

GE
Intelligent Platforms

GFK-1860B

Programmable Control Products

VersaMax[®] System
Ethernet Network Interface Unit
User's Manual

Sept 2014



These instructions do not purport to cover all details or variations in equipment, nor to provide for every possible contingency to be met during installation, operation, and maintenance. The information is supplied for informational purposes only, and GE makes no warranty as to the accuracy of the information included herein. Changes, modifications, and/or improvements to equipment and specifications are made periodically and these changes may or may not be reflected herein. It is understood that GE may make changes, modifications, or improvements to the equipment referenced herein or to the document itself at any time. This document is intended for trained personnel familiar with the GE products referenced herein.

This document is approved for public disclosure.

GE may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not provide any license whatsoever to any of these patents.

GE provides the following document and the information included therein as is and without warranty of any kind, expressed or implied, including but not limited to any implied statutory warranty of merchantability or fitness for particular purpose.

For further assistance or technical information, contact the nearest GE Sales or Service Office, or an authorized GE Sales Representative.

Issued: Sept 2014

Copyright © 2014 General Electric Company, All rights reserved.

*** Indicates a trademark of General Electric Company and/or its subsidiaries. All other trademarks are the property of their respective owners.**

Refer to the section, [Contact Information](#) for support on this product.

Please send documentation comments or suggestions to controls.doc@ge.com

Safety Symbol Legend



Warning

Indicates a procedure, condition, or statement that, if not strictly observed, could result in personal injury or death.



Caution

Indicates a procedure, condition, or statement that, if not strictly observed, could result in damage to or destruction of equipment.



Attention

Indicates a procedure, condition, or statement that should be strictly followed to improve these applications.

Contents

Chapter 1	Introduction, Description, and Specifications	1-1
	Other VersaMax Manuals.....	1-1
	Ethernet NIU Specifications	1-4
	Network Interface Unit Operation.....	1-5
Chapter 2	Installation.....	2-1
	Section 1 – Installation Instructions	2-2
	Section 2 – Setting the Network IP Address	2-14
Chapter 3	Configuring an Ethernet NIU and I/O Station.....	3-1
	Using Autoconfiguration or Programmer Configuration	3-2
	Configuring “Racks” and “Slots”	3-4
	Software Configuration of the Ethernet NIU and I/O Station	3-6
	Autoconfiguration of the Ethernet NIU and I/O Station	3-15
	Configuring the ENIU Produced Exchange.....	3-18
	Configuring the ENIU Consumed Exchange.....	3-24
	EGD Exchange Status and Control Bytes for Fault Handling.....	3-32
Chapter 4	Modbus	4-1
	Modbus Protocol.....	4-2
	Modbus Tables	4-3
	Supported Function Codes.....	4-5
Chapter 5	Ethernet Global Data	5-1
	EGD Overview.....	5-1
	EGD Protocol.....	5-3
	EGD Exchange Definition	5-3
Chapter 6	Troubleshooting	6-1
	Checking Status and Operation with the ENIU’s LEDs	6-2
	ENIU Fault Table.....	6-7
	Using FTP to Obtain Network Status and Version Information.....	6-9
	EGD Troubleshooting	6-10
	Determining the MAC Address of the ENIU.....	6-11
	Reading the Stored IP Address of the ENIU.....	6-13
Chapter 7	VersaMax Product Overview.....	7-1
	The VersaMax™ Family of Products	7-1
	VersaMax Products for Ethernet Networks	7-2
	Power Supplies	7-3
	I/O Modules.....	7-4

Contents

	Carriers.....	7-7
	Expansion Modules.....	7-9
	VersaMax General Product Specifications	7-11
Appendix A	Glossary	A-1
Appendix B	IP and MAC Addresses.....	B-1
Appendix C	Number Conversion Table.....	C-1
Appendix D	Compatibility Matrix	D-1

Chapter 1

Introduction, Description, and Specifications

This manual describes installation and operation of the VersaMax™ Ethernet Network Interface Unit (ENIU). This manual has the following layout:

Chapter 1: Description and Specifications of the Ethernet Network Interface Unit.

Chapter 2: Installation procedures.

Chapter 3: Configuration procedures for the Ethernet NIU and I/O Station.

Chapter 4: Modbus/TCP Communications.

Chapter 5: Ethernet Global Data Communications.

Chapter 6: Troubleshooting.

Chapter 7: An overview of VersaMax products.

Other VersaMax Manuals

<i>VersaMax Modules, Power Supplies, and Carriers User's Manual</i> (catalog number GFK-1504).	Describes the many VersaMax I/O and option modules, power supplies, and carriers. This manual also provides detailed system installation instructions.
<i>VersaMax PLC User's Manual</i> (catalog number GFK-1503).	Describes the installation and operation of the VersaMax CPU.
<i>Remote I/O Manager User's Guide</i> (catalog number GFK-1847).	Gives step-by-step instructions for using the Remote I/O Manager configuration software.
<i>VersaMax Profibus Communications Modules User's Manual</i> (catalog number GFK-1534).	Describes the installation and operation of the Profibus Network Interface Unit module and the Profibus Network Communications Module.
<i>VersaMax DeviceNet Communications Modules User's Manual</i> (catalog number GFK-1533).	Describes the installation and operation of the DeviceNet NIU.
<i>VersaMax Genius NIU User's Manual</i> (catalog number GFK-1535).	Describes the installation and operation of the Genius NIU.

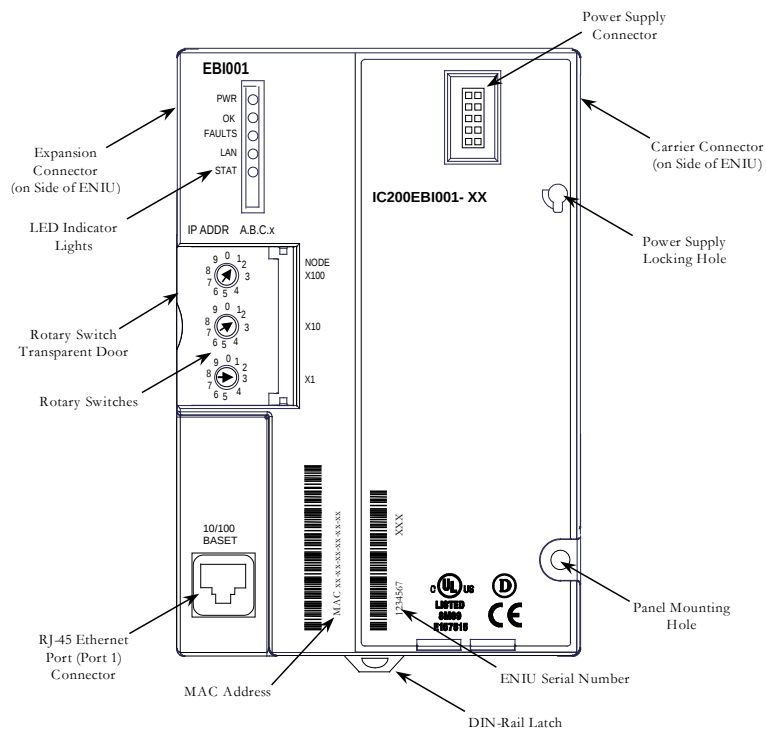
The Ethernet Network Interface Unit (IC200EBI001) operates as either a Modbus TCP server or an Ethernet Global Data (EGD) station on an Ethernet network, providing I/O data, status, and diagnostic data.

The ENIU interfaces VersaMax I/O modules to an Ethernet network. The ENIU and its expansion racks form an I/O station capable of handling up to 64 I/O modules. The memory space of the ENIU will support I/O data consisting of up to 256 bytes of discrete input, 256 bytes of discrete output, 256 bytes of analog input, and 256 bytes of analog output data. The maximum I/O that can be supported by an ENIU station is limited by the modules that are installed.

In Modbus mode, up to 10 Modbus TCP connections can be made to the ENIU at one time.

In EGD mode, the ENIU has exactly one produced and one consumed EGD exchange.

The ENIU may be configured to run either Modbus or EGD. It cannot run both modes at the same time.



The Network Interface Unit installs on a 35mm x 7.5mm conductive DIN-rail. A VersaMax power supply module¹ mounts directly on the right-hand side of the ENIU's front surface. LEDs on the left-hand side of the ENIU's front surface indicate the presence of power and show the operating mode and status of the ENIU. Three rotary switches beneath a transparent protective door can be used to configure the ENIU's address on the Ethernet network. The unshielded RJ-45 connector is used to connect the Ethernet network.

¹ Power supply must be an "Expanded 3.3V" model. See power supply specs in Chapter 7.

Ethernet NIU Specifications

Number of racks	8 per station.
Number of modules	8 per rack for a total of 64 per station.
I/O data	1024 bytes maximum. %I: 2048 points %Q: 2048 points %AI: 128 channels %AQ: 128 channels
Fault table data	32 Faults (128 bytes)
Ethernet network address	Any valid Class A, B or C address.
Ethernet network data rate	10/100Mbit auto-detect.
Ethernet Duplex	Full/Half auto-detect
Indicators (5)	PWR LED to indicate power OK LED to indicate health of the ENIU FAULTS LED to indicate presence of faults LAN LED to indicate traffic on the Ethernet network STAT LED to indicate presence of a Modbus connection or EGD is consuming.
Power Consumption*	+5V at 175mA; +3.3V at 425mA

* Requires "Expanded 3.3V" type power supply.

Configuration Software

There are two choices of configuration software for the Ethernet NIU:

- **VersaPro 2.0 (or later version).** Allows you to (1) configure all VersaMax products and (2) program PLC ladder logic.
- **Remote I/O Manager 2.0 (or later version).** Allows you to configure all VersaMax I/O products, but does not allow PLC ladder logic programming. If you will be using GE I/O products with a third-party CPU, you can use this software to configure the I/O products. This software is a stand-alone version of the VersaPro 2.0 (or later version) configuration tool.

Network Interface Unit Operation

Warning

Be sure you understand the operation of outputs at power up and with communications upsets for the mode (Default, Hold Last State) that you are using. Failure to heed this warning could result in unexpected operation, possibly leading to injury to personnel and damage to equipment.

NOTE: The following topics should be understood before starting an ENIU setup:

Modbus Mode

- Operation of discrete outputs and analog outputs upon first communication
- How much data can be exchanged in a single message
- Choosing between “Default” and “Hold Last State” configuration
- Operation at power-up
- Operation if the Ethernet Cable is disconnected, the Ethernet connection is lost, or a communication error occurs

EGD Mode

- The ordering of data in an EGD packet
- Operation if the Ethernet Cable is disconnected, the Ethernet connection is lost, or a communication error occurs

“Default” and “Hold Last State” Configuration Options

Each ENIU Output Module’s output mode must be configured individually using the configuration software. The two output mode choices are “Default” or “Hold Last State.” The following two tables show Output Table and Real Output states/values for both of these configuration choices; one table is for power up and the other for recovery from upsets. The Output Table is part of the ENIU’s internal memory; Real Outputs are the values on the output modules’ output terminals. The Output Table is segmented into two parts: Discrete Outputs and Analog Outputs.

The Power up Sequence Table

The ENIU Output Table is stored in volatile memory; therefore, all Output Table values are initially zero upon power up. Configuration settings, including the table of default values for each output, are stored in non-volatile Flash memory.

Power up Sequence				
ENIU Condition	Configured for “Default”		Configured for “Hold Last State”	
	Output Table	Real Outputs	Output Table	Real Outputs
Power up	Zeros	Default table	Zeros	Zeros
First Write to segment of Output Table after Power up (Modbus only)	Zeros plus written values	Output table	Zeros plus written values	Output table
Operational (Modbus only)	Zeros plus sum of all writes	Output table	Zeros plus sum of all writes	Output table
After first communication (EGD only)	Values from EGD exchange	Output table	Values from EGD exchange	Output table

The After Upset Sequence Table

This table shows the sequence if power remains on but the connection is upset (by such things as disconnection or communication error) and then restored.

Sequence After Ethernet Connection Loss, Cable Removal, or Communication Error				
ENIU Condition	Configured for “Default”		Configured for “Hold Last State”	
	Output Table	Real Outputs	Output Table	Real Outputs
After disconnect, cable removal, or communication error	Last values in Output table	Default table	Last values in Output table	Output table
First write to segment of Output Table after recovery (Modbus only)	Last values in Output table plus written value	Output table	Last values in Output table plus written values	Output table
Operational (Modbus only)	Last values in Output table plus sum of all writes	Output table	Last values in Output table plus sum of all writes	Output table
After first communication (EGD only)	Values from EGD exchange	Output table	Values from EGD exchange	Output table

Note: In Modbus mode, after disconnect, cable removal, or communication error, and before the first Write message is implemented, reads of the ENIU, which return values from the Output Table, may not be the same as the values of the Real Outputs for modules configured for “Default.”

Modbus Operation

Operation at Power up

Upon power up, the ENIU will send the values for discrete outputs and analog outputs to the output modules based on the configuration choice of “Defaults” or “Hold Last State” that is configured for each module. The output modules will continue using these values until the first output message is sent to the appropriate segment of the Output Table. Both segments must be written to drive both segments. Upon receiving this output message, the ENIU will update its output table, then write the entire output table to the output modules.

Operation if Ethernet Cable is Disconnected, Ethernet Connection is Lost, or Communication Error Occurs

Upon upset, the discrete and analog output modules will go to either the default state or hold last state, depending on which was configured, on a module-by-module basis. The output modules will continue using these values until the first output message is sent to the ENIU. Upon receiving this output message, the ENIU will update its output table, then write the entire output table to the output modules.

Reading Values from the ENIU after Power up but Before a Write has occurred.

If after power up, an Output Table Read is performed before a Write is done in Modbus mode, the values returned for modules configured for Default may not be the same as the Real output values (which were set from the Default table values). The Output Table and Real Outputs will remain at these initial values until changed by the first Write.

Reading Values from the ENIU after an Upset, but Before a Write has occurred.

This applies to the situation where power continues to be applied to the ENIU, but a communications upset occurs in Modbus mode. If, after the communications upset, an Output Table Read is performed before a Write is done, the Output Table values returned may not be the same as the Real Output values for any discrete modules configured for “Default.”

Warning

For both discrete and analog Output or Mixed modules, if the ENIU loses power but the Output or Mixed modules do not lose power, the Hold Last State parameter does not work as might be expected. Upon ENIU power loss, outputs configured for Hold Last State go to the Last State value on modules that still have power. When power is restored to the ENIU, outputs go to the Default value. When communications are reestablished and the ENIU receives a Write message, outputs go to the values in the ENIU’s Output Table.

EGD Operation

Operation at Power up

Default is used until the first communication, then the exchange values are used.

Operation on Upset

Default or Hold Last State used. On first communication, exchange values are used.

Maximum Data Transfer per Message in Modbus Mode

Reading Inputs (Modbus Mode)

The ENIU only responds to direct requests from clients. All values, both input and output, can be requested. Messages sent from the ENIU to the client are limited to 250 bytes of data per request as per the Modbus TCP specification. Read requests are for contiguous values.

Writing Outputs (Modbus Mode)

The maximum overall length of output message data sent to the ENIU from the client is 200 bytes per message as per the Modbus TCP specification. As noted in the following caution message, the first message to the ENIU after power up or after Ethernet connection loss, cable removal, or communication error should contain all discrete and analog outputs used in the I/O map. This can be accomplished by using a Write Multiple Registers command that writes to bytes that overlap both the discrete and analog tables of the ENIU.

Caution

It is very important that the first output message contain all discrete and analog outputs used in the I/O map; otherwise, spurious outputs will be produced in the time between the first message releasing the output tables to the modules and the arrival of the second message.

How to write to all Discrete and Analog Outputs in a Single Message

The segments for Discrete Outputs and Analog Outputs are contiguous in memory. By using only the higher numbered Discrete Outputs and the lower numbered Analog Output references, a contiguous block of memory that contains all the outputs and will fit in a single write can be achieved. The example in the following table shows how this can be done.

No. of Discrete	No. of Analog	%Q Range	%AI Range	Write Start Address	Length (Bytes)
256	32	%Q1793-2048	AQ1-32	368	96

Maximum Data Transfer per Message in EGD Mode

Reading Inputs – (all input values contained in each message)

The ENIU has one produced exchange that contains all of the configured discrete and analog input module data. The maximum configurable discrete input module data is 2048 bits or 256 bytes and the maximum configurable analog input data is 128 words or 256 bytes for a combined total of 512 bytes. There is an additional 4 bytes of status information at the beginning of the data in the exchange for a total of 516 total bytes allowed in one produced exchange.

Writing Outputs – (all output values, %Q & %AQ, are contained in each message)

The ENIU has one consumed exchange that contains all of the configured discrete and analog output module data. The maximum configurable discrete output module data is 2048 bits or 256 bytes and the maximum configurable analog output data is 128 words or 256 bytes for a combined total of 512 bytes. There is an additional 4 bytes of control information at the beginning of the data in the exchange for a total of 516 total bytes allowed in one consumed exchange.

Maximum Number of I/O for ENIU Release 1.1

Maximum No. of I/O Modules per Rack	8
Maximum No. of I/O Racks	8 (1 local + 7 expansion)
Maximum No. of Discrete I/O Points ¹	2048 (8 x 8 x 32)
Maximum No. of Analog I/O Channels ²	256 (8 x 8 x 4) - this figure is equal to 128 in and 128 out.

- Notes:**
1. Currently, the maximum discrete point count for a VersaMax module is 32
 2. Starting with release 1.1, the ENIU supports high density analog modules.

NOTE: A full system of all %AI and all %AQ will not fit in the %AI or %AQ table, each of which is limited to 128.

Caution

In Modbus mode, reading the output tables (%Q and %AQ) may not return the actual state of the outputs if defaults are active.

Chapter 2

Installation

This chapter gives basic installation instructions. For more information, please refer to the *VersaMax Modules, Power Supplies, and Carriers Manual*, GFK-1504.

This chapter is divided into two sections:

Section 1 – Installation Instructions

Discusses mounting (and removing) the various components, wiring the system, and powering up the system

Section 2 – Setting the IP Address

Discusses various methods of setting temporary and non-volatile working IP addresses for the ENIU.

Section 1 – Installation Instructions

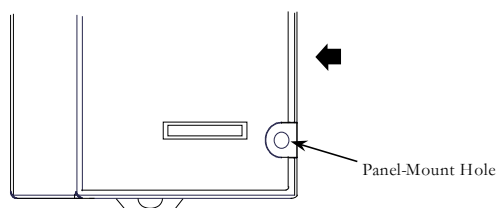
Mounting Instructions

All VersaMax modules and carriers in the same PLC “rack” must be installed on the same section of 7.5mm x 35mm DIN-rail, which must be electrically grounded to provide EMC protection. The rail must have a conductive (unpainted) corrosion-resistant finish. DIN-rails compliant with DIN EN50032 are preferred.

For vibration resistance, the DIN-rail should be installed on a panel using screws spaced approximately 5.24cm (6 inches) apart. DIN-rail clamps (available as part number IC200ACC313) can also be installed at both ends of the station to lock the modules in position.

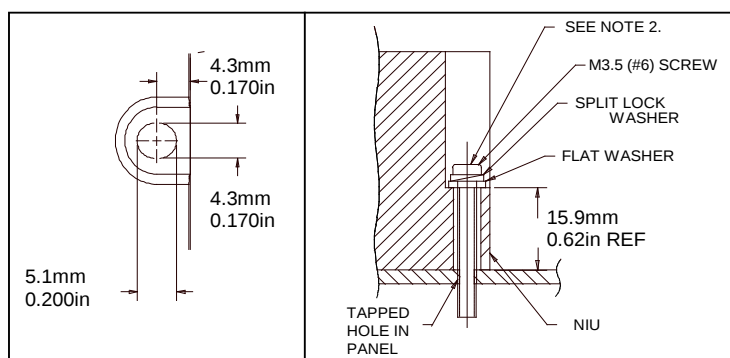
Panel-Mounting

For maximum resistance to mechanical vibration and shock, the DIN-rail-mounted module must also be installed on a panel. Using the module as a template, mark the location of the module’s panel-mount hole on the panel. Drill and tap an M3.5 (6-32) hole in the panel at the marked location. Install the module using an M3.5 (#6) screw in the panel-mount hole.



Note 1. Tolerances on all dimensions are +/-0.13mm (0.005in) non-cumulative.

Note 2. 1.1-1.4Nm (10-12 in/lbs) of torque should be applied to M3.5 (#6-32) steel screw threaded into material containing internal threads and having a minimum thickness of 2.4mm (0.093in). See figure below.

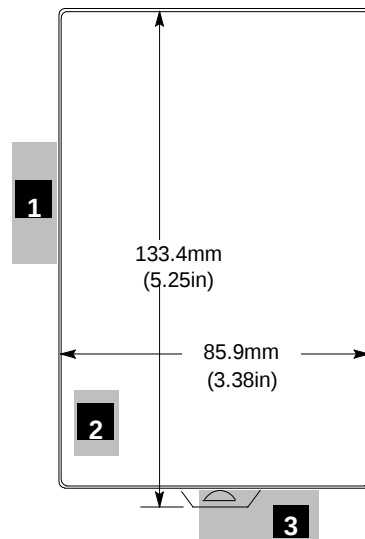


Installing an Ethernet Network Interface Unit

If the I/O Station will have more than one expansion rack or one expansion rack that uses an Isolated Expansion Receiver Module (IC200ERM001) as its interface to the expansion bus, an Expansion Transmitter Module must be installed to the left of the NIU. The Expansion Transmitter Module must be installed on the same section of DIN-rail as the rest of the modules in the main “rack” (rack 0). See the instructions for installing expansion equipment at the end of this chapter.

Clearance Required for the ENIU

Maintain a clearance of 2 inches (5.1cm) above and below the equipment and 1 inch (2.54cm) to the left. Additional clearance requirements are shown below.

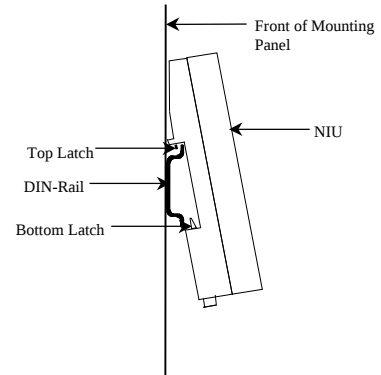


(The numbered items below refer to the numbered areas in the figure above.)

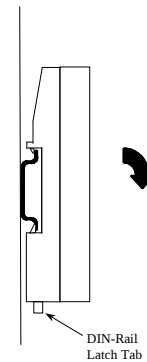
1. Allow sufficient finger clearance for opening ENIU rotary switch door.
2. Allow adequate clearance for communications cable, which connects to the RJ-45 jack in area 2.
3. Allow adequate space for power supply wiring. The power supply mounts on the front of the ENIU on the right-hand side. Also, allow sufficient space in front of the ENIU for the depth of the power supply.

Installing the ENIU on the DIN-rail

1. Turn off power to the system
2. Place the stationary top latch of the ENIU over the top edge of the DIN-rail.



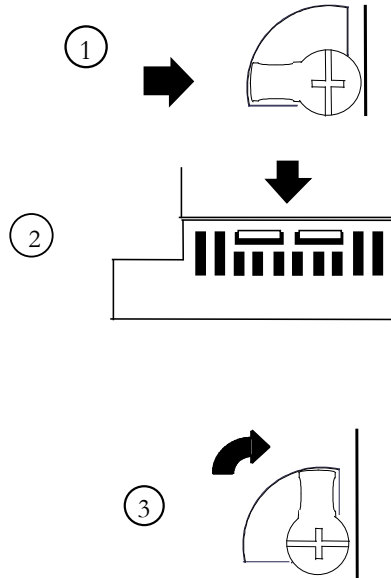
3. Pivot the ENIU downward until the ENIU's spring-loaded bottom latch snaps around the bottom edge of the DIN-rail.



Removing the ENIU from the DIN-rail

1. Turn off power to the power supply.
2. If the ENIU is attached to the panel with a screw, remove the power supply module and then remove the panel-mount screw.
3. Slide the ENIU to the left along the DIN-rail away from the other modules until its carrier connector disengages from the carrier on its right. (If the ENIU will not slide freely along the DIN-rail, pull downward on the DIN-rail latch tab with a small screwdriver while sliding the ENIU.)
4. Use a small flathead screwdriver to pull the DIN-rail latch tab downward, then pull the bottom of the ENIU forward until it clears the bottom edge of the DIN-rail. Finally, lift the ENIU off the top edge of the DIN-rail.

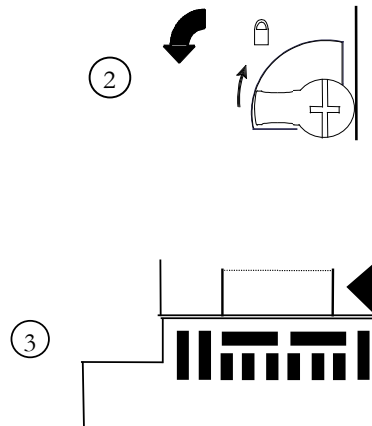
Installing the Power Supply on the ENIU



1. The power supply module installs directly on top of the ENIU. The latch on the power supply must be in the unlocked position as shown in the picture on the left.
2. Align the connectors and the latch post and press the power supply module down firmly, until the two tabs on the bottom of the power supply click into place. Be sure the tabs are fully inserted in the holes in the bottom edge of the ENIU.
3. Turn the latch to the locked position to secure the power supply to the top of the ENIU module.

Removing the Power Supply from the ENIU

Exercise care when working around operating equipment. Devices may become very hot and could cause injury.

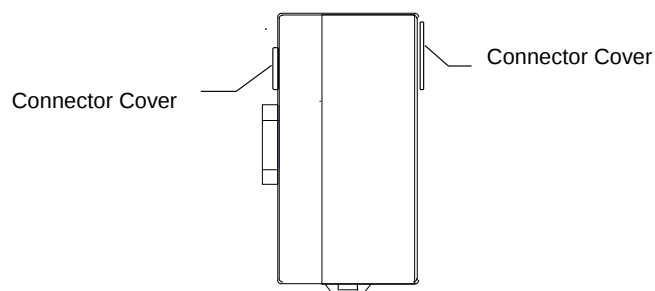


1. Remove power.
2. Turn the latch to the unlocked position as illustrated.
3. Press the flexible panel on the lower edge of the power supply to disengage the tabs on the power supply from the holes in the carrier.
4. Pull the power supply straight off.

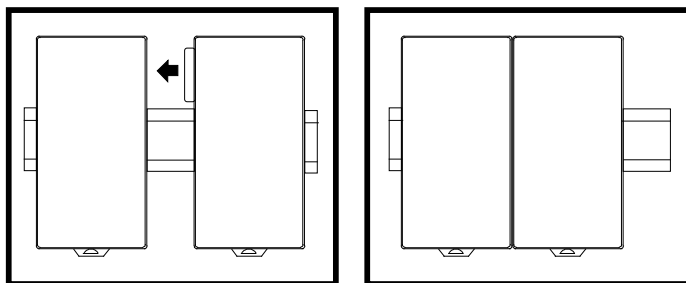
Installing Additional Modules

Before joining carriers to the ENIU, remove the connector cover on the right-hand side of the ENIU. Do not discard this cover; you will need to install it on the last carrier. It protects the connector pins from damage and ESD during handling and use.

Do not remove the connector cover on the left-hand side.



Install additional modules by mounting modules on their carriers and sliding them along the DIN-rail to fully engage the connectors in the sides of the carriers.



Installing the Ethernet Cable

The 10Base-T / 100Base-TX RJ-45 port on the Ethernet NIU connects directly to a network without an external transceiver. Connect the port to an external 10Base-T / 100Base-TX hub or switch using a twisted pair cable. Category 5 cable is required for 100Base-TX operation. 10Base-T / 100Base-TX cables are readily available from commercial distributors. GE recommends purchasing rather than making cables. Cables must meet the applicable IEEE 802.3 or 802.3u standard, noted in the table below.

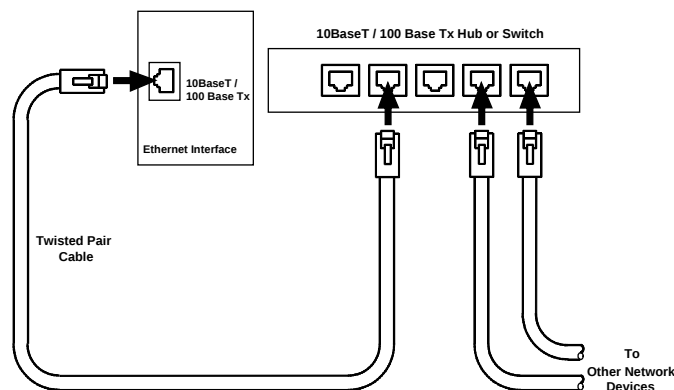
The Ethernet NIU automatically senses whether it is connected to a 10BaseT or 100BaseTX network, and whether communications are half-duplex or full duplex.

NOTE: We recommend you use Category 5 cable even if using 10Base-T because (1) it is a higher quality cable, (2) it will support a later system upgrade to 100Base-TX, and (3) cable cost is low compared to installation labor cost.

Unshielded Twisted Pair (UTP) Ethernet Cables				
Cable Category	Rating	Max. Drop Length	Standard	Suitable for:
Category 5	100 Mbits/sec.	100 Meters	IEEE 802.3u	100Base-TX 10Base-T
Category 4	20 Mbits/sec.	100 Meters	IEEE 802.3	10Base-T
Category 3	16 Mbits/sec.	100 Meters	IEEE 802.3	10Base-T

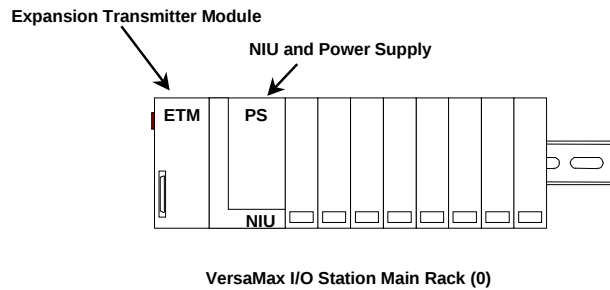
Network Connection

Connection of the Ethernet Interface to a 10Base-T or 100Base-TX network is shown below (each cable drop can be up to 100 meters long):

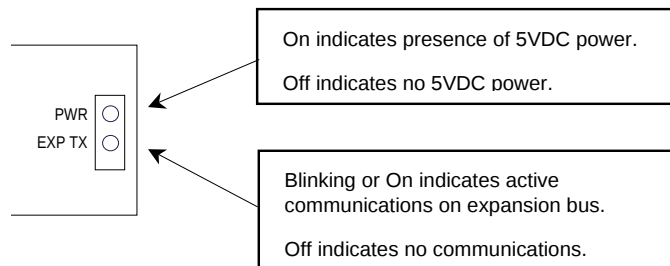


Installing an Expansion Transmitter Module

If the I/O Station will have more than one expansion rack or one expansion rack that uses an Isolated Expansion Receiver Module (IC200ERM001) as its interface to the expansion bus, an Expansion Transmitter Module must be installed to the left of the NIU. The Expansion Transmitter Module must be installed on the same section of DIN-rail as the rest of the modules in the main “rack” (rack 0).



1. Make sure rack power is off.
2. Attach the Expansion Transmitter to DIN-rail to the left of the NIU position.
3. Install the NIU as instructed. Connect the modules and press them together until the connectors are mated.
4. After completing any additional system installation steps, apply power and observe the module LEDs.



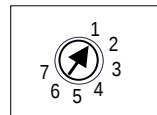
Removing an Expansion Transmitter Module

1. Make sure rack power is off.
2. Slide module on DIN-rail away from the NIU in the main rack.
3. Using a small screwdriver, pull down on the tab on the bottom of the module and lift the module off the DIN-rail.

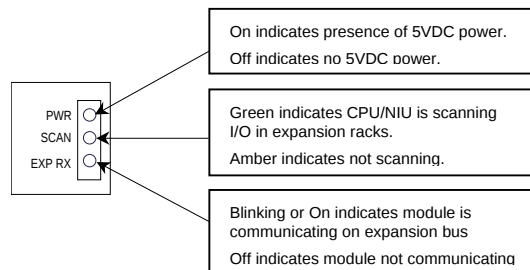
Installing an Expansion Receiver Module

An Expansion Receiver Module (IC200ERM001 or 002) must be installed in the leftmost slot of each VersaMax expansion “rack”.

1. Insert the label inside the access door at the upper left corner of the module.
2. Attach the module to the DIN-rail at the left end of the expansion rack.
3. Select the expansion rack ID (1 to 7) using the rotary switch under the access door at upper left corner of the module.



4. Install the Power Supply module on top of the Expansion Receiver.
5. Attach the cables. If the system includes an Expansion Transmitter Module, attach the terminator plug to the EXP2 port on the last Expansion Receiver Module.
6. After completing any additional system installation steps, apply power and observe the module LEDs.



Removing an Expansion Receiver Module

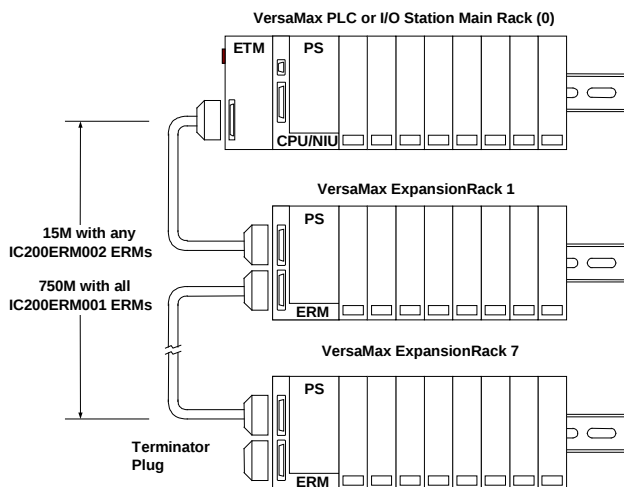
1. Make sure rack power is off.
2. Separate the Power Supply module from the Expansion Receiver Module.
3. Slide the Expansion Receiver Module on DIN-rail away from the other modules.
4. Using a small screwdriver, pull down on the tab on the bottom of the module and lift the module off the DIN-rail.

Expansion Rack Power Sources

Power for module operation comes from the Power Supply installed on the Expansion Receiver Module. If the expansion rack includes any Power Supply Booster Carrier and additional rack Power Supply, it must be tied to the same source as the Power Supply on the Expansion Receiver Module.

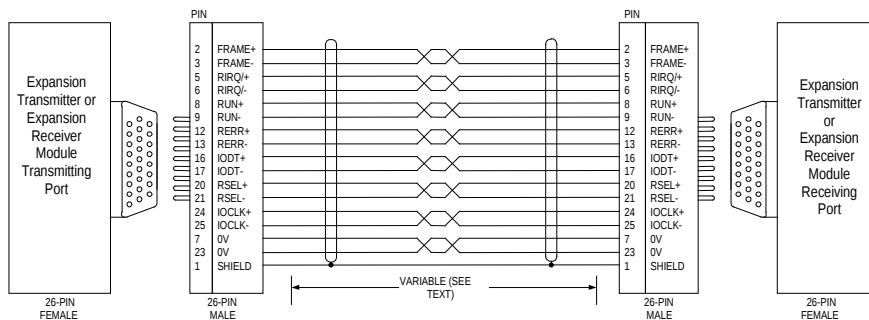
Connecting the Expansion Cable: RS-485 Differential

For a multiple-rack expansion system, connect the cable from the expansion port on the Expansion Transmitter (ETM) to the Expansion Receivers (ERM) as shown below. If all the Expansion Receivers are the Isolated type (IC200ERM001), the maximum overall cable length is 750 meters. If the expansion bus includes any non-isolated Expansion Receivers (IC200ERM002), the maximum overall cable length is 15 meters.



Install the Terminator Plug (supplied with the Expansion Transmitter module) into the lower port on the last Expansion Receiver. Spare Terminator Plugs can be purchased separately as part number IC200ACC201 (Qty 2).

RS-485 Differential Inter-Rack Connection (IC200CBL601, 602, 615)



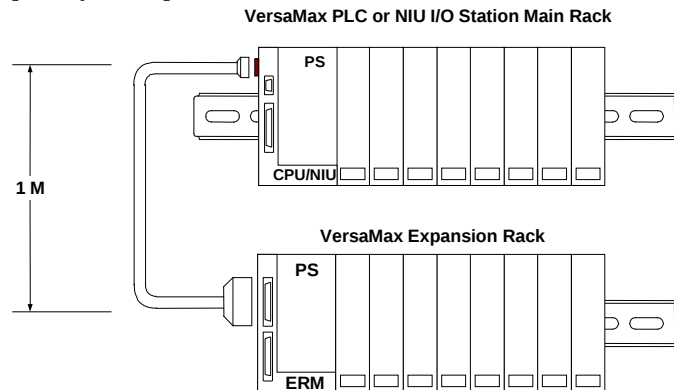
Building a Custom Expansion Cable

Custom expansion cables can be built using Connector Kit IC200ACC202, Crimper AMP 90800-1, and Belden 8138, Manhattan/CDT M2483, Alpha 3498C, or equivalent AWG #24 (0.22mm²) cable.

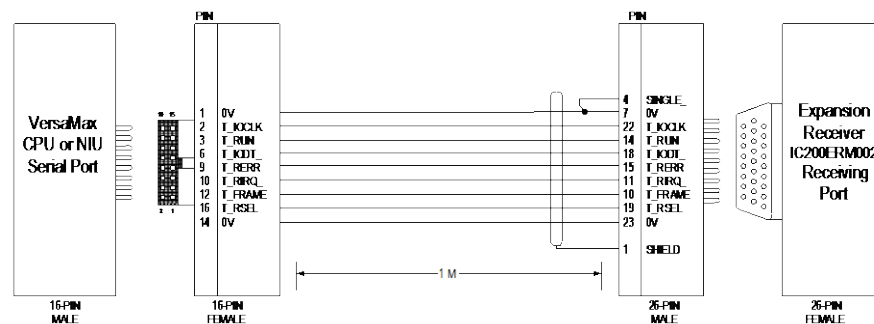
Connecting the Expansion Cable: Single-ended

For a system with one non-isolated expansion rack (IC200ERM002) and no Expansion Transmitter, connect the expansion cable from the serial port on the VersaMax NIU to the Expansion Receiver (ERM) as shown below. The maximum cable length is one meter. Cables cannot be fabricated for this type of installation; cable IC200CBL600 must be ordered separately.

NOTE: No Terminator Plug is needed in a single-ended installation; however, it will not impede system operation if installed.



Single-Ended Inter-Rack Connection (IC200CBL600)



Power Sources for Single-Ended Expansion Rack Systems

When operating the system in single-ended mode, the power supplies for the main rack and expansion rack must be fed from the same main power source. The main rack and expansion racks cannot be switched ON and OFF separately; either both must be ON or both must be OFF for proper operation.

Power for module operation comes from the Power Supply installed on the Expansion Receiver Module. If the expansion rack includes any Power Supply Booster Carrier and additional rack Power Supply, it must be tied to the same source as the Power Supply on the Expansion Receiver Module.

Powering up the ENIU

When power is applied to the ENIU, you should observe the following conditions if there are no problems:

- PWR LED should be ON steady
- OK LED should be ON steady
- FAULTS LED should be OFF
- LAN LED should be flashing or ON steady if network communications are occurring.
- STAT LED should be ON if at least one Modbus TCP master is connected or if EGD consumption is active; otherwise it should be OFF.

If there is an abnormal indication, please refer to Chapter 5 for troubleshooting information.

CE Mark Installation Requirements

The following requirements for surge, electrostatic discharge (ESD), and fast transient burst (FTB) protection must be met for applications that require CE Mark listing:

- The VersaMax I/O Station is considered to be open equipment and should therefore be installed in an enclosure (IP54).
- This equipment is intended for use in typical industrial environments that utilize anti-static materials such as concrete or wood flooring. If the equipment is used in an environment that contains static material, such as carpets, personnel should discharge themselves by touching a safely grounded surface before accessing the equipment.
- If the AC mains are used to provide power for I/O, these lines should be suppressed prior to distribution to the I/O so that immunity levels for the I/O are not exceeded. Suppression for the AC I/O power can be made using line-rated MOVs that are connected line-to-line, as well as line-to-ground. A good high-frequency ground connection must be made to the line-to-ground MOVs.
- AC or DC power sources less than 50V are assumed to be derived locally from the AC mains. The length of the wires between these power sources and the PLC should be less than a maximum of approximately 10 meters.
- Installation must be indoors with primary facility surge protection on the incoming AC power lines.

Section 2 – Setting the Network IP Address

The IP Address of the ENIU needs to be permanently set by “Storing” a configuration to the ENIU from VersaPro or the Remote I/O config tool. The configuration is “Stored” over Ethernet.

The ENIU needs to be given a temporary IP address to allow “Storing” of the permanent IP Address.

There are two methods for setting a temporary IP Address

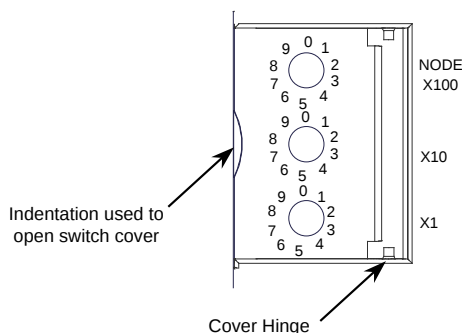
- Use the rotary switches on the ENIU (this only works if the ENIU does not have a permanent IP address stored to it (a new ENIU).
- Create an entry in the ARP table of your PC that sets an IP address for the MAC Address of the ENIU, then Telnet to Port 1 which causes the ENIU to temporarily use the IP address. A utility to do this is available on the GE Support website.

Note: If your network will be connected to another network, you must set an address that will be compatible with the other network. In that case, you must obtain a unique IP address for the Ethernet NIU from your system administrator.

Initially Setting the Network IP Address using the rotary switches (only works if no IP address is set)

Open the clear protective hinged cover by pulling forward at the indentation in the left side of the ENIU (see figure below). Use a 2.44mm (3/32in) flat screwdriver to adjust the rotary switches. **Always cycle power after changing the switch settings.**

These switches, marked Node X100, X10 and X1 select the hundreds, tens, and units digits of the last octet of the network IP address. Select any valid address in the range 1-254. The full IP address will be 195.0.0.X, with the X being set by the rotary switches.



At this point, the subnet mask defaults to 255.255.255.0 and there is no configured gateway IP address. So, to communicate to the ENIU (which now has an IP address of 195.0.0.x), an Ethernet device or the programmer must also have a unique IP address in the range of 195.0.0.X , which places it in the same subnet as the ENIU.

See Chapter 3 for details on setting up communications parameters and storing the configuration to the ENIU.

Once a software configuration that contains a valid IP address (0.0.0.0 should not be used) is stored to the ENIU, the IP address setting on the rotary switches is ignored.

Forcing a Temporary Network IP Address

If there are no Modbus™ TCP connections, EGD consumption is not active and the programmer is not connected, the IP address can be temporarily forced to any valid IP address. Forcing the IP address will place the ENIU at the specified address with a default subnet mask matching the class of the forced IP address and no configured gateway IP address.

In the following procedure, an example is given to illustrate the commands used. The example shows the use of the Windows Telnet and ARP (Address Resolution Protocol) utility programs from a Windows® 95/98/ME/NT™ command prompt to assign an IP address of 3.16.27.5 to an ENIU that has a MAC address of 08-00-19-01-48-64. (NOTE: MAC address is in hex characters.)

Note that if your computer is connected to the same network (subnet) as the ENIU, they both must have the same network identification (net id) to be able to communicate (see Appendix B for further information). The computer's IP address in this example is 3.16.88.139. To force a specific IP address, use the following procedure:

- 1) Determine the Media Access Control (MAC) address of the ENIU. The ENIU's MAC Address is laser marked on the ENIU's plastic case and is of the form 080019xxxxxx. If the MAC address can not be read from the ENIU's case, see the section "Determining the MAC Address of the ENIU" later in this chapter for an alternate method.
- 2) On the programmer computer, we will use the Windows ARP utility program (ARP.exe) to set the IP address.
- 3) Start the MS-DOS® command prompt from the Windows Start/Programs menu.
- 4) Determine the IP address of your computer. You can use two possible methods: (1) use the following commands at the command prompt: WINIPCFG in Windows 95/98/ME or IPCONFIG in Windows NT/2000 or (2) look it up in the Windows Start/Settings/Control Panel/Network dialog box

under Protocol (TCP/IP) Properties. The configuration of your computer system will determine which of these methods will work successfully for you.

- 5) At the MS-DOS prompt, type in the following, then press the Enter key:

arp -s (IP address you want to assign to ENIU) (MAC address of the ENIU) (IP address of your computer)

For this example we will use the following:

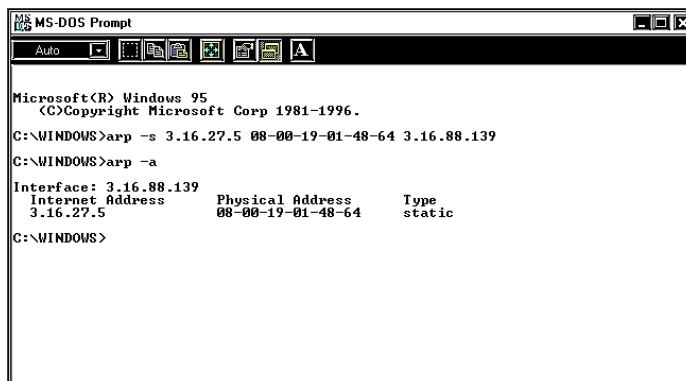
arp -s 3.16.27.5 08-00-19-01-48-64 3.16.88.139

NOTE: You will not see any reply on the screen (see next figure).

- 6) To verify that the ARP table entry was accepted, type the following, then press the Enter key:

arp -a

An entry matching the desired ENIU IP address (“Internet Address”) and MAC address (“Physical Address”) should be seen in the table, with the “Type” listed as 'static'. This is shown in the following figure:



```

Microsoft(R) Windows 95
(C)Copyright Microsoft Corp 1981-1996.
C:\WINDOWS>arp -s 3.16.27.5 08-00-19-01-48-64 3.16.88.139
C:\WINDOWS>arp -a
Interface: 3.16.88.139
Internet Address      Physical Address      Type
3.16.27.5             08-00-19-01-48-64    static
C:\WINDOWS>

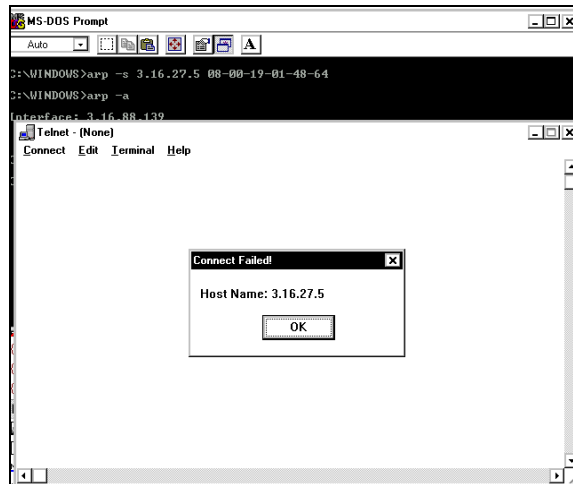
```

- 7) Next, create a Telnet connection to the ENIU (Port 1) by typing in the following command at the computer’s MS-DOS prompt, then pressing the Enter key:

telnet (IP Address) 1

For this example, the command is **telnet 3.16.27.5 1**

A Telnet window will appear. After several seconds, a “Connect Failed!” dialog box will appear (see next figure); regardless, the ENIU will change its IP address to the one designated in the Telnet command. (If it takes more than 15 seconds for the Connect Failed box to appear, the Telnet command probably didn’t work.) Note: don’t wait too long to do the Telnet command because the ARP table static entry will time out in two minutes and the Telnet command will not work.



At this point, the forced IP address is **temporary** and will be lost if the ENIU is power-cycled. The following steps show how to make this IP address permanent.

- 8) Configure the network settings of the ENIU to the final desired values using the VersaPro or Remote I/O Manager configuration software. See the configuration software documentation or on-line help for detailed instructions.
- 9) Store the configuration to the ENIU, using the forced IP address in your communications settings. The stored settings will be permanent. See Chapter 3 for details on communications settings and instructions on storing a configuration.

Notes

1. *The forced IP address is lost after every power reset of the ENIU. Be sure to store the ENIU configuration parameters using the configuration software as detailed above to make the IP address permanent.*
2. *NOTE: Static ARP table entries are kept in your computer's ARP table, but dynamic entries are automatically cleared after about two minutes. If you wish to remove the static listing once the ENIU has been configured with the configuration program, type "arp -d [IP_ADDRESS]" from the command prompt. Note that all ARP table entries are removed automatically when your computer is shut down.*

Chapter 3

Configuring an Ethernet NIU and I/O Station

This chapter explains how an Ethernet NIU and the modules in an I/O Station can be configured. Configuration determines certain characteristics of module operation and also establishes the program references to be used by each module in the system.

This chapter describes:

- **Using autoconfiguration or programmer configuration**

The Ethernet NIU and I/O Station can be either autoconfigured or configured from a programmer using the Remote I/O Manager configuration software.

- **Configuring racks and slots**

Even though a VersaMax I/O Station does not have a module rack, both autoconfiguration and software configuration use the traditional convention of “racks” and “slots” to identify module locations.

- **Software configuration of the Ethernet NIU and I/O Station**

Software configuration provides greater flexibility than autoconfiguration in setting up an I/O Station. Software configuration is done using VersaPro version 1.5 (or later) or the Remote I/O Manager configuration software. (See Appendix D for a version vs. feature table.)

- **Autoconfiguration of the Ethernet NIU and I/O Station**

Autoconfiguration provides a default configuration for the ENIU in Modbus mode and I/O Station and does not require the use of a programmer. I/O modules that have software-configurable features always use their default settings when autoconfigured.

Autoconfiguration using EGD mode is not supported.

- **Configuring EGD Exchanges and Status/Control Bytes**

Uses examples to show how to configure consumed and produced exchange data including status and control data.

Using Autoconfiguration or Programmer Configuration

The Ethernet NIU and I/O Station can be either (1) autoconfigured in Modbus mode, or (2) configured using the Remote I/O Manager or VersaPro 1.5 (or later) programming software in either Modbus or EGD mode. (See Appendix D for a version vs. feature table.) The choice of which configuration method to use depends on the requirements of the application.

Autoconfiguration (Modbus Mode Only)

Autoconfiguration is done by the ENIU itself. It provides a default configuration for the ENIU and I/O Station and does not require the use of a programmer. If there is not a stored configuration already present at powerup, the ENIU sees which modules are installed and automatically creates a configuration for the I/O Station. I/O modules that have software-configurable features can only use their default settings when the I/O Station is autoconfigured. Under autoconfiguration, discrete output data defaults to 0 while analog output data holds its last state. Under Autoconfiguration EGD functionality is disabled. Autoconfiguration is described later in this chapter.

An autoconfiguration can be used as is for Modbus mode if the default settings for software-configurable features are acceptable. For EGD mode and for Modbus mode where default settings are not acceptable, auto configuration can be used as a starting point, and the default configuration can be loaded to VersaPro or Remote I/O Manager and modified.

Software Configuration

Using the configuration software makes it possible to reassign I/O table addresses and to configure many I/O module features. The configuration software runs on a computer that connects to the ENIU via Ethernet.

The configuration software can be used to:

- Create a customized configuration
- Store (write) a configuration to the ENIU
- Load (read) an existing configuration or autoconfiguration from an ENIU
- Compare the configuration in an ENIU with a configuration file stored in the programmer
- Clear a configuration that was previously stored to the ENIU
- Enable EGD functionality

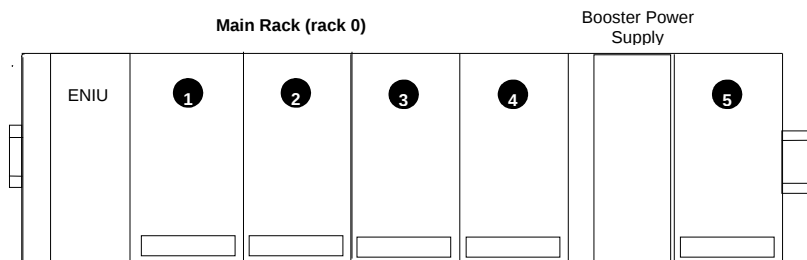
The ENIU retains a software configuration across power cycles. Storing a configuration disables autoconfiguration, so the ENIU will not overwrite the configuration during subsequent startups.

However, clearing a configuration with the programmer does cause a new autoconfiguration to be generated. In that case, autoconfiguration is enabled until a configuration is stored from the programmer again.

Software configuration is summarized later in this chapter. Instructions for installing and using the configuration software are found in the *Remote I/O Manager Software User's Guide* (GFK-1847) or the *VersaPro Programming Software User's Guide* (GFK-1670).

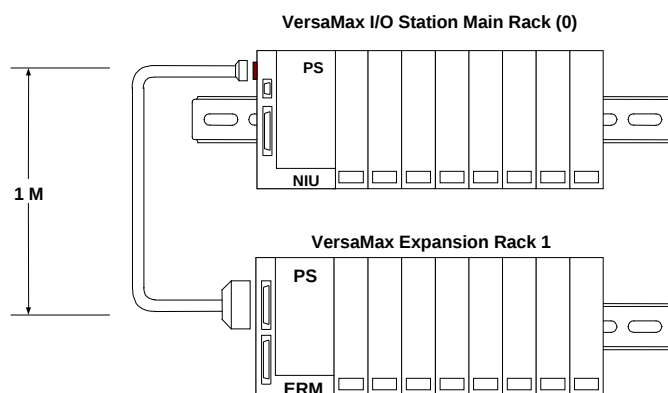
Configuring “Racks” and “Slots”

Even though a VersaMax I/O Station does not have a rack, both autoconfiguration and software configuration use the traditional convention of “racks” and “slots” to identify module locations. Each logical rack consists of the ENIU or an Expansion Receiver module plus up to 8 additional I/O and option modules mounted on the same DIN-rail. Each I/O or option module occupies a “slot”. The module next to the ENIU or Expansion Receiver module is in slot 1. Booster power supplies do not count as occupying slots.

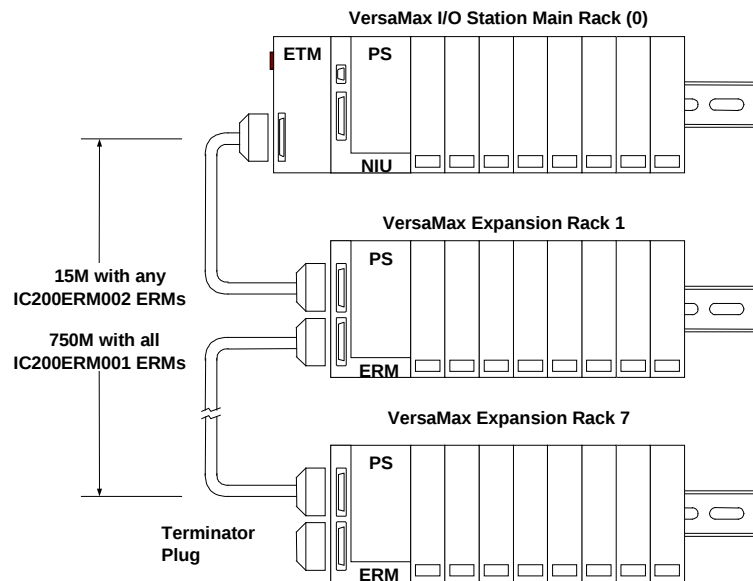


The main rack (containing the ENIU) is always called rack 0.

In an I/O Station that has one expansion rack attached to the expansion bus by a non-isolated Expansion Transmitter Module (IC200ERM002), the expansion rack must be configured as rack 1.



In an I/O Station with an Expansion Transmitter Module (IC200BTM001) and up to seven expansion “racks”, each with an Expansion Receiver Module (IC200ERM001 or IC200ERM002), the additional racks are configured as rack 1 through rack 7.



Configuring I/O References

As I/O modules are added to the configuration, the configuration software keeps a running total of input/output memory. If the modules added consume more than the maximum memory available, the configuration software displays (1) the reference address of the module that caused the error and (2) an error message.

The I/O Station, including all expansion racks, can include up to 1024 total bytes of data.

You can change the I/O references assigned to a module when configuring that module.

Software Configuration of the Ethernet NIU and I/O Station

Software configuration provides greater flexibility than autoconfiguration in setting up an I/O Station. Software configuration is performed using one of the following:

- **VersaPro 2.0 (or later version).** This software allows you to (1) configure all Series 90-30 and VersaMax products, including CPUs and NIUs, and (2) program PLC ladder logic. For details of installing and using this software, refer to the *VersaPro Programming Software User's Guide* (GFK-1670).
- **Remote I/O Manager 2.0 (or later version).** Allows you to configure all Series 90-30 and VersaMax I/O products, including CPUs and NIUs. If you will only be using GE I/O products (no CPUs), you can use this software for configuration purposes. Ladder logic programming is not supported. The software is available as catalog number IC641CFG100. For details of installing and using this software, refer to the *Remote I/O Manager Software User's Guide* (GFK-1847).

Note: The Remote I/O Manager comes bundled with a serial programmer cable if ordered as catalog number IC641CFG110; however, the *Ethernet NIU does not support this serial cable*. The IC641CFG110 package was created for the convenience of Genius and Profibus NIU users, since those two products require the serial cable for software configuration.

The Remote I/O Manager software runs on a computer equipped with Windows 95/98, Windows NT 4.0, or Windows 2000. Note that VersaPro 1.1 and the Remote I/O Manager software cannot be installed on the same computer. If VersaPro 1.1 is present, you will be prompted to un-install it during the Remote I/O Manager installation. If both VersaPro and the Remote I/O Manager software are required, you must upgrade to VersaPro version 1.5 (or later), which includes a built-in version of the Remote I/O Manager. However, VersaPro 2.0 or later is required to support high density analog modules. Please see Appendix D, which contains a feature to software version compatibility matrix.

Notes on Using the Configuration Software

1. The same Remote I/O Manager software can configure different types of VersaMax NIUs and all supported IO modules.
2. Empty slots are allowed in an ENIU software configuration (unlike an autoconfiguration). Note: If empty slots are later filled when using EGD mode, the EGD exchange on the controller will need to be updated.
3. The I/O Station cannot include the following communication modules: IC200BEM002 (Profibus Slave module) and IC200BEM103 (DeviceNet Master module).
4. The reference addresses assigned to modules in the I/O Station can be edited. Addresses do not need to be consecutive.

Basic Steps of Software Configuration

The Remote I/O Manager software provides a simple default configuration that you edit to match the actual system modules. The default configuration consists of a power supply (PWR001 – see Note below) and a CPU001. Carriers and modules are then added in the same sequence as the hardware installation.

Note

The default PWR001 power supply is not powerful enough for as ENIU system. Therefore, you must use one of the higher rated VersaMax power supplies such as the IC200PWR002.

The basic configuration steps are listed below.

- Configure the expansion rack system (local, local single rack, multiple remote rack). This automatically adds the appropriate types of expansion modules to the racks.
- Configure the power supply type and any booster power supplies and carriers.
- Configure the ENIU. This includes changing the NIU type, if necessary, and assigning its parameters as described on the next page.
- Configure the expansion modules if the system has expansion racks.
- Add module carriers and define wiring assignments.
- Place modules on carriers and select their parameters. Configurable parameters of I/O modules are described in the VersaMax Modules, Power Supplies, and Carriers User's Manual (GFK-1504).
- Save the configuration file so that it can be stored to the ENIU.

For step-by-step instructions, please refer to the *Remote I/O Manager Software User's Guide* (GFK-1847).

Configuring ENIU Network Parameters

ENIU Network configuration establishes the basic operating characteristics of the Network Interface Unit. When a programmer is first connected, either the rotary switches or the forced IP address are used as the temporary IP address. The Subnet Mask defaults to match the class of the IP address, and the Gateway IP address defaults to Unused. Once a configuration is stored, the network settings of the ENIU immediately change to those in the stored configuration, and any IP address set with the rotary switches or by the forcing method is ignored.

If your ENIU will be communicating with devices on other networks, the parameters in the following table must be set appropriately. These values should be assigned by the person in charge of your network (the network administrator). See Appendix B for more detailed information about IP Addressing and network setup.

Feature	Description	Config. Default	Choices
IP Address	The IP Address is the unique address of the Ethernet interface as a node on the network.	0.0.0.0	A valid Class A, B, or C address
Subnet Mask	Subnet mask of the ENIU used to identify the section of the overall network the ENIU is on.	0.0.0.0	A valid dotted-notation mask.
Gateway IP Address	IP address of the default gateway (router) device to be used when the ENIU is unable to locate the desired remote device on the local sub-network.	0.0.0.0	A valid Class A, B, or C address in the same subnet as the ENIU.

Selecting EGD or Modbus

Configuring Modbus Parameters

There are no parameters that are specific to the Modbus protocol. There are, however, three EGD parameters that must be set correctly to disable EGD and enable Modbus. The following table shows how to enable Modbus:

Tab	Feature	Values*
Produced Exchange	Exchange Type	IP Address
Produced Exchange	Consumer IP Address	0.0.0.0
Consumed Exchange	Producer ID	0.0.0.0

* These are the default settings for these parameters

If any of these parameters are not set, the ENIU assumes EGD is trying to be configured and will verify the remaining EGD parameters.

Configuring EGD Parameters

To use EGD on the ENIU, the EGD configuration parameters must be set correctly. The following tables show these parameters and their acceptable values.

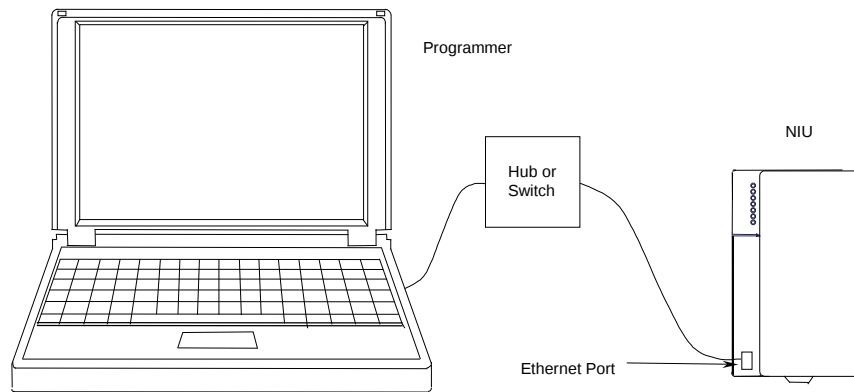
Produced Exchange Parameters			
Feature	Description	Config. Default	Choices
Exchange ID	Uniquely identifies a particular exchange on a specific producing node.	1	Cannot be changed from default of 1
Exchange Type	Determines whether the exchange will be produced directly to an IP address or to a group address (multicast).	IP Address	IP Address or Group ID
Consumer IP Address	The destination IP address of the exchange. The Consumer IP Address field will only be present if the Exchange Type is IP Address	0.0.0.0	0.0.0.0 to 255.255.255.255 0.0.0.0 is only valid in Modbus mode
Group ID	The destination group of the exchange. The Group ID field will only be present if the Exchange Type is Group ID	1	1 To 32
Producer Period (in ms)	The period at which the exchange is produced on the network	200	5 To 3,600,000

Consumed Exchange Parameters			
Feature	Description	Config. Default	Choices
Exchange ID	Uniquely identifies a particular exchange on a specific producing node	0	1 To 16383
Producer ID	Uniquely identifies the producer of an exchange on a give network	0.0.0.0	0.0.0.0 to 255.255.255.254 0.0.0.0 is only valid in Modbus mode
Group ID	The group IP address the exchange is to be consumed from	0	0 to 32 0 Means no group
Consumed Period (in ms)	The period at which the exchange is to be consumed	200	5 To 3,600,000
Update Timeout (in ms)	The time allowed for the consumption of an exchange before it is deemed inactive. Upon timeout, outputs go to their Default or Hold Last State values.	2000	10 To 3,600,000

Software Configuration: Load, Store, Verify, Clear

To transfer and check the contents of a configuration, use the Load/Store/Verify functions from the Tools menu. A configuration file must be saved in the programmer (computer) before using the load/store/verify functions.

The computer connects to the Ethernet NIU via the Ethernet network.

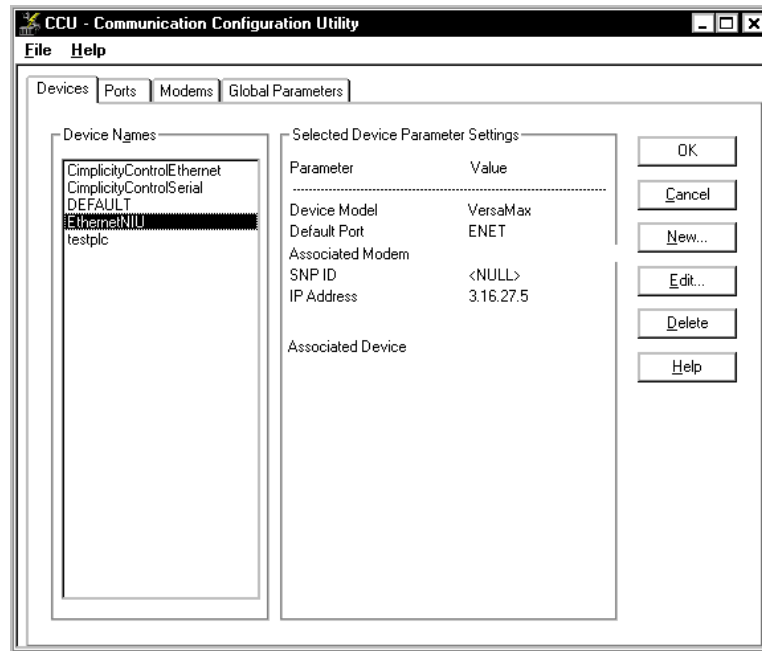


To use the Load/Store/Verify functions, the programmer must be able to communicate with the ENIU. The configuration software has a set of communications parameters that need to be correctly set for communicating with the Ethernet NIU. The following section discusses this:

Creating a New Communications Setup

In VersaPro or Remote I/O Manager, create a communications setup for the Ethernet NIU using the ENIU's IP address. Use these steps:

1. First, choose Communications Setup from the Tools menu. The Communication Configuration Utility dialog box will appear, shown below.
2. Second, click the New button and create your new setup. Configure the following parameters:
 - Enter a Device Name. In the example shown below, the Device Name of the newly-created setup is "EthernetNIU":
 - Device Model as *VersaMax* (Optional)
 - Default Port as *ENET*
 - IP address as the address of your ENIU (3.16.27.5 in the example)
 - Leave the other parameters blank



Modifying the ENIU Communications Settings

If your communications parameter settings are not correct, you can change them in the Communications Configuration Utility dialog box, shown in the previous figure. Click the Device Name of your ENIU communications setup to select it (it is called EthernetNIU in the previous figure). Click the Edit button to make changes. Make sure that:

- The IP address is set to the ENIU's current IP address
- The Default Port is ENET
- The Device Model is VersaMax (optional)

Note

The programmer must be able to communicate with the ENIU at the current ENIU settings. This means that if the IP address is set to 195.0.0.X, the Subnet Mask is at 255.255.255.0, and there is no Gateway IP address, the programmer must be at a different 195.0.0.X address in order to communicate with the ENIU.

Storing a Configuration to the Ethernet NIU

After completing a configuration in the programmer (and saving it in the programmer), the configuration must be *Stored* to the Ethernet NIU. In the Tools menu, select Load/Store/Verify and click on Store. Since you cannot store a configuration to the ENIU while it is communicating with a controller, you must take the controller off-line to perform the store.

Storing a software configuration disables autoconfiguration, so the ENIU will not overwrite a software configuration with an autoconfiguration during subsequent startups.

If any mismatched, missing, or extra modules are detected, the ENIU turns on its FAULTS LED.

The ENIU FAULTS LED will flash a repeating fatal error code (two blinks, pause, five blinks, pause, repeat) and will autoconfigure on a power cycle for either of the following conditions:

- If the ENIU rejects the configuration due to corrupted or unacceptable data
- If the Ethernet network is disconnected during the store operation

Please see Chapter 6 for information on troubleshooting these errors.

Loading a Configuration from the ENIU to the Programmer

The programming software can *Load* a previously stored configuration from the Ethernet NIU back to the programmer. In the Tools menu, select Load/Store/Verify and click Load.

Note that the following modules share hardware module IDs:

IC200MDL650 loads as IC200MDL636
IC200MDL750 loads as IC200MDL742
IC200MDL331 loads as IC200MDL329
IC200MDD844 loads as IC200MDD842
IC200MDL141 loads as IC200MDL140

If an Autoconfiguration containing any of the above modules is loaded, the software may display an incorrect catalog number and description for them. Edit any incorrect modules using the programmer before storing the configuration back to the ENIU. Once this has been done, you will be able to load the configuration properly.

Comparing Configurations in the Programmer and ENIU

Use the *Verify* function to compare a configuration file in the programmer with a configuration that was previously stored to the Ethernet NIU. In the Tools menu, select Load/Store/Verify and click Verify.

Deleting a Software Configuration from the ENIU

Use the software's *Clear* function to remove a previously stored configuration from the ENIU. Clearing a software configuration causes a new autoconfiguration to be generated. Autoconfiguration remains enabled until a software configuration is stored from the programmer again. A clear function will not erase the network settings of the ENIU. After a *Clear* function, the ENIU will remain at the same IP address with the same subnet mask and gateway IP address. To invalidate the IP address and revert back to using the ENIU's rotary switches, store an IP address of 0.0.0.0 to the ENIU.

NOTE 1: If an IP address of 0.0.0.0 is stored to the ENIU and then the configuration is cleared, the ENIU will cease referring to the rotary switches (unless a new configuration is stored). The IP address of the ENIU will be that of the switch settings at the time of the 0.0.0.0 store. For example if an IP address of 0.0.0.0 is stored to the ENIU and the switches are set so that the IP address is 195.0.0.128 and then a clear is performed, the ENIU will be 195.0.0.128 regardless of the switch settings until a new configuration is stored.

NOTE 2: Clearing the configuration will erase EGD parameter information and cause the ENIU to change to Modbus mode.

Autoconfiguration of the Ethernet NIU and I/O Station

Autoconfiguration is done by the ENIU itself. It provides a default configuration for the ENIU and I/O Station and does not require the use of a programmer. I/O modules that have software-configurable features always use their default settings when autoconfigured. Under autoconfiguration, discrete output data defaults to 0 while analog output data holds its last state.

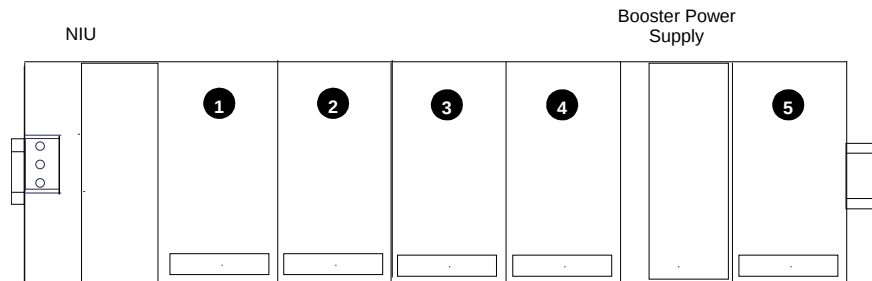
When no previous autoconfiguration exists, the ENIU automatically reads, at powerup, the default configuration of the modules installed in the system.

Once this autoconfiguration is complete as described below, the ENIU retains this configuration until it is either (1) cleared or (2) powered up with a changed I/O module configuration, in which case, a new autoconfiguration will be generated that reflects the changes.

Autoconfiguration disables EGD functionality and sets the ENIU to Modbus mode.

Autoconfiguration Sequence

Each module is considered to occupy a “slot”. The position adjacent to the ENIU is slot #1. Booster power supplies do not count as occupying slots.



Autoconfiguration starts at slot 1 of rack 0 (the main rack) and continues in the same order the modules occupy in the I/O Station.

Autoconfiguration stops at the first empty slot or faulted module. For example, if there are modules in slots 1, 2, 3, 5 and 6 but slot 4 is empty, the modules in slots 5 and 6 are not autoconfigured. The ENIU reports *Extra I/O Module* faults.

Note

For the autoconfiguration process to work as expected, any additional power supplies in the I/O Station must be powered up at “exactly” the same time as or before the main power supply.

Autoconfiguration Assigns Reference Addresses

The ENIU stores data internally as discrete input bits, discrete output bits, analog input words, and analog output words.

The NIU Data Memories	
%I	discrete input bits
%AI	analog input words
%Q	discrete output bits
%AQ	analog output

During autoconfiguration, the ENIU automatically looks at the modules installed in the I/O Station and assigns them to addresses in this internal I/O map. Reference addresses are assigned in ascending order. For modules that have multiple data types (for example, mixed I/O modules), each data type is assigned reference addresses individually. Chapter 4 contains related information on memory mapping and the Modbus Function Codes used to access ENIU memory.

Modules that have software-configurable features use their default settings when autoconfigured. Under autoconfiguration, discrete output data defaults to 0 while analog output data holds its last state. I/O module features are described in the *VersaMax Modules, Power Supplies, and Carriers Manual* (GFK-1504).

Adding I/O modules to an Autoconfigured I/O Station

If additional I/O modules are added to an existing I/O Station, they do not become part of the autoconfiguration until the ENIU is power-cycled.

Clearing an Autoconfiguration

To clear an existing autoconfiguration, power down the ENIU, disconnect the ENIU from the first I/O module, and power up the ENIU. The autoconfiguration in the ENIU is then cleared. (An existing software configuration can only be cleared using the programmer, as described previously in this chapter.)

Hot Inserting I/O Modules

It is possible to hot insert I/O modules in an I/O Station. If the module being replaced already exists in the configuration, no other action is necessary to make the module operable.

Autoconfiguring an I/O Station with Expansion Racks

- The Expansion Receiver modules must have their rack ID selection dials set correctly.
- Any available rack number can be used for a new expansion rack but they must all be unique (no duplicate rack numbers). It is best to assign expansion rack numbers from lowest (1) to highest (7) as they are installed.
- If a new expansion rack is added in the future, it should be assigned a rack number that is higher than the racks that are already installed. If a new expansion rack with a lower rack number is added and the system is then auto-configured, the racks numbered higher than the new rack number have their I/O reference addresses shifted in the reference tables. Any existing program logic using those references would need to be adjusted to use the new references.
- When autoconfiguring an I/O Station with expansion racks, either all racks must be powered from the same source or the expansion racks must be powered up before the main rack.
- To add another expansion rack to the I/O Station, the I/O Station must be powered down. After adding the expansion rack, power up the I/O Station. It will then autoconfigure.
- To force autoconfiguration for expansion racks, first power down the ENIU. Remove the transmitter module from the ENIU or remove the expansion cable at the transmitter. Power up the ENIU and let it autoconfigure. Power the ENIU down again, reattach the transmitter or cable, and power up the ENIU again.

Configuring the ENIU Produced Exchange

The ENIU Produced Exchange contains input module and fault status data. Even if your ENIU system does not use any input modules, you must configure a Produced Exchange for the transmission of fault status data.

Phase 1 – Planning the Exchange

This phase should be done first. Planning the exchange will save time and mistakes during the configuration phase. A worksheet should be filled out and kept as part of your system documentation. Blank worksheets are provided at the end of this section.

ENIU Rack Configuration Example

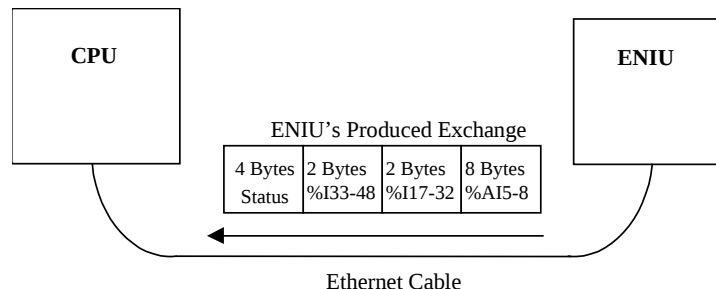
In this example, the ENIU rack is configured as illustrated below.

ENIU	Rack 0 Module 1	Rack 0 Module 2	Rack 0 Module 3	Rack 0 Module 4	Rack 0 Module 5	Rack 0 Module 6
	%Q33-48	%I33-48	%Q17-32	%I17-32	%AQ5-8	%AI5-8

ENIU Produced Exchange Details

An ENIU can only have one Produced Exchange. It automatically configures its own Produced Exchange, which contains only input (%I and %AI) data; the produced exchange is automatically generated from the hardware configuration stored in the ENIU. The ENIU's Produced Exchange data will be sent in the same order that the input modules are configured in the rack, from left to right. For this example, 16 bytes of data will be sent in the following order:

- The first four bytes contain Status data.
 - The next two bytes contains data from Module 2 (%I0033 - %I0048)
 - The next word contains data from Module 4 (%I0017 - %I0032)
 - The final eight bytes contain data from Module 6 (%AI0005 - %AI0008)
- Note that each analog input produces two bytes of data.



ENIU Produced Exchange Memory Mapping Worksheet Example

This example worksheet shows the mapping of CPU memory to ENIU memory. Note that the organization of memory addresses in the ENIU's Produced Exchange is based on the ENIU rack's physical configuration. In automatically configuring its Produced Exchange, the ENIU uses its stored configuration, starting with Slot 1 (from left to right). As it encounters input addresses, it adds them to the exchange in the order in which it encounters them. Therefore, the ENIU's Produced Exchange is not configurable in any way by the user. For example, the first input module the ENIU encounters when scanning its configuration from left to right is the module in slot 2; therefore, the first byte of data the ENIU sends after the Status data will be from the input module in slot 2.

Note that from the CPU's standpoint this is a Consumed Exchange. Addresses in CPU memory must be configured by the user. The first byte of an exchange is called byte 0. All discrete addresses used in an exchange must be byte-aligned.

ENIU Produced Exchange Memory Mapping Worksheet				
Produced by: ENIU			Consumed by: CPU	
Exchange Byte No.	CPU Memory (assigned by user)	Data Type	ENIU Rack Location	ENIU Memory (assigned automatically by ENIU)
0-3	%R0050 - %R0051	Status	N/A	N/A
4-5	%I0033 - %I0048	Discrete	Rack 0, Slot 2	%I0033 - %I0048
6-7	%I0017 - %I0032	Discrete	Rack 0, Slot 4	%I0017 - %I0032
8-15	%AI0005 - %AI0008	Analog	Slot 6, inputs 1-4	%AI0005 - %AI0008

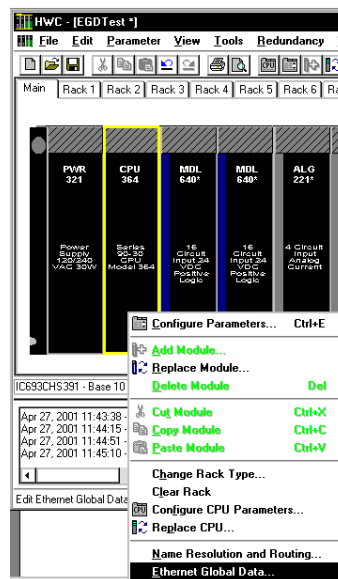
Note

To avoid confusion, we recommend that ENIU memory addresses be mapped to the same addresses in CPU memory; however, this is not a requirement (any applicable CPU memory addresses can be used).

Phase 2 – Configuring the PLC CPU to Receive the ENIU Produced Exchange

This section uses the data from the planning phase in the example shown in the previous section. This example uses an IC693CPU364 (which is a combination CPU/Ethernet Interface module) as the master device in the system that will be exchanging EGD data with the ENIU. The screens in the following figures were copied from VersaPro Release 2.00.

- To begin, open the Hardware Configuration File in VersaPro
- Configure the CPU364 assigning all applicable CPU and Ethernet Interface settings.
- Next, right-click the CPU364 module. A short cut menu will appear as shown in the next figure.



- On the short cut menu, choose Ethernet Global Data. The Ethernet Global Data configuration window will appear.
- Select the Consumed Exchanges tab because the ENIU's Produced exchange becomes the CPU364's Consumed exchange. See the next figure.

Ethernet Global Data

Produced Exchanges Consumed Exchanges

Exchanges:

Exchange ...	Adapter Name	Producer Id	Group Id	Consumed Period	Update Timeout

Ranges for Selected Exchange: Exchange Size in Bytes: 0

Offset	Reference	Low Point	Hi Point	Description

Local Producer: 3.16.32.1

OK Cancel Help

Add Exch. Delete Exch.

Add Range Insert Range Delete Range

- Note that the ENIU only supports one Produced exchange and one Consumed exchange. Begin configuring the CPU's Consumed exchange (to receive the ENIU's Produced exchange) by clicking the Add Exch. button.
- In the Exchange field, you must enter 1 for the Exchange Number. See next figure.

Ethernet Global Data

Produced Exchanges Consumed Exchanges

Exchanges:

Exchange ...	Adapter Name	Producer Id	Group Id	Consumed Period	Update Timeout
1	EGD1	0	0	200	0

Ranges for Selected Exchange: Exchange Size in Bytes: 0

Offset	Reference	Low Point	Hi Point	Description
Status	%I	1	16	Status: This is where the PLC is to put...
Time Stamp	NOT USED			Time Stamp: Optional place for the PL...

Local Producer: 3.16.32.1

OK Cancel Help

Add Exch. Delete Exch.

Add Range Insert Range Delete Range

Exchange Identifier: Enter a unique number to identify the exchange

Local Producer's (CPU364) IP Address

Exchange Table

Range Table

- Configure the following parameters:
 - ☐ **Exchange** Enter the exchange number. The ENIU only produces one exchange, and it must be set to a value of 1.
 - ☐ **Adapter Name** This is the configured Adapter Name of the CPU364. It was configured previously in the CPU configuration window.
 - ☐ **Producer ID** The IP address of the ENIU (3.16.32.2 for this example).
 - ☐ **Group ID** Since the exchange is only being sent to the CPU364 and not to a group, this is set to 0.
 - ☐ **Consumed Period** This field sets how frequently this exchange is produced. The default is 200 ms. This is probably too slow for many applications. We will set it to 30 ms for this example. The lower the value, the more frequently the exchanges will occur; however this also increases network traffic. (Whether the network traffic increase is an issue depends on whether other devices besides the CPU364 and ENIU are using this network. If these are the only two devices on the network, then it's not an issue.)
 - ☐ **Update Timeout** Should be set to a value at least twice the value of the ENIU producer exchange period value. If the CPU364's Ethernet Interface does not receive an exchange from the ENIU within the time set in this parameter, it will declare the occurrence of a refresh error.

Once configured, the Exchange Table should look similar to the following figure.

Exchange ...	Adapter Name	Producer Id	Group Id	Consumed Period	Update Timeout
1	EGD1	3.16.32.2	0	30	70

- Next, configure the Range Table for the CPU364's consumer memory addresses that will be used to store the ENIU's produced data. The data you entered in the ENIU Produced Exchange Memory Mapping Worksheet will be used to configure the Range Table.
 - ☐ Set the Status reference address first (in the first row of the Range Table). You can fill in the Reference field by clicking it and selecting the memory type from the drop-down list.
 - ☐ Click the Low Point field and enter a starting address number for this range. The Hi Point field will be filled in by the software based on the Reference and Low Point values.

- ☐ Enter a Description to help you identify this data range.
- ☐ Click the Add Range button. This will active a row in the Range Table.
- ☐ Click the Reference field and choose a memory type (such as %R, %AI, %I, etc.) from the drop-down menu.
- ☐ Click the Low Point field and enter a start address number for this range. Then click the High Point field and enter an end address number for this range.
- ☐ Click the Description field and enter a description to help identify this data range.
- ☐ Continue to add ranges until the exchange is fully configured. For this example the data from the example worksheet is used. When finished, the Ethernet Global Data window looks like the figure below.

The screenshot shows the 'Ethernet Global Data' window with two tabs: 'Produced Exchanges' and 'Consumed Exchanges'. The 'Produced Exchanges' tab is active, showing a table of exchanges. The first exchange, 'EGD1', has a Producer Id of '3.16.32.2', Group Id of '0', Consumed Period of '30', and Update Timeout of '70'. Below this, the 'Ranges for Selected Exchange' table is shown, listing various data ranges with their offsets, references, low and high points, and descriptions. The 'Status' range is highlighted, showing a reference of '%R' and a description of 'Status: This is where the PLC is to'. Other ranges include 'Time Stamp', 'ENIU Status Data', and 'ENIU Rack 0, Slot 2' through 'Slot 6'. A note at the bottom states 'A Memory Range to be passed in the exchange'.

Exchange ...	Adapter Name	Producer Id	Group Id	Consumed Period	Update Timeout
1	EGD1	3.16.32.2	0	30	70

Offset	Reference	Low Point	Hi Point	Description
Status	%R	501	501	Status: This is where the PLC is to
Time Stamp	NOT USED			Time Stamp: Optional place for the
0.0	%R	50	51	ENIU Status Data
4.0	%I	33	48	ENIU Rack 0, Slot 2
6.0	%I	17	32	ENIU Rack 0, Slot 4
8.0	%AI	5	8	ENIU Rack 0, Slot 6

A Memory Range to be passed in the exchange

Note that the Status word in the first row of the Range Table is for PLC exchange status use and is not part of the exchange.

Configuring the ENIU Consumed Exchange

The ENIU Consumed Exchange contains output module and fault command data from the CPU. Even if your ENIU system does not use any output modules, you must configure a Consumed Exchange for the transmission of fault command data.

Phase 1 – Planning the Exchange

This phase should be done first. Planning the exchange will save time and mistakes during the configuration phase. A worksheet should be filled out and kept as part of your system documentation. Blank worksheets are provided at the end of this section

ENIU Rack Configuration Example

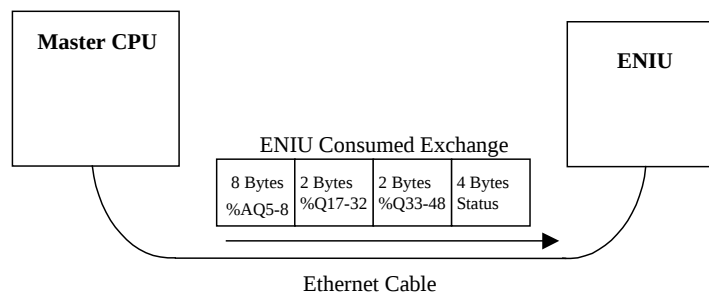
In this example, the ENIU rack is configured as illustrated below

ENIU	Rack 0 Slot 1	Rack 0 Slot 2	Rack 0 Slot 3	Rack 0 Slot 4	Rack 0 Slot 5	Rack 0 Slot 6
	%Q33-48	%I33-48	%Q17-32	%I17-32	%AQ5-8	%AI5-8

ENIU Consumed Exchange Example Overview

The ENIU Consumed Exchange contains only output (%Q and %AQ) data. The exchange data will be automatically written to the output modules in the order in which the output modules are configured in the ENIU rack, from left to right; no user configuration is required at the ENIU end. For this example, the 16 bytes of data consumed will be written in the following order:

- The first four bytes contain Control data that instruct the ENIU how to act on fault data (if faults are present).
- The next two bytes will be written to %Q0033 - %Q0048 (from module in Slot 1)
- The next two bytes will be written to %Q0017 - %Q0032 (from module in Slot 3)
- The final eight bytes will be written to %AQ0005 - %AQ0008 (from module in Slot 5). Note that each analog input requires two bytes in the exchange.



ENIU Consumed Exchange Example Memory Mapping Worksheet

This example worksheet shows the mapping of CPU memory to ENIU memory. Note that the memory addresses in the ENIU are based on the ENIU rack's physical configuration. In automatically writing consumed data to its output modules, the ENIU scans its rack configuration starting with Slot 1, and writes the Consumed Exchange data it received to the output modules in the order in which it encounters them. Therefore, the ENIU exchange addresses are not selectable by the user. For example, the first output module encountered when scanning from left to right is the module in slot 1; therefore, the first byte of data received (other than Control data), will be written to the module in slot 1.

Note that from the CPU's standpoint this is a Produced Exchange. CPU addresses must be configured by the user. The first byte of an exchange is called byte 0. All discrete addresses used in an exchange must be byte-aligned.

ENIU Consumed Exchange Memory Mapping Worksheet				
Produced by: CPU		Consumed by: ENIU		
Exchange Byte No.	CPU Memory (assigned by user)	Data Type	ENIU Rack Location	ENIU Memory (assigned automatically by ENIU)
0-3	%R0200 - %R0201	Status	N/A	N/A
4-5	%Q0033 - %Q0048	Discrete	Slot 1	%Q0033 - %Q0048
6-7	%Q0017 - %Q0032	Discrete	Slot 3	%Q0017 - %Q0032
8-15	%AQ0005 - %AQ0008	Analog	Slot 5, outputs 1-4	%AQ0005 - %AQ0008

Note

To avoid confusion, we recommend that ENIU memory addresses be mapped to the same addresses in CPU memory; however, this is not a requirement (any applicable CPU memory addresses can be used).

Phase 2 - Configuring the CPU to Send to the ENIU's Consumed Exchange

The CPU364's Produced Exchange (which is the ENIU's Consumed Exchange) is configured in a similar manner as the CPU364's Consumed Exchange, described in the previous section, "Configuring the ENIU's Produced Exchange."

- To begin, access the Ethernet Global Data configuration window, as in the previous example. Select the Produced Exchanges tab. See next figure.

Ethernet Global Data

Produced Exchanges | Consumed Exchanges

Exchanges:

Exchange ...	Adapter Name	Cons Type	Cons Addr...	Send Type	Prod Period	Reply Rate

Ranges for Selected Exchange: Exchange Size in Bytes: 0

Offset	Reference	Low Point	Hi Point	Description

Local Producer: 8.16.32.1

Buttons: OK, Cancel, Help, Add Exch, Delete Exch, Add Range, Insert Range, Delete Range

Local Producer Identifier: A unique number (in IP Address Format) to identify the PLC

- Note that the ENIU only supports one Produced Exchange and one Consumed Exchange. Begin configuring the Produced Exchange by clicking the Add Exch button.
- In the Exchange field, you must enter 1 for the Exchange number. See the next figure.

Ethernet Global Data

Produced Exchanges | Consumed Exchanges

Exchanges:

Exchange	Adapter Name	Cons Type	Cons Addr...	Send Type	Prod Period	Reply Rate
EGD1	Group ID			Always	200	0

Local Producer: 3.16.32.1

OK Cancel Help Add Exch Delete Exch

Exchange Table

Range Table

Ranges for Selected Exchange: Exchange Size in Bytes: 0

Offset	Reference	Low Point	Hi Point	Description
Status	21	1	16	Status: This is where the PLC is to put

Add Range Insert Range Delete Range

Exchange Identifier: Enter a unique number to identify the exchange

- Configure the following Produced Exchange fields:
 - **Adapter Name** This field is based on the Adapter Name you already configured for the Ethernet Interface (in the CPU364 configuration window).
 - **Cons Type** This field specifies the consumer type. Three choices are offered: Group ID, IP Address, or Name. For this example, the choice is IP Address, since only the ENIU is to receive this exchange. We will use the ENIU's IP address in the Cons Addr field to identify it as the consumer of the exchange.
 - **Cons Addr** Here we will enter the ENIU's IP address; this is based on our choice for the Cons Type field.
 - **Send Type** This field is not currently configurable. It is "permanently" set to Always.
 - **Prod Period** This field sets how frequently this exchange is produced. The default is 200 ms. This is probably too slow for many applications. We will set it to 30 ms for this example, which is the same value we set for the Consumed Period in the Consumed Exchange in the previous section. Setting these to the same value is recommended for best efficiency.

Note

The **Prod. Period** parameter sets the network production time and is independent of I/O scan time. For example, if you process changes at a 20 ms rate and you produce at 10 ms, you may still not see every change

if the I/O is scanning at a 25 ms. rate. I/O scan is affected by number and type of I/O modules, TAN configuration, and network loading.

- ☐ **Reply Rate.** This field is not currently used. It is “permanently” set to 0.

Once configured, the Exchange Table should look similar to the following figure.

Exchange ...	Adapter Name	Cons Type	Cons Addr...	Send Type	Prod Period	Reply Rate
1	EGD1	IP Address	3.16.32.2	Always	30	0

- Next, configure the Producer's memory addresses that will be the source of the data shared in the exchange.
 - ☐ Set the Status reference address first (in the first row of the Range Table). You can fill in the Reference field by clicking it and selecting the memory type from the drop-down list.
 - ☐ Click the Low Point field and enter a starting address number for this range. The Hi Point field will be filled in by the software based on the Reference and Low Point values.
 - ☐ Enter a Description to help you remember what the data range is used for.
 - ☐ Click the Add Range button. This will active a row in the Range Table.
 - ☐ Click the Reference field and choose a memory type (such as %R, %AI, %I, etc.) from the drop-down menu.
 - ☐ Click the Low Point field and enter a start address number for this range. Then click the High Point field and enter an end address number for this range.
 - ☐ Click the Description field and enter a description to help identify this data range.
 - ☐ Continue to add ranges until the exchange is fully configured. For this example the data shown in the example worksheet is used. When finished, the Ethernet Global Data window looks like the figure below.

The screenshot shows the 'Ethernet Global Data' window with two tabs: 'Produced Exchanges' and 'Consumed Exchanges'. The 'Produced Exchanges' tab is active, showing a table with one exchange (Exchange 1) configured with Adapter Name 'EGD1', Cons Type 'IP Address', Cons Addr... '3.16.32.2', Send Type 'Always', Prod Period '30', and Reply Rate '0'. Below this, the 'Ranges for Selected Exchange' table is shown for Exchange 1, with columns: Offset, Reference, Low Point, Hi Point, and Description. The ranges are:

Offset	Reference	Low Point	Hi Point	Description
Status	%R	500	500	Status: PLC Internal Status Word
0.0	%R	200	201	ENIU Status Command Words
4.0	%Q	33	48	To ENIU slot 1
6.0	%Q	17	32	To ENIU slot 3
8.0	%AQ	5	8	To ENIU slot 5, outputs 1-4

At the bottom, there is a text field labeled 'A Memory Range to be passed in the exchange'.

Note that the Status word in the first row of the Range Table is for PLC internal status use and is not part of the exchange.

Blank Memory Mapping Continuation Worksheet

If your exchange will not fit on the previous worksheet, copy this blank continuation worksheet and use it to continue your exchange data.

[illegible]

EGD Exchange Status and Control Bytes for Fault Handling

Both ENIU exchanges have four bytes dedicated to Status or Control data. These four bytes are the first data sent in the exchanges. This section describes how to configure this portion of the exchanges. In the exchanges produced by the ENIU, the status bytes contain fault status information from the ENIU. In the exchanges consumed by the ENIU, the status bytes contain Control commands from the master device.

Control Bytes for Consumed Exchange

For the ENIU's Consumed Exchange, only Control Byte 1 is active; the other three bytes are always set to 0. Control Byte 1 is used to acknowledge and clear ENIU faults.

Control Byte 1	7	6	5	4	3	2	1	0
	CLR	Reserved (always 0)						ACK

Bit(s)	Value	Meaning
0	0 or 1	Fault acknowledged command. When this bit changes from 0 to 1, the ENIU updates the ENIU status data to contain the next fault. If there is no next fault, the ENIU clears the ENIU status data.
1-6	always 0	Reserved (always 0)
7	0 or 1	Clear all faults command. When this bit changes from 0 to 1, the ENIU clears its internal fault table. It also turns off its FAULT LED unless a new fault is immediately logged or an existing fault condition continues to exist.

Control Byte 2	7	6	5	4	3	2	1	0
	Reserved (always 0)							

Control Byte 3	7	6	5	4	3	2	1	0
	Reserved (always 0)							

Control Byte 4	7	6	5	4	3	2	1	0
	Reserved (always 0)							

Consumed Exchange Control Byte Examples

Value of Control Byte 1	Description
0	No faults acknowledged (NACK). The ENIU takes no resulting action.
Changes from 0 to 1	Indicates that the last fault sent by the ENIU is acknowledged (ACK) by the master. As a result, the ENIU sends the next fault in its fault table. If no additional faults exist, the ENIU sends the "NO FAULT" indication. Note that the ENIU does not clear any faults due to an acknowledgement; use the Clear Fault Table (128) command to clear the fault table.
128	Commands the ENIU to clear its fault table.
Changes from 1 to 0	No fault is acknowledged by the master. If the ENIU sent a fault in its previous exchange, it will send the same fault again. If the ENIU has sent all of its faults and they have all been acknowledged, it will send a NO FAULT status in its next exchange.

Status Bytes for ENIU's Produced Exchange

The first four bytes of the ENIU's produced exchange are used to report fault status information from the ENIU to the master device. The first three bytes refer to the location of the fault. The fourth byte contains the status code. A table of fault status codes can be found in Chapter 6. If a fault exists, bit 7 of Status Byte 0 will be set to 1 (value = 128). If no fault exists, bit 7 of Status Byte 0 will be set to 0. If the ENIU is consuming exchanges, bit 6 of Status Byte 0 will be set to 0; if not consuming, bit 6 will be set to 1.

Status Byte 0

7	6	5	4	3	2	1	0
FLT	NC	Reserved (all bits=0)			Rack Number (0-7)		

Bit(s)	Value	Meaning
0-2	0-7	The physical "rack" location of the faulted I/O module. The value 0 refers to the ENIU main rack.
3-5	0	Reserved
6	0 or 1	0 = Normal operation – consuming. 1 = Not consuming (NC)
7	0 or 1	0 = no fault data present. The remaining fields in bytes 0-3 may be ignored. 1 = a fault is present. The remaining fields in bytes 0-3 provide the rack, slot, point, and fault code of the fault.

Status Byte 1

7	6	5	4	3	2	1	0
Reserved (all bits=0)				Slot Number (0-8)			

Bit(s)	Value	Meaning
0-3	0-8	The "slot" location of the faulted I/O module. The value 0 refers to the ENIU itself.
4-7	0	Reserved

Status Byte 2

7	6	5	4	3	2	1	0
Point Number(0-63)							

Bit(s)	Value	Meaning
0-7	0, 1-63	If no faults exist (Bytes 1 and 2 = 0), this byte will default to 0. If faults exist (Bytes 1 and 2 indicate faulted rack and slot location) and this byte = 0, it indicates a module-level fault on the module indicated by Bytes 1 and 2. If this byte equals a value between 1 and 63 inclusive, it indicates the faulted point number on the module indicated by Bytes 1 and 2.

Status Byte 3

7	6	5	4	3	2	1	0
Fault code (0-63)							

Bit(s)	Value	Meaning
0-7	0-63	See Chapter 6 for a list of the fault codes.

Examples

The following examples show the first four bytes (Status bytes) of the ENIU produced exchange and the interpretation of the values.

Binary	Decimal	Meaning
1000000 00000100 00000001 00000101	128,4,1,5	Module in Rack 0, Slot 4 has a blown fuse fault (fault code 5) on point 1
10000001 00000110 00000010 00001011	129,6,2,11	Module in Rack 1, Slot 6 has a high alarm fault (fault code 11) on point 2
01000000 00000000 00000000 00000000	64,0,0,0	ENIU is not consuming
00000000 00000000 00000000 00000000	0,0,0,0	There are no faults

Fault Handling Example (User Programming Required)

In the example below, the ENIU detects three faults after it sends its second produced exchange.

- **Master Exchanges 1, 2, and 3.** Master's Control Byte is 0 since it has not received any fault data from the ENIU.
- **ENIU Exchanges 1 and 2.** ENIU detects no faults, so sends a NO FAULT status.
- **ENIU Exchange 3.** ENIU has detected three faults, so it sends the data for Fault #1.
- **Master Exchange 4.** The Master changes its Control Byte status from 0 to 1, acknowledging receipt of the ENIU's Fault #1 data.
- **ENIU Exchange 4.** In response, ENIU sends Fault #2 data in its Status bytes.
- **Master Exchange 5.** In response, the Master resets its Control Byte to logic 0.
- **ENIU Exchange 5.** The Master didn't acknowledge, so ENIU sends Fault #2 data again.
- **Master Exchange 6.** The Master acknowledges by setting its Control Byte to logic 1.
- **ENIU Exchange 6.** In response to this acknowledgement, the ENIU sends Fault #3 data.
- **Master Exchange 7.** In response, the Master resets its Control Byte to logic 0.
- **ENIU Exchange 7.** The Master didn't acknowledge, so ENIU sends Fault #3 data again.
- **Master Exchange 8.** The Master acknowledges by setting its Control Byte to logic 1.
- **ENIU Exchange 8.** ENIU recognizes acknowledgement, but has no more faults, so sends the NO FAULT status, and continues to do so until another fault is detected.
- **Master Exchange 9.** Master sends the Clear Fault Table command (128), which causes the ENIU to clear its fault table and turn off its FAULT LED.

Master		ENIU		
Produced Exchange No.	Control Data	Status Data	Produced Exchange No.	FAULT LED
1	NACK (0)			OFF
		← NO FAULT	1	OFF
2	NACK (0)			OFF
		← NO FAULT	2	OFF
3	NACK (0)			ON
		← Fault #1 Data	3	ON
4	ACK (1)			ON
		← Fault #2 Data	4	ON
5	NACK (0)			ON
		← Fault #2 Data	5	ON
6	ACK (1)			ON
		← Fault #3 Data	6	ON
7	NACK (0)			ON
		← Fault #3 Data	7	ON
8	ACK (1)			ON
		← NO FAULT	8	ON
9	CLEAR (128)			OFF

Chapter 4

Modbus

This section describes the implementation of Modbus/TCP communications on the Ethernet NIU.

- Modbus Protocol
 - Modbus Connections
 - Modbus Port
 - Modbus Conformance Classes
 - Modbus Message Format
- Modbus Tables
 - Register/Input Register Table
 - Input Discrete Table
 - Coil Table
- Supported Function Codes
 - Read Multiple Registers
 - Write Multiple Registers
 - Read Coils
 - Read Input Discretes
 - Read Input Registers
 - Write Coil
 - Write Single Register
 - Read Exception Status

Modbus Protocol

The Ethernet NIU supports a Modbus/TCP server with the following features:

Modbus Connections

The Ethernet NIU supports up to 10 simultaneous connections. This allows for the fast re-establishment of a connection. If a Modbus connection is lost, an immediate attempt at reconnecting by the client will be successful.

Modbus Port

Modbus communication on the Ethernet NIU is supported on the Modbus industry standard port 502.

Modbus Conformance Classes

The Ethernet NIU supports Modbus Conformance classes 0 and 1.

Modbus Message Format

The Modbus/TCP protocol has a specific message format as follows:

Table 4-1. Modbus Message Format

BYTE 0 – 1	Transaction Identifier – unique ID generated by the client.
BYTE 2 – 3	Protocol Identifier = 0
BYTE 4	Length Field (upper byte) = 0 (all msgs < 256)
BYTE 5	Length Field (lower byte) = number of bytes following
BYTE 6	Unit Identifier
BYTE 7	Modbus Function Code
BYTE 8 on	Data as needed

NOTE: The ‘CRC-16’ or ‘LRC’ check fields normally associated with Modbus are not needed in Modbus/TCP since the TCP/IP and link layer checksum mechanisms are used to verify packet delivery.

Modbus Byte Order

MODBUS uses a ‘big-endian’ representation for addresses and data items. This means that when a numerical quantity larger than a single byte is transmitted (as in a word or double-word), the MOST significant byte is sent first. So, for example:

The quantity 0x1234 would be transmitted in the order 0x12 0x34

The quantity 0x12345678L would be transmitted in the order 0x12 0x34 0x56 0x78

Modbus Bit Order

If a series of bits is read as a register, such as %I1 to %I16, the highest numbered bit (%I16 in this example) is the least significant, and the lowest numbered bit (%I1 in this example) is the most significant.

Modbus Tables

The Modbus protocol's reference table definition is different from the internal structure of the Ethernet NIU tables. Modbus refers to a Register, Input Register, Input Discrete and Coil table; the Ethernet NIU refers to a Discrete Input (%I), Discrete Output (%Q), Analog Input (%AI), Analog Output (%AQ) and Fault Table. The following table shows how each Modbus table has been mapped to the ENIU tables. Note that everything in this table refers to physical memory inside the ENIU. In effect, ENIU memory has been given Modbus names. For example, if we issue a Read Inputs Discrete command to read inputs in the Modbus Input Discrete table, we are actually reading from the ENIU's %I internal table, which is mapped to the Modbus Input Discrete table.

Table 4-2. Modbus Reference Tables

Modbus Register Table	Modbus Input Register Table	Modbus Input Discrete Table	Modbus Coil Table	ENIU Internal Tables
0 – 127 (16-bit words)	0 – 127 (16-bit words)	0 – 2047 (bits)	---	%I1 – 2048 (bits)
128 – 255 (16-bit words)	128 – 255 (16-bit words)	---	---	%AI1 – 128 (16-bit words)
256 – 383 (16-bit words)	256 – 383 (16-bit words)	---	0 – 2047 (bits)	%Q1 – 2048 (bits)
384 – 511 (16-bit words)	384 – 511 (16-bit words)	---	---	%AQ1 – 128 (16-bit words)
1024 – 1087 (16-bit words)	1024 – 1087 (16-bit words)	---	---	Fault Table (32 Faults x two 16-bit words per fault)

Modbus Register/Input Register Table

The Ethernet NIU makes no distinction between the Modbus Register and Modbus Input Register tables. The Modbus Register and Modbus Input Register tables are identically mapped to all four NIU I/O tables as well as the Fault table.

Applicable Functions

- Read Multiple Registers

All NIU tables can be read with this function.

- Write Multiple Registers

Only the %Q and %AQ memory types may be written to with this function.

- Read Input Registers

All NIU tables can be read with this function.

- Write Single Register

Only the %Q, %AQ, and Fault tables may be written to with this function.

Modbus Input Discrete Table

The Modbus Input Discrete table is mapped exclusively to the NIU Discrete Input (%I) table.

Applicable Functions

- Read Input Discretes

Modbus Coil Table

The Modbus Coil table is mapped exclusively to the NIU Discrete Output (%Q) table.

Applicable Functions

- Read Coils
- Write Coil

Supported Function Codes

The Function Codes are defined for the Modbus memory mapping, so to determine what area of NIU memory is affected, refer back to Table 4-2, which maps the Modbus table names to their corresponding NIU table names.

The Ethernet NIU supports the following Modbus function codes:

- Read Multiple Registers (Function Code 3)
- Write Multiple Registers (Function Code 16)
- Read Coils (Function Code 1)
- Read Input Discretes (Function Code 2)
- Read Input Registers (Function Code 4)
- Write Coil (Function Code 5)
- Write Single Register (Function Code 6)
- Read Exception Status (Function Code 7)

NOTE: The following function request and response message descriptions start with the Modbus Function Code (byte 0 is actually byte 7 of the Modbus message format). See Table 4-1, “Modbus Message Format.”

Read Multiple Registers

This command reads from 1 to 125 16-bit words from the Modbus Register table. Any part of the Modbus Register table can be read from using this function. When reading from the fault table, however, the entire fault table must be read.

The Read Multiple Registers request is in the following form:

BYTE 0	Function Code = 3
BYTE 1-2	Register Table Offset
BYTE 3-4	Word Count (1-125)

The Read Multiple Registers response is in the following form:

BYTE 0	Function Code = 3
BYTE 1	Byte Count of response ($B = 2 \times \text{word count of request}$)
BYTE 2 – (B + 1)	Register Values

If the request accesses an invalid offset or it contains an invalid length, the response is an Exception Response in the following form:

BYTE 0	Function Code = 0x83
BYTE 1	Exception Code = 2

Read Multiple Registers Examples:

Register Table Offset = 0 and word count = 2 returns %I1-32.

Register Table Offset = 383 and word count = 2 returns %Q2033-2048 and %AQ1.

Register Table Offset = 1024 and word count = 64 returns the fault table.

Any combination of Register Table Offset and Word Count that accesses an offset > 511 and < 1024 produces an Exception Response.

Trying to read the fault table and giving a Register Table Offset > 1024 or a Word Count <> 64 produces an Exception Response.

Write Multiple Registers

This command writes from 1 to 100 16-bit words to the Modbus Register table. Only the part of the Modbus Register table mapped to the %Q and %AQ I/O tables may be written to using this function.

The Write Multiple Registers request is in the following form:

BYTE 0	Function Code = 0x10
BYTE 1-2	Register Table Offset
BYTE 3-4	Word Count(1-100)
BYTE 5	Byte Count (B = 2 x word count)
BYTE 6 – (B + 5)	Register Values

The Write Multiple Registers response is in the following form:

BYTE 0	Function Code = 0x10
BYTE 1-2	Register Table Offset (same as request)
BYTE 3-4	Word Count (same as request)

If the request accesses an invalid offset or it contains an invalid length, the response is an Exception Response in the following form:

BYTE 0	Function Code = 0x90
BYTE 1	Exception Code = 2

Write Multiple Registers Examples:

Register Table Offset = 256 and Word Count = 2 writes the Register Values into %Q1-32.

Register Table Offset = 383 and Word Count = 2 writes the Register Values into %Q2033-2048 and %AQ1.

Any combination of Register Table Offset and Word Count that accesses an offset < 256 or > 511 produces an Exception Response.

Read Coils

This command reads from 1 to 2000 bits from the Modbus Coil table.

The Read Coils request is in the following form:

BYTE 0	Function Code = 1
BYTE 1-2	Coil Table Offset
BYTE 3-4	Bit Count (1-2000)

The Read Coils response is in the following form:

BYTE 0	Function Code = 1
BYTE 1	Byte count of response; $B = (\text{bit count of request} + 7)/8$
BYTE 2 – (B + 1)	Bit Values (least significant bit is first coil)

If the request accesses an invalid offset or it contains an invalid length, the response is an Exception Response in the following form:

BYTE 0	Function Code = 0x81
BYTE 1	Exception Code = 2

Read Coils Examples:

Coil Table Offset = 0 and Bit Count = 1 returns coil %Q1.

Coil Table Offset = 0 and Bit Count = 2000 returns coil values %Q1-2000.

Coil Table Offset = 4 and Bit Count = 13 returns coil values %Q5-17

Any combination of Coil Table Offset and Bit Count that accesses an offset > 2047 produces an Exception Response.

Read Input Discretes

This command reads from 1 to 2000 bits from the Modbus Input Discrete table.

The Read Input Discretes request is in the following form:

BYTE 0	Function Code = 2
BYTE 1-2	Input Discrete Table Offset
BYTE 3-4	Bit Count (1-2000)

The Read Coils response is in the following form:

BYTE 0	Function Code = 2
BYTE 1	Byte count of response ($B = (\text{bit count of request} + 7)/8$)
BYTE 2 – (B + 1)	Bit Values (least significant bit is first coil)

If the request accesses an invalid offset or it contains an invalid length, the response is an Exception Response in the following form:

BYTE 0	Function Code = 0x82
BYTE 1	Exception Code = 2

Read Input Discrete Examples:

Input Discrete Table Offset = 0 and Bit Count = 1 returns input discrete %I1.

Input Discrete Table Offset = 0 and Bit Count = 2000 returns input discrete values %I1-2000.

Input Discrete Table Offset = 4 and Bit Count = 13 returns input discrete values %Q5-17

Any combination of Input Discrete Table Offset and Bit Count that accesses an offset > 2047 produces an Exception Response.

Read Input Registers

This command reads from 1 to 125 16-bit words from the Modbus Register table. This command is handled exactly as the Read Multiple Registers command.

The Read Input Registers request is in the following form:

BYTE 0	Function Code = 4
BYTE 1-2	Register Table Offset
BYTE 3-4	Word Count (1-125)

The Read Coils response is in the following form:

BYTE 0	Function Code = 4
BYTE 1	Byte Count of response (B = 2 x word count of request)
BYTE 2 – (B + 1)	Register Values

If the request accesses an invalid offset or it contains an invalid length, the response is an Exception Response in the following form:

BYTE 0	Function Code = 0x84
BYTE 1	Exception Code = 2

Read Input Registers Examples:

See Read Multiple Registers section for examples.

Write Coil

This command writes 1 bit to the Modbus Coil table.

The Write Coil request is in the following form:

BYTE 0	Function Code = 5
BYTE 1-2	Coil Table Offset
BYTE 3	= 0xFF to turn coil ON, = 0 to turn coil OFF
BYTE 4	= 0

The Write Coil response is in the following form:

BYTE 0	Function Code = 5
BYTE 1-2	Coil Table Offset (same as request)
BYTE 3	= 0xFF to turn coil ON, = 0 to turn coil OFF (same as request)
BYTE 4	= 0

If the request accesses an invalid offset, the response is an Exception Response in the following form:

BYTE 0	Function Code = 0x85
BYTE 1	Exception Code = 2

Write Coil Examples:

Coil Table Offset = 0 and value = 0xFF turns coil %Q1 ON.

Coil Table Offset = 0 and value = 0 turns coil %Q1 OFF.

Any Coil Table Offset > 2047 produces an Exception Response.

Write Single Register

This command writes one 16-bit word to the Modbus Register table. Only the part of the Modbus Register table mapped to the %Q and %AQ I/O tables or the first word of the Fault table may be written to using this function.

The Write Single Register request is in the following form:

BYTE 0	Function Code = 6
BYTE 1-2	Register Table Offset
BYTE 3-4	Register Value

The Write Single Register response is in the following form:

BYTE 0	Function Code = 6
BYTE 1-2	Register Table Offset (same as request)
BYTE 3-4	Register Value (same as request)

If the request accesses an invalid offset, the response is an Exception Response in the following form:

BYTE 0	Function Code = 0x86
BYTE 1	Exception Code = 2

Write Single Register Examples:

Register Table Offset = 256 writes the register value into %Q1-32.

Register Table Offset = 384 writes the register value into %AQ1.

Register Table Offset = 1024 and register value = 0 clears the Fault table.

Any Register Table Offset < 256 or (> 511 and < 1024) or > 1024 produces an Exception Response.

Read Exception Status

This command reads one 8-bit status of the Ethernet NIU.

The Read Exception Status request is in the following form:

BYTE 0	Function Code = 7
--------	-------------------

The Read Exception Status response is in the following form:

BYTE 0	Function Code = 7
BYTE 1	Exception Status

Exception Status Data Format

7	6	5	4	3	2	1	0
Unused							Fault Present

Ethernet Global Data

Chapter 5

This section describes the implementation of Ethernet Global Data (EGD) communications on the Ethernet NIU.

- EGD Protocol
- EGD Exchange Definition

EGD Overview

Ethernet Global Data is a protocol of GE that provides efficient connectionless periodic data transfer over an Ethernet network. It operates over the industry standard User Datagram Protocol (UDP). The UDP protocol works at the ISO Transport layer. It supports fast, efficient communications because it is connectionless and is not acknowledged.

Caution

Since Ethernet Global Data (EGD) communications is connectionless and is not acknowledged, error-checking and interlocking circuitry must be designed into the application to ensure the safety of personnel and equipment in the event that EGD data is lost. Failure to heed this warning could result in injury to personnel and damage to equipment.

In EGD communications, a device (called a **producer**) shares a portion of its memory contents periodically with one or more other devices (called **consumers**). This sharing of memory between devices is called an **exchange**.

Exchange Parameters

When an exchange is configured, several key pieces of information must be included:

- Exchange ID. The unique identification of the exchange configuration
- Producer ID. The unique identification of the producing (sending) device
- The area of memory in the producing device to be exchanged
- How frequently the exchange will be produced
- Consumer ID. The unique identification of the consuming (receiving) device or devices
- The area of memory where the received data is to be stored for each consuming device

Note that a device can be configured to be both a producer and a consumer of exchanges.

EGD Protocol

The Ethernet NIU supports Ethernet Global Data communication with the following features:

Number of EGD Exchanges

The Ethernet NIU supports one EGD consumed data exchange and one EGD produced data exchange. The consumed data exchange is comprised of ENIU control data and the discrete and analog module output data being sent to the ENIU. The produced data exchanged is comprised of ENIU status data and the discrete and analog module input data being sent from the ENIU.

EGD Port

EGD communication on the Ethernet NIU is supported on the EGD defined UDP Data Port 18246 (4746H).

EGD Destination Addresses

The Ethernet NIU allows EGD exchanges to be sent to or received from a single destination address (IP Unicast addressing), a group of addresses (IP Multicast addressing), or all EGD nodes (IP Broadcast addressing). NOTE: Unicast addressing is recommended unless there are special requirements. The following table shows the defined Multicast addresses for transmission of data exchanges to a group of nodes:

Table 5-1. EGD Multicast Address Assignments

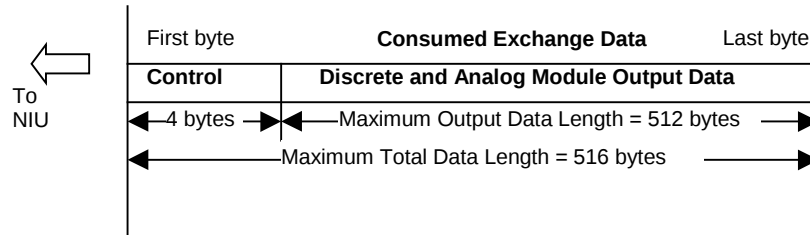
Parameter	IP Address
Group 1	224.0.7.1
Group 2	224.0.7.2
:	:
:	:
Group 32	224.0.7.32

EGD Exchange Definition

As stated earlier the Ethernet NIU supports two EGD exchanges; one consumed, and one produced. The ENIU consumed exchange will be sent from a master to the ENIU, and the ENIU produced exchange will be sent from the ENIU to a master.

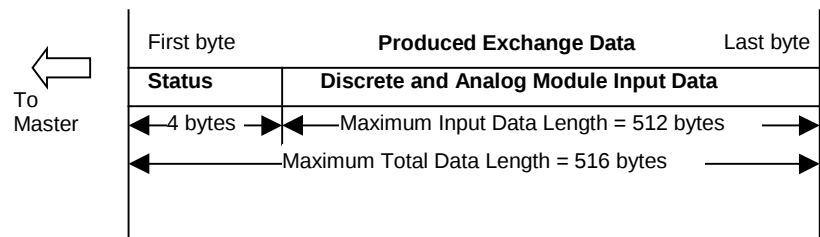
Consumed Exchange

The ENIU consumes one exchange containing all of the output data for discrete output (Q) and analog output (AQ) areas configured in the ENIU's network I/O map. The data is received in the same sequence the modules occupy in the I/O Station. If a single module receives both discrete and analog output data, its discrete data is located before its analog data. The maximum overall length of this output data is 512 bytes. An additional 4 bytes at the start of the data are used by the master for control operations. The total maximum length of the data portion of the exchange is 516 bytes.



The following is an example of how the byte ordering is arranged in an EGD consumed exchange. Please note that the order is based on module order not reference (memory address) order.

Control Data				Discrete and Analog Module Data						
Byte 1	Byte 2	Byte 3	Byte 4	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
ENIU										



The following is an example of how the byte ordering is arranged in an EGD produced exchange. Please note that the order is based on module order and not on reference (memory address) type or order. As implied in the following figure, each analog channel required two bytes of memory.

Status Data				Discrete and Analog Module Exchange Data										
Byte 1	Byte 2	Byte 3	Byte 4	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8	Byte 9	Byte 10	Byte 11
ENIU														
				4-Channel Analog Input Module %AI5				16-Bit Discrete Input Module %I9				8-Bit Discrete Input Module %I1		
				Slot 1				Slot 2				Slot 3		

Notes

If a discrete output does not use a multiple of bytes, extra bits in the byte will be padded with zeros.

If an empty slot is left in the I/O (carrier present, but no I/O module), it will not affect the EGD exchanges. If a module is added later in the empty slot and configured, the ENIU exchange(s) will change to add the new module in the appropriate place in the data. The controller EGD exchange(s) will need to be changed to match.

Chapter 6

Troubleshooting

Overview

This chapter discusses the following topics:

- **Checking status and operation with the ENIU's LEDs.** The ENIU has five LED indicator lights on its front panel that indicate both normal and fault conditions. This section discusses how to interpret these LED indicators.
- **Reading the ENIU's fault codes.** The ENIU has a fault table that stores up to 32 fault codes. This section describes how to read and interpret the fault table codes.
- **Using FTP to obtain network status and version information.** Using the FTP utility to obtain status and version files from the ENIU.
- **EGD Troubleshooting.** Checking common EGD communication problems.
- **Determining the ENIU's MAC address.** The ENIU's MAC address is printed on the front cover of the ENIU. However, if it becomes illegible or you suspect it is incorrect, you can determine it using the method described.
- **Determining the ENIU's IP address.** If the ENIU's stored IP address is not known, you can determine it using the method described.

Checking Status and Operation with the ENIU's LEDs

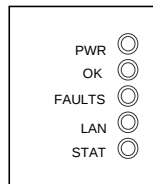
When power is applied to the ENIU, you can verify status and operation by checking the module LEDs.

LED Power up Sequences

- Normally, the PWR LED will light immediately when power is applied. After a self-test, which takes about 5 seconds, the OK LED should light and, if the ENIU has an active network connection, the LAN LED should begin flashing (or could be ON steady if communications traffic is heavy) at that time also.
- If the ENIU's rotary switches are set to either 900 (IP address check) or 901 (MAC address check), upon power up, the PWR LED will light immediately, but the other four LEDs will cycle through an address identification sequence before assuming their normal roles as status indicators. An address identification sequence takes about 45 seconds to complete. For normal operation, set the rotary switches to a number in the range of 000 to 254 (000 would be a good choice).

NOTE: The rotary switches may also be set within the range of 1-254 to set an IP address, as discussed earlier in Chapter 2 in the section "Setting the Network IP Address." Setting the switches in the range of 1-254 will not delay the normal LED status operation at power up.

LED Descriptions



PWR	Green when power is applied to the ENIU.
OK	Green when the ENIU is operational.
FAULTS	If no faults are detected, this LED is OFF Amber if the ENIU has detected a fault with itself or an I/O module. Blinks Amber pattern if the ENIU has encountered a fatal error.
LAN	Green, solid or blinking, when network packets are received or transmitted. Indicates received data addressed to the ENIU as well as network broadcast data (sent to all network devices). OFF when the ENIU senses no communication on the network.
STAT (Modbus Mode)	Green indicates at least one Modbus TCP Master connected. Amber blinking at 1/sec rate indicates an IP address problem (see the next section "IP Address Problems").
STAT (EGD Mode)	Green indicates exchanges are being received. Amber blinking at 1/sec rate indicates an IP address problem (see the next section "IP Address Problems").

Troubleshooting Using LED Indicator Lights

PWR LED

ON

The ENIU's power comes from the power supply that is mounted on the front of the ENIU's case. If power is applied to the power supply, the ENIU's PWR LED should be ON. (NOTE: A power supply with expanded 3.3V capability is required by the ENIU, such as the IC200PWR002 model.)

OFF

If this LED is OFF, check the following

- Check that the input power to the power supply is present.
- If input power is present, try substituting a known good power supply.
- If the power supply checks OK, the problem may be a defective ENIU, or a problem with one of the modules connected to the ENIU. Turn off input power to the power supply, and then separate the ENIU from all of the connected modules. Now try powering up the ENIU again. If the PWR LED is still OFF, the problem is in the ENIU. If the PWR LED is ON, the problem is in one of the other modules. Reconnect them one-by-one to determine which one is "killing" the power supply.

OK LED

ON

Normally, upon power up, the ENIU will conduct a series of self-tests. If these tests are passed, the ENIU turns on its OK LED. This LED should normally come on within a few seconds after power is applied.

OFF

If this LED is OFF, it means the ENIU did not pass its power-up self-tests and has an internal problem.

FAULTS LED

OFF

This LED should be OFF for normal operation.

ON Steady or Flashing

If this LED is not OFF, check the following

- If **ON steady**, the ENIU has detected a non-fatal fault. The fault may be related to the ENIU or to a connected module. Check the ENIU fault table to determine which fault or faults are present. See the section "ENIU Fault Table" for details.

- If **FLASHING** with amber (yellow) color, a fatal fault is present. The ENIU will be disabled by the presence of this fatal fault. The flashing will be a repeating 2,5 pattern (flashes twice, pauses, flashes five times, pauses, repeats). This can indicate either a software error (very unlikely) or, more commonly, a configuration error. A configuration error occurs if a configuration being stored from the programmer is rejected as unusable (see next topic “Causes for Configuration Store Fault”), or if a network disconnect occurs during a configuration store. There are a number of rules that are checked on a configuration. It is suggest that if a user is having problems with a store being rejected, autoconfigure the ENIU into a temporary folder and alter one setting at a time until the problem is isolated.

NOTE: When a store is rejected or a disconnect occurs during a store, a power cycle is required to restart the ENIU and it will autoconfigure after the power cycle.

Causes for Configuration Store Fault

The following rules are likely causes of a configuration being rejected:

General Configuration Problems

- Subnet mask of 0.0.0.0 or 255.255.255.255
- Gateway IP address of 0 in subnet or broadcast address (unless 0.0.0.0)
- Gateway IP not in subnet (unless 0.0.0.0)

EGD Configuration Problems

If (1) the produced exchange type is changed from the default IP Address, or (2) the produced IP is changed from the default of 0.0.0.0, or (3) the consumed IP is changed from the default of 0.0.0.0, then it is assumed the device is to run EGD instead of Modbus TCP, and the following rules are checked during configuration store:

- If produced exchange type is IP, the produced IP address must not be 0.0.0.0
- If produced exchange type is Group, group ID must be between 1 and 32 inclusive
- Consumed exchange cannot be 0.0.0.0 or 255.255.255.255.
- Consumed group ID cannot be above 32

NOTE: If EGD is configured, both consumption and production must be configured, even if one side is not used. It is suggested that the time periods be adjusted for the unused side to minimize network traffic. However, even if one side is “not used,” its exchange still contains four bytes of Fault or Control Status data.

Example: If the EGD ENIU system is used only for outputs (Consumed exchanges), the Produced exchange must be configured for the stored configuration to be accepted, but the production period can be set at the default of 200 ms.

LAN LED

Flashing or ON Steady

This LED will be flashing or appear to be ON steady to indicate the ENIU is communicating over the network.

OFF

If this LED is OFF, it indicates that the ENIU is not communicating over the network. Check the following

- Determine if there are any other operational devices connected to the network. Check the STAT LED (see “STAT LED”).
- Ensure that the Ethernet cable is connected firmly in the ENIU’s RJ-45 connector. Also, this cable may be defective; try substituting another cable. Also check the other end of the cable to ensure that the device it connects to (hub, switch, etc.) is connected and powered up.
- There may be a problem with some other network device or in the network configuration (ensure that the ENIU’s IP address and MAC address were configured correctly in the applicable network devices).
- The ENIU’s IP address may be invalid. Check to see if the STAT LED is flashing (see the “STAT LED” section below).
- Even if the ENIU has a valid IP address, it may not be compatible with the network. Ensure that the ENIU’s netid matches the netid of the network/subnetwork it is installed on. Netid information can be found in Appendix B.

STAT LED (Modbus Mode)

ON

This LED will be ON with a green color to indicate the presence of at least one Modbus TCP Master on the network.

OFF

If OFF, it indicates that no Modbus TCP Master device has been detected on the network.

Flashing

If this LED is flashing an amber color at a one per second rate, it indicates an invalid ENIU IP address. Check the following

- If using the rotary switches, you may have the switches set outside the valid range of 1- 254.
- You may have set the ENIU to an IP address that is the same as the base address of a subnet (such as 3.16.32.0).
- You may have set the ENIU to an IP address that is the same as the broadcast address of a subnet (such as 3.16.32.255)

STAT LED (EGD Mode)**ON**

This LED will be ON with a green color to indicate that the ENIU is consuming the EGD exchange and controlling its outputs.

OFF

If OFF, this LED indicates that the ENIU is not consuming EGD exchanges and that its outputs have reverted to their configured states of Hold Last State or Default.

Flashing

If this LED is flashing an amber color at a one per second rate, it indicates an invalid ENIU IP address. Check the following

- If using the rotary switches, you may have the switches set outside the valid range of 1- 254. Note: 900 and 901 are also valid settings.
- You may have set the ENIU to an IP address that is the same as the base address of a subnet (such as 3.16.32.0).
- You may have set the ENIU to an IP address that is the same as the broadcast address of a subnet (such as 3.16.32.255)

ENIU Fault Table

Fault Table Data Format

The ENIU's internal fault table, which can store up to 32 fault codes, can be accessed by a client application using either Modbus or EGD. This internal fault table operates as a First-In-First-Out (FIFO) stack. When fault 33 occurs, fault 1 is dropped from the table. These faults can include both faults provided by the I/O modules and diagnostic information provided by the ENIU itself. Faults may be read as an extension to the normal register table. A client application may request all faults or clear all faults by sending the applicable message to the ENIU.

Each fault is represented by 2 words, starting from reference 1024 in the register table. Whenever a fault is present in the fault table, the FAULTS LED will be on (solid).

All fault table faults are informational only and do not halt operation of the ENIU. The presence of one or more fault table faults will cause the ENIU's Fault LED to turn ON steady. If a fatal fault occurs, the Fault LED will blink ON and OFF.

Modbus Mode

Reading the ENIU Fault Table

The entire fault table can be read using a *Read Multiple Registers* command, starting at the beginning of the fault table (1024), with a length 64 registers. No partial read access of the fault table is permitted. Empty fault table entries are all zeros. Note that faults are shifted so that the most recent fault is always located at offset 1024.

Clearing the ENIU Fault Table

When desired, the client application may clear all faults in the table by using a *Write Single Register* command to write a zero to the first register (1024) of the fault table. No other write to the fault table by a client is permitted.

Fault Table Entries

Each fault is two words in length and is formatted as follows:

Fault Entry (2 Words)		
Word No.	Upper byte	Lower byte
Word 0	Rack Number (0-7)	Slot Number (0-15)
Word 1	Module I/O Point Number (0-63)	Fault Code (see table)

EGD Mode

The top fault in the FIFO fault stack is sent in the status message. The top fault or all faults can be cleared by sending an ACK or clear in the control section of a consumed exchange. See the "EGD Exchange Status and Control Bytes" section of Chapter 3 for details and examples on configuring and handling fault codes in EGD mode.

Fault Table Codes for Modbus and EGD Modes

Fault Table Codes	
Fault Description	Fault Number
UNKNOWN_FAULT	0
CORRUPTED_CONFIGURATION_FAULT	1
UNSUPPORTED_FEATURE_FAULT	2
CONFIG_MISMATCH_FAULT	4
FUSE_BLOWN_FAULT	5
LOSS_OF_IO_MODULE_FAULT	6
ADDITION_OF_IO_MODULE_FAULT	7
EXTRA_IO_MODULE_FAULT	8
LOSS_OF_USER_POWER_FAULT	9
OPEN_WIRE_FAULT	10
HIGH_ALARM_FAULT	11
LOW_ALARM_FAULT	12
OVERRANGE_FAULT	13
UNDERRANGE_FAULT	14
SHORT_CIRCUIT_FAULT	15
NONVOLATILE_STORE_FAULT	16
LOSS_OF_NON_IO_MODULE_FAULT	17
ADDITION_OF_NON_IO_MODULE_FAULT	18
INSUFFICIENT_CONFIG_MEMORY_FAULT	19
MODULE_NOT_CONFIGURED_FAULT	20
INPUT_POINT_FAULT	21
WIRING_FAULT	22
THERMISTOR_FAULT	23
A_TO_D_CONVERTOR_FAULT	24
MAIL_QUEUE_FULL	25
MAIL_LOSS_FAULT	26
MODULE_IN_BOOT_MODE	27
LOSS_OF_RACK_FAULT	28
ADDITION_OF_RACK_FAULT	29
RACK_NOT_CONFIGURED_FAULT	30
LOSS_OF_TAN_XMIT_FAULT	31
ADDITION_OF_TAN_XMIT_FAULT	32
EXTRA_TAN_XMIT_FAULT	33
TAN_SPEED_CHANGE_FAULT	34
LOSS_OF_MOD_UNSUPP_FEATURE	35

Using FTP to Obtain Network Status and Version Information

Network Status

To obtain network status information such as number of packets transmitted and received, number of collisions, etc.

1. Start at an MS-DOS prompt.
2. If using a networked computer, change to a directory where you have write privileges.
3. FTP to the ENIU.
4. Log into the ENIU as **programmer** (all lower case)
5. Enter the word **password** (all lower case) as the password
6. Type **get netstat.txt**. This copies the text file **netstat.txt** into the current directory of your personal computer. This file contains formatted Ethernet statistics.
7. When the operation completes, enter the word **bye** to terminate the FTP session.

Version Information

Use the same procedure as above except substitute the file name **version.txt** for **netstat.txt** in step 6.

EGD Troubleshooting

The following are common EGD communications problems and solutions.

Wrong network setup. Check exchange configuration on both the ENIU and master device. If using a GE CPU, try pinging the ENIU from the Station Manager. Obtain network status from the ENIU (see previous section “Using FTP to Obtain Network Status and Version Information”) and see if UDP packets are being sent and received by the ENIU at appropriate rate for the exchange speed.

Wrong exchange size. EGD exchange length must match expected length. Obtain version information from ENIU (see previous section “Using FTP to Obtain Network Status and Version Information”) to see length of produced and consumed exchange ENIU is expecting. Make sure four bytes were added to the start of the exchanges for Status and Control data.

Data Going to the wrong Place. If the exchanges are transferring data, but the data seems to be going to the wrong places, verify the exchange definitions in configuration and check the modules in the ENIU starting with Rack 0, Slot 1 onward until the discrepancy is found.

Determining the MAC Address of the ENIU

A MAC address is a unique six-byte number (written as six 2-digit hex numbers separated by dashes, such as 08-00-19-01-24-B6) that identifies each Ethernet hardware device. These numbers are assigned by manufacturers and are often not configurable by users. The ENIU's MAC address is printed on the front of the ENIU. However, if it becomes illegible or appears to be incorrect, you can check it using the following procedure:

To determine the MAC address of the ENIU before it is on the network, set the Node Address switches to 901 and power cycle the ENIU. The OK, Fault, LAN, and Status LED's will flash 12 patterns (two for each byte) spelling out the MAC address digit by digit. This follows binary encoding with the exception of the use of the green Status LED to represent zero. The following table contains the coding for these patterns.

Use the worksheet on the next page to record the 12 LED patterns, then return to the following table to decode them. The numbers in parentheses represent the binary value of the condition, and Logic 1 conditions are shown in bold type for improved readability.

LED Decoding Table				
Hex Value	OK LED	Fault LED	LAN LED	Status LED
0	Off (0)	Off (0)	Off (0)	Green (0)
1	Off (0)	Off (0)	Off (0)	Amber (1)
2	Off (0)	Off (0)	On (1)	Off (0)
3	Off (0)	Off (0)	On (1)	Amber (1)
4	Off (0)	On (1)	Off (0)	Off (0)
5	Off (0)	On (1)	Off (0)	Amber (1)
6	Off (0)	On (1)	On (1)	Off (0)
7	Off (0)	On (1)	On (1)	Amber (1)
8	On (1)	Off (0)	Off (0)	Off (0)
9	On (1)	Off (0)	Off (0)	Amber (1)
A	On (1)	Off (0)	On (1)	Off (0)
B	On (1)	Off (0)	On (1)	Amber (1)
C	On (1)	On (1)	Off (0)	Off (0)
D	On (1)	On (1)	Off (0)	Amber (1)
E	On (1)	On (1)	On (1)	Off (0)
F	On (1)	On (1)	On (1)	Amber (1)

Note

All GE ENIU MAC addresses start with 08-00-19

Worksheet for Determining the MAC Address

If the rotary switches are set to 901 when the Ethernet ENIU powers up, it displays its MAC address in a series of 12 patterns on the OK, FAULT, LAN, and STATUS LED lights. Each pattern will be separated by a brief “no-number” condition, in which all four LEDs are off. As each of the 12 patterns is displayed, record them in the table below. Then, decode the binary patterns into hexadecimal numbers using either (1) the “Weight” column values in the table below, or (2) the “LED Decoding Table” from the previous page. An additional “Value” row is provided below the table for you to write in the decoded numbers. Use the following guidelines for recording the patterns:

1. For the OK, FAULT, and LAN LEDs, record one of these two states: 1 if On or leave blank if Off (or, you may wish to record zeros if Off, instead of blanks).
2. For the STATUS LED, record one of these three states: 1 if On with an amber (yellow) color, 0 if On with a green color, or leave blank if Off. The “On with a green color” state of this LED is used to distinguish the number zero from a no-number indication.

NOTE: Since GE MAC addresses always begin with 08-00-19, these first six numbers have been filled in already.

LED ↓	1	2	3	4	5	6	7	8	9	10	11	12	Weight
OK		1				1							8
FAULT													4
LAN													2
STATUS	0		0	0	1	1							1
Value	0	8	0	0	1	9							N/A

Example

LED ↓	1	2	3	4	5	6	7	8	9	10	11	12	Weight
OK		1				1					1		8
FAULT										1		1	4
LAN									1		1	1	2
STATUS	0		0	0	1	1	0	1			1		1
Value	0	8	0	0	1	9	0	1	2	4	B	6	N/A

So the MAC address in this example is 08-00-19-01-24-B6

Reading the Stored IP Address of the ENIU

If the stored IP address of an ENIU is lost or forgotten, you can cause it to be displayed on the ENIU's front panel LEDs by (1) setting the Node Address switches to 900 and (2) power-cycling the ENIU. The OK, Fault, LAN, and Status LED's will flash 8 patterns (two for each byte) spelling out the IP address digit by digit. As each of the 8 patterns is displayed, record them in the table below. Then, decode the binary patterns into hexadecimal (hex) numbers using the Weight column values or the "LED Decoding Table." Two additional (Value) rows are provided below the table for you to write in the decoded hex numbers and equivalent decimal numbers. Use the following guidelines for recording the patterns:

- For the OK, FAULT, and LAN LEDs, record either 1 if On or blank if Off (or, if preferred, write a zero if Off instead of a blank).
- For the STATUS LED, record one of these three conditions: 1 if On with an amber (yellow) color, 0 if On with a green color, or blank if Off.

If no IP address has been stored, the LEDs will indicate the default address of 195.0.0.0.

LED ↓	1	2	3	4	5	6	7	8	Weight
OK									8
FAULT									4
LAN									2
STATUS									1

Hex. Values									N/A
Decimal Values									N/A

IP Address Example

In the following example, the bit patterns have been recorded and then converted to the two-byte hex number, then to an equivalent decimal number:

IP Address Example, Continued

LED ↓	1	2	3	4	5	6	7	8	Weight
OK						1			8
FAULT								1	4
LAN		1				1			2
STATUS	0	1	1	0	1	1	0	1	1

Hex. Values	0	3	1	0	1	B	0	5	N/A
Decimal Values	3		16		27		5		N/A

So the decoded IP address is 3.16.27.5

Chapter 7

VersaMax Product Overview

The VersaMax Family of Products

The VersaMax family of products provides distributed I/O that spans PLC and PC-based architectures. Designed for industrial and commercial automation, VersaMax I/O provides a common, flexible I/O structure for local and remote control applications. The VersaMax PLC provides big-PLC power with a full range of I/O and option modules. VersaMax I/O Stations with Network Interface Modules make it possible to add VersaMax I/O to other types of networks. VersaMax meets UL, CUL, CE, Class1 Zone 2 and Class I Division 2 requirements.

The 70-mm depth and small footprint of VersaMax I/O enables easy, convenient mounting as well as space-saving benefits. Modules can accommodate up to 32 points of I/O each.

VersaMax products feature DIN-rail mounting with up to eight I/O and option modules per “rack” and up to 8 racks per VersaMax PLC or VersaMax I/O Station system. Expansion racks can be located up to 750 meters from the main VersaMax PLC or VersaMax I/O Station rack. Expansion racks can include any VersaMax I/O, option, or communications module.

VersaMax provides automatic addressing that can eliminate traditional configuration and the need for hand-held devices. Multiple field wiring termination options provide support for two, three, and four-wire devices.

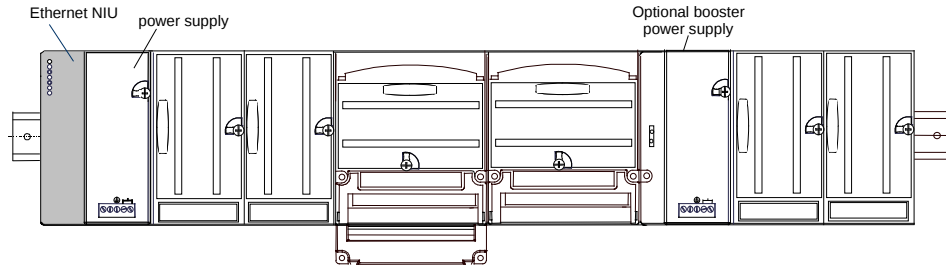
For faster equipment repair and shorter Mean-Time-To-Repair, the hot insertion feature enables addition and replacement of I/O modules while a machine or process is running and without affecting field wiring.

VersaMax Products for Ethernet Networks

There are two VersaMax products for Ethernet networks. One is the IC200CPUE05, which is a CPU with an embedded Ethernet interface, and the other is the IC200EBI001 Ethernet NIU.

Ethernet Network Interface Unit

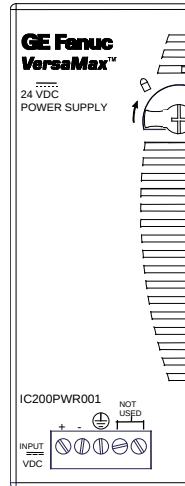
The Ethernet Network Interface Unit acts as controller for an I/O Station of VersaMax modules. Many types of modules can be combined to suit the needs of the application. I/O modules install on individual “carriers”. Power for module operation is provided by a power supply that installs directly on the NIU. Additional “booster” power supplies can be included in the system if needed for modules with high current requirements.



For more information about modules and system installation instructions, please see the *VersaMax Modules, Power Supplies and Carriers User's Manual* (GFK-1504).

Power Supplies

A VersaMax Power Supply provides +5V and +3.3V power to the modules in the station. VersaMax power supplies are available for various AC or DC input voltages as shown in the table below. Additional power supplies can be installed on special booster carriers if needed for systems where the number of modules creates the need for a booster. No booster supply is needed to power conventional I/O modules.



Available Power Supplies and Carrier

The following VersaMax power supplies and carrier are available:

24VDC Power Supply	IC200PWR001
24VDC Expanded 3.3V Power Supply	IC200PWR002
120/240VAC Power Supply	IC200PWR101
120/240VAC Expanded 3.3V Power Supply	IC200PWR102
12VDC Power Supply	IC200PWR201
12VDC Expanded 3.3V Power Supply	IC200PWR202
Power Supply Booster Carrier	IC200PWB001

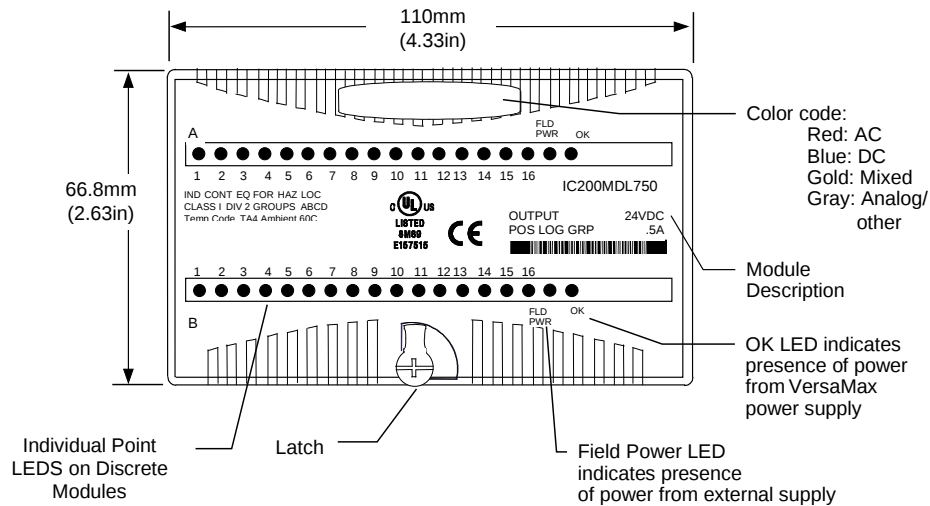
Note

The **IC200PWR001** power supply does **not** have sufficient power capacity to support a typical ENIU system.

Power supplies are described in the *VersaMax Modules, Power Supplies, and Carriers User's Manual* (GFK-1504).

I/O Modules

VersaMax I/O and option modules are approximately 110mm (4.33in) by 66.8mm (2.63in) in size. Modules can be mounted either horizontally or vertically on several types of available I/O Carriers. Modules are 50mm (1.956 in) in depth, not including the height of the carrier or the mating connectors. The following figure shows an example of a discrete output module.



VersaMax I/O modules are described in the *VersaMax Modules, Power Supplies, and Carriers User's Manual* (GFK-1504).

I/O Modules

The following types of VersaMax I/O Modules are available:

Discrete Input Modules	
Input 120VAC 8 Point Grouped Module	IC200MDL140
Input 240VAC 8 Point Grouped Module	IC200MDL141
Input 120VAC 8 Point Isolated Module	IC200MDL143
Input 240VAC 4 Point Isolated Module	IC200MDL144
Input 120VAC (2 Groups of 8) 16 Point Module	IC200MDL240
Input 240VAC (2 Groups of 8) 16 Point Module	IC200MDL241
Input 120VAC 16 Point Isolated Module	IC200MDL243
Input 240VAC 8 Point Isolated Module	IC200MDL244
Input 125VDC Positive/Negative Logic Grouped 8 Point Module	IC200MDL631
Input 125VDC Positive/Negative Logic Grouped 16 Point Module	IC200MDL632
Input 48VDC Positive/Negative Logic Grouped 16 Point Module	IC200MDL635
Input 48VDC Positive/Negative Logic Grouped 32 Point Module	IC200MDL636
Input 24VDC Positive/Negative Logic (2 Groups of 8) 16 Point Module	IC200MDL640
Input 5/12VDC (TTL) Positive/Negative Logic 16 Point Module	IC200MDL643
Input 5/12VDC (TTL) Positive/Negative Logic Grouped 32 Point Module	IC200MDL644
Input 24VDC Positive/Negative Logic (4 Groups of 8) 32 Point Module	IC200MDL650
Discrete Output Modules	
Output 120VAC 0.5A per Point Isolated 8 Point Module	IC200MDL329
Output 120VAC 0.5A per Point Isolated 16 Point Module	IC200MDL330
Output 120VAC 2.0A per Point Isolated 8 Point Module	IC200MDL331
Output 24VDC Positive Logic 2.0A per Point (1 Group of 8) w/ESCP 8 Point Module,	IC200MDL730
Output 12/24VDC Positive Logic 0.5A per Point (1 Group of 16) 16 Point Module	IC200MDL740
Output 24VDC Positive Logic 0.5A per Point (1 Group of 16) w/ESCP 16 Point Module	IC200MDL741
Output 24VDC Positive Logic 0.5A per Point (2 Groups of 16) w/ESCP 32 Point Module	IC200MDL742
Output 5/12/24VDC Negative Logic 0.5A per Point (1 Group of 16) 16 Point Module	IC200MDL743
Output 5/12/24VDC Negative Logic 0.5A per Point (2 Groups of 16) 32 Point Module	IC200MDL744
Output 12/24VDC Positive Logic 0.5A per Point (2 Groups of 16) 32 Point Module	IC200MDL750
Output Relay 2.0A per Point Isolated Form A 8 Point Module	IC200MDL930
Output Relay 2.0A per Point Isolated Form A 16 Point Module	IC200MDL940

Discrete Mixed I/O Modules		
Mixed 24VDC Positive Logic Input 20 Points / Output Relay 2.0A 12 Point Module		IC200MDD840
Mixed 24VDC Positive Logic Input 20 Point / Output 12 Point / (4) High Speed Counter, PWM, or Pulse Train Configurable Points		IC200MDD841
Mixed 16 Point Grouped Input 24VDC Pos/Neg Logic / 16 Pt Grouped Output 24VDC Pos. Logic 0.5A w/ESCP		IC200MDD842
Mixed 24VDC Positive Logic Input Grouped 10 Point / Output Relay 2.0A per Point 6 Point Module		IC200MDD843
Mixed 24 VDC Pos/Neg Logic Input Grouped 16 Point / Output 12/24VDC Pos. Logic 0.5A 16 Point Module		IC200MDD844
Mixed 16 Point Grouped Input 24VDC Pos/Neg Logic / 8 Pt Relay Output 2.0A per Pt Isolated Form A		IC200MDD845
Mixed 120VAC Input 8 Point / Output Relay 2.0A per Point 8 Point Module		IC200MDD846
Mixed 240VAC Input 8 Point / Output Relay 2.0A per Point 8 Point Module		IC200MDD847
Mixed 120VAC Input 8 Point / Output 120VAC 0.5A per Point Isolated 8 Point Module		IC200MDD848
Mixed 120VAC In Isolated 8 Point / Output Relay 2.0A Isolated 8 Point Module		IC200MDD849
Mixed 240VAC In Isolated 4 Point / Output Relay 2.0A Isolated 8 Point Module		IC200MDD850
Analog Input Modules		
Analog Input Module, 12 Bit Voltage/Current 4 Channels		IC200ALG230
Analog Input Module, 16 Bit Voltage/Current, 1500VAC Isolation, 8 Channels		IC200ALG240
Analog Input Module, 12 Bit Voltage/Current 8 Channels		IC200ALG260
Analog Input Module, 15 Bit Voltage Differential 8 Channels		IC200ALG261
Analog Input Module, 15 Bit Current Differential 8 Channels		IC200ALG262
Analog Input Module, 15 Bit Voltage 15 Channels		IC200ALG263
Analog Input Module, 15 Bit Current 15 Channels		IC200ALG264
Analog Input Module, 16 Bit RTD, 4 Channels		IC200ALG620
Analog Input Module, 16 Bit Thermocouple, 7 Channels		IC200ALG630
Analog Output Modules		
Analog Output Module, 12 Bit Current, 4 Channels		IC200ALG320
Analog Output Module, 12 Bit Voltage 4 Channels. 0 to +10VDC Range		IC200ALG321
Analog Output Module, 12 Bit Voltage 4 Channels. -10 to +10VDC Range		IC200ALG322
Analog Output Module, 13 Bit Voltage 8 Channels		IC200ALG325
Analog Output Module, 13 Bit Voltage 12 Channels		IC200ALG327
Analog Output Module, 16 Bit Voltage/Current, 1500VAC Isolation, 4 Channels		IC200ALG331
Analog Mixed I/O Modules		
Analog Mixed Module, Input Current 4 Channels, Output Current 2 Channels		IC200ALG430
Analog Mixed Module, 0 to +10VDC Input 4 Channels, Output 0 to +10VDC 2 Channels		IC200ALG431
Analog Mixed Module, 12 Bit ± 10 VDC, Input 4 Channels and Output 2 Channels		IC200ALG432

Carriers

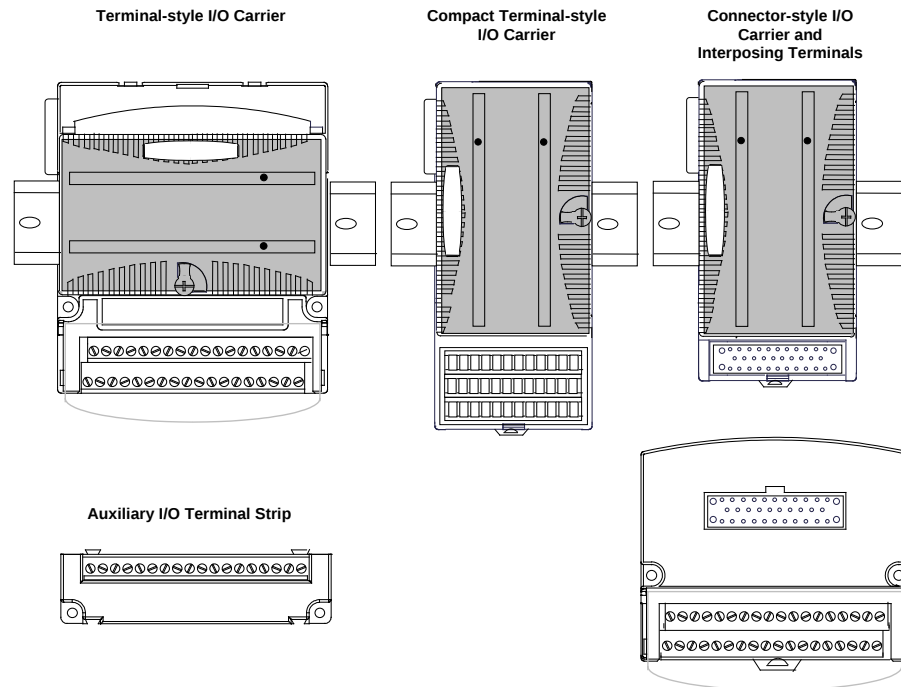
Carriers provide mounting, backplane communications, and field wiring connections for all types of VersaMax modules. I/O modules can be installed on carriers or removed without disturbing field wiring.

There are three basic I/O Carrier types:

- Terminal-style I/O carriers. Modules mount parallel to the DIN rail.
- Compact Terminal-style I/O Carriers. Modules mount perpendicular to the DIN rail.
- Connector-style I/O Carriers. Modules mount perpendicular to the DIN rail. These carriers are normally used with Interposing I/O Terminals as illustrated below.

See the *VersaMax Modules, Power Supplies, and Carriers User's Manual* (GFK-1504) for information about VersaMax I/O Carriers.

Terminal-style I/O carriers have 36 individual terminals for direct connection of field wiring. Auxiliary I/O Terminal Strips are available for applications requiring additional wiring terminals. See GFK-1504 for information about the VersaMax Interposing Terminals and Auxiliary I/O Terminal Strips.



VersaMax Carriers and Terminal Strips

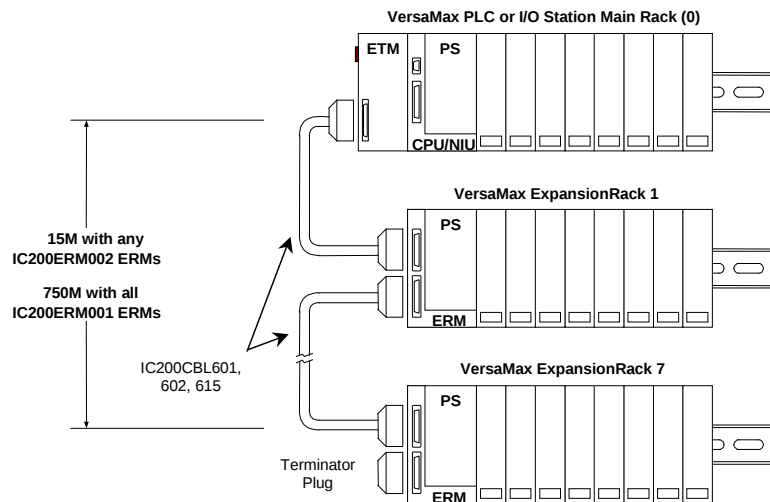
The following types of Carriers, terminals, and cables are available:

Terminal-Style I/O Carriers	
Barrier-Style Terminal I/O Carrier	IC200CHS001
Box-Style Terminal I/O Carrier	IC200CHS002
Spring-Style Terminal I/O Carrier	IC200CHS005
Compact Terminal-Style I/O Carriers	
Compact Box-Style I/O Carrier	IC200CHS022
Compact Spring-Style I/O Carrier	IC200CHS025
Connector-Style I/O Carrier	
Connector-Style I/O Carrier	IC200CHS003
Interposing Terminals for use with Connector-Style Carrier	
Barrier-Style Interposing I/O Terminals	IC200CHS011
Box-Style Interposing I/O Terminals	IC200CHS012
Thermocouple-Style Interposing I/O Terminals	IC200CHS014
Spring-Style Interposing I/O Terminals	IC200CHS015
Cables for use with Connector-Style I/O Carriers	
2 connectors, 0.5m, no shield	IC200CBL105
2 connectors, 1.0m, no shield	IC200CBL110
2 connectors, 2.0m, no shield	IC200CBL120
1 connector, 3.0m, no shield	IC200CBL230
Auxiliary I/O Terminal Strips for use with Terminal-style I/O Carriers and Interposing Terminals	
Barrier-Style Auxiliary I/O Terminal Strip	IC200TBM001
Box-Style Auxiliary I/O Terminal Strip	IC200TBM002
Spring-Style Auxiliary I/O Terminal Strip	IC200TBM005
Other Carriers	
Communications Carrier	IC200CHS006
Power Supply Booster Carrier	IC200PWB001

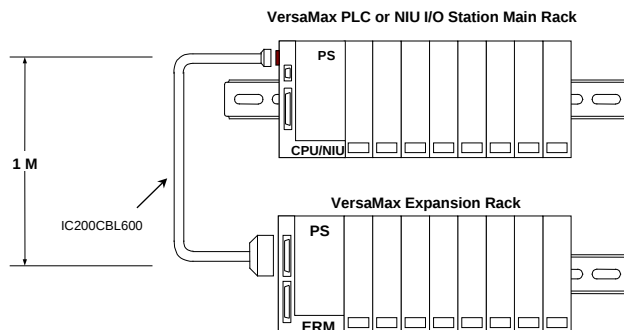
Expansion Modules

Expansion modules can be used to extend the I/O Station and add more modules. There are two basic types of VersaMax I/O expansion systems, Multi-Rack and Single-ended:

- Multi-Rack:** A VersaMax PLC or NIU I/O Station with an IC200ETM001 Expansion Transmitter Module (ETM) and one to seven expansion “racks”, each with an Expansion Receiver Module (ERM), catalog number IC200ERM001 or IC200ERM002. If all the Expansion Receivers are the Isolated type (IC200ERM001), the maximum overall cable length is 750 meters. If the expansion bus includes any non-isolated Expansion Receivers (IC200ERM002), the maximum overall cable length is 15 meters.



- Single-ended:** A PLC or NIU I/O Station connected directly to one expansion rack with non-isolated Expansion Transmitter Module (IC200ERM002). Maximum cable length is 1 meter. (No terminator is required.)



VersaMax Modules for Expansion Racks

All types of VersaMax I/O and communications modules can be used in expansion racks. Some VersaMax analog modules require specific module revisions as listed below:

Module	Module Revision
IC200ALG320	B or later
IC200ALG321	B or later
IC200ALG322	B or later
IC200ALG430	C or later
IC200ALG431	C or later
IC200ALG432	B or later

Available Expansion Modules, Cables, and Related Products

The following Expansion Modules and related products are available:

<i>Expansion Modules</i>	
Expansion Transmitter Module	IC200ETM001
Expansion Receiver Module, Isolated	IC200ERM001
Expansion Receiver Module, Non-isolated	IC200ERM002
<i>Cables</i>	
Expansion Cable, 1 meter	IC200CBL601
Expansion Cable, 2 meters	IC200CBL602
Expansion Cable, 15 meters	IC200CBL615
Terminator Plug (included with ETM)	IC200ACC201
Connector Kit	IC200ACC302

See the *VersaMax Modules, Power Supplies, and Carriers User's Manual* (GFK-1504) for information about VersaMax Expansion modules.

VersaMax General Product Specifications

VersaMax products should be installed and used in conformance with product-specific guidelines as well as the following specifications:

Environmental		
Vibration	IEC68-2-6	1G @57-150Hz, 0.012in p--p @10-57Hz
Shock	IEC68-2-27	15G, 11ms
Operating Temp.		0 deg C to +60 deg C ambient
Storage Temp.		-40 deg C to +85 deg C
Humidity		5% to 95%, noncondensing
Enclosure Protection	IEC529	Steel cabinet per IP54: protection from dust & splashing water
EMC Emission		
Radiated, Conducted	CISPR 11/EN 55011	Industrial Scientific & Medical Equipment (Group 1, Class A)
	CISPR 22/EN 55022	Information Technology Equipment (Class A)
	FCC 47 CFR 15	referred to as FCC part 15, Radio Devices (Class A)
EMC Immunity		
Electrostatic Discharge	EN 61000-4-2	8KV Air, 4KV Contact
RF Susceptibility	EN 61000-4-3	10V _{rms} /m, 80Mhz to 1000Mhz, 80% AM
	ENV 50140/ENV 50204	10V _{rms} /m, 900MHz +/-5MHZ 100%AM with 200Hz square wave
Fast Transient Burst	EN 61000-4-4	2KV: power supplies, 1KV: I/O, communication
Surge Withstand	ANSI/IEEE C37.90a	Damped Oscillatory Wave: • 2.5KV power supplies, I/O [12V-240V] • 1KV communication
	IEC255-4	Damped Oscillatory Wave: Class II, power supplies, I/O [12V-240V]
	EN 61000-4-5	2 kV cm(P/S); 1 kV cm (I/O and communication modules)
Conducted RF	EN 61000-4-6	10V _{rms} † 0.15 to 80Mhz, 80%AM
Isolation		
Dielectric Withstand	UL508, UL840, IEC664	1.5KV for modules rated from 51V to 250V
Power Supply		
Input Dips, Variations	EN 61000-4-11	During Operation: Dips to 30% and 100%, Variation for AC +/-10%, Variation for DC +/-20%

ARP	Address Resolution Protocol. A system used to determine the MAC address of a device whose IP address is known. In practice, an ARP request message is sent to a particular IP address, requesting the device's MAC address. The device's reply, containing its MAC address, is called an ARP response message . It is also a computer utility program (such as ARP.EXE in Windows 95) that can be used in conjunction with Telnet software to assign IP addresses to devices.
Byte	A group of eight bits. Also called by the data communications term, octet .
Carrier	A device that provides mounting, backplane communications, and field wiring connections for VersaMax modules. Carriers, in turn, mount to DIN-rails.
DIN-rail	A metal rail measuring a standard 7.5mm x 35mm that serves as a mounting device for various products, such as the VersaMax products, which usually snap on or off the DIN-rail. DIN-rails should be compliant, preferably, with the DIN EN50032 standard.
EGD	Ethernet Global Data. A proprietary GE protocol that provides efficient connectionless periodic data transfer over an Ethernet network. It operates over the industry standard User Datagram Protocol (UDP).
ENIU	Ethernet Network Interface Unit
ERM	Expansion Receiver Module. A VersaMax module that mounts in an expansion rack. It is used to interface the expansion rack to the expansion bus.
ETM	Expansion Transmitter Module. A VersaMax module that mounts in an I/O station. It is used to interface the I/O station to the expansion bus.
Gateway	See Router
Host	See Node
Hostid	The portion of an IP Address that identifies the host. Each host on the same network must have a unique hostid .
I/O	Input/Output
IP	Internet Protocol
IP Address	A unique address number assigned to a network device. It has a network portion,

called the **netid**, and a host portion called the **hostid**. According to the current standard, it is a 32-bit value. To make it more readable to humans, the 32 binary bits are often divided into four eight-bit segments (“octets”); then each segment is converted to a decimal number (in the range of 0-255), and each decimal number is separated by dots. For example, 192.2.1.123. This type of notation is called “dotted decimal” notation. In practice, a block of IP address numbers is assigned to an organization, which then assigns unique numbers from within its assigned number block to devices that it manufactures or implements.

LAN	Local Area Network
MAC	Media Access Control.
MAC Address	A unique address number assigned to a network device. According to the current standard, it is a 48-bit value, divided into twelve four-bit segments, with each segment converted to a hexadecimal number, and each pair of hexadecimal number separated by a dash. For example, 08-00-19-01-24-B6. In practice, an organization is assigned a unique 24-bit initial number sequence (GE’s sequence is 08-00-19). The organization uses its assigned sequence with 24 additional bits to assign a unique number to the network devices it produces. MAC Addresses conform to the 802.3 MAC Layer Standard.
Modbus™	A serial communications protocol, originally developed by Modicon, that has come into widespread use worldwide. Often called “Modbus RTU”, or just “RTU.” Modbus is now a trademark of Gould, Inc.
Netid	The portion of a host’s IP Address that identifies what network the host is located on. Each host on a given network/subnet must have the same netid . A router, which serves as a host on two or more networks, has a different IP address for each network it is connected to, with each IP address being based upon the netid for its network.
NIC	Network Interface Card
NIU	Network Interface Unit
Node	A device connected to a network. Also called a “host.” Examples of Ethernet nodes are (1) a personal computer with an Ethernet NIC, (2) a PLC with an Ethernet interface module, or (3) an Ethernet router. Each node must have its own unique IP Address.
Octet	A group of eight bits. Octet is the data communications term for byte .
PLC	Programmable Logic Controller
Router	A device that routes messages between/among two or more networks. A router uses a message’s destination IP Address to determine what network to route the message to. A router has a different IP Address for each network it connects to (see also, Netid). A router is also known by the term “gateway.”

Subnet	Shortened version of the term subnetwork. It is the most basic or lowest level of network. Subnets are formed by subdividing an organization's assigned Network ID using a subnet mask . This subdividing involves reassigning bits from the host portion to the network portion of the address.
Subnet mask	A 32-bit binary number, assigned to each network or subnetwork, that is used in conjunction with IP addresses. It allows an organization to use their assigned IP addresses, which otherwise would be limited to one network, to form additional internal subnetwork addresses.
TCP	Transmission Control Protocol
UDP	User Datagram Protocol. UDP is an ISO transport layer protocol that is very efficient because it is connectionless and uses a minimum of error checking.
UTP	Unshielded Twisted Pair.
VersaMax	A GE Automation product series, consisting of PLC and I/O devices.
VersaPro	A GE Automation programming and configuration software product.

Appendix B

IP and MAC Addresses

This appendix gives an overview of IP addresses, routers (gateways), subnet masks and MAC addresses.

IP Addresses

Each TCP/IP *node* on a network must have a unique *IP address*. The *Ethernet NIU* is such a node, as is a Personal Computer running TCP/IP. There may be other nodes on the network that are not involved with communications to the PLCs, but no matter what their function, each TCP/IP node must have its own *IP address*. It is the unique *IP address* that identifies each node on the network (or system of connected networks). (Note that Internet terminology often uses the term “host” to identify a node on a network.)

The *IP address* is 32 bits long and has a *netid* part and a *hostid* part. Each network is a Class A, Class B, or Class C network. The class of a network determines how the IP Address bits are apportioned between the *netid* and *hostid* parts:

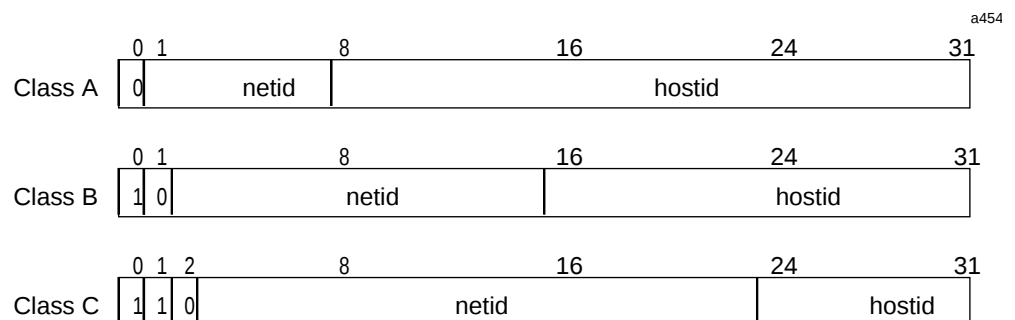


Figure B-1. IP Address Format for Network Classes A, B, C

Each node on the same local network must have an *IP address* of the same class and each must have the same *netid*. Each node on the same network must have a different *hostid* thus giving it a unique *IP address*.

IP addresses are written as four decimal integers (0–255) separated by periods (called “dotted-decimal”) where each integer gives the value of one byte of the IP address. For example, the 32-bit IP address

00000011 00000000 00000000 00000001

is written in dotted-decimal format as:

3.0.0.1

Network Classes

As shown in the previous section, there are three major network classes, A, B, and C. You can distinguish the class of an IP address from its first integer if it’s written in dotted-decimal format, or from the leading bit or bits (called “Class ID Bits Patterns” in the following table) if it’s written in binary format. Using the leading bits method, we see that any IP address (in binary format) that starts with a zero is a Class A address, any IP address starting with 10 is a Class B address, and any starting with 110 is a Class C address. To illustrate this, the following example IP address is shown in both formats:

Dotted-decimal format: 191.12.3.77

Binary format: 10111111 00001100 00000011 01001101

In the table below, we see from the “Range of First Integer” column, that the first of the four integers, 191, indicates this to be a Class B address. Additionally, we see from the “Class ID Bit Patterns” column that the leading bits 10 also indicate that this is a Class B address.

Be aware that utilizing these leading bits (which are in the netid portion of the IP address) for the Class ID leaves less bits for network identification. Compare the “Size of Netid” column with the “Remaining Netid Bits” column in the table, which reflects the netid size after the class ID bits are subtracted.

The following table reveals how many total network numbers are available. As shown, the total number is 126+16,382+2,097,150 which equals 2,113,658 possible networks for the entire world. That seemed like a large number 20 years ago, but due to the enormous growth of the Internet, there is a danger of running out of network addresses. This assessment is based upon the current 32-bit standard known as IP Version 4. A new IP address standard, IP Version 6, will, when implemented, address this problem by increasing IP address length to 128 bits.

Class	Size of Netid	Size of Hostid	Range of First Integer	Class ID Bit Patterns	Remaining Netid Bits	Max. Networks per Class	Max. no. of Hosts per Network	Default Subnet Mask
A	8 bits	24 bits	0 – 127	0	7	126	16,777,216	255.0.0.0
B	16 bits	16 bits	128 – 191	10	14	16,382	65,536	255.255.0.0
C	24 bits	8 bits	192 – 223	110	21	2,097,150	254	255.255.255.0
Other	N/A	N/A	224 – 255	N/A	N/A	N/A	N/A	N/A

Routers and Gateways

Routers (also known as gateways) connect individual physical networks into a system of networks. When a node needs to communicate with a node on another network, a router transfers the data between the two networks. Each router port must have an appropriate IP address for the network (or subnetwork) it connects to.

The following figure shows Router R1 connecting Network 1 with Network 2. Note that each router port has its own IP address (the Network 1 port is 128.1.0.2, and the Network 2 port is 128.2.0.3). A router port's IP address must contain the netid of the network it connects to in keeping with the rule that all nodes on a given network must have the same netid.

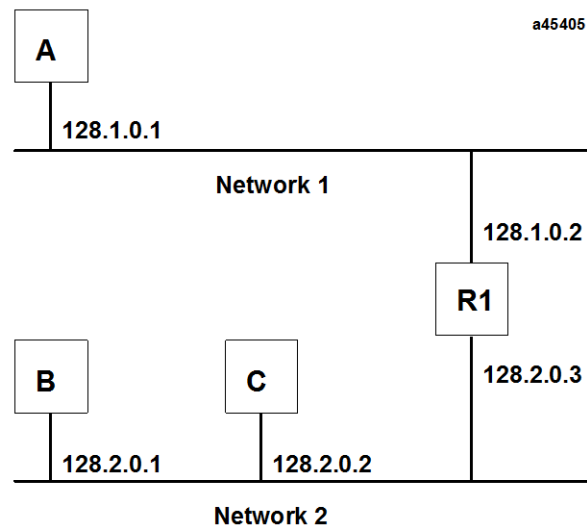


Figure B-2. Connecting Two Networks with a Router

When host B communicates with host C, it knows from C's IP address that C is on the same network. In an Ethernet environment, B can then resolve C's IP address to a MAC address (via ARP) and communicate with C directly.

When host B communicates with host A, it knows from A's IP address that A is on another network (the *netids* are different). In order to send data to A, B must have the IP address of the router connecting the two networks. In this example, the router's IP address on Network 2 is 128.2.0.3.

Subnets

When an organization is assigned a block of IP addresses, it receives one network ID (designated by the *netid* portion of the IP address) and a block of host IDs. The *netid* portion of the assignment is fixed; it cannot be changed. All IP addresses that an organization assigns from its block have the same *netid*, so they are all assigned to one basic network. At the time this arrangement was conceived, it was not foreseen that organizations might want or need to divide a single network into many subnetworks, so no provision was originally made for this. Later, because the need for more networks arose, the subnet arrangement was devised. Since the *netid* portion of an IP address is fixed, it was necessary to use a portion of the *hostid* to create subnetwork IDs.

So, subnet addressing is an extension of the IP address scheme that allows a site to use a single *netid* for multiple networks (subnets). Routing outside the site continues as usual by dividing the IP address into a *netid* and a *hostid* based on the Network Class (A, B, or C) definition. However, inside a site, the IP address is rearranged into custom *netid* and *hostid* portions by a *subnet mask*, discussed next.

Subnet Masks

A subnet mask is a 32-bit binary number that is assigned to a network to indicate how many of the *hostid* bits will be reallocated for use as subnet ID bits. Each class of network, A, B, or C, has a default subnet mask. A default subnet mask does not create subnet ID capability. You must create a custom subnet mask to be able to create subnets. Let's use a Class B network address as an example. The normal bit allocation of a class B IP address is that the first 16 bits are the *netid* and the last 16 bits are the *hostid*. The following is the default subnet mask for a class B network:

Default Class B subnet mask: 11111111 11111111 00000000 00000000

In a subnet mask, the ones indicate how many bits are in the *netid* and the zeros indicate how many bits are in the *hostid*. So the default Class B subnet mask shown (16 one-bits and 16 zero-bits) does not change the allocation of the bits in a Class B IP address, which means we cannot create any subnets if we have a default subnet mask. We must create a "custom" subnet mask in order to create subnets. This consists of taking the default subnet mask and replacing some zeros with ones, starting with the left-most zero. For example, let's change the default subnet mask above by changing the first two zeros to ones. Here is the resulting subnet mask:

Custom Class B subnet mask: 11111111 11111111 11000000 00000000

The two extra *netid* bits, taken from the *hostid* bits, can be used to designate subnets. These two bits have four possible combinations:

00 01 10 11

Therefore, we might assume that we could designate four subnets with these two bits. However, the current subnet standard states that the all-zeros and all-ones combinations are not allowed to be used as subnet IDs. Therefore, our two bits only allow us to name two subnets. (Note that the two bits are the two most significant bits in their byte, so they have values of 128 and 64.) If we needed more than two subnets, we would have to reallocate more *hostid* bits to the subnet ID by changing more of the zeros to ones in the subnet mask. The possible number of subnets for a given number of reallocated *hostid* bits is found by this formula:

$$\text{Subnets} = 2^N - 2 \quad \text{where } N = \text{the number of reallocated hostid bits}$$

So, using this formula, we see that 2 bits gives 2 subnets, 3 bits gives 6 subnets, 4 bits gives 14 subnets, etc. Note that by reallocating *hostid* bits for use in the subnet ID, there are fewer host bits left to use for assigning host IDs. Therefore, depending on your application, you may need to carefully balance the number of subnets you create against the number of hosts required to ensure that you have sufficient of each.

Subnet Example

Lets create a pair of subnets from Network 2 (a Class B network) in the previous figure. As we discussed in the last section, using the following custom subnet mask would add two additional *netid* bits, allowing us to create two subnetworks:

11111111 11111111 11000000 00000000

(In dotted-decimal format, our subnet mask looks like this: 255.255.192.0)

The new configuration for our network system would be as shown in the following figure. Notice that Network 2 has been subdivided into subnetworks 2.1 and 2.2. Hosts on these subnets still use the basic *netid* of 128.2.as the first numbers in their IP addresses, but note the additional subnet numbers added to the addresses:

128.2.64.x and 128.2.128.x

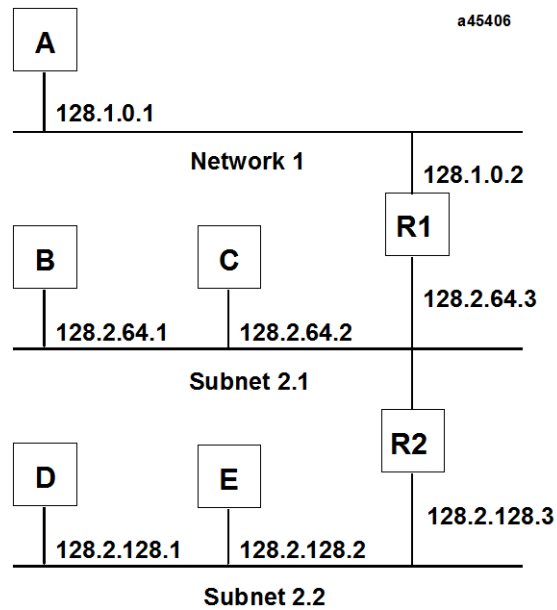


Figure B-3. Network Configuration Using a Subnet Mask

In our new arrangement, shown above, Network 2 has been subdivided into two subnets and Hosts D and E have been added. Router R2 connects Subnet 2.1 with Subnet 2.2. Hosts D and E will use Router R2 to communicate with hosts not on Subnet 2.2. Hosts B and C will use Router R1 to communicate with hosts not on Subnet 2.1. When B is communicating with D, R1 (the configured Router for B) will route the data from B to D through Router R2.

In practice, a network's routers (gateways) use the subnet masks (each subnetwork has one) in conjunction with a message's destination IP Address (which is located at the beginning of each message) to determine which subnet to route the message to.

MAC Addresses

Each byte of the MAC Address is an 8-bit binary number. Thus, the 12-digit hex address is really a 48-bit binary number. A typical MAC Address, 08-00-19-00-53-12, is represented as a binary number in the following example:

Byte	1		2		3		4		5		6	
Hex	0	8	0	0	1	9	0	0	5	3	1	2
Binary	0000	1000	0000	0000	0001	1001	0000	0000	0101	0011	0001	0010

Another characteristic that is important, especially for multi-vendor networks, is the order of address-bit transmission on the physical medium. MAC Addresses are transmitted in ascending byte order, with the least significant bit of each byte transmitted first.

The example above is shown including bit transmission order as follows:

Byte	1		2		3		4		5		6	
Hex	0	8	0	0	1	9	0	0	5	3	1	2
Binary	0000	1000	0000	0000	0001	1001	0000	0000	0101	0011	0001	0010
Bit Order	8765	4321	...9									
	</											

Appendix C

Number Conversion Table

Hexadecimal digits have decimal values from 0..15, represented as 0...9, A (for 10), B (for 11) ... F (for 15). Use the following table to convert binary-to-decimal-to-hex:

Decimal	Binary	Hexadecimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Figure C-1. Decimal, Binary, and Hexadecimal Conversion Table

To convert a binary value that has more than four bits into a hexadecimal value, first separate it into groups of four bits. (If the number of binary bits is not an even multiple of four, add leading zeroes to make it an even multiple.) Each group of four binary bits represents a single hexadecimal number. For example, let's convert the binary number 1011110000 to the hexadecimal number 2F0. First separate the binary number into groups of four bits (add two leading zeroes): 0010 1111 0000. Then use the above conversion table to find the hexadecimal value for each group: 0010 = 2, 1111 = F, 0000 = 0.

Appendix D

Compatibility Matrix

The following table identifies the major features of each VersaMax ENIU release, the software required to fully use the features of each release, and the publications that document each release. Note that later versions of the ENIU and software support the features of the previous releases.

Features	ENIU Release Version	Software Versions that Support the Features ¹	ENIU User's Manual Version	IPI Version
Initial ENIU release Modbus RTU support	1.0	VersaPro 1.5 (or later) Remote I/O Manager 1.5 (or later)	GFK-1860	GFK-1861
Ethernet Global Data (EGD) Expansion Rack support Hot insertion of I/O modules	1.1	VersaPro 1.5 (or later) Remote I/O Manager 1.5 (or later) Logic Developer PLC 2.10 (or later)	GFK-1860A	GFK-1861A
Support for high density analog I/O modules	1.1	VersaPro 2.0 (or later) Remote I/O Manager 2.0 (or later) Logic Developer PLC 2.10 (or later)	GFK-1860A	GFK-1861A

¹ VersaPro and Logic Developer PLC 2.10 provide both configuration and PLC programming capability. Remote I/O Manager only provides configuration capability; it is normally used to configure applications where the GE ENIU is controlled by a third-party CPU.

A

- Add modules to autoconfiguration, 3-16
- Addresses, IP and MAC, B-1
- ARP command, 2-16
- Autoconfiguration, 3-1, 3-2, 3-15

B

- Baud rates, 1-4
- Binary
 - conversion, C-1
- Bootloader mode, 2-14

C

- Cable
 - Ethernet types, 2-7
 - installing Ethernet, 2-7
- Cables, VersaMax, 7-8
- Carriers
 - figures, 7-7
 - table, 7-7
- Catalog number, 1-2
- CE Mark installation requirements, 2-13
- Classes
 - network, B-2
- Clear All Faults bit, 6-7
- Clearance required, 2-3
- Color code on modules, 7-4
- Communications settings
 - creating, 3-11
 - editing, 3-12
- Configuration
 - basic steps, 3-8
 - clearing, 3-16
 - comparing, 3-14
 - deleting, 3-14
 - EGD consumed exchange, 3-24
 - EGD produced exchange, 3-18, 3-24
 - loading, 3-13
 - software, 3-6, 3-11
 - storing, 3-13
- Configuring
 - ENIU parameters, 3-9
 - I/O references, 3-5
- Connecting

- network devices, 2-7
- Consumed exchange
 - configuring, 3-24
- Conversion
 - number table, C-1

D

- Description, 1-3
- DeviceNet NIU User's Manual, 1-1
- DIN-rail, 2-2
 - mounting, 1-3, 2-2
 - type, 2-2
- Documentation, 1-1

E

- EGD
 - data size, 1-9
 - exchange config, 3-18, 3-24
 - fault example, 3-36
 - status and control, 3-32
 - worksheet, 3-30, 3-31
- ESD protection
 - CE Mark requirements, 2-13
- Extra I/O Module fault, 3-15

F

- Fault table, 6-7
- Faults
 - Extra I/O Module, 3-15
 - fatal, 3-13
 - format, 6-7
 - LED, 3-13
- Field Power LED, 7-4
- FTB protection
 - CE Mark requirements, 2-13

G

- Gateways, B-3
- General product specs., 7-11
- Genius NIU User's Manual, 1-1
- Glossary, A-1, D-1

Index

H

- Hexadecimal
 - conversion table, C-1
- Hot inserting modules, 3-17
- Hot insertion, 7-1
- Hub connections, 2-7
- Humidity, 7-11

I

- I/O
 - max. number, 1-9
- I/O carriers, 7-4
 - installation, 2-2
- I/O data sizes, 1-4
- I/O modules
 - autoconfiguration, 3-16
 - figure, 7-4
 - table, 7-5
- I/O, configuring, 3-5
- Input data, 1-8, 1-9
- Inserting modules, 3-17
- Installing
 - additional modules, 2-6
- IP address
 - determining, 6-13
 - forcing, 2-15
 - setting local, 2-14
- IP Addresses, B-1

K

- Keying dials on carrier, 7-4

L

- LEDs, 1-4
 - descriptions, 6-2
- Load, Store, Verify, Clear, 3-11

M

- MAC Address, B-7
 - determining, 6-11
- Manuals, 1-1
- Mapping

- modbus to ENIU memory, 4-3

- Mask, subnet, B-4

- Modbus

- function codes, 4-5

- Protocol, 4-2, 5-3

- reference tables, 4-3

- Module color code, 7-4

- Module dimensions, 7-4

- Module keying, 7-4

- Module latch, 7-4

- Module orientation on I/O carriers, 7-7

- Modules per station, 1-4, 7-1

- Mounting

- DIN-rail, 1-3

- instructions, 2-2

- Mounting holes, 2-2

N

- Network address, 1-4

- Network address, setting, 2-14

- Network classes, B-2

O

- OK LED, 7-4

- Output data, 1-8, 1-9

P

- Panel mounting, 2-2

- Power supplies, 7-3

- Power supply installation, 2-5

- Power-up sequence, 6-2

- Produced exchange

- configuring, 3-18, 3-24

- Profibus NIU User's Manual, 1-1

- Protocol

- Modbus, 4-2, 5-3

R

- Racks and slots, 3-4

- Read_DP_Slave_Diagnostic_Information, 4-3, 4-5

- Reference address assignment, 3-16

- Remote I/O Manager

- software, 1-4

- Removing

from DIN-rail, 2-4
Rotary switches, 2-14
Routers, B-3

S

Screws, 2-2
Shock, 7-11
Slots, 3-1, 3-2, 3-15
Software
 configuration, 1-4
Software configuration, 3-2
Specifications, 1-4
 System, 7-11
Status
 checking with LEDs, 6-2
Subnet, B-4
 Example, B-5
 masks, B-4
Surge protection, 2-13

T

Telnet command, 2-16
Temperature, 7-11
Terminal strips, 7-7

V

VersaMax PLC User's Manual, 1-1
VersaPro
 software, 1-4
Vibration, 7-11
Vibration resistance, 2-2

