II.

If you are using the configuration extension to change the framing to IEEE 802.3, do not forget to designate the backplane slot number on the next line. Without the slot number, the system will not record the change in framing.

Assigning a Slot Number

To activate the configuration extension screen, you must enter the backplane slot number on the second line. This is the slot where you have mounted or intend to mount the Ethernet module. The slots are numbered from left to right, from one to x.

Note: If you do not enter the slot number, the system will ignore any other data you enter on this screen.

Assigning the IP Network Address

The Internet Protocol (IP) network address is a 32-bit address in the form xxx.xxx.xxx.xxx, where each group xxx is a decimal number ranging from 0 to 255.
If the module will be communicating on an open network or if the module's derived IP address is already being used, consult your network administrator to obtain a unique address. Type the new address in fields B4 through B1.
A space is provided for recording the IP network address on the label inside the front door panel.
If you input the address before installing the module or if you hot swap the module, it will automatically recognize the address you have already specified and will identify itself accordingly.

⚠ **CAUTION**

DUPLICATE ADDRESS HAZARD

Be sure to register the module's IP network address with your system administrator to avoid duplication.

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

Note: If you are using the configuration extension to change the IP network address, you also must input the backplane slot number. Without the slot number, the system will not recognize your changes.

Assigning the Default Gateway Address and Subnet Mask

Consult your network administrator to determine whether you need to specify a default gateway address and subnet mask. If this data is required, the network administrator should supply it. Input the gateway address in fields G4 through G1. Input the subnet mask at the bottom of the screen.

> **Note:** If you are using the configuration extension to assign a gateway address and subnet mask, remember to input a slot number as well. The slot number is required to activate the configuration extension.

Resetting the Module

If you change the default configuration after installing the module, you must reset the module for your changes to take effect. The module may be reset through a command in the MSTR block in Modsoft (see *MSTR: Reset Option Module, p. 60*), by cycling power or by lifting the module off the backplane and then setting it back in its slot.

Configuring More Than One Ethernet Module

You may configure from two to six Ethernet modules in a single controller, depending on the model. A 140 CPU 113 or 213 will accept a total of two network option modules, including NOE, NOM, NOP, CRP 811and other modules. A 140 CPU 424, 434 or 534 will accept six. The first Ethernet TCP/IP module configured requires 20 words of memory. Each additional module requires an additional 16 words of memory.

The modules may be placed in any slot in the local backplane. They do not have to be placed next to each other.

To configure the modules, simply page down to an unused configuration extension screen. Enter the backplane slot number to activate the screen.

Configuring The Module with Concept

Introduction

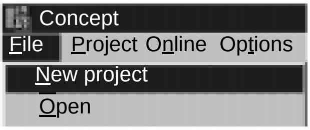
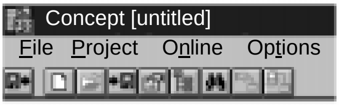
The following describes how to configure the NOE 211 00/NOE 251-00 module from your programming panel using Concept 2.2 or later. This is used to configure the module's IP parameters using Concept. The module can function as a network interface to the CPU without I/O services, as long as the IP parameters are provided by a BOOTP server, or with the module's default IP address.

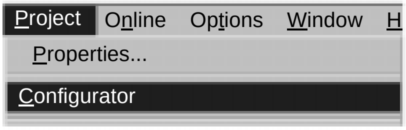
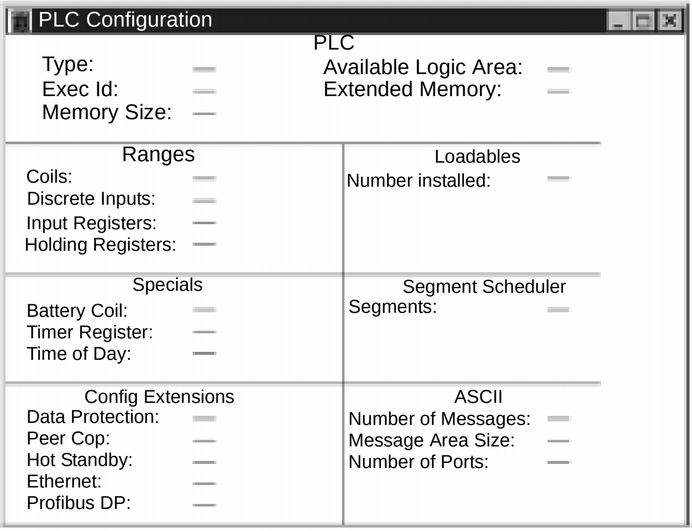
Selecting your PLC

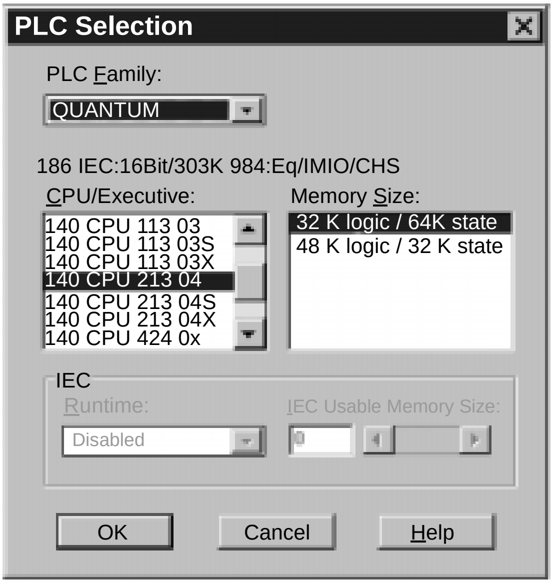
After the NOE module has been installed in a Quantum backplane (refer to *Front Panel Components, p. 9*), you can begin to configure it using Concept. To begin configuring first select your CPU (PLC).

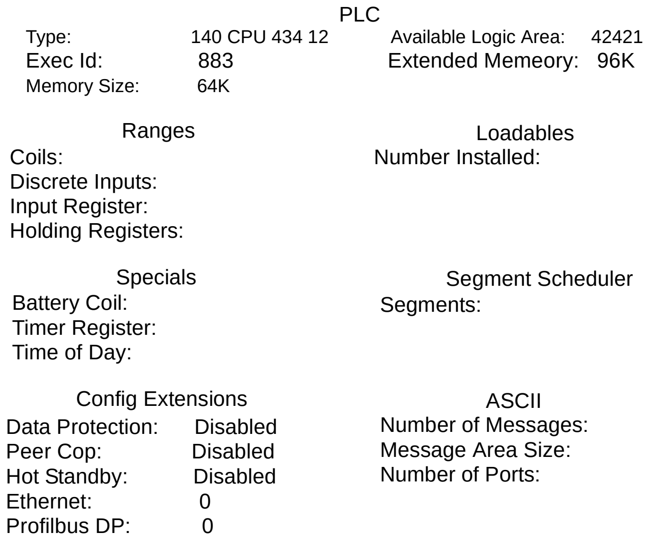
Procedure for Selecting a CPU

Perform the following steps to select a CPU..

Step	Action
1	Open Concept 2.2 or later on your programming panel (PC)
2	As shown in the following figure, from the File menu, select **New project**.  Concept File Project Online Options New project Open **Result**: As shown in the following figure, a new project is opened and the file name (untitled) appears over the menu bar. 

Step	Action
3	As shown in the following figure, from the Project menu, select **Configurator**.  Project Online Options Window H Properties... Configurator **Result**: As shown in the following figure, the PLC Configuration screen displays when a new project is selected.  PLC Configuration PLC Type: — Exec Id: — Memory Size: — Available Logic Area: — Extended Memory: — Ranges Coils: — Discrete Inputs: — Input Registers: — Holding Registers: — Loadables Number installed: — Specials Battery Coil: — Timer Register: — Time of Day: — Segment Scheduler Segments: — Config Extensions Data Protection: — Peer Cop: — Hot Standby: — Ethernet: — Profibus DP: — ASCII Number of Messages: — Message Area Size: — Number of Ports: —

Step	Action
4	Double click on the PLC selection in the left window pane. **Result**: As shown in the following figure, the PLC Selection dialog box displays. The default selection is Quantum.  PLC Selection PLC Family: QUANTUM 186 IEC:16Bit/303K 984:Eq/IMIO/CHS CPU/Executive: 140 CPU 113 03, 140 CPU 113 03S, 140 CPU 113 03X, 140 CPU 213 04, 140 CPU 213 04S, 140 CPU 213 04X, 140 CPU 424 0x Memory Size: 32 K logic / 64K state, 48 K logic / 32 K state IEC Runtime: Disabled IEC Usable Memory Size: 0
5	From the **CPU/Executive** scroll box, select the CPU that is installed in your Quantum backplane NOTE: Depending on the CPU selected, you may need to select the correct memory size applicable to it in the **Memory Size** dialog box.

Step	Action
6	Click the **<OK>** button. **Result:** As shown in the following figure, your PLC type and default configuration parameters are displayed in the PLC Configuration screen.  PLC Type: 140 CPU 434 12 Available Logic Area: 42421 Exec Id: 883 Extended Memeory: 96K Memory Size: 64K Ranges Coils: Discrete Inputs: Input Register: Holding Registers: Loadables Number Installed: Specials Battery Coil: Timer Register: Time of Day: Segment Scheduler Segments: Config Extensions Data Protection: Disabled Peer Cop: Disabled Hot Standby: Disabled Ethernet: 0 Profilbus DP: 0 ASCII Number of Messages: Message Area Size: Number of Ports:

Next

Next, you must configure the number of Ethernet modules (NOEs) that your system will contain, as shown in the procedure.

Setting the Number of NOEs

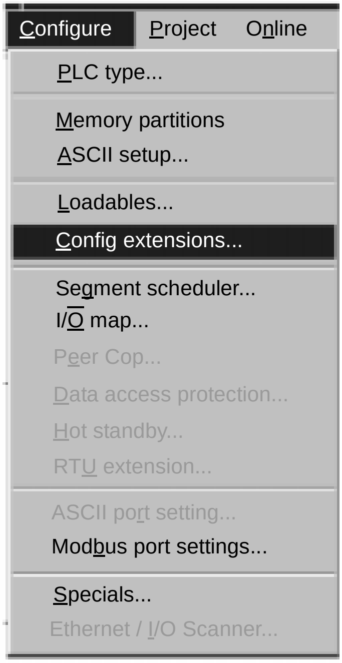
The following information describes how many NOEs you can configure in a single controller and how to configure that number.
You may configure from two to six Ethernet modules in a single controller, depending on the model. A 140 CPU 113 or 213 will accept a total of two network option modules, including NOE, NOM, NOP, and CRP 811. A 140 CPU 424, 434 or 534 will accept six.

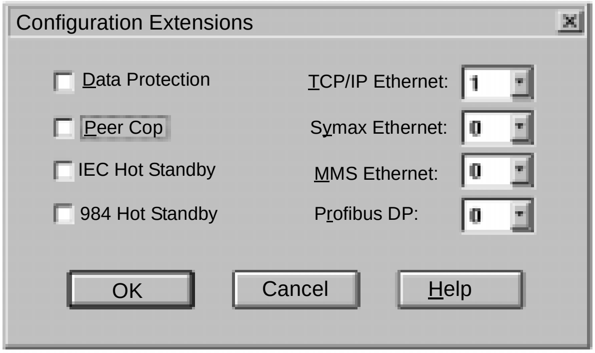
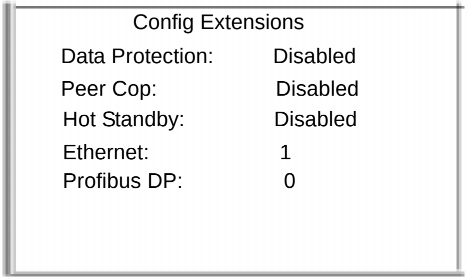
Memory Requirements

The first Ethernet TCP/IP module configured requires 20 words of memory. Each additional module requires an additional 16 words of memory.

Procedure for Setting the Number of NOEs

From the PLC Configuration screen, follow the steps below to select the number of NOE modules.

Step	Action
1	As shown in the following figure, from the Configure menu, select Config extensions, or double-click anywhere in the **Config Extensions** region of the screen.  Configure Project Online PLC type... Memory partitions ASCII setup... Loadables... Config extensions... Segment scheduler... I/O map... Peer Cop... Data access protection... Hot standby... RTU extension... ASCII port setting... Modbus port settings... Specials... Ethernet / I/O Scanner... **Result**: The Configuration Extension dialog box displays.

Step	Action
2	As shown in the following figure, in the TCP/IP Ethernet scroll box, select the number of NOE modules to be configured.  Configuration Extensions □ Data Protection TCP/IP Ethernet: 1 □ Peer Cop Symax Ethernet: 0 □ IEC Hot Standby MMS Ethernet: 0 □ 984 Hot Standby Profibus DP: 0
3	Click on the <**OK**> butt **Result**: As shown in the following figure, the Ethernet status changes from 0 to the number selected in Step 2.  Config Extensions Data Protection: Disabled Peer Cop: Disabled Hot Standby: Disabled Ethernet: 1 Profibus DP: 0

Next

Next, you need to create an I/O map for the NOEs in your configuration.

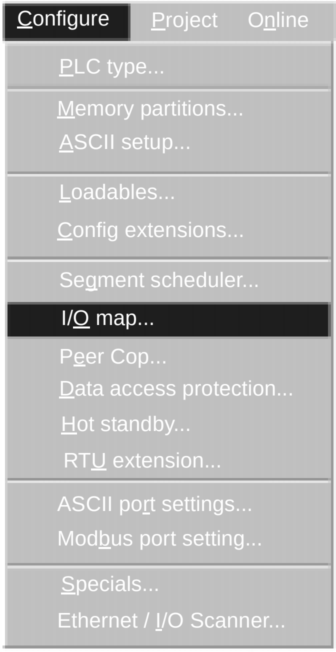
Accessing and Editing the I/O Map

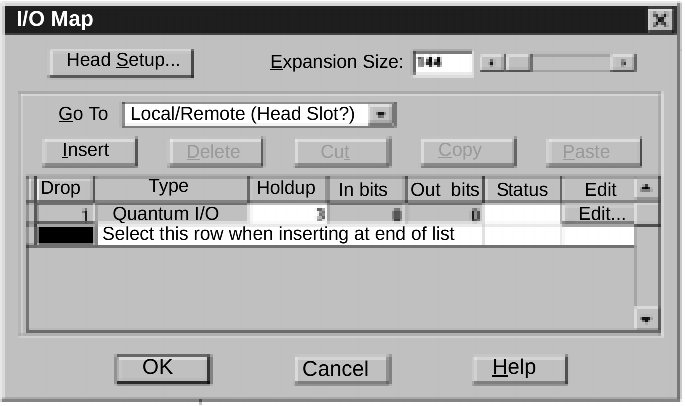
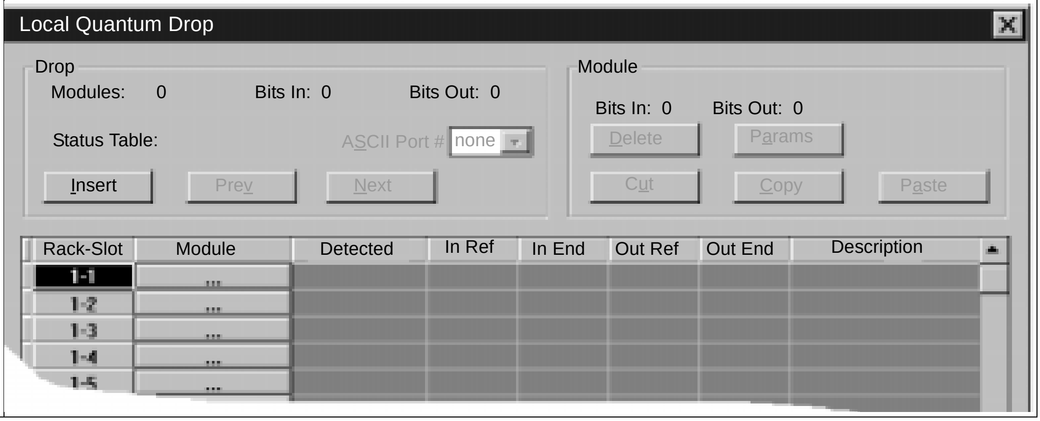
This procedure is required to determine the number of NOEs in the system and the slot numbers in which they reside.

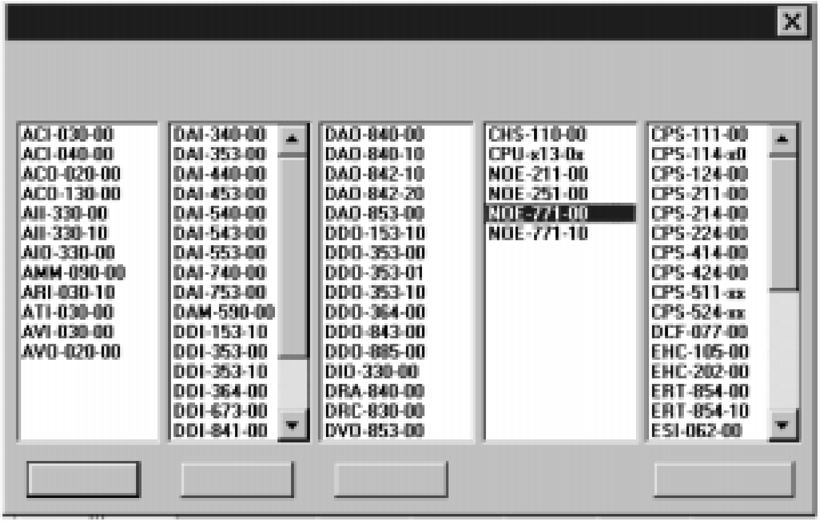
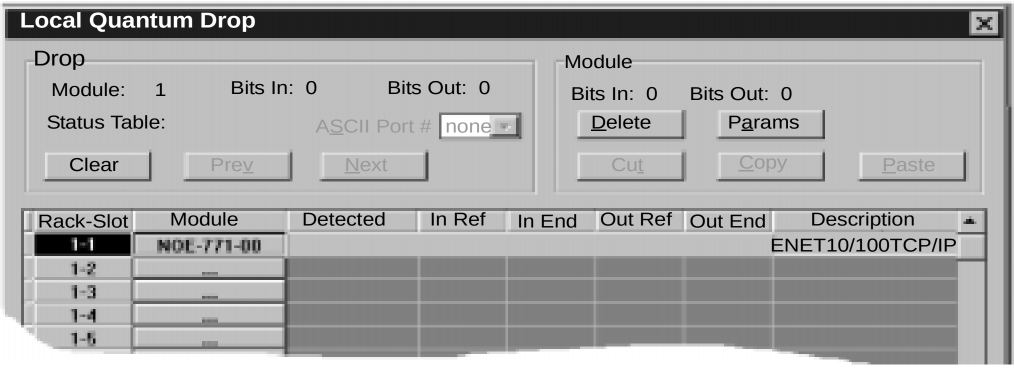
As part of the configuration process, you need to create an I/O Map for the local backplane including the NOE 2 X1 00 module. This step is required to determine the number of NOEs in the system and their slot assignments.

Steps

Perform the following steps to access and edit an I/O Map from the PLC. Configuration screen.

Step	Action
1	As shown in the following figure, from the Configure menu, select **I/O map**.  Configure Project Online PLC type... Memory partitions... ASCII setup... Loadables... Config extensions... Segment scheduler... I/O map... Peer Cop... Data access protection... Hot standby... RTU extension... ASCII port settings... Modbus port setting... Specials... Ethernet / I/O Scanner... **Result**: As shown in the following figure, the I/O Map dialog box displays.

Step	Action
2	Click the <**Edit**> button at the end of the Quantum I/O row.  I/O Map Head Setup... Insert Delete Cut Copy Paste OK Cancel Help Expansion Size: 144 Go To: Local/Remote (Head Slot?) Drop | Type | Holdup | In bits | Out bits | Status | Edit 1 | Quantum I/O | 3 | 0 | 0 | | Edit... Select this row when inserting at end of list **Result**: As shown in the following figure, the Local Quantum Drop I/O dialog box displays.  Local Quantum Drop Insert Prev Next Delete Params Cut Copy Paste Drop Modules: 0 Bits In: 0 Bits Out: 0 Status Table: ASCII Port # none Module Bits In: 0 Bits Out: 0 Rack-Slot | Module | Detected | In Ref | In End | Out Ref | Out End | Description 1-1 | ... 1-2 | ... 1-3 | ... 1-4 | ... 1-5 | ...

Step	Action
3	Click on the **...** button under **Module**. **Result**: As shown in the following figure, the I/O Module Selection drop down menu displays.  ACI-030-00 ACI-040-00 ACO-020-00 ACO-130-00 AII-330-00 AII-330-10 AIO-330-00 AMM-090-00 ARI-030-10 ATI-030-00 AVI-030-00 AVO-020-00 DAI-340-00 DAI-353-00 DAI-440-00 DAI-453-00 DAI-540-00 DAI-543-00 DAI-553-00 DAI-740-00 DAI-753-00 DAM-590-00 DDI-153-10 DDI-353-00 DDI-353-10 DDI-364-00 DDI-673-00 DDI-841-00 DAO-840-00 DAO-840-10 DAO-842-10 DAO-842-20 DAO-853-00 DDO-153-10 DDO-353-00 DDO-353-01 DDO-353-10 DDO-364-00 DDO-843-00 DDO-885-00 DIO-330-00 DRA-840-00 DRC-830-00 DVO-853-00 CHS-110-00 CPU-x13-0x NOE-211-00 NOE-251-00 NOE-771-00 NOE-771-10 CPS-111-00 CPS-114-x0 CPS-124-00 CPS-211-00 CPS-214-00 CPS-224-00 CPS-414-00 CPS-424-00 CPS-511-xx CPS-524-xx DCF-077-00 EHC-105-00 EHC-202-00 ERT-854-00 ERT-854-10 ESI-062-00
4	Select NET Adapter
5	Click on **NOE-2X1-00**, which appears in the Special column, and then click on the <**OK**> button. **Result**: The Local Quantum Drop I/O dialog box redisplays and the **NOE-2X1-00** is now listed under Module and described in the Description column.  Local Quantum Drop Drop Module: 1 Bits In: 0 Bits Out: 0 Status Table: ASCII Port # none Clear Prev Next Module Bits In: 0 Bits Out: 0 Delete Params Cut Copy Paste Rack-Slot | Module | Detected | In Ref | In End | Out Ref | Out End | Description 1-1 | NOE-771-00 | | | | | | ENET10/100TCP/IP 1-2 | ... 1-3 | ... 1-4 | ... 1-5 | ...
6	Repeat Steps 3 and 4 if other modules need to be added to the I/O map.
7	Click the <**OK**> buttons to return to the PLC Configuration screen.

Next

Next, you will configure the Ethernet address parameters from the Ethernet/ I/O Scanner screen as shown in.

Configuring the Ethernet Address Parameters

The NOE 771 x0 module’s Ethernet address parameters, consisting of Internet, Subnet mask, and Gateway addresses, are accessible from the Ethernet/ I/O Scanner dialog box. Prior to performing the following procedure, consult your system administrator to determine if you must configure new Ethernet address parameters, or whether the module will obtain them from the BOOTP server.

CAUTION

DUPLICATE ADDRESS HAZARD

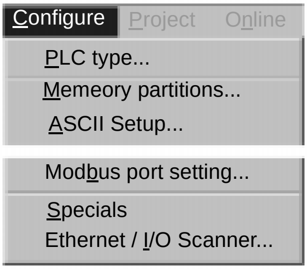
Always obtain your IP addresses from your system administrator to avoid the possibility of duplicate addresses. Having two devices with the same IP address can cause unpredictable operation of your network.

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

Steps

Perform the following steps to configure the Ethernet Address Parameters..

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>As shown in the following figure, from the Configure menu, select <b>Ethernet/ I/O Scanner</b>



<b>Result</b>: The Ethernet/ I/O Scanner dialog box displays.</td></tr>
<tr><td>2</td><td>As shown in the following figure, to configure new Ethernet address parameters, click on the <b>Specify IP Address</b> radio button.</td></tr>
<tr><td>3</td><td>Type in the new IP, Subnet Mask, and Gateway addresses in the applicable text boxes.</td></tr>
<tr><td>4</td><td>Select the correct Internet frame type from the <b>Frame Type</b> scroll box.</td></tr>
<tr><td>5</td><td>As shown in the following figure, if the module’s BOOTP server will assign Ethernet address parameters, click on the Use Bootp Server radio button Note that if you select this option, the address parameter text boxes will be grayed out and will not display the addresses.</td></tr>
</table>

How the Module Derives Its IP Address

During initialization, the NOE 771 module attempts to read the address parameter information from the PLC and determines its IP Address in the following fashion

- If the PLC has the IP Address and the BOOTP server is not selected, the module will use the configured IP address that you assigned in Step 2 of the above procedure.
- If the BOOTP server was selected in Step 5 of the above procedure, the module will send BOOTP requests to receive its IP Address.
- If no Configuration Extension exists, the NOE sends out BOOTP requests. If the module does not receive its IP Address from the BOOTP server after 2 minutes, it will then use the IP Address derived from its MAC address.

Note: The MAC address is assigned at the factory and is recorded on a label on the front panel, above the cable connector. This is a unique 48-bit global assigned address. It is set in PROM. The Ethernet address is recorded on the label in hexadecimal, in the form 00.00.54.xx.xx.xx.

The MSTR Instruction

3

At a Glance

Overview

This chapter describes the MSTR ladder logic instruction that can be used to read or write controller information in PLCs that support networking communication capabilities over Modbus Plus and Ethernet.

What's in this Chapter?

This chapter contains the following topics:

MSTR: Introduction

At a Glance

All NOE 2X1 00 Quantum Ethernet TCP/IP modules provide the user with the capability of transferring data to and from nodes on a Modbus Plus or TCP/IP network through the use of a special MSTR (master instruction). All PLCs that support networking communication capabilities over Modbus Plus and Ethernet can use the MSTR ladder logic instruction to read or write controller information.

Overview of MSTR Operations

The MSTR instruction allows you to initiate one of 12 possible network communications operations over the network. Each operation is designated by a code.

The following table lists the 12 operations and indicates those that are supported on an Ethernet TCP/IP network.

MSTR Operation	Code	TCP/IP Ethernet Support
Write data	1	supported
Read data	2	supported
Get local statistics	3	supported
Clear local statistics	4	supported
Write global database	5	not supported
Read global database	6	not supported
Get remote statistics	7	supported
Clear remote statistics	8	supported
Peer Cop health	9	not supported
Reset Option Module	10	supported
Read CTE(config extension)	11	supported
Write CTE (config extension)	12	supported

Up to four MSTR instructions can be simultaneously active in a ladder logic program. More than four MSTRs may be programmed to be enabled by the logic flow as one active MSTR block releases the resources it has been using and becomes deactivated, the next MSTR operation encountered in logic can be activated.

Characteristics

Size The MSTR block is three nodes high.

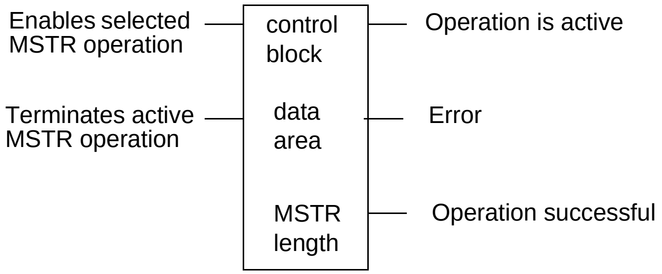
PLC Compatibility The MSTR block is available for the following types of controllers.

- The MSTR is standard in PLCs that have built-in Modbus Plus capabilities (Modbus Plus functionality only).
- The MSTR is standard in all Quantum PLCs with Modbus Plus functionality and/ or TCP/IP Ethernet option modules.
- The MSTR is available as a loadable in chassis mount PLCs (Modbus Plus functionality only).

Op Code BF hex

Representation

The MSTR block has two inputs and three outputs.
The block's layout is shown below.



Inputs The MSTR instruction has two control inputs.

- The input to the top node enables the instruction when it is ON.
- The input to the middle node terminates the active operation when it is ON.

Outputs The MSTR instruction can produce three possible outputs.

- The output from the top node echoes the state of the top input - it goes ON while the instruction is active.
- The output from the middle node echoes the state of the middle input - it goes ON if the MSTR operation is terminated prior to completion or if an error occurs in completing the operation.
- The output from the bottom node goes ON when an MSTR operation has been completed successfully.

If all outputs are zero, this indicates a condition in which four MSTR instructions are already in progress.

Top Node Content The 4x register entered in the top node is the first of several (network dependent) holding registers that comprise the network control block. The control block structure differs according to the network in use.

For the TCP/IP Ethernet network the control block structure is as follows:

Register	**Content**
Displayed	Identifies one of ten MSTR operations legal for TCP/IP (1 ... 4 and 7 ... 12).
First implied	Displays error status.
Second implied	Displays length (number of registers transferred).
Third implied	Displays MSTR operation-dependent information.
Fourth implied	High byte: Destination index. Low byte: Quantum backplane slot address of the NOE module.
Fifth implied	Byte 4 of the 32-bit destination IP Address.
Sixth implied	Byte 3 of the 32-bit destination IP Address.
Seventh implied	Byte 2 of the 32-bit destination IP Address.
Eighth implied	Byte 1 of the 32-bit destination IP Address.

Middle Node Content The 4x register entered in the middle node is the first in a group of contiguous holding registers that comprise the data area. For operations that provide the communication processor with data such as a Write operation, the data area is the source of the data. For operations that acquire data from the communication processor, such as a Read operation, the data area is the destination for the data. In the case of the Ethernet Read and Write CTE operations, the middle node stores the contents of the Ethernet configuration extension table in a series of registers.

Bottom Node Content The integer value entered in the bottom node specifies the length - the maximum number of registers in the data area. The length must be in the range 1 ... 100.

MSTR: Error Codes

At a Glance

If an error occurs during an MSTR operation, a hexadecimal error code will be displayed in the first implied register in the control block (the top node). Function error codes are network-specific.

MSTR Function Error Codes

TCP/IP Ethernet Error Codes An error in an MSTR routine over TCP/IP Ethernet may produce one of the following errors in the MSTR control block:

Hex Error Code	Meaning
1001	User has aborted the MSTR element.
2001	An unsupported operation type has been specified in the control block.
2002	One or more control block parameters has been changed while the MSTR element is active (applies only to operations that take multiple scans to complete). Control block parameters may be changed only when the MSTR element is not active.
2003	Invalid value in the length field of the control block.
2004	Invalid value in the offset field of the control block.
2005	Invalid values in the length and offset fields of the control block.
2006	Invalid slave device data area.
3000	Generic Modbus fail code.
30ss*	Modbus slave exception response.
4001	Inconsistent Modbus slave response.
F001	Option Module not responding.

** The ss subfield in error code 30ss is shown in the following table:

ss Hex Value	Meaning
01	Slave device does not support the requested operation.
02	Nonexistent slave device registers requested.
03	Invalid data value requested.
05	Slave has accepted long-duration program command.
06	Function can’t be performed now; a long-duration command is in effect.
07	Slave rejected long-duration program command.

An error on the TCP/IP Ethernet network itself may produce one of the following errors in the MSTR control block:

Hex Error Code	Meaning
5004	Interrupted system call.
5005	I/O error.
5006	No such address.
5009	The socket descriptor is invalid.
500C	Not enough memory.
500D	Permission denied.
5011	Entry exists.

Hex Error Code	Meaning
5016	An argument is invalid.
5017	An internal table has run out of space.
5020	The connection is broken.
5023	This operation would block and the socket is nonblocking.
5024	The socket is nonblocking and the connection cannot be completed.
5025	The socket is nonblocking and a previous connection attempt has not yet completed.
5026	Socket operation on a nonsocket.
5027	The destination address is invalid.
5028	Protocol not supported.
5029	Protocol wrong type for socket.
502A	Protocol not available.
502B	Protocol not supported.
502C	Socket type not supported.
502D	Operation not supported on socket.
502E	Protocol family not supported.
502F	Address family not supported.
5030	Address is already in use.
5031	Address is not available.
5032	Network is down.
5033	Network is unreachable.
5034	Network dropped connection on reset.
5035	The connection has been aborted by the peer.
5036	The connection has been reset by the peer.
5037	An internal buffer is required, but cannot be allocated.
5038	The socket is already connected.
5039	The socket is not connected.
503A	Can’t send after socket shutdown.
503B	Too many references; can’t splice.
503C	Connection timed out.
503D	The attempt to connect was refused.
5040	Host is down.
5041	The destination host could not be reached from this node.
5042	Directory not empty.
5046	NI_INIT returned.

Hex Error Code	Meaning
5047	The MTU is invalid.
5048	The hardware length is invalid.
5049	The route specified cannot be found.
504A	Collision in select call: these conditions have already been selected by another task.
504B	The task id is invalid.

CTE Error Codes The following error codes are returned if there is a problem with the Ethernet configuration extension table (CTE) in your program configuration.

Hex Error Code	Meaning
7001	There is no Ethernet configuration extension.
7002	The CTE is not available for access.
7003	The offset is invalid.
7004	The offset + length is invalid.
7005	Bad data field in the CTE.

MSTR: Read and Write Operations

At a Glance

An MSTR Write operation (type 1 in the displayed register of the top node) transfers data from a master source device to a specified slave destination device on the network. An MSTR Read operation (type 2 in the displayed register of the top node) transfers data from a specified slave source device to a master destination device on the network. Read and Write use one data master transaction path and may be completed over multiple scans.

Note: TCP/IP Ethernet routing must be accomplished via standard third-party Ethernet IP router products.

Read and Write Operations Control Block

The registers in the MSTR control block (the top node) contain the Read or Write information as described in the following table.

Register	Function	Content	
Displayed	Operation Type	1 = Write, 2 = Read.	
First Implied	Error status	Displays a hex value indicating an MSTR error.	
		Exception response, where response size is incorrect.	Exception code + 3000
		Exception response where response size is incorrect.	4001
		Read Write	
Second implied	Length	Write = number of registers to be sent to slave. Read = number of registers to be read from slave.	
Third implied	Slave device data area	Specifies starting 4x register in the slave to be read from or written to (1 = 4001, 49 =40049).	
Fourth implied	Low byte	Quantum backplane slot address of the NOE module.	
	High Byte	Destination Index.	
Fifth ... eighth implied	Destination	Each register contains one byte of the 32-bit IP address.	

MSTR: Get Local Statistics

At a Glance

The Get Local Statistics operation (type 3 in the display register of the top node) obtains information related to the local node where the MSTR has been programmed.

Control Block

The registers in the MSTR control block (the top node) contain the Get Local Statistics information as described in the following table.

Register	**Function**	**Content**
Displayed	Operation Type	3
First implied	Error status	Displays a hex value indicating an MSTR error, when relevant.
Second implied	Length	Starting from offset, the number of words of statistics from the local processor's statistics table; the length must be > 0 < data area.
Third implied	Offset	An offset value relative to the first available word in the local processor's statistics table. If the offset is specified as 1, the function obtains statistics starting with the second word in the table.
Fourth implied	Low byte	Quantum backplane slot address of the NOE module.
	High Byte	Destination Index.
Fifth .. Eighth implied	Not applicable	

TCP/IP Ethernet Statistics

A TCP/IP Ethernet board responds to "Get Local Statistics" and "Get Remote Statistics" commands with the following information:

Word	Meaning
00 ... 02	MAC address
03	Board Status
04 and 05	Number of receiver interrupts
06 and 07	Number of transmitter interrupts
08 and 09	Transmit_timeout error count
10 and 11	Collision_detect error count
12 and 13	Missed packets
14 and 15	Memory error
16 and 17	Number of times driver has restarted
18 and 19	Receive framing error
20 and 21	Receiver overflow error
22 and 23	Receive CRC error
24 and 25	Receive buffer error
26 and 27	Transmit silo underflow
28 and 29	Late collision
30 and 31	Lost carrier
32 and 33	Number of retries
34 and 35	IP address

MSTR: Clear Local Statistics

At a Glance

The Clear Local Statistics operation (type 4 in the display register of the top node) obtains information related to the local node where the MSTR has been programmed.

Control Block

The registers in the MSTR control block (the top node) contain the Clear Local Statistics information as described in the following table.

Register	Function	Content
Displayed	Operation Type	4
First implied	Error status	Displays a hex value indicating an MSTR error, when relevant.
Second implied	Not applicable	
Third implied	Not applicable	
Fourth implied	Low byte	Quantum backplane slot address of the NOE module.
	High byte	Destination Index.
Fifth .. Eighth implied	Not applicable	

MSTR: Get Remote Statistics

At a Glance

The Get Remote Statistics operation (type 7 in the displayed register of the top node) obtains information relative to remote nodes on the network. This operation may require multiple scans to complete and does not require a master data transaction path.

The remote comm processor always returns its complete statistics table when a request is made, even if the request is for less than the full table. The MSTR instruction then copies only the amount of words you have requested to the designated 4x registers.

Note: TCP/IP Ethernet routing must be accomplished via standard third-party Ethernet IP router products.

Control Block

The registers in the MSTR control block (the top node) contain the Get Remote Statistics information as described in the following table.

Register	Function	Content
Displayed	Operation Type	7
First implied	Error status	Displays a hex value indicating an MSTR error, when relevant.
Second implied	Length	Starting from offset, the number of words of statistics from the local processor's statistics table; the length must be > 0 < data area.
Third implied	Offset	An offset value relative to the first available word in the local processor's statistics table. If the offset is specified as 1, the function obtains statistics starting with the second word in the table.
Fourth implied	High byte	Destination index.
	Low Byte	Quantum backplane slot address of the NOE module.
Fifth .. Eighth implied	Destination	Each register contains one byte of the 32-bit IP address.

TCP/IP Ethernet Statistics

A TCP/IP Ethernet board responds to "Get Local Statistics" and "Get Remote Statistics" commands with the following information:

Word	Meaning
00 ... 02	MAC address
03	Board Status
04 and 05	Number of receiver interrupts
06 and 07	Number of transmitter interrupts
08 and 09	Transmit_timeout error count
10 and 11	Collision_detect error count
12 and 13	Missed packets
14 and 15	Memory error
16 and 17	Number of times driver has restarted
18 and 19	Receive framing error
20 and 21	Receiver overflow error
22 and 23	Receive CRC error
24 and 25	Receive buffer error
26 and 27	Trasnmit Buffer error count.
28 and 29	Transmit silo underflow
30 and 31	Late collision
32 and 33	Lost carrier
34 and 35	Number of retries
36 and 37	IP address

MSTR: Clear Remote Statistics

At a Glance

The Clear Remote Statistics operation (type 8 in the displayed register of the top node) clears statistics relative to a remote network node from the data area in the local node. This operation may require multiple scans to complete and uses a single data master transaction path.

Control Block

The registers in the MSTR control block (the top node) contain the Clear Remote Statistics information as described in the following table.

Register	Function	Content
Displayed	Operation Type	8
First implied	Error status	Displays a hex value indicating an MSTR error, when relevant.
Second implied	Not applicable	
Third implied	Not applicable	
Fourth implied	High byte	Destination index.
Fifth .. Eighth implied	Destination	Each register contains one byte of the 32-bit IP address.

MSTR: Peer Cop Health

At a Glance

The peer cop health operation (type 9 in the displayed register of the top node) reads selected data from the peer cop communications health table and loads that data to specified 4x registers in state RAM. The peer cop communications health table is 12 words long, and the words are indexed via this MSTR operation as words 0 ... 11.

Control Block

The registers in the MSTR control block (the top node) contain the information for a Peer Cop Health operation as described in the following table.

Register	Function	Content
Displayed	Operation Type	9
First implied	Error status	Displays a hex value indicating an MSTR error, when relevant.
Second implied	Data size	Number of words requested from peer cop table (range 1 ... 12).
Third implied	Index	First word from the table to be read (range 0 ... 11, where 0 = the first word in the peer cop table and 11 = the last word in the table).
Fourth implied	High byte	Destination index.
Fifth .. Eighth implied	Destination	Each register contains one byte of the 32-bit IP address.

Peer Cop Health Information

The peer cop communications health table (shown below) comprises 12 contiguous registers that can be indexed in an MSTR operation as words 0 ... 11. Each bit in each of the table words is used to represent an aspect of communications health relative to a specific node on the TCP/IP network:

- The bits in words 0 ... 3 represent the health of the global input communication expected from nodes 1 ... 64. Since global input is not supported these bits are set to zero.
- The bits in words 4 ... 7 represent the health of the output from a specific node.
- The bits in words 8 ... 11 represent the health of the input to a specific node.

The following table shows the relationship of the health bits to network nodes.

Type of Status	**Word Index**	**Bit-To-Network Node Relationship**															
Global Input	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Specific Output	4	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
	5	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17
	6	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33
	7	64	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49
Specific Input	8	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
	9	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17
	10	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33
	11	64	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49

The state of a peer cop health bit reflects the current communication status of its associated node:

- A health bit is set when data is successfully exchanged with its corresponding node.
- A health bit is cleared when no communication has occurred with the corresponding node within the configured peer cop health time-out period.
- All health bits are cleared at PLC start time. The health bit for a given node is always zero when its associated peer cop entry is null.
- All global health bits are always reported as zero.

MSTR: Reset Option Module

At a Glance

The Reset Option Module operation (type 10 in the displayed register of the top node) causes a Quantum NOE option module to enter a reset cycle to reset its operational environment.

Control Block

The registers in the MSTR control block (the top node) contain the Reset Option Module information as described in the following table:

Register	Function	Content
Displayed	Operation Type	10
First Implied	Error status	Displays a hex value indicating an MSTR error, when relevant.
Second Implied	Not applicable	
Third Implied	Not applicable	
Fourth Implied	Low byte	Quantum backplane slot address of the NOE module.
Fifth ... Eighth Implied	Not applicable	

MSTR: Read CTE (Configuration Extension Table)

At a Glance

The Read CTE operation (type 11 in the displayed register of the top node) reads a given number of bytes from the Ethernet configuration extension table to the indicated buffer in PLC memory. The bytes to be read begin at a byte offset from the beginning of the CTE. The content of the Ethernet CTE table is displayed in the middle node of the MSTR block.

Control Block

The registers in the MSTR control block (the top node) contain the Read CTE information as described in the following table:

Register	Function	Content
Displayed	Operation Type	11
First Implied	Error status	Displays a hex value indicating an MSTR error, when relevant.
Second Implied	Not applicable	
Third Implied	Not applicable	
Fourth Implied	Low byte	Quantum backplane slot address of the NOE module.
Fifth ... Eighth Implied	Not applicable	

CTE Display Information

The values in the Ethernet configuration extension table (CTE) are displayed in a series of registers in the middle node of the MSTR instruction when a Read CTE operation is implemented. The middle node contains the first of 11 contiguous 4x registers. The registers display the following CTE data:

Parameter	Register	Content
Frame type	Displayed	1 = 802.3
		2 = Ethernet
IP Address	First implied	First byte of the IP address
	Second implied	Second byte of the IP address
	Third implied	Third byte of the IP address
	Fourth implied	Fourth byte of the IP address
Subnetwork mask	Fifth implied	Hi word
	Sixth implied	Lo word
Gateway	Seventh implied	First byte of the gateway
	Eighth implied	Second byte of the gateway
	Ninth implied	Third byte of the gateway
	Tenth implied	Fourth byte of the gateway

MSTR: Write CTE (Configuration Extension Table)

At a Glance

The Write CTE operation (type 12 in the displayed register of the top node) writes an indicated number of bytes from PLC memory, starting at a specified byte address, to an indicated Ethernet configuration extension table at a specified offset. The starting register for the data to be written into the Ethernet CTE table is identifed in the middle node of the MSTR block.

Control Block

The registers in the MSTR control block (the top node) contain the Write CTE information as described in the following table:

Register	Function	Content
Displayed	Operation Type	12
First Implied	Error status	Displays a hex value indicating an MSTR error, when relevant.
Second Implied	Not applicable	
Third Implied	Not applicable	
Fourth Implied	Low byte	Quantum backplane slot address of the NOE module.
Fifth ... Eighth Implied	Not applicable	

CTE Write Information

The values to be written into the Ethernet configuration extension table (CTE) are contained in a series of registers identified in the middle node of the MSTR instruction when a Write CTE operation is implemented. The middle node identifies the first of 11 contiguous 4x registers. The registers contain the following CTE data:

Parameter	**Register**	**Content**
Frame type	Displayed	1 = 802.3
		2 = Ethernet
IP Address	First implied	First byte of the IP address
	Second implied	Second byte of the IP address
	Third implied	Third byte of the IP address
	Fourth implied	Fourth byte of the IP address
Subnetwork mask	Fifth implied	Hi word
	Sixth implied	Lo word
Gateway	Seventh implied	First byte of the gateway
	Eighth implied	Second byte of the gateway
	Ninth implied	Third byte of the gateway
	Tenth implied	Fourth byte of the gateway

Retrieving Data via the World Wide Web

4

At a Glance

Overview

This chapter describes how the Ethernet TCP/IP module accesses data via the World Wide Web.

What's in this Chapter?

This chapter contains the following topics:

Retrieving Data via the World Wide Web

At a Glance

Each Ethernet TCP/IP module contains a World Wide Web server. Pages on the embedded web site display:

- the Ethernet statistics for the node
- the controller's configuration
- the controller's 4X register values
- the controller's personality
- the status, configuration and register values of remote I/O
- the status, configuration and register values of distributed I/O

The web pages can best be viewed across the World Wide Web using version 3.0 or greater of either Netscape Navigator or Internet Explorer.

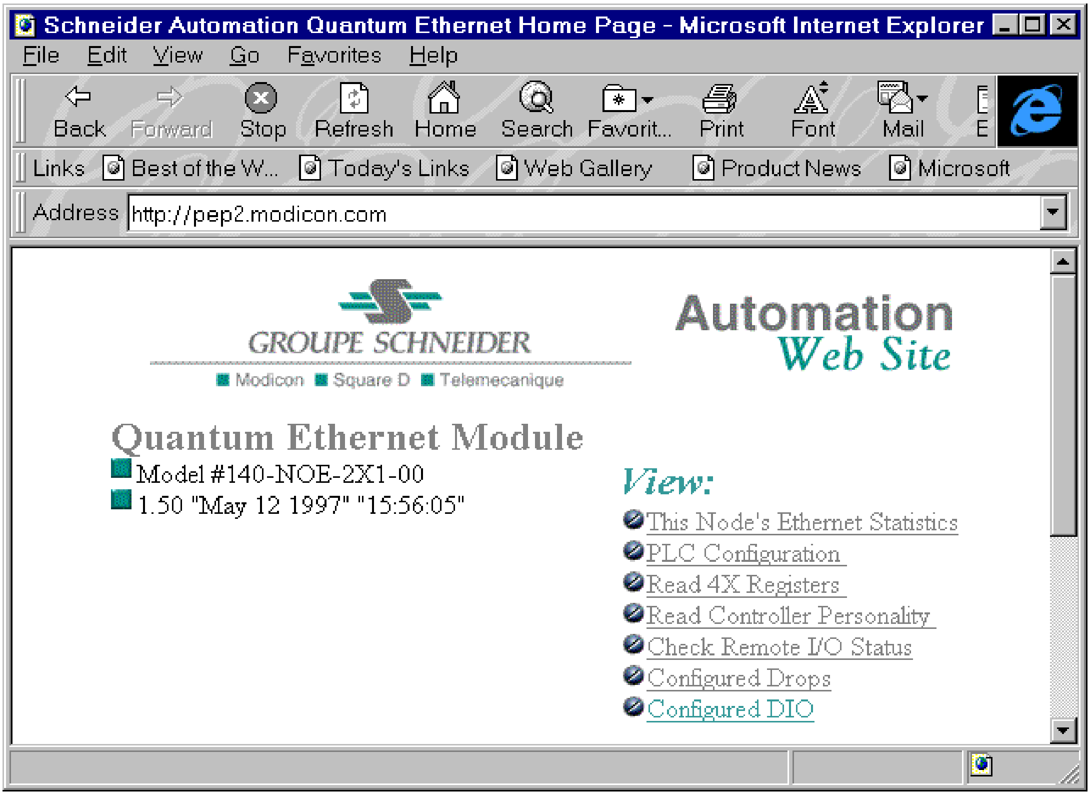
Accessing the Ethernet Home Page

At a Glance

Before you can access the module’s home page, you must learn its full IP address or URL from your system administrator. Then you can access the home page on your Web browser.

How to Access the Home Page

To access the module’s home page, type the address or URL in the Address or Location box in the browser window.
The home page then displays the module’s model number and software version as in the example below.


Schneider Automation Quantum Ethernet Home Page - Microsoft Internet Explorer
File Edit View Go Favorites Help
Back Forward Stop Refresh Home Search Favorit... Print Font Mail E
Links Best of the W... Today's Links Web Gallery Product News Microsoft
GROUPE SCHNEIDER
Modicon Square D Telemecanique
Automation Web Site

Address http://pep2.modicon.com

Quantum Ethernet Module

- Model #140-NOE-2X1-00
- 1.50 "May 12 1997" "15:56:05"

View:

- This Node's Ethernet Statistics
- PLC Configuration
- Read 4X Registers
- Read Controller Personality
- Check Remote I/O Status
- Configured Drops
- Configured DIO

Home Page Options

At a Glance

The Ethernet module’s home page contains hyperlinks to seven pages of data:

- This Node’s Ethernet Statistics
- PLC Configuration
- Read 4X Registers
- Read Controller Personality
- Check Remote I/O Status
- Configured Drops
- Configured DIO

Selecting the Options

Click on any hyperlink on the home page to view the module’s data.

The data on these pages is static. If you click the Enable automatic refresh link at the bottom of certain pages, the data on that page will be updated every 10 seconds. To change the refresh rate, edit the last two digits in the URL from refresh=10 to refresh=x, where x is the new refresh rate in seconds. Press Enter.

To return to a static display, click the new Disable automatic refresh link.

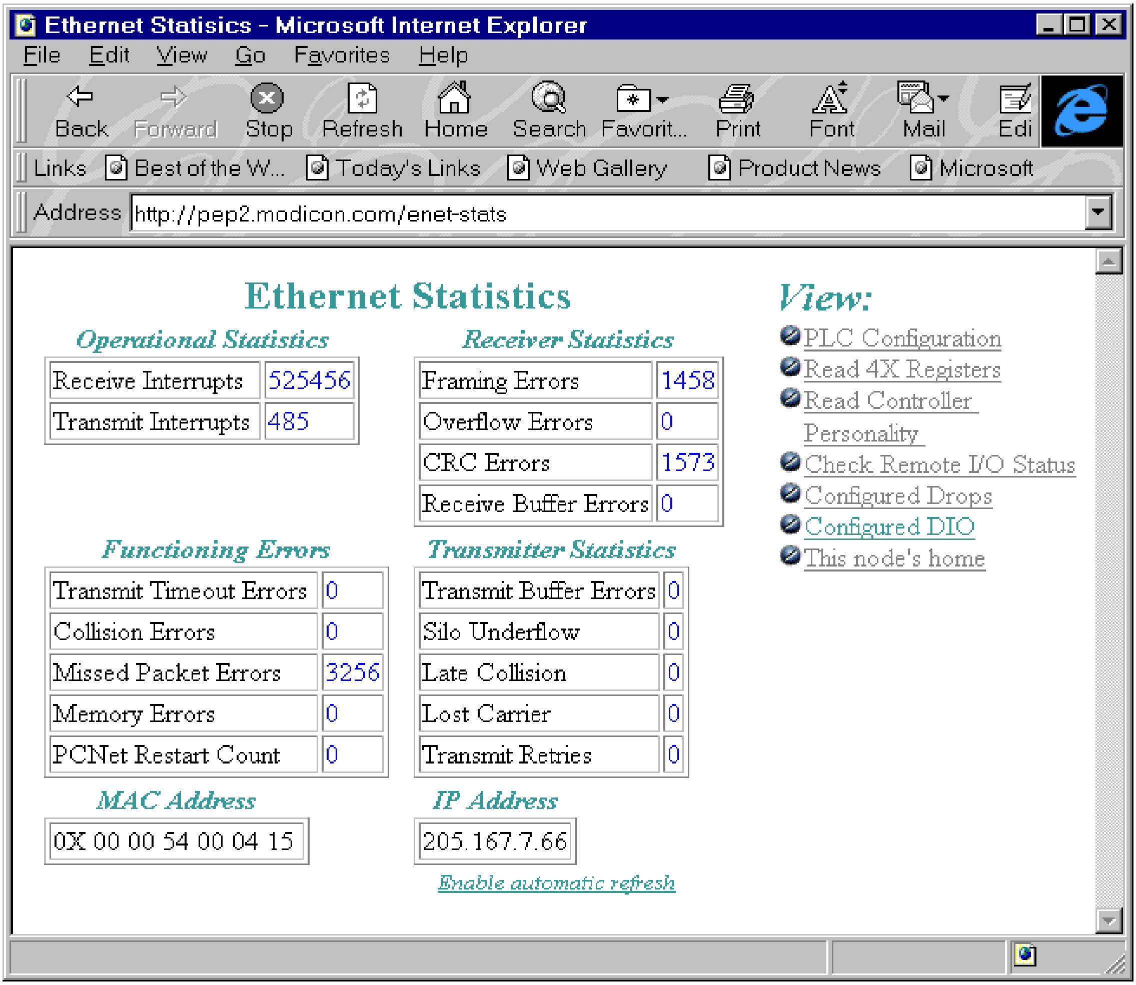
Ethernet Statistics

At a Glance

The Ethernet Statistics page displays tables containing the operational statistics, functioning errors, receiver statistics, transmitter statistics, MAC address and IP address.

Viewing the Statistics

A typical statistics display is shown below.


Ethernet Statisics - Microsoft Internet Explorer
File Edit View Go Favorites Help
Back Forward Stop Refresh Home Search Favorit... Print Font Mail Edi
Links Best of the W... Today's Links Web Gallery Product News Microsoft
Address

http://pep2.modicon.com/enet-stats

Ethernet Statistics

Operational Statistics

Receive Interrupts	525456
Transmit Interrupts	485

Receiver Statistics

Framing Errors	1458
Overflow Errors	0
CRC Errors	1573
Receive Buffer Errors	0

Functioning Errors

Transmit Timeout Errors	0
Collision Errors	0
Missed Packet Errors	3256
Memory Errors	0
PCNet Restart Count	0

Transmitter Statistics

Transmit Buffer Errors	0
Silo Underflow	0
Late Collision	0
Lost Carrier	0
Transmit Retries	0

MAC Address

0X 00 00 54 00 04 15

IP Address

205.167.7.66

Enable automatic refresh

View:

- PLC Configuration
- Read 4X Registers
- Read Controller Personality
- Check Remote I/O Status
- Configured Drops
- Configured DIO
- This node's home

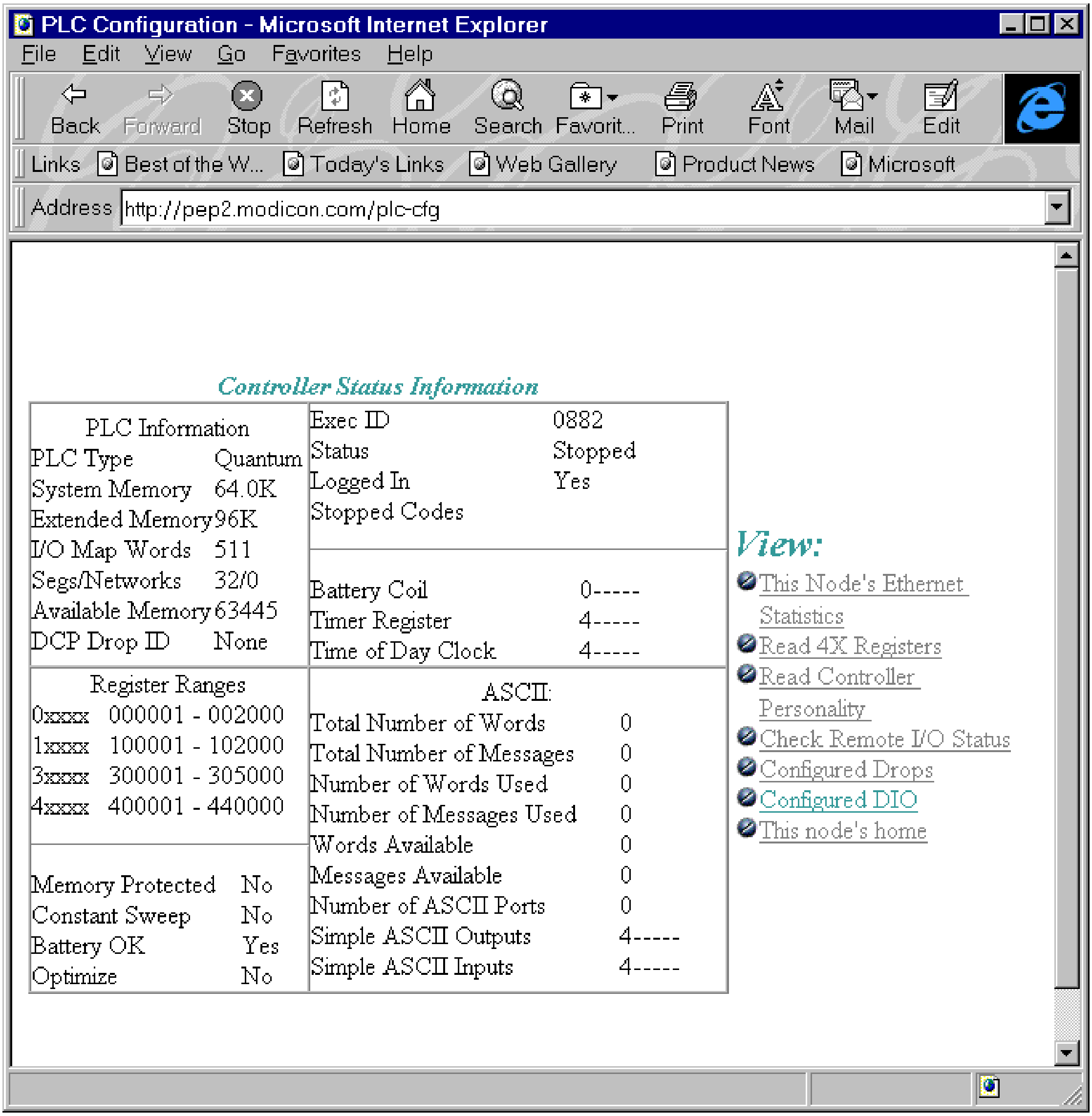
PLC Configuration

At a Glance

The configuration page displays detailed information about the controller status, including the address range of the 0x, 1x, 3x and 4x registers.

Viewing the PLC Configuration

A typical PLC configuration display is shown below.


PLC Configuration - Microsoft Internet Explorer
File Edit View Go Favorites Help
Back Forward Stop Refresh Home Search Favorit... Print Font Mail Edit
Links Best of the W... Today's Links Web Gallery Product News Microsoft

Address http://pep2.modicon.com/plc-cfg

Controller Status Information

PLC Information	
PLC Type	Quantum
System Memory	64.0K
Extended Memory	96K
I/O Map Words	511
Segs/Networks	32/0
Available Memory	63445
DCP Drop ID	None

Exec ID	0882
Status	Stopped
Logged In	Yes
Stopped Codes	

Battery Coil	0-----
Timer Register	4-----
Time of Day Clock	4-----

Register Ranges	
0xxxx	000001 - 002000
1xxxx	100001 - 102000
3xxxx	300001 - 305000
4xxxx	400001 - 440000

ASCII:	
Total Number of Words	0
Total Number of Messages	0
Number of Words Used	0
Number of Messages Used	0
Words Available	0
Messages Available	0
Number of ASCII Ports	0
Simple ASCII Outputs	4-----
Simple ASCII Inputs	4-----

Memory Protected	No
Constant Sweep	No
Battery OK	Yes
Optimize	No

View:

- This Node's Ethernet Statistics
- Read 4X Registers
- Read Controller Personality
- Check Remote I/O Status
- Configured Drops
- Configured DIO
- This node's home

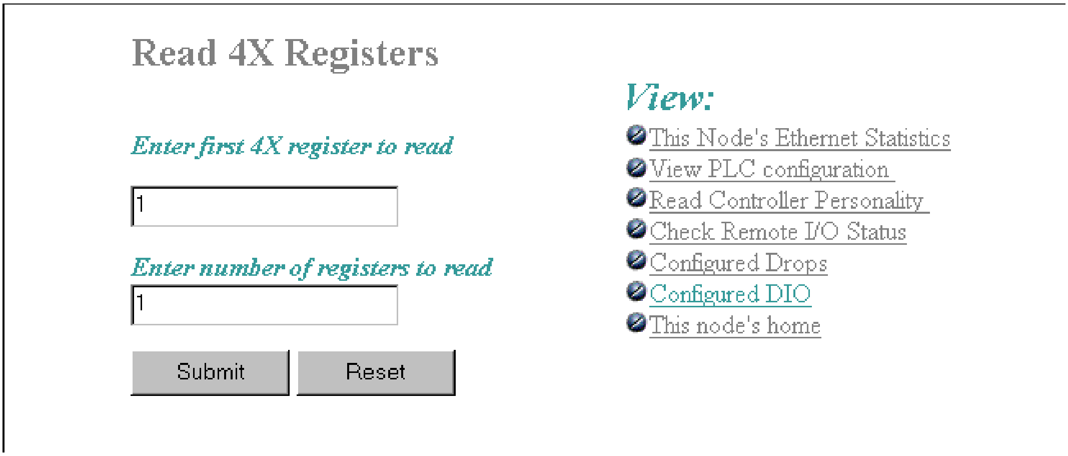
Read 4X Registers

At a Glance

This page allows you to specify a range of 4X registers to be read.

Viewing the 4X Registers

A typical 4X registers display is shown below.



Read 4X Registers

Enter first 4X register to read: 1

Enter number of registers to read: 1

For information about automatic refresh, see *Home Page Options, p. 68*.

Enter the number of the first 4x register number in the desired range in the first box, omitting the leading 4 and any zeros. For example, if the first register you would like to display is 40001, type 1 in the box. If the first register in your range is 40115, type 115 in the box.

Enter the number of registers to be read in the second box. Then click the submit button. The browser should display a 4X Register Values page, giving the decimal value for each register in the specified range.

The browser may display one of the following error messages.

Error Message	Cause
Read error from controller = 0x3	Number of registers to read is greater than 125
	Number of registers to read is set at zero
Read error from controller = 0x2	Number of first 4x register to read is greater than the number of 4x registers configured.
	Number of registers to read is greater than the number of 4x registers configured
	Number of registers to read is set at zero

Example

The figure below demonstrates values for a starting register of 400001 and a length of 40 registers.

To view another range, return to the Read 4X Registers page by clicking on the Read 4X Registers link or pressing the Back button on your browser. Click the reset button to reset the form. Type a starting register and length for the new range. Press the submit button.

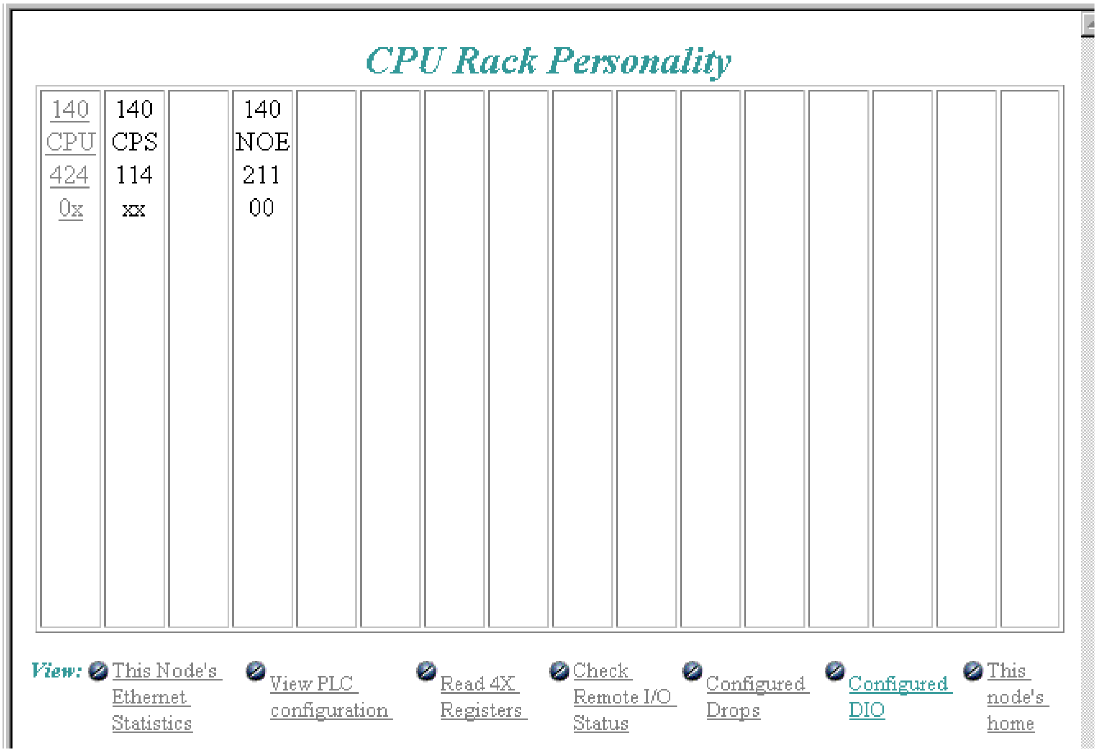
Read Controller Personality

At a Glance

This page displays a table showing the modules installed in the backplane, which may not be the same as those configured. The controller name is a hyperlink to the PLC Configuration page. This page may only be viewed when the controller is stopped.

Viewing the Controller Personality

A typical controller personality display is shown below.



CPU Rack Personality

140 CPU 424 0x	140 CPS 114 xx		140 NOE 211 00												

View: This Node's Ethernet Statistics | View PLC configuration | Read 4X Registers | Check Remote I/O Status | Configured Drops | Configured DIO | This node's home

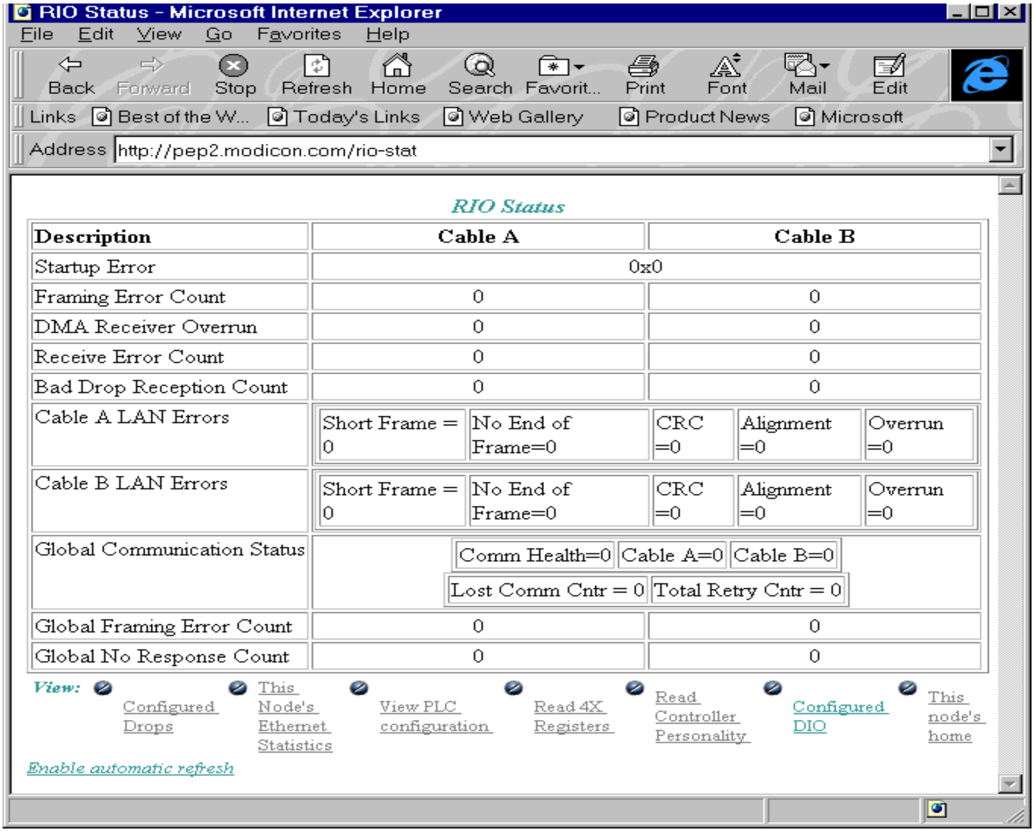
Check Remote I/O Status

At a Glance

The RIO Status page displays a table with error statistics for Cable A and Cable B. The data is static. For information about automatic refresh, see *Home Page Options, p. 68*.

Viewing the Remote I/O Status

A typical Remote I/O Status display is shown below.


RIO Status - Microsoft Internet Explorer
File Edit View Go Favorites Help
Back Forward Stop Refresh Home Search Favorit... Print Font Mail Edit
Links Best of the W... Today's Links Web Gallery Product News Microsoft
View: Configured Drops This Node's Ethernet Statistics View PLC configuration Read 4X Registers Read Controller Personality Configured DIO This node's home
Enable automatic refresh

Address http://pep2.modicon.com/rio-stat

RIO Status

Description	Cable A	Cable B
Startup Error	0x0	
Framing Error Count	0	0
DMA Receiver Overrun	0	0
Receive Error Count	0	0
Bad Drop Reception Count	0	0

Cable A LAN Errors	Short Frame = 0
	No End of Frame=0
	CRC =0
	Alignment =0
	Overrun =0	

Cable B LAN Errors	Short Frame = 0
	No End of Frame=0
	CRC =0
	Alignment =0
	Overrun =0	

Global Communication Status	Comm Health=0 Cable A=0 Cable B=0 Lost Comm Cntr = 0 Total Retry Cntr = 0	
Global Framing Error Count	0	0
Global No Response Count	0	0

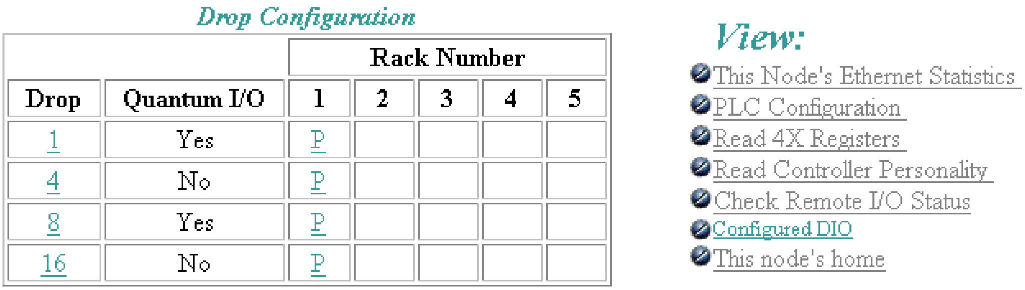
Configured Drops

At a Glance

The Drop Configuration page displays the configured drops, indicates whether they are Quantum I/O or 800 series I/O and whether the racks are present (P) in the I/O map. Any SyMax I/O drop will be displayed as Quantum I/O.

Viewing the Configured Drops

A typical Configured Drops display is shown below.



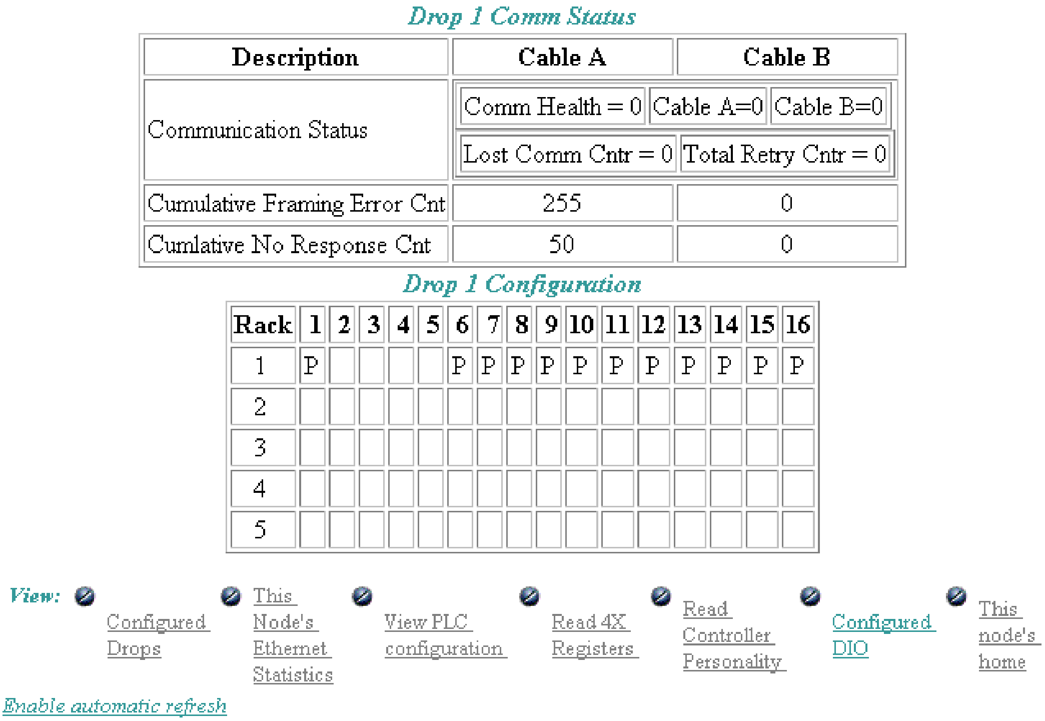
Drop Configuration

		Rack Number				
Drop	Quantum I/O	1	2	3	4	5
1	Yes	P				
4	No	P				
8	Yes	P				
16	No	P				

View:

- This Node's Ethernet Statistics
- PLC Configuration
- Read 4X Registers
- Read Controller Personality
- Check Remote I/O Status
- Configured DIO
- This node's home

The drop number is a hyperlink to a Drop Communication Status and Configuration page. A typical Drop Communication Status and Configuration display is shown below.



Drop 1 Comm Status

Description	Cable A	Cable B
Communication Status	Comm Health = 0 Cable A=0 Cable B=0 Lost Comm Cntr = 0 Total Retry Cntr = 0	
Cumulative Framing Error Cnt	255	0
Cumlative No Response Cnt	50	0

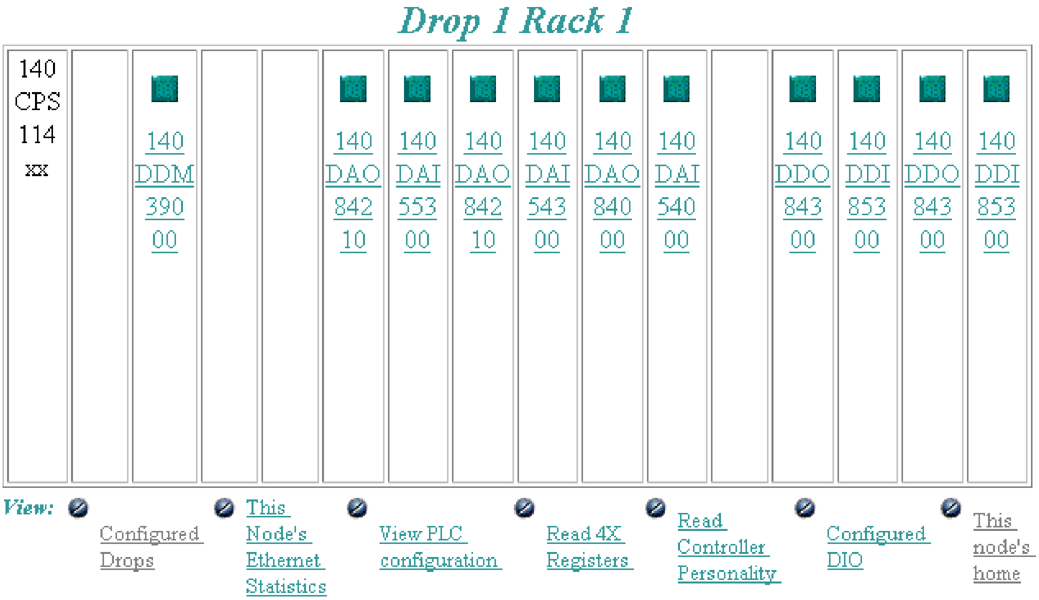
Drop 1 Configuration

Rack	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	P					P	P	P	P	P	P	P	P	P	P	P
2																
3																
4																
5																

View: Configured Drops · This Node's Ethernet Statistics · View PLC configuration · Read 4X Registers · Read Controller Personality · Configured DIO · This node's home

Enable automatic refresh

If a rack is present in the Configuration chart on the Drop Communication Status and Configuration Page, the P is a hyperlink to a Drop Rack Configuration page. A typical Drop Rack Configuration display is shown below. For information about automatic refresh, see *Home Page Options, p. 68*.



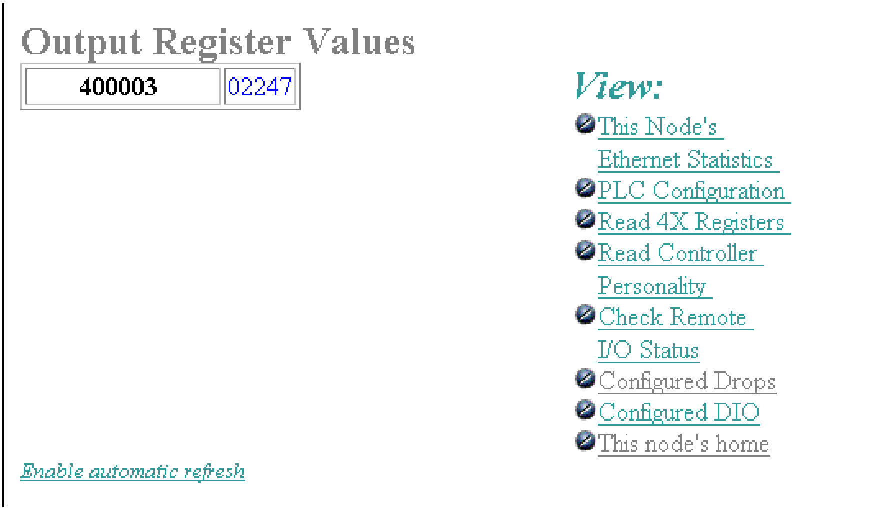
Drop 1 Rack 1

140 CPS 114 xx		🟩 140 DDM 390 00			🟩 140 DAO 842 10	🟩 140 DAI 553 00	🟩 140 DAO 842 10	🟩 140 DAI 543 00	🟩 140 DAO 840 00	🟩 140 DAI 540 00		🟩 140 DDO 843 00	🟩 140 DDI 853 00	🟩 140 DDO 843 00	🟩 140 DDI 853 00

View: Configured Drops | This Node's Ethernet Statistics | View PLC configuration | Read 4X Registers | Read Controller Personality | Configured DIO | This node's home

This page displays all the modules in the rack. The box above the module number is a health bit indicator: if green, the module is healthy; if red, the module either is not present or is not responding properly.

The module number may be hyperlinked to a Register Value page, displaying the current decimal value for input and/or output registers. A typical Register Value display is shown below.



Output Register Values

400003	02247

Enable automatic refresh

Configured DIO

At a Glance

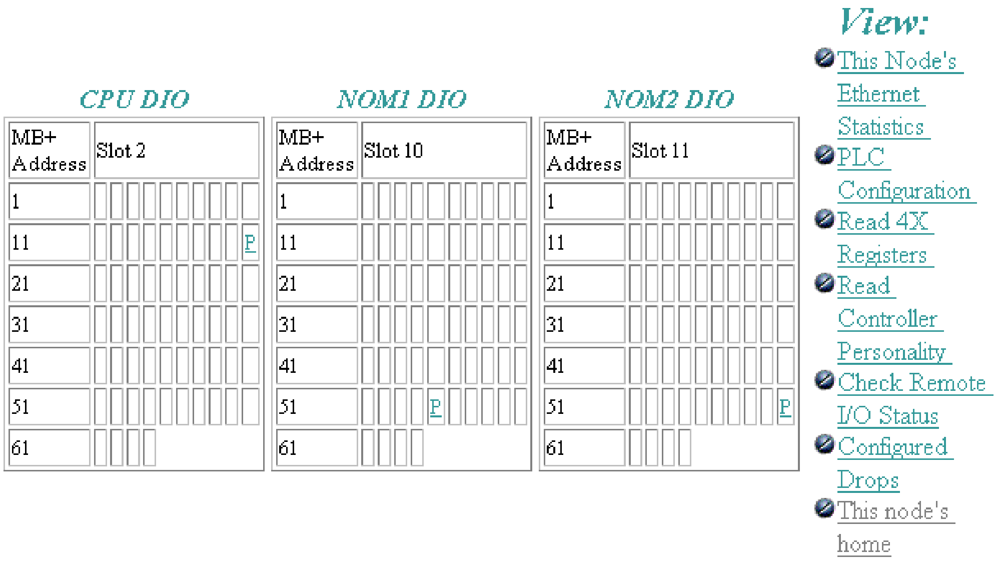
The Distributed I/O Configuration page indicates whether the controller or any Modbus Plus Network Option modules (NOM) are configured for DIO. The display shows the slot number of the controller or NOM module in the main backplane. It also shows the Modbus Plus address of any distributed I/O.

Viewing the Configured DIO

The chart is arranged in rows of 10 Modbus Plus addresses. The number in the MB+ Address column indicates the first address in that row.

A P in any row indicates that Modbus Plus DIO is configured at that address.

A typical Configured DIO display is shown below. In this example the DIO associated with the controller has a Modbus Plus address of 20. The DIO associated with NOM1 has an address of 55. The DIO associated with NOM2 has an address of 60.



CPU DIO

MB+ Address	Slot 2									
1										
11										P
21										
31										
41										
51										
61										

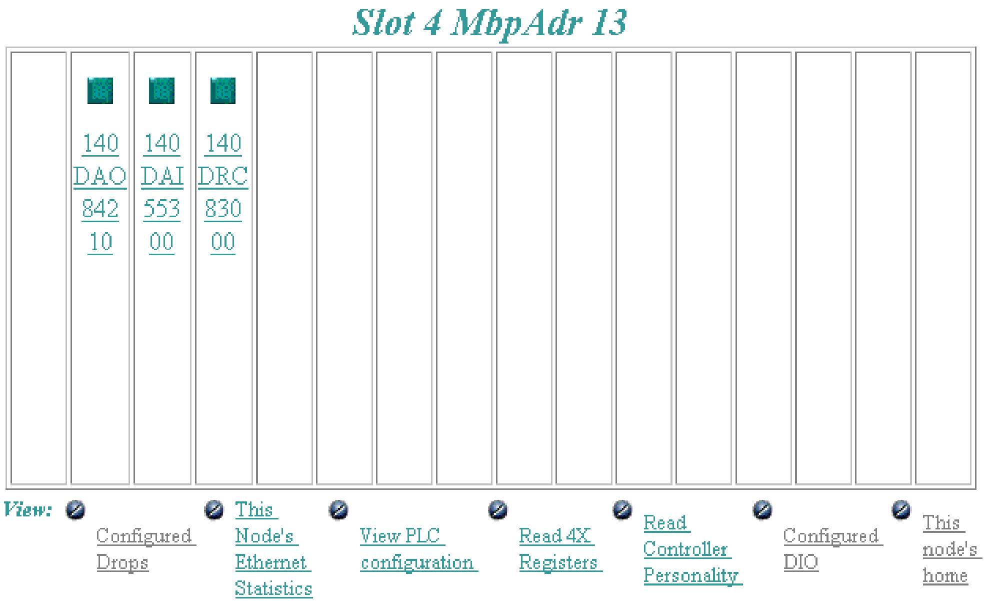
NOM1 DIO

MB+ Address	Slot 10									
1										
11										
21										
31										
41										
51					P					
61										

NOM2 DIO

MB+ Address	Slot 11									
1										
11										
21										
31										
41										
51										P
61										

If DIO is present, the P is a hyperlink to a Modules Configuration Page. A typical Modules Configuration display is shown below.



The configuration page shows the modules present in the backplane. The box above the module number is a health bit indicator: if green, the module is healthy; if red, the module either is not present or is not responding properly.
The module number may be hyperlinked to a Register Value page, displaying the current decimal value for input and/or output registers.
A typical Register Values display is shown below.

Using the Network Options Ethernet Tester

5

At a Glance

Overview

The Network Options Ethernet Tester enables you to establish communication with an Ethernet module, get and clear statistics in the module, and read and write registers at the module. The following pages describes this function.

What's in this Chapter?

This chapter contains the following topics:

Ethernet Tester: Introduction and Installation

At a Glance

The Ethernet Tester enables you to establish communication with an Ethernet module, get and clear statistics in the module, and read and write registers at the module.

Introduction

An Ethernet module may act as a client or as a server.

If it will be acting as a client -- that is, initiating transactions on the network for its Quantum controller -- then you must program an MSTR block in ladder logic.

The Ethernet module may also act as a server, responding to requests and commands from devices on the network for its Quantum controller.

The Network Options Ethernet Tester utility allows you to get and clear statistics and to read and write registers over the network, using a Windows-based PC.

You may also create your own program using the Ethernet module as a server.

Note: In its capacity as server, the Ethernet module can only accept 20 connections at any one time. If a new connection is attempted and the server has already reached its limit, it will terminate the least used connection in order to make room for the new one.

Installing the Ethernet Tester

To install the Ethernet Tester, insert the utility diskette in your disk drive.

Run a:\setup.exe.

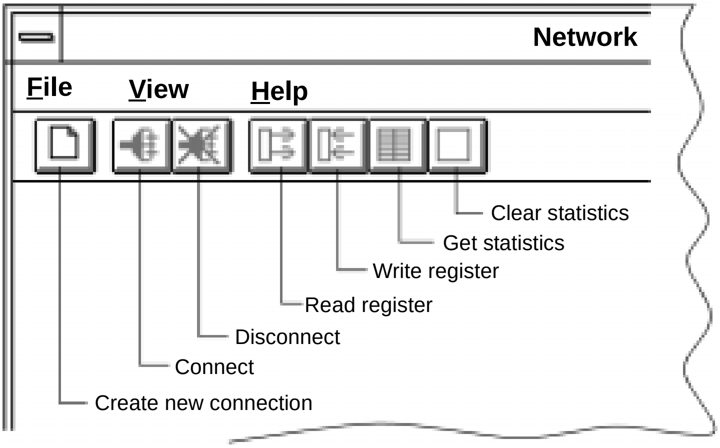
Ethernet Tester: Establishing a Connection with an Ethernet Module

At a Glance

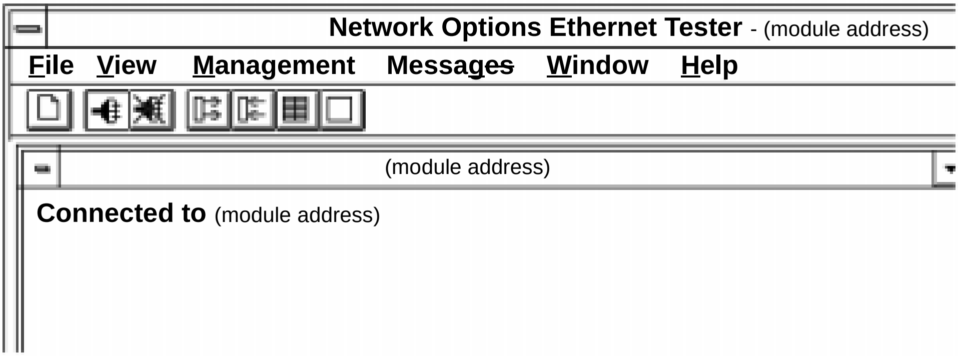
The information below describes how to use the Ethernet Tester's menus to establish a connection with an Ethernet module, and how to disconnect from the module.

Establishing the Connection

To establish a connection with an Ethernet module using the Network Options Ethernet Tester, you must know the module's IP network address or host name. Refer to the tester's opening screen below. From the inital menu, select File and choose New from the options in the pulldown menu or click on the new connection button in the toolbar.



Type the module's IP network address or host name in the box provided. Click the OK button. This dedicates a connection from your PC to the designated Ethernet module and brings you to the Ethernet Tester's main menu.
Refer to the menu shown below.



Connected to (module address)

To activate the connection, select Management and choose Connect from the pulldown menu or click on the connect button in the toolbar.

When you are ready to disconnect, select Management and choose Disconnect from the pulldown menu or click on the disconnect button in the toolbar.

You may establish several connections with the same module or with other modules by selecting New from the File pulldown menu or by clicking on the create new connection button in the toolbar. Each connection has its own window within the main window.. The Window pulldown menu gives you options for arranging connection windows and allows you to select one. The options available on the pulldown menus and toolbar in the main window apply to the selected connection.

After disconnecting from one module, you may reassign its dedicated connection by selecting Management and choosing Set IP Address from the pulldown menu. Type the new IP network address or host name in the box provided.

Ethernet Tester: Getting and Clearing Statistics

At a Glance

The information below describes how to use the Ethernet Tester to get and clear statistics in the Ethernet module.

You can also get and clear statistics using the MSTR block.

Getting and Clearing Statistics

To get statistics from the Ethernet module do the following.

- Select Messages and choose Get Statistics from the Ethernet Tester’s pulldown menu or click on the get statistics button in the toolbar.

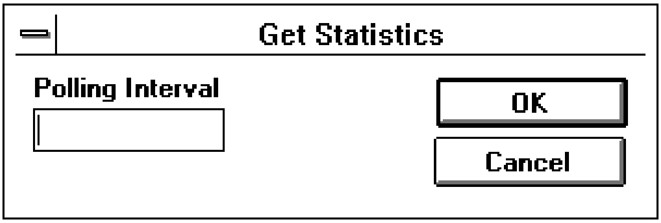
To clear statistics do the following.

- select Messages and choose Clear Statistics from the pulldown menu or click on the clear statistics button in the toolbar.

You can specify a polling interval for continuous gathering of statistics. The polling interval is the number of seconds between transactions. To set a polling interval do the following.

- Type a polling interval in the box provided and click OK. Complete statistics for the module will be printed in the window for this connection.

The Polling Interval dialog is shown below.



To change the polling interval without interrupting communication with the Ethernet module do the following.,

- Select Messages and choose Poll Interval and type the new polling interval in the box.

Available Statistics

The Network Options Ethernet Tester will provide the following statistics. An illustration of the statistics display is provided at the end of this list:

- Total Transaction Count. How many transactions have been completed.
- IP Address.
- MAC Address.
- Status. The hex value displayed may be 0001, 8001 or C001.
  - 0001 indicates that the module is running, the Link indicator is not lit and no entry exists in the crash log.
  - 8001 indicates that module is running and the Link indicator is lit. No entry exists in the crash log.
  - C001 indicates that the module is running, the Link indicator is lit and an entry exists in the crash log.
- Receive Interrupts and Transmit Interrupts. The number of times the PCNET controller chip has generated interrupts.
- Transmit timeout errors. The number of times the transmitter has been on the channel longer than the interval required to send the maximum length frame of 1519 bytes. This is also known as a babble error.
- Collision errors. The number of collisions detected by the Ethernet chip.
- Missed packet errors. The number of times a received frame was dropped because a receive descriptor was not available.
- Memory errors. The number of times an Ethernet controller chip experienced an error accessing shared RAM. A memory error will cause a restart.
- PcNet restart count. The number of times the Ethernet controller chip was restarted due to fatal runtime errors, including memory errors, transmit buffer errors and transmit underflow.
- Framing error. The number of times an incoming frame contained a non-integer multiple of eight bits.
- Overflow errors. The number of times the receiver has lost part or all of an incoming frame, due to an inability to store the frame in memory before the internal FIFO overflowed.
- CRC errors. The number of times a CRC (FCS) error was detected on an incoming frame.
- Receive buffer errors. The number of times a receive buffer was not available while data chaining a received frame.
- Transmit buffer errors. The number of times the end packet flag on the current buffer was not set and the Ethernet controller did not own the next buffer. A transmit buffer error causes a restart.
- Silo Underflow. The number of times a packet was truncated due to data late from memory. A Silo Underflow will cause a restart.
- Late Collision. The number of times a collision was detected after the slot time of the channel had elapsed.
- Lost Carrier. The number of times a carrier was lost during a transmission.
- Transmit retries. The number of times the transmitter has failed after 16 attempts to transmit a message, due to repeated collisions.

A typical statistics display is shown below.

Ethernet Tester: Reading and Writing Registers

At a Glance

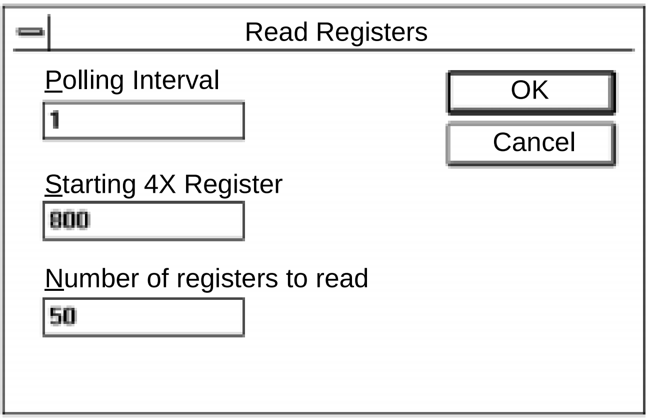
When you have established a connection with an Ethernet module, you can use the Ethernet Tester to read and write data in that module's registers. You can establish a time interval for automatic polling of the module's registers. You can automatically increment data in selected registers when writing.

Reading Registers

To read registers do the following. Refer to the figure below.

Step	Action	Comment
1	Select Messages and choose Read Registers from the pulldown menu or click on the read register button in the toolbar.	The Read Registers dialog will appear.
2	Type in a polling interval.	The polling interval is the number of seconds between transactions. When typing the 4x register number, omit the leading 40 or 400, as shown below.
3	Type in the first 4x register you want to read.	
4	Type the number of registers to read.	
5	Click OK.	The register values will be displayed in the window for this connection. Five values will be listed in each row, with the number of the first register at the beginning of the row.

The Read Register dialog box.



Read Registers

Polling Interval: 1

Starting 4X Register: 800

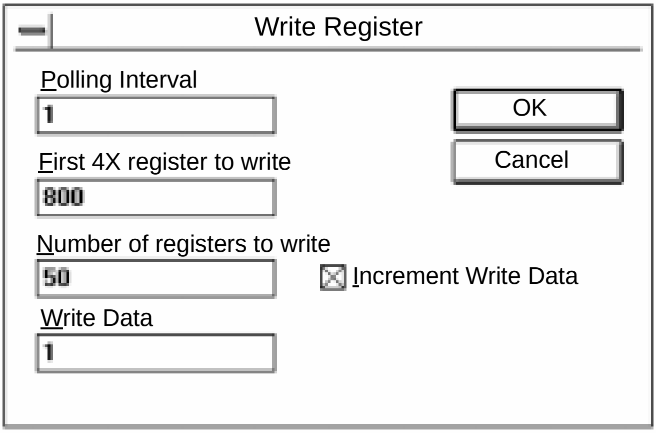
Number of registers to read: 50

Writing Registers

To write registers do the following. Refer to the figure below.

Step	Action	Comment
1	select Messages and choose Write Registers from the pulldown menu or click on the write register button in the toolbar.	The Write Register dialog will appear. A typical dialog is shown below.
2	Type in a polling interval	The polling interval is the number of seconds between transactions. When typing the 4x register number, omit the leading 40 or 400, as shown above. To change the polling interval without interrupting communication with the Ethernet module, select Messages and choose Poll Interval. Type the new polling interval in the box.
3	Type in the first register you want to write	
4	Type in the number of registers to write	If you try to read or write registers and an error occurs, the NOE Tester will display a Read Request Error or Write Request Error. The error codes correspond with MSTR block error codes.

The Write Register dialog box.



Write Register

Polling Interval: 1

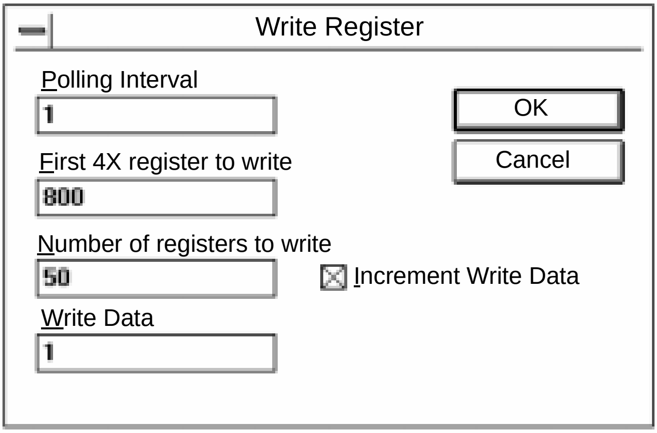
First 4X register to write: 800

Number of registers to write: 50

☒ Increment Write Data

Write Data: 1

Step	Action	Comment
5	Type in the data to be written to those registers	If you select the Increment Write Data box, the value of the data you have entered will be increased by one with each transaction. The write data will be displayed in the window for this connection.
The Write Register dialog box.		



Write Register

Polling Interval: 1

First 4X register to write: 800

Number of registers to write: 50

☒ Increment Write Data

Write Data: 1

Maintenance

6

At a Glance

Overview

This chapter details maintenance of the Ethernet TCP/IP module regarding responding to errors, hot swapping an Ethernet module, and downloading a enw software image.

What's in this Chapter?

This chapter contains the following topics:

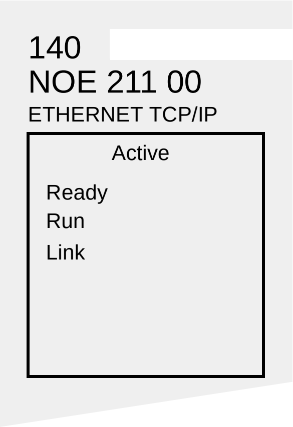
Responding to Errors

Responding to Errors

When faults occur in communication with the NOE module, the LED display can help you determine what went wrong. In addition, if the module crashes it will note its error condition in an error log file. Information for reading the LED display and error log is contained below.

Normal Operation of the LED Display

During normal operation, the LEDs should display the following pattern.



In normal operation the Run indicator will flash. The Coll LED also may flash, indicating that collisions are occurring on the Ethernet network. Such collisions are normal.

If a fault has occurred, the normal LEDs may be extinguished or other indicators may light. This section will discuss errors reported by the Active, Ready, Coll, Link, Kernel, Appl and Fault indicators.

For each type of error, try the suggested remedies in the order given. If no remedy suggested here overcomes the error, consult your Schneider Automation customer service representative.

Certain error codes are recorded in the MSTR block. Refer to the description of the MSTR block for instructions on how to read and interpret those codes.

Active LED Error

When the Active LED fails to light, then the NOE 771 00 module is not communicating with with the backplane. Perform the following checks

Step	Action
1	Make sure the module and the controller are installed properly.
2	Verify that the controller is functioning. If it isn't, replace it.
3	If neither the new controller nor the module will function, replace the backplane.
4	Make sure that no more than two network option modules -- including NOE, NOM, NOP and CRP 811 modules -- have been installed in the backplane with a 140 CPU 113 or 213; no more than six network option modules with a 140 CPU 424 or 534.
5	Check the version of the controller executive. You must have version 2.0 or greater to support the Ethernet module. Earlier versions do not recognize the module.
6	If steps 4 and 5 above check ok, replace the module.

Ready LED Error

The Ready LED fails to light, the module has failed internal diagnostic tests. Perform the following checks.

Step	Action
1	Make sure that power has been applied to the backplane.
2	If step 1 checks ok, replace the module.

Link LED Error

If the Link LED fails to light, the module is not communicating with the Ethernet hub/switch. Perform the following checks

Step	Action
1	Make sure that the cable has been installed correctly and the module is functioning properly.
2	Verify that the hub/switch is working properly.
3	If steps 1 and 2 above check ok , replace the module.

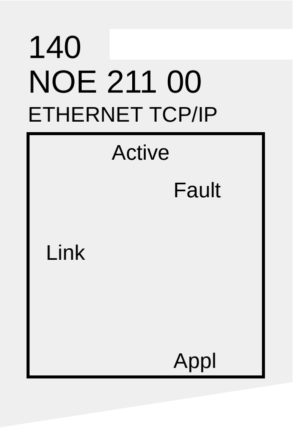
Kernel LED Error

Check for the following Kernel LED error conditions:

If...	Then ...
The Ready LED is on and the Kernel LED is flashing,	the module has detected an invalid software image.
The Ready LED is on and the Kernel LED is shining steadily,	an attempt to download a software image has failed and the module is in kernel mode.
Either of the above conditions exist,	download a new *Ethernet Tester: Establishing a Connection with an Ethernet Module, p. 85*

Fault LED

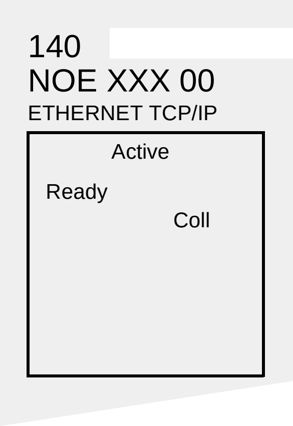
The location of the Fault LED is shown below.



The Fault LED will flash briefly following an error as the module attempts to recover. The Fault indicator will remain on only when the error log is full. In that case, the module will be unable to recover. Use the ERRLOG utility to clear the error log.

Collision LED

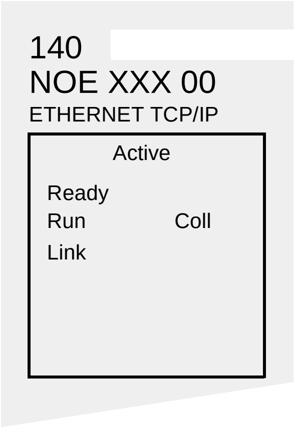
If the twisted pair cable has not been connected properly, the Coll LED will shine steadily and the Link LED will be extinguished. (This condition does not occur with fiber optic modules.)



Perform the following checks:

Step	Action
1	Make sure the cable has been installed properly and is working properly.
2	Verify that the Ethernet hub/switch is functioning properly.

If the Coll LED is flashing, the module is reporting collisions on the Ethernet network. While such collisions are normal, the frequency of the flashes is an indication of the volume of traffic on the network. The flashes may be so frequent that the LED appears to be shining steadily. Heavy traffic will slow communications. If response time is important to your application, you should consider segmenting your network to reduce the frequency of collisions.



Application LED

If the module crashes, it will note the reason in a log. If the module is able to recover, the Appl LED will light, indicating that an entry has been made in the error log. To learn how to read and clear the error log, refer to the section below.

Reading and Clearing the Crash Log

The crash log provides you with the ability to capture conditions that lead to an anomalous condition. By providing the crash log to Schneider Electric technical support, you can facilitate their assistance in resolving your problems.

Note: The crash log is provided with the understanding that with a complex product in thousands of customer applications, there may be conditions that require advance diagnostics. The crash log is one of the tools used to solve complex problems.

The Crash Log: If the Appl indicator is lit, entries have been made in the crash log. The log may hold up to 64K of entries.

Reading the Crash Log: The crash log can be read from the Embedded Web Pages (see Chapter 7) or via FTP.

Reading the Crash Log via FTP: Perform the following steps to access the crash log via FTP

Step	Action
1	Log into the module’s FTP Server.
2	Change the directory to wwwroot/conf/diag.
3	Perform an FTP to get the crash log: get crash log.

Clearing the Crash Log: The crash log can be cleared from the Embedded Web Pages (see Chapter 7) or via FTP

Clearing the Crash Log via FTP: Perform the following steps to clear the crash log via FTP.

Step	Action
1	Log into the module’s FTP Server.
2	Change the directory to wwwroot/conf/diag.
3	Perform an FTP rm crash.log to delete the crash log file.

Hot Swapping an Ethernet Module

At a Glance

You may replace your Ethernet module while the controller is running, however you must observe the precautions described below.

Hot Swapping the Module

Observe these precautions in hot swapping your Ethernet module.

⚠	CAUTION
	PROCESS INTERRUPTION Before swapping the module, make sure that the IP network address of the replacement module will be unique on your network. **Failure to observe this precaution can result in injury or equipment damage.**

The new Ethernet module will inherit any configuration changes you had made. If the original Ethernet module was given a user-configured address, the new module will assume that address. If you will be using the default address, check with your system administrator to ensure that address is not already in use on your network.

Note: If you are replacing the module because it failed, be aware that you may have lost several transactions. These transactions are not captured in memory and cannot be recovered by the new module.

Hot Swapping a Module

Follow these steps to hot swap a module.

Step	Action
1	Disconnect the cable and remove the old module from the backplane.
2	Insert the new module in the slot and reconnect the cable.

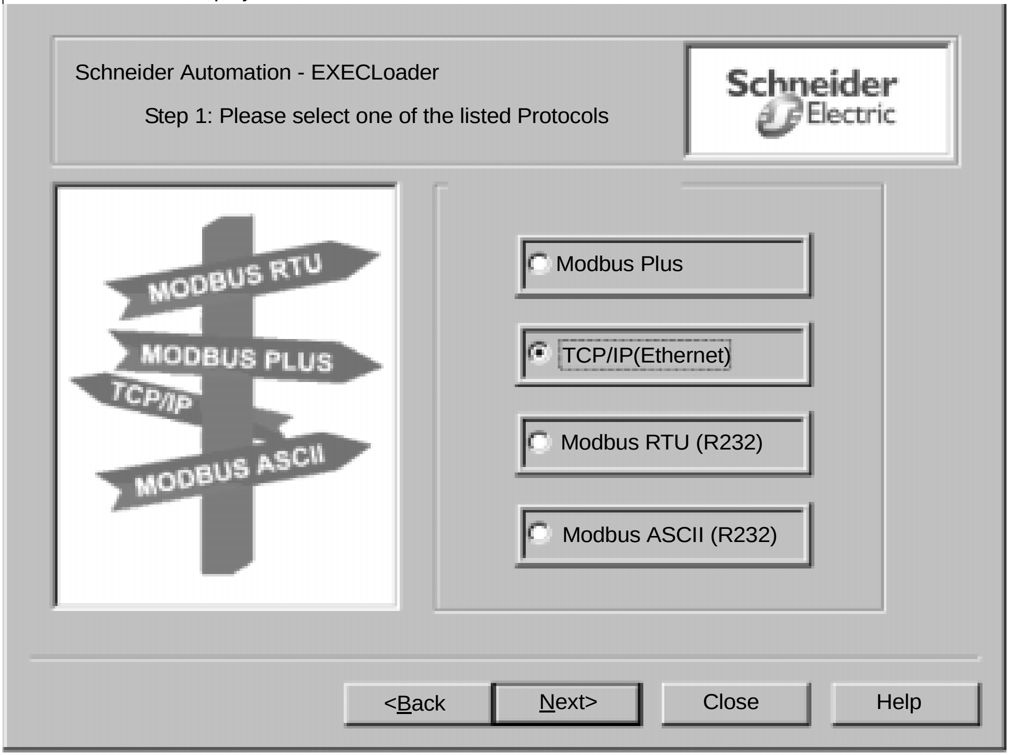
Downloading a New Software Image

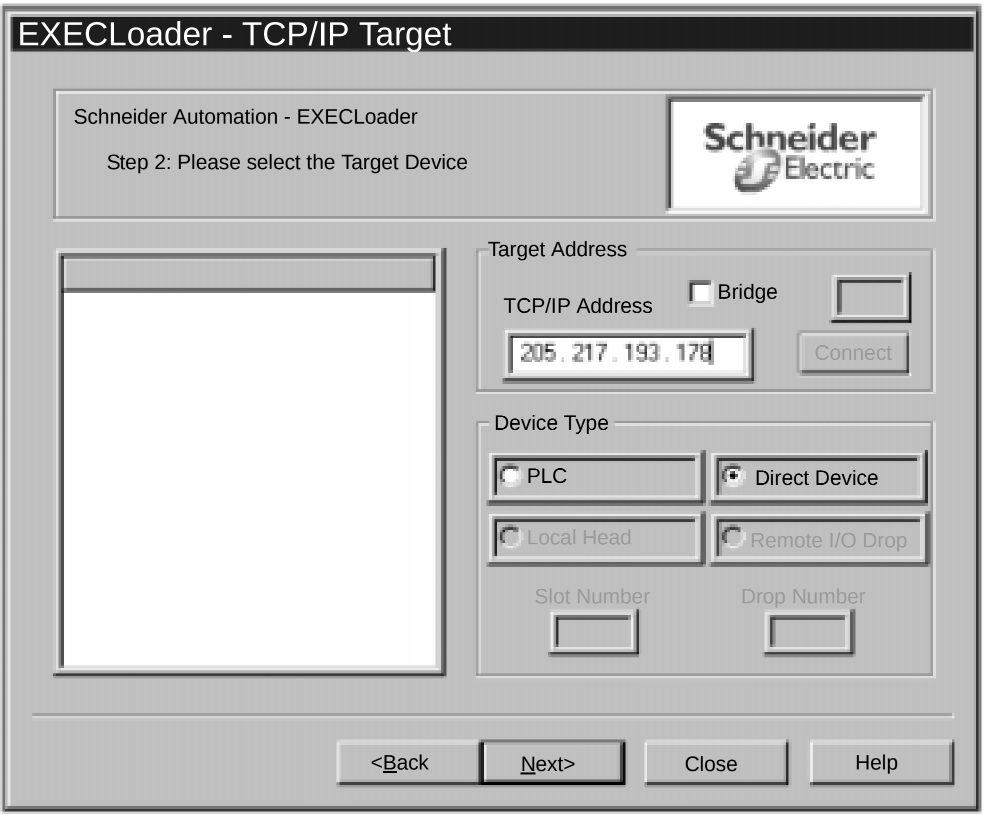
At a Glance

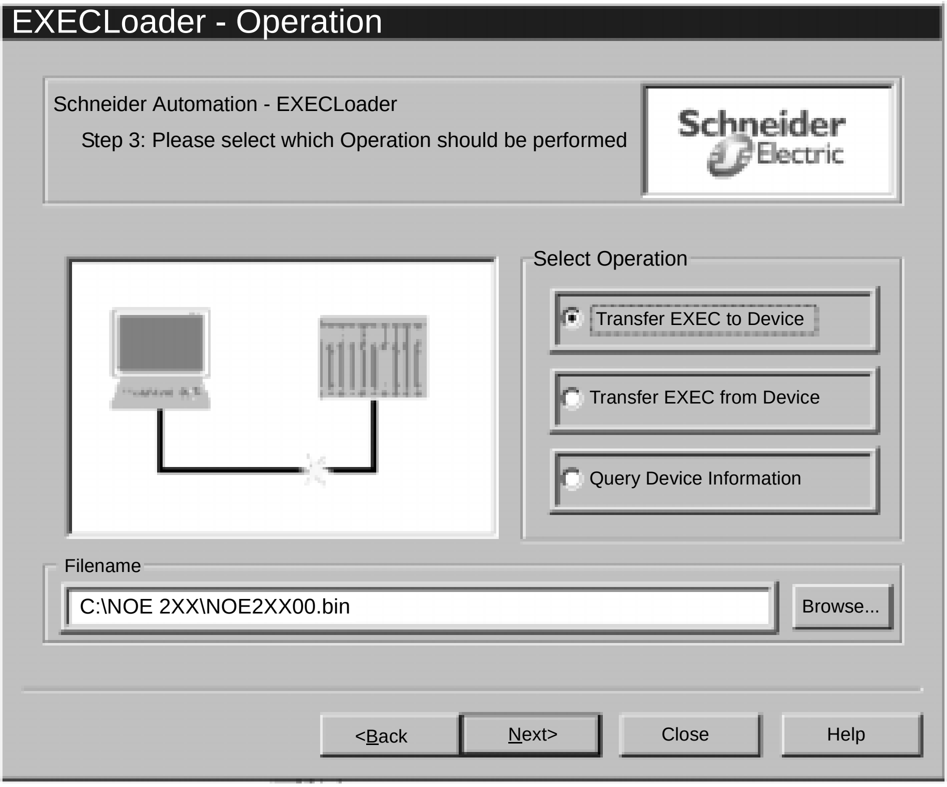
From time to time, Schneider Automation may release improved versions of the Quantum Ethernet TCP/IP software. These new software images may be downloaded through Modsoft using the following procedure.

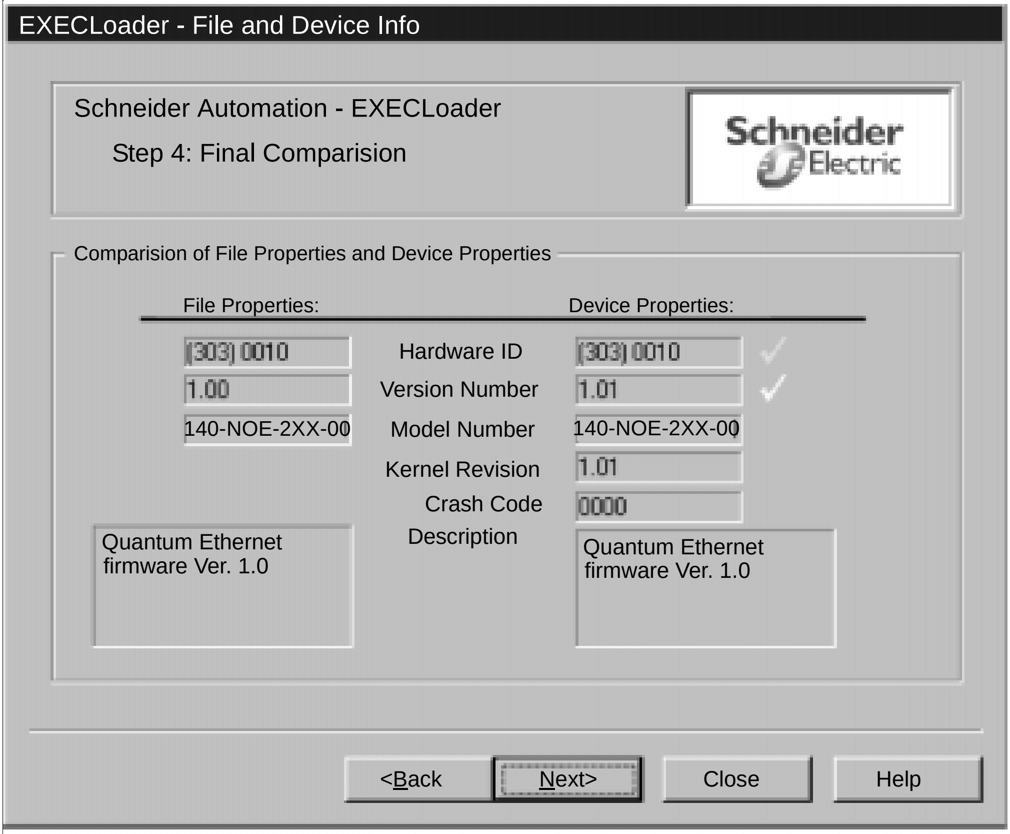
The Concept Exec Loader

The following steps detail downloading a new NOE Exec using the Concept Exec Loader

Step	Action
1	Activate the Exec Loader program.
2	Click on the Next button to start the Exec Loader processResult: The EXECLoader - Communication Protocol screen is displayed.  Schneider Automation - EXECLoader Step 1: Please select one of the listed Protocols ○ Modbus Plus ● TCP/IP(Ethernet) ○ Modbus RTU (R232) ○ Modbus ASCII (R232) MODBUS RTU MODBUS PLUS TCP/IP MODBUS ASCII <Back Next> Close Help
3	Click on TCP/IP [Ethernet

Step	Action
4	Click on the Next button Result: The EXECLoader - TCP/IP Target screen is displayed.  EXECLoader - TCP/IP Target Schneider Automation - EXECLoader Step 2: Please select the Target Device Target Address TCP/IP Address 205 . 217 . 193 . 178 □ Bridge Connect Device Type ○ PLC ● Direct Device ○ Local Head ○ Remote I/O Drop Slot Number Drop Number <Back Next> Close Help
5	Click on Direct Device

Step	Action
6	Click on the Next button Result: The EXECLoader - Operation screen is displayed.  EXECLoader - Operation Schneider Automation - EXECLoader Step 3: Please select which Operation should be performed Select Operation ● Transfer EXEC to Device ○ Transfer EXEC from Device ○ Query Device Information Filename C:\NOE 2XX\NOE2XX00.bin Schneider Electric Browse... <Back Next> Close Help
7	Click on Transfer EXEC to Device
8	Use Browser to select the file name.

Step	Action
9	Click on the Next button Result: The EXECLoader - File and Device Info screen is displayed. 

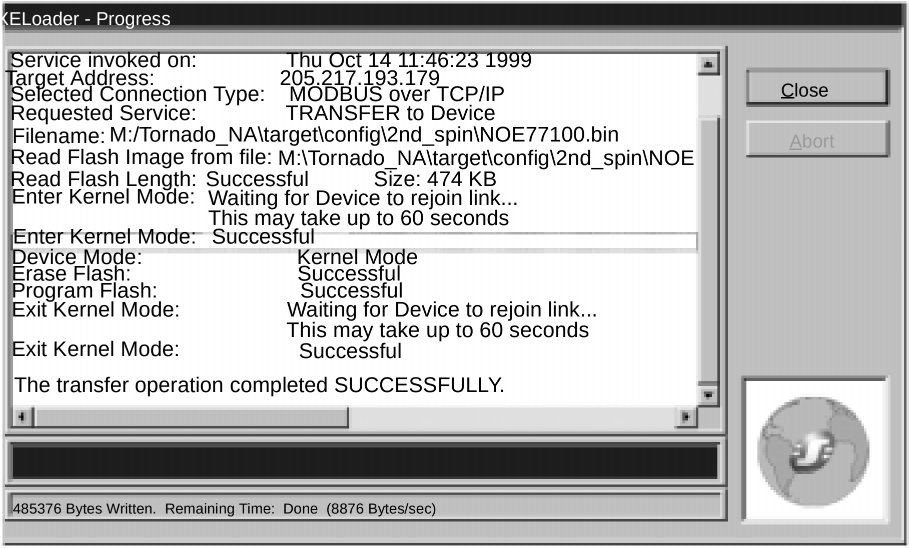
EXECLoader - File and Device Info

Schneider Automation - EXECLoader

Step 4: Final Comparision

Comparision of File Properties and Device Properties

File Properties:		Device Properties:	
(303) 0010	Hardware ID	(303) 0010	✓
1.00	Version Number	1.01	✓
140-NOE-2XX-00	Model Number	140-NOE-2XX-00	
	Kernel Revision	1.01	
	Crash Code	0000	
Quantum Ethernet firmware Ver. 1.0	Description	Quantum Ethernet firmware Ver. 1.0	

Step	Action
10	Click on the Next button Result: The EXECLoader - Progress screen is displayed.  ELoader - Progress Service invoked on: Thu Oct 14 11:46:23 1999 Target Address: 205.217.193.179 Selected Connection Type: MODBUS over TCP/IP Requested Service: TRANSFER to Device Filename: M:/Tornado_NA\target\config\2nd_spin\NOE77100.bin Read Flash Image from file: M:\Tornado_NA\target\config\2nd_spin\NOE Read Flash Length: Successful Size: 474 KB Enter Kernel Mode: Waiting for Device to rejoin link... This may take up to 60 seconds Enter Kernel Mode: Successful Device Mode: Kernel Mode Erase Flash: Successful Program Flash: Successful Exit Kernel Mode: Waiting for Device to rejoin link... This may take up to 60 seconds Exit Kernel Mode: Successful The transfer operation completed SUCCESSFULLY. 485376 Bytes Written. Remaining Time: Done (8876 Bytes/sec)
11	When the process is completed, you can click on the Close button.

Appendices

What's in this Appendix?

The appendix contains the following chapters:

A

Specifications

Ethernet TCP/IP Modules: Specifications

At a Glance

Specifications are shown below for these Ethernet modules:

- NOE 211 00
- NOE 251 00

Specifications

Ethernet Module Specifications

Specification	Description
Communication Ports	
Ethernet ports transmit and receive Modbus commands encapsulated in TCP/IP protocol: TCP/UDP system port number 502 used with ASA protocol_id of 0	
NOE 211 00	1 10BASE-T Ethernet network (RJ-45) port
NOE 251 00	1 10BASE-FL Ethernet network (ST-style) port
Operating Power	
Power Dissipation	5 W
Bus Current Required	1 A
Operating Conditions	
Temperature	0 to 60C
Humidity	0 to 95% Rh noncondensing @ 60C
Altitude	15,000 ft (4500 m)
Vibration	10-57 Hz @ 0.0075 mm d.a. 57-150 Hz @ 1 g
Storage Conditions	
Temperature	-40 to +85C
Humidity	0 to 95% Rh noncondensing @ 60C
Free Fall	1 m unpackaged
Shock	3 shocks / axis, 15 g, 11 ms

B

Ethernet Developer’s Guide

Ethernet Modules: Developer’s Guide

At a Glance

This Guide describes the design of the sample TCP/IP application named Network Options Ethernet Tester (NOET). The NOET application is a Windows MDI (Multiple Document Interface) application that verifies the installation of the Quantum Ethernet TCP/IP module and also serves as a sample application for developers. TCP/UDP system port number 502 is used with ASA protocol_id of 0.

References

Inside Visual C++, Second Edition, David J. Kruglinski
Window Sockets, An Open Interface for Network Programming under Microsoft® Windows Version 1.1

Application Overview

The sample application performs the following steps:

- Calls the window socket function **socket** to create a socket.
- Calls the window socket function **setsockopt** to set the socket attributes.
- Calls the window socket function **connect** to establish a connection.
- Encodes the request. The request consists of a header followed by a Modbus message. The header contains an invoke identifier, a protocol type, the command length, and a destination identifier.

Invoke Identifier	Protocol Type	Command Length	Destination ID	Modbus Message

- Calls the window socket function **send** to transmit the request to the remote node.
- Calls the window socket function **recv** to receive the response from the remote node.
- Calls the window socket function **closesocket** to close the connection and release the socket.

The winsock.lib import library provided by the installation is used to link the window socket calls.

Development Environment

The sample application was developed with Microsoft Visual C++, version 1.52. The sample application uses Microsoft Foundation Class. The initial application was generated by the Visual C++ application wizard.

Class Descriptions

The following list describes the different classes.

- **CSample_app.** The Csample_app is the application class. This application was generated by the application wizard, and the source is in the file sam_app.cpp. The class declaration is in sam_app.h.
- **CMainFrame.** The CMainFrame is derived from the MFC class CMDIFrameWnd and is the application's main window frame. The source for CMainFrame is in mainfrm.cpp, and the declaration is in mainfrm.h. The code for CMainFrame was initially generated by the application wizard, and was modified to process window timer messages.
- **CSample_doc.** The CSample_doc is the document class. The declaration is in sam_doc.h and the implementation is in sam_doc.cpp.
- **CSample_View.** The CSample_View is the view of the document. It is derived from the CScrollView class. The declaration is in the sam_vw.h class, and it is implemented in the sam_vw.cpp, disp.cpp, tcp_hlp.cpp, and the tx_rx.cpp files.
- **CIP_dig.** The CIP_dlg class is the dialog class for getting the IP address. It is derived from the CDialog class. The declaration is in the cip_dlg.h file and the implementation is in the cip_dlg.cpp file. Both of these files were generated by The Visual C++ class wizard.
- **ClrStatsDlg.** The ClrStatsDlg class is the dialog class for clearing statistics. It is derived from the CDialog class. The declaration is in the cstatdlg.h file and the implementation is in the cstatdlg.cpp. Both of these files were generated by The Visual C++ class wizard.
- **GetStatsDlg.** The GetStatsDlg class is the dialog class for get statistics. It is derivied from the CDialog class. The declaration is in the gstatdlg.h file and the implementation is in the gstatdlg.cpp file. Both of these files were generated by The Visual C++ class wizard.
- **CPollDlg.** The CPollDlg class is the dialog class for determining the poll period. It is derived from the CDialog class. The declaration is in the polldlg.h file, and the implementation is in the polldlg.cpp file. Both of these files were generated by The Visual C++ class wizard.
- **CReadDlg.** The CReadDlg class is the dialog class for determining the registers to read. It is derived from the CDialog class. The declaration is in the readdlg.h file, and the implementation is in the readdlg.cpp file. Both of these files were generated by The Visual C++ class wizard.
- **CWriteDlg.** The CWriteDlg class is the dialog class for determining the registers to write and the write data. It is derived from the Cdialog class. The declaration is in the writedlg.h and the implementation is in the writedlg.cpp file. Both of these files were generated by The Visual C++ class wizard.

- **CAboutDlg.** The CAboutDlg class is the dialog class for about. Both the declaration and its implementation are in the sam_app.cpp file.

The CSample_doc Class

The CSample_doc (the document class) contains the user data used by the CSample_View class. The user data consists of the remote node’s IP address, the transaction type and its associated values. The different transaction types are read register, write register, clear statistics, and get statistics. In addition to the transaction type and the associated values, the document class also contains the poll interval.

A user modifies the user data via a menu or tool bar. The CSample_doc processes the menu or tool bar window command message by invoking the corresponding dialog. The state of the various menu items and tool bar buttons depends on the connection state between the application and the remote node. The CSample_View class maintains the connection state, and hence sets the state of the menu items and tool bar buttons.

The CSample_View Class

The CSample_View class manages the TCP/IP connection, sends requests to remote nodes, and displays either connection state, or the results of a transaction. In addition it sets the states of the tool bar buttons and menu items.

Accessing TCP/IP

The CSample_View interfaces with window sockets via its application programming interface, and via messages sent by the window sockets DLL to the CSample_View window. The reference for the window socket API is given above. The first call made to the window sockets DLL must be WSAStartup. This call is made by InitInstance member function of the CSample_app class. The last call to the window socket DLL must be WSACleanup. This call is made by the ExitInstance member function of the Csample_app class.

The CSample_View allocates and sets the socket attributes. The attributes it sets are:

- Set Linger to cause a hard close
- Receive out of band data in the normal data stream
- Disable Nagel algorithm for send coalescing.

When Nagel algorithm is disabled, if the stack receives an application message, it will immediately pass the message to the application and will send a TCP/IP acknowledgment message. Although this can generate more traffic, the application receives the message sooner then if Nagel algorithm is enabled. The member function tcpip_setsocket_options sets the socket attributes.

The window socket interface provides the WSAAsyncSelect function which notifies the window of network events. The member function tcpip_setsocket_options calls WSAAsyncSelect function.

The different events are given by the following table.

Event	Description
FD_READ	A socket can read data
FD_WRITE	A socket can write data
FD_OOB	A socket can read out of band data
FD_CONNECT	A connect response has been received
FD_CLOSE	The connection has been closed

One of the parameters to the WSAAsyncSelect is a user defined message the window socket DLL sends to the window. The sample application user message is WM_TCPIP_EVENT and is defined in the file wn_msh.h. MFC architectural framework calls the CSample_View tcpip_event member function to process this message. Like all functions which process messages, tcpip_event parameters are a word and a long word. The word parameter is the socket, and the long word parameter contains the network event , and an error code.

Tcpip_event examines the network event and calls the member function indicated in the following table.

Network Event	Member Function
FD_READ	OnTcpIpRead()
FD_WRITE	OnTcpIpWrite()
FD_OOB	OnTcpIpOob()
FD_CONNECT	OnTcpIpConnect()
FD_CLOSE	OnTcpIpClose()

Application Message Format

TCP/IP transmits a message as a stream. There is no indication of the start of a message nor the end of the message. The NOE option module adds a header to determine the message boundaries. The message is a Modbus message. The header contains the following fields.

- Invoke Identifier. This two byte field associates a request with the response. The client application picks the invoke identifier, and server returns the same invoke identifier in the response.
- Protocol Type. This two byte field identifies the protocol type. Currently, the only protocol supported is Modbus.
- Command Length. This two byte field is the size of the rest of the message.
- Destination Identifier. This one byte field is reserved for future use.

The Modbus message follows the header. The message does not contain the address field, instead, the first byte is the Modbus function code.

The data structure for the header is declared in modbus.h and the CSample_View encode_header function encodes the header. The member functions encode_clear_stats, encode_read_stats, encode_read_rq, and encode_write_rq encode the corresponding Modbus messages.

Timers

CSample_View requires to periodically receive a timer message. This message triggers the CSample_View to transmit a message. Since window timers are a limited resource, the window associated with CMainFrame class receives the timer messages. CMainFrame member AddTimerList function will place a window on its timer list. When CMainFrame processes the WM_TIMER message, it sends each window on its time list the user defined WM_POLL_INTERVAL message.

MFC calls CSample_View member OnInitalUpdate function when it is first being created. OnInitialUpdate calls CMainFrameís AddTimerList in order to receive the WM_POLL_INTERVAL message. MFC architectural framework calls CSample_View OnPollInterval member function to process this message.

Transaction Processing

CSample_View transaction processing consists of establishing a connection, transmitting the request, receiving the response, and displaying the response. CSample_View uses both a transmit and a receive state machine to advance a transaction.

Transmit State Machine

The transmit state machine establishes a connection, and periodically transmits a request. The different states for the transmit state machine are as follows.

- IDLE. In the IDLE state, there is no connection.
- RESOLVING_NAME. In the RESOLVING_NAME state, CSample_View is waiting for the window socket DLL to convert a node’s name into an IP address.
- CONNECTING. In the CONNECTING state, CSample_View is waiting for the window socket DLL to generate the FD_CONNECT event. This event indicates if the attempt to establish a connection succeeded or failed.
- CONNECTED. The CONNECTED state indicates that a connection has been successfully established.
- WAIT_TO_TX. In the WAIT_TO_TX state, CSample_View is waiting to transmit the message. It transmits the message, when the time from the last transmit exceeds the specified poll interval.
- BLOCKED. When CSample_View attempts to send a message, the window socket DLL may not be able to transmit the complete message. This is a flow control condition, and CSample_View enters the BLOCKED state. The window socket DLL generates the FD_WRITE event when it can send more data.
- TX_DONE. CSample_View enters the TX_DONE when it has completed transmitting the request.

If the CSample_View is in the IDLE state, and user selects either the connect menu item, or the connect tool bar button, CSample_View OnManagConnect function attempts to establish connect with its tcpip_initate_connection function. This function examines the remote destination and determines if it’s a name or an IP address.

If the remote destination is a name, OnMangConnect changes the transmit state to RESOLVING_NAME, and it invokes the window sockets DLL WSAAsyncGetHostByName function to resolve the name. Window sockets DLL will generate the user defined WM_TCPIP_NAME_RESOLVED message which indicates if the name has been resolved. The OnTcpIpNameResolved member function process the WM_TCPIP_NAME_RESOLVED message. If the name is not resolved, OnTcpIpNameResolved changes the transmit state back to IDLE.

If the remote node is an IP address, or if it’s a name that has been resolved, then CSample_View tcpip_connect_rq function is called to initiate a connect request to the remote node. The listen port for the connect request is five hundred and two, and is defined by the constant MBAP_LISTEN_PORT in modbus.h. If tcpip_connect_rq succeeded in initiating a connect request, then tcpip_connect_rq changes the transmit state to CONNECTING, otherwise it changes the transmit state to IDLE.

The window sockets DLL generates a FD_CONNECT event which indicates if the connect request succeeded or failed. CSample_View OnTcpIpConnect function processes the FD_CONNECT event. If the connect request succeeded, OnTcpIpConnect changes the transmit state to CONNECTED, otherwise it changes the state to IDLE.

Recall that MFC architectural framework calls CSample_View OnPollInterval member function to processes WM_POLL_INTERVAL message sent as result of CMainFrame class processing a WM_TIMER message. OnPollInterval examines the transmit state. If the transmit state is CONNECTED, and the user has selected a transaction type, then OnPollInterval calls CSample_View TransmitUserRequest function.

TransmitUserRequest encodes a request based on the transaction type, saves the current time, and calls CSample_View TransmitMessage function. OnPollInterval uses the saved time to determine when to transmit the next request. TransmitMessage attempts to send a message to the remote side. To send the message, TransmitMessage enters a loop. In the body of the loop transmit message calls the window socket DLL send function. The following lists the outcomes of the send function and the actions taken.

- The message was sent successfully. TransmitMessage changes the transmit state to TX_DONE and exits the loop.
- Only part of the message was sent. TransmitMessage reenters the loop.
- Send function returns an error indicating there is no buffer space within the transport system. TransmitMessage changes the transmit state to BLOCKED and exists the loop.
- Send function returns some other error. TransmitMessage closes the connection, changes the transmit state to IDLE, and exits the loop.

When buffer space within the transport system becomes available to transmit messages, the window socket DLL generates a FD_WRITE event. CSample_View OnTcpWrite function processes the FD_WRITE function by calling TransmitMessage.

The receive state machine (which is described below) processes the response to a request. When the receive state machine has completed receiving the response, it changes the transmit state machine from the TX_DONE state to the WAIT_TO_TX state.

Recall that the TransmitUserRequest saves the time. CSample_View OnPollInterval uses this saved time to determine if a new request needs to be transmitted. OnPollInterval is called by MFC architectural framework to process the WM_POLL_INTERVAL sent when CMainFram class processes the window timer message, WM_TIMER. OnPollInterval examines the transmit state. If the transmit state is WAIT_TO_TX, and the elapsed time from the previous transmit request exceeds the poll interval, then OnPollInterval calls TransmitUserRequest to start another transaction.

Receive State Machine

The receive state machine receives a response to a transaction by first reading the header, determining the size of the rest of the message, and then reading the body of the message. The different states of the receive state machine are as follows.

- RX_HEADER. In the RX_HEADER state, the receive machine is receiving the message header.
- RX_BODY. In the RX_HEADER state, the receive machine is receiving the response message associated to the requested transaction.
- DUMP_BODY. In the DUMP_BODY state, the receive message is receiving a message, but there is no associated transaction with respect to this message.

The window socket DLL generates the FD_READ event whenever there is data to be read. If only part of the data is read, it generates another event. CSample_View OnTcpIpRead function processes the FD_READ event, and drives the receive state machine.

When a FD_READ event is generated it is possible that the complete message is not present. The remote node may have attempted to send a 100 byte response, but the transport system may have only had buffer space to transmit three bytes. The receiver will get a FD_READ for the three bytes. OnTcpIpRead calls CSample_View rx_msg to read the receive data into the buffer. There are three parameters to rx_msg. The first parameter is a pointer to a receive buffer. The second input parameter is the receive size. The third parameter is both an input and output parameter. On both input and output the third parameter is the number of bytes read. These parameters allow the processing of a partially received message.

The receive state machine maintains a variable which is the number of bytes received. Initially the receive state machine is in the RX_HEADER state, and the number of bytes received is zero.

When OnTcpIpRead is called and the receive state is RX_HEADER OnTcpIpRead calls rx_msg with receive size equal to the header size. On return OnTcpIpRead examines the number of bytes received. If the number of bytes received is not equal to the header size, then receive machine remains in the RX_HEADER state, and OnTcpIpRead returns.

If upon return, the number bytes received is the same size as the header size, then the header has been received. OnTcpIpRead sets the number of bytes received to zero, and the receive size is obtained from the header. These two values will be used the next time rx_msg is called. OnTcpIpRead also obtains the transaction identifier and the protocol type from the header. If the transaction identifier matches the transmit request identifier and the protocol type is MODBUS, then OnTcpIpRead changes the receive state to RX_BODY. However if either transaction identifier does not match or the protocol is not MODBUS, then OnTcpIpRead changes the receive state to DUMP_BODY.

When OnTcpIpRead is called and the receive state is RX_BODY, OnTcpIpRead calls rx_msg with receive size equal to the value obtained from the header. On return OnTcpIpRead examines the number of bytes received. If the number of bytes received is not equal to the receive size, then the receive machine remains in the RX_HEADER state, and OnTcpIpRead returns.

If upon return the number of bytes received is the same as the receive size, then OnTcpIpRead has read the response to a transaction. OnTcpIpRead saves the results and invalidates the client area which causes the results to be display. OnTcpIpRead also changes the transmit state to WAIT_TO_TX, and resets the state receive state machine by setting the state to RX_HEADER and the number of bytes received to zero. It then returns.

When OnTcpIpRead is called and the receive state is DUMP_BODY, OnTcpIpRead calls rx_msg with receive size equal to the value obtained from the header. On return OnTcpIpRead examines the number of bytes received. If the number of bytes received is not equal to the receive size, then the receive machine remains in the RX_HEADER state, and OnTcpIpRead returns.

If upon return the number of bytes received is the same as the receive size, the OnTcpIpRead has completed reading the message. Since this message does not correspond to an transaction, the only processing OnTcIpRead performs is resetting the receive state machine.

The member function rx_msg calls the window socket recv function to read data. The recv function either returns a non negative number that is the number of bytes read or it returns an error. If the number bytes read is zero, then the connection no longer exits, and rx_msg closes the socket, and sets the transmit state to IDLE. If the recv function returns the error indicating that no receive data is available, then rx_msg just returns. For any other recv function error, rx_msg closes the socket, and sets the transmit state to IDLE.

Displaying on the Screen

CSample_View m_display member indicates the display type. The different types of the displays and the CSample_View member functions for showing the display are as follows.

- Displaying the connection state. The different connection states displayed are IDLE, RESOLVING NAME, and CONNECTING. ConnPaint member function displays the connection state.
- GetStatsPaint member function displays the results of a get statistics request.
- ClearStatsPaint member function displays the results of a clear statisitics request.
- ReadRegPaint member function displays the results of a read register request.
- WriteRegPaint member function displays the results of a write register request.

MFC architectual framework calls CSample_View OnDraw member function to process the window WM_PAINT message. OnDraw examines m_display member variable and calls the corresponding member function described in the previous paragraph. Whenever CSample_View needs to display a result, it calls Cview Invalidate function which causes a WM_PAINT message.

CSample_View is derived from MFC CScrollView class. This class handles the scroll logic. To perform the scroll logic, CScrollView requires the size of the document. It is informed of the document size via its SetScrollSizes member function.

CSample_View UpdateScrollSizes member function based on the display type calculates the document size, and then calls SetScrollSizes. CSample_View calls UpdateScrollSizes when the display type changes or when the user changes the window size.

C

Quantum Ethernet TCP/IP Modbus Application Protocol

At a Glance

Overview

This chapter describes the Quantum Ethernet TCP/IP Modbus Application Protocol, which constructs message transactions between controllers and other networked node devices.

What's in this Chapter?

This chapter contains the following topics:

Modbus Application Protocol: Introduction

At a Glance

The information below introduces the application of Modbus protocol for message transactions between controllers and other networked node devices. For details about how Modbus messages are constructed, see the Modbus Protocol Reference Guide (PI-MBUS-300).

Modbus Protocol in Message Transactions

The Modbus Application Protocol (MBAP) is a layer-7 protocol providing peer-to-peer communication between programmable logic controllers (PLCs) and other host-based nodes on a LAN. Collectively these nodes implement all or part of a control application used for industrial automation applications in the automotive, tire and rubber, food and beverage, and utilities industries, to name a few.

Modbus protocol transactions are typical request-response message pairs. Modbus requests contain function codes representing several classes of service including data access, online programming, and program download and upload classes. Modbus responses can be ACKs with and without data, or NACKs with error information.

The Modbus Application Protocol can be transmitted over any communication system that supports messaging services. However, the current Quantum implementation transports Modbus Application Protocol PDUs over TCP/IP. Both Ethernet II and IEEE 802.3 framing are accommodated, although Ethernet II framing is the default.

Modbus Application Protocol: PDU Introduction

At a Glance

The following information defines the structure and content of the Modbus Application Protocol PDU, and provides an example.

PDU Layout

The Modbus Application Protocol PDU, mbap_pdu, is received at TCP port number 502. The current maximum size of the mbap_pdu for this class of services is 256 bytes.

The structure and content of the mbap_pdu are defined to be:

mbap_pdu ::={ inv_id[2], proto_id[2], len[2],dst_idx[1], data=mb_pdu

Header of PDU

The header is seven bytes long and includes the following fields:

inv_id:	[2 bytes] Invocation id used for transaction pairing.
proto_id:	[2 bytes] Used for intra-system multiplexing, default is 0 for Modbus services.
len:	[2 bytes] The len field is a byte count of the remaining fields and includes the dst_id and data fields.

Remainder of PDU

The remainder of the pdu consists of two fields:

dst_idx:	[1 byte] Destination index is used for intra-system routing of packets.
service portion:	[n bytes] The service portion of the Modbus pdu is defined as mb_pdu and is described below.

Service Portion of PDU

The service portion of the Modbus Application Protocol, called mb_pdu, contains two fields:

func_code:	[1 byte] Modbus function code.
modbus_data:	n bytes] This field is function code dependent and usually contains information such as variable references, variable counts and data offsets.

The size and content of the modbus_data field are dependent on the value of the function code.

PDU Example

Here are the values for a sample mbap_pdu for reading a register:
00 01 00 00 00 06 01 03 00 00 00 01
The values above have the folowing structure and content:
inv_id: 00 01
proto_id: 00 00
len: 00 06
dst_idx: 01
func_code: 03
modbus_data: 00 00 00 01

Modbus Application Protocol: Service Classes

At a Glance

The Modbus Application Protocol performs several classes of service for accessing PLCs. They are described below.

Service Classes

Data Access
Services to read/write discrete and analog data values in PLC registers.

Online Programming
Services that make relatively minor alterations to a ladder logic program, with a user-controlled introduction of these changes into the program.

Software Image Downloading and Uploading
Image download services support the downloading of a ladder logic control program to the PLC. Image upload services support the uploading of a ladder logic control program from a PLC to a PC host for archival/backup purposes.

Configuration
Configuration services allow the user to define parameter values which affect the PLC's operations, including defining register files, I/O map, communication port configuration, and scan attributes.

Device Execution State Control
The class of service allows the user to start/stop the PLC scan execution. These services require the user to be in an application login context which is obtained through other Modbus services.

Modbus Application Protocol: PDU Analysis

At a Glance

The information below gives a detailed analysis of the structure of the Modbus Application Protocol PDU.

PDU Analysis

The Modbus Application Protocol PDU is transmitted over a TCP/IP Ethernet stack. Both Ethernet II and IEEE 802.3 framing will be accommodated. Ethernet II framing is the default.

- from the wire in for IEEE 802.3 framing
- is IEEE 802.3 framing if length <=1500

802.3_pdu ::= {dst_addr[6], src_addr[6], length[2], data=802.2_pdu}

- an IEEE 802.3 PDU has a maxFrameSize of 1518 octets
- an IEEE 802.3 PDU has a minFrameSize of 64 octets

802.2_pdu : {dsap[1], ssap[1], frm_cntrl[1], snap_hdr[5], data=ip_pdu}

- the snap_hdr is associated with a "well-known" 802.2 sap snap_hdr ::={org_code[3], ethertype[2]
- the snap hdr (sub network access protocol) allows the older style Ethernet protocols to run on the newer IEEE 802.2 interface. The ethertype parameter indicates the service, ex. ip or arp. IP has a value 0x800.
- from the wire in for Ethernet II framing
- is Ethernet II framing if length >1500

802.3_pdu ::= {dst_addr[6], src_addr[6], length[2], data=ip_pdu}

- the common part of the packet begins here

ip_pdu ::= {ip_hdr[20], data=tcp_pdu}

tcp_pdu ::= {tcp_hdr[24], data=appl_pdu=mbap_pdu}

The mbap_pdu is the Modbus Application Protocol whose messages are received at a well-known port. The current maximum size of the mbap_pdu for this class of services in 256 bytes.

The structure and content of the mbap_pdu is defined to be:

mbap_pdu ::={ inv_id[2], proto_id[2], len[2], dst_idx[1], data=mb_pdu }

The header is 7 bytes long, and includes the following fields:

- inv_id [2 bytes] invocation id used for transaction pairing
- proto_id [2 bytes] used for intra-system multiplexing,default is 0 for Modbus services
- len [2 bytes] the len field is a byte count of the remaining fields and includes the dst_id and data fields.

The remainder of the pdu includes two fields:

- dst_idx [1 byte] destination index is used for intra-system routing of packets. (currently not implemented)
- data [n bytes] this is the service portion of the Modbus pdu, mb_pdu, and is defined below

The service portion of the Modbus Application Protocol, called mb_pdu, contains 2 fields:

mb_pdu ::= { func_code[1], data[n] }

- func_code [1 byte] MB function code
- data [n bytes] this field is function code dependent and usually contains information such as variable references, variable counts, and data offsets.

The size and content of the data field are dependent on the value of the function code.

Modbus Application Protocol: TCP/IP Specific Issues

At a Glance

The information below describes issues that are specific to Schneider Automation's implementation of Modbus Application Protocol in TCP/IP communication.

Broadcast/ Multicast

Although broadcast and/or multicast are supported by both IP network address and IEEE 802.3 MAC address, the Modbus Application Protocol does not support either broadcast or multicast at the application layer.

Schneider Automation's Quantum PLCs use broadcast addressing because they use ARP as the means of locating the destination node. The client interface to the Modbus Application Protocol service on the PLC, the MSTR block, requires the user to provide the destination IP address. Also the embedded stack does use a pre-configured default gateway IP address in the case where ARP does not succeed.

TCP Port Number

Schneider Automation has obtained a well-known system port from an Internet Authority. Schneider Automation's well-known system port number is 502. The Internet Authority assigned the system port number 502 to asa-appl-proto with Dennis Dubé as the company point of contact.

This port number allows Schneider Automation to transport various application protocols over with TCP or UDP. The particular protocol is indicated by the value of the proto_id parameter in the mbap_pdu. Currently the only assignment is 0 meaning Modbus Application Protocol.

Modbus Application Protocol: Reference Documents

At a Glance

Following is a listing of reference documents related to Schneider Automation's implementation of TCP/IP communications.

Reference Documents

[1] ANSI/IEEE Std 802.3-1985, ISO DIS 8802/3, ISBN - 0-471-82749-5, May 1988
[2] ANSI/IEEE Std 802.2-1985, ISO DIS 8802/2, ISBN 0-471-82748-7, Feb 1988
[3] RFC793, TCP (Transmission Control Protocol) DARPA Internet Program Protocol Specification, Sep 1981
[4] RFC 791, IP (Internet Protocol) DARPA Internet Protocol Specification, Sep 1981
[5] RFC826, An Ethernet Address Resolution Protocol (ARP), David Plummer, NIC Sep 1982
[6] RFC1042, A Standard for the Transmission of IP Datagrams over IEEE 802.2 Networks, Postel & Reynolds, ISI, Feb 1988
[7] RFC 792, ICMP (Internet Control Message Protocol) DARPA Internet C Control Message Protocol Specification, Jon Postel, Sep 1981
[8] RFC951, BOOTSTRAP PROTOCOL (BOOTP), Bill Croft and John Gilmore , September 1985
[9] RFC783, The Trivial File Transfer Protocol (TFTP) rev 2, K.R. Sollins MIT, June 1981

D

Suppliers

Suppliers of Materials

At a Glance

The information below offers suggestions for locating materials you may need for installing and testing your Ethernet network.

Schneider Automation has not qualified and does not endorse any of these products.

Suppliers

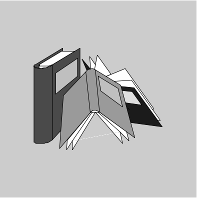
A variety of Ethernet installation tools, cable diagnostic tools, cables, connectors and other related equipment is readily available from mail order suppliers or at your local computer supply store.

Cable testing equipment is available from:

● Datacom Technologies	1-800-468-5557
● Microtest, Inc.	1-800-526-9675
● Scope Communications, Inc.	1-508-393-1236
● Wavetek, Inc.	1-800-854-2708

Schneider Automation has not qualified and does not endorse any of these products.

Glossary



A

address On a network, the identification of a station. In a frame, a grouping of bits that identifies the frame's source or destination.

API Application Program Interface. The specification of functions and data used by one program module to access another; the programming interface that corresponds to the boundary between protocol layers.

ARP Address Resolution Protocol. A network layer protocol used to determine the physical address which corresponds to the IP address for a host on the network. ARP is a sub-protocol which operates under TCP/IP.

B

bps Bits per second.

bridge A device that connects two or more physical networks which use the same protocol. Bridges read frames and decide whether to transmit or block them based on their destination address.

C

client A computer process requesting service from other computer processes.

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.

DNS Domain Name System. A protocol within TCP/IP used to find IP addresses based on host names.

F

field A logical grouping of contiguous bits that convey one kind of information, such as the start or end of a message, an address, data or an error check.

frame A group of bits which 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 types Two common framing types are Ethernet II and IEEE 802.3.

FTP File Transfer Protocol. A networking protocol used to exchange files between stations on a network or over the Internet.

G

gateway A device which connects networks with dissimilar network architectures and which operates at the Application Layer. This term may refer to a router.

H

host A node on a network.

hostname A domain name given to a specific computer on a network and used to address that computer.

HTTP HyperText Transport Protocol. A protocol used to deliver hypertext documents.

hub A device which connects a series of flexible and centralized modules to create a network.

I

ICMP Internet Control Message Protocol. A protocol within TCP/IP used to report errors in datagram transmission.

Internet The global interconnection of TCP/IP based computer communication networks.

IP Internet Protocol. A common network layer protocol. IP is most often used with TCP.

IP Address Internet Protocol Address. A 32-bit address assigned to hosts using TCP/IP.

L

layer In the OSI model, a portion of the structure of a device which provides defined services for the transfer of information.

M

MAC Address Media Access Control address. The hardware address of a device. A MAC address is assigned to an Ethernet TCP/IP module in the factory.

N

network Interconnected devices sharing a common data path and protocol for communication.

node An addressable device on a communications network.

O

OSI model Open System Interconnection model. A reference standard describing the required performance of devices for data communication. Produced by the International Standards Organization.

P

packet The unit of data sent across a network.

PING Packet Internet Groper. A program used to test whether a destination on a network can be reached.

port An access point for data entry or exit within a host using TCP services.

protocol Describes message formats and a set of rules used by two or more devices to communicate using those formats.

R

repeater A device that connects two sections of a network and conveys signals between them without making routing decisions or filtering packets.

router A device that 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 will attempt to send the packet through the network by the most efficient path.

S

server Provides services to clients. This term may also refer to the computer on which the service is based.

socket The association of a port with an IP address, serving as an identification of sender or recipient.

stack The software code which implements the protocol being used. In the case of the NOE modules it is TCP/IP.

STP Shielded Twisted Pair. A type of cabling consisting of several strands of wire surrounded by foil shielding, twisted together.

subnet A physical or logical network within an IP network, which shares a network address with other portions of the network.

subnet mask Used to indicate which bits in an IP address identify a subnet.

switch A network device which 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 Transmission Control Protocol.

TCP/IP A protocol suite consisting of the Transmission Control Protocol and the Internet Protocol; the suite of communications protocols on which the Internet is based.

U

UDP User Datagram Protocol. A protocol which transmits data over IP.

URL Uniform Resource Locator. The network address of a file.

UTP Unshielded Twisted Pair. A type of cabling consisting of insulated cable strands which are twisted together in pairs.

W

Winsock The Microsoft implementation of the Windows Sockets networking API based on the Berkeley UNIX Sockets interface for supporting TCP/IP.

WWW World Wide Web. A hypertext-based, distributed information system in which clients and servers are freely available.

Index