



INDUSTRIAL CONTROL COMMUNICATIONS, INC.



# INVERTER

Plug-in option

## FR-A7N-ETH

### INSTRUCTION MANUAL

*Ethernet multiprotocol communication interface*



July 2015  
ICC #10653-2.333-000

PRE-OPERATION INSTRUCTIONS	1
INSTALLATION	2
INVERTER SETTINGS	3
FINDER APPLICATION	4
WEB SERVER	5
PARAMETER NUMBERING	6
FILESYSTEM & FIRMWARE	7
PROTOCOL-SPECIFIC INFO	8

Thank you for choosing this ICC, Inc. plug-in option for the Mitsubishi 700 Series Inverter. This instruction manual provides handling information and precautions for use of this equipment. Incorrect handling may cause unexpected failures or damage. In order to ensure optimal performance, please read this manual carefully prior to use of the equipment. Please forward this manual to the end user of the equipment.

### This section pertains specifically to safety issues

Do not attempt to install, operate, maintain or inspect this product until you have read through this instruction manual and any related documents carefully, and can use the equipment properly. Do not use this product until you have a full working knowledge of the equipment, safety information and instructions. In this instruction manual, the safety instruction levels are classified into "WARNING" and "CAUTION" levels.



Assumes that incorrect handling may cause hazardous conditions resulting in death or severe injury.



Assumes that incorrect handling may cause hazardous conditions resulting in moderate or slight injury, or may cause physical damage only.



Please note that even the **CAUTION** level may lead to serious consequence depending on conditions. Please be sure to follow the instructions of both levels as they are critical to personnel safety.

## SAFETY INSTRUCTIONS

### 1. Electrical Shock Prevention



- Do not open the front cover of the inverter while power is on or while the inverter is running, as an electrical shock may result.
- Do not operate the inverter with the front cover or wiring cover removed, as accidental contact with exposed high-voltage terminals and internal components may occur, resulting in an electrical shock.
- If power is off, do not remove the front cover except when necessary for wiring or periodic inspection. While the front cover is removed, accidental contact with exposed high-voltage terminals and internal components may occur, resulting in an electrical shock.
- Prior to starting wiring or inspection, confirm that input power to the inverter has been switched off via observation of the inverter's display panel. Additionally, wait for at least 10 minutes after removal of input power, and then confirm that all residual voltage has been dissipated by using a voltage meter. Internal DC bus capacitors may contain high voltages for several minutes after removal of input power, resulting in a dangerous situation should anything come into contact with them.
- All personnel involved in the installation or inspection of this equipment should be fully competent to perform the required work.
- Always install plug-in options prior to wiring main power.
- Do not touch the plug-in option with wet hands.
- Do not subject the cables to scratches, excessive stress, heavy loads or pinching.

## 2. Injury Prevention

### CAUTION

- To prevent explosions or similar damage, apply only the voltages specified in the instruction manual to each terminal.
- To prevent explosions or similar damage, ensure that all cables are properly connected to the correct terminals.
- To prevent explosions or similar damage, observe all wiring polarity indicators.
- To prevent burns from hot components, do not touch the inverter while power is on, or for some time after power is removed.

## 3. Additional Instructions

Please note the following points to prevent equipment damage, injury or electrical shock.

### 1) Transportation and Mounting

#### CAUTION

- Do not install or operate the plug-in option if it is damaged or has parts missing.
- Do not stand on or rest heavy objects on the equipment.
- Check that the mounting orientation is correct.
- Prevent conductive items such as screws and metal fragments, or flammable substances such as oil from entering the inverter.

### 2) Trial Run

#### CAUTION

- To prevent unexpected equipment movement, confirm and adjust all required parameters prior to starting operation.

## 3) Usage

### WARNING

- Do not modify the equipment.
- Do not remove any inverter or option parts unless specifically instructed to do so in this manual.

### CAUTION

- Performing a “parameter clear” or “all parameter clear” will reset all inverter parameters to their factory default settings. After performing one of these operations, remember to reenter any custom parameter values prior to starting operation.
- To prevent damage from electrostatic discharge, always touch a grounded piece of metal prior to touching any equipment.

## 4) Maintenance, Inspection and Parts Replacement

### CAUTION

- Do not perform hi-pot tests on the equipment.

## 5) Disposal

### CAUTION

- Contact the local or state environmental agency in your area for details on the disposal of electrical components and packaging.

## 6) General Instructions

For clarity purposes, illustrations in this manual may be drawn with covers or safety guards removed. Ensure all covers and safety guards are properly installed prior to starting operation.

– CONTENTS –

<b>1</b>	<b>PRE-OPERATION INSTRUCTIONS</b>	<b>1</b>
<b>1.1</b>	<b>Product Overview</b>	<b>1</b>
<b>1.2</b>	<b>Unpacking and Product Confirmation</b>	<b>5</b>
1.2.1	Shipment Confirmation	5
1.2.2	Component Overview	7
<b>1.3</b>	<b>LED Indicators</b>	<b>8</b>
<b>1.4</b>	<b>Environmental Specifications</b>	<b>10</b>
<b>2</b>	<b>INSTALLATION</b>	<b>11</b>
<b>2.1</b>	<b>Pre-Installation Instructions</b>	<b>11</b>
<b>2.2</b>	<b>A700 &amp; F700 Installation Procedure</b>	<b>12</b>
<b>2.3</b>	<b>E700 Installation Procedure</b>	<b>15</b>
<b>2.4</b>	<b>Wiring</b>	<b>17</b>
<b>3</b>	<b>INVERTER SETTINGS</b>	<b>18</b>
<b>3.1</b>	<b>RS-485 Communication Settings</b>	<b>20</b>
3.1.1	RS-485 Communication Station (Pr. 331)	20
3.1.2	RS-485 Communication Speed (Pr. 332)	21

3.1.3	RS-485 Communication Parity Check Selection (Pr. 334).....	21
3.1.4	PU Communication Station Number (Pr. 117).....	22
3.1.5	PU Communication Speed (Pr. 118).....	22
3.1.6	PU Communication Parity Check (Pr. 120).....	23
3.1.7	PU Communication Check Time Interval (Pr. 122).....	23
3.1.8	PU Communication Waiting Time Setting (Pr. 123).....	24
3.1.9	Protocol Selection (Pr. 549).....	24
<b>3.2</b>	<b>Operation Mode Setting .....</b>	<b>25</b>
3.2.1	Operation Mode Indication.....	25
3.2.2	Operation Mode Switching & Comm. Startup Mode (Pr. 79, Pr. 340).....	26
<b>3.3</b>	<b>Operation &amp; Speed Command Source (Pr.338, Pr.339, Pr.550) .....</b>	<b>30</b>
<b>3.4</b>	<b>Communication EEPROM Write Selection (Pr. 342) .....</b>	<b>34</b>
<b>4</b>	<b>FINDER APPLICATION .....</b>	<b>35</b>
<hr/> <hr/>		
<b>4.1</b>	<b>Overview .....</b>	<b>35</b>
<b>4.2</b>	<b>Configuring the IP Address .....</b>	<b>37</b>
4.2.1	Via the Finder Utility.....	37
4.2.2	Via the Web Page.....	38
<b>5</b>	<b>EMBEDDED WEB SERVER .....</b>	<b>39</b>
<hr/> <hr/>		
<b>5.1</b>	<b>Overview .....</b>	<b>39</b>

<b>5.2</b>	<b>Authentication</b> .....	<b>42</b>
<b>5.3</b>	<b>Page Select Tabs</b> .....	<b>43</b>
<b>5.4</b>	<b>Monitor Tab</b> .....	<b>44</b>
5.4.1	Information Window .....	44
5.4.2	Parameter Group Selection List .....	44
5.4.3	Parameter List .....	46
5.4.4	Parameter List Filter.....	48
5.4.5	Non-Scanned Parameter Refresh .....	49
5.4.6	Radix Selection.....	50
<b>5.5</b>	<b>PROFINET Tab</b> .....	<b>51</b>
5.5.1	Information Window .....	52
5.5.2	Device Identification and Configuration.....	52
5.5.3	I/O Data Configuration Arrays .....	53
5.5.4	Submitting Changes.....	55
<b>5.6</b>	<b>BACnet Tab</b> .....	<b>56</b>
5.6.1	Information Window .....	57
5.6.2	Device Identifiers .....	57
5.6.3	Submitting Changes.....	58
<b>5.7</b>	<b>Config Tab</b> .....	<b>59</b>
5.7.1	Information Window .....	60
5.7.2	Authentication Configuration.....	60
5.7.3	IP Address Configuration .....	61
5.7.4	MAC Address Configuration.....	61

5.7.5	Timeout Configuration .....	62
5.7.6	Submitting Changes .....	64
<b>5.8</b>	<b>EtherNet/IP Tab .....</b>	<b>65</b>
5.8.1	Information Window .....	66
5.8.2	Device Identification .....	66
5.8.3	Run/Idle Flag Behavior .....	66
5.8.4	Class 1 (I/O) Data Configuration Arrays .....	67
5.8.5	Submitting Changes .....	69
<b>5.9</b>	<b>Alarm Tab .....</b>	<b>70</b>
5.9.1	Information Window .....	71
5.9.2	Email Configuration .....	72
5.9.3	Alarm Configuration .....	74
5.9.4	Submitting Changes .....	76
<b>5.10</b>	<b>Modbus Tab .....</b>	<b>78</b>
5.10.1	Information Window .....	79
5.10.2	Supervisory Timer Selection .....	79
5.10.3	Connection Timer Selection .....	80
5.10.4	Register Remap Configuration .....	80
5.10.5	Submitting Changes .....	83
<b>5.11</b>	<b>MELSEC Tab .....</b>	<b>84</b>
5.11.1	Information Window .....	85
5.11.2	TCP Port .....	85
5.11.3	Submitting Changes .....	86

<b>5.12</b>	<b>Dashboard Tab</b> .....	<b>87</b>
5.12.1	Information Window .....	88
5.12.2	Gauge Window Navigation.....	88
5.12.3	Gauge Window Configuration .....	88
5.12.4	Submitting Changes.....	94
<b>6</b>	<b>PARAMETER NUMBERING AND BEHAVIOR</b>	<b>95</b>
<hr/> <hr/>		
<b>6.1</b>	<b>Register Numbers</b> .....	<b>95</b>
<b>6.2</b>	<b>Scanned and Non-Scanned Registers</b> .....	<b>95</b>
<b>6.3</b>	<b>Inverter Command and Status Register</b> .....	<b>98</b>
<b>6.4</b>	<b>Inverter Operation Mode Setting and Status Register</b> .....	<b>100</b>
<b>6.5</b>	<b>Universal Access Registers</b> .....	<b>101</b>
6.5.1	Overview .....	101
6.5.2	Universal Access Reads .....	103
6.5.3	Universal Access Writes .....	104
6.5.4	Disabling .....	105
<b>7</b>	<b>FILESYSTEM &amp; FIRMWARE</b>	<b>106</b>
<hr/> <hr/>		
<b>7.1</b>	<b>Overview</b> .....	<b>106</b>
<b>7.2</b>	<b>Initiating FTP via the Finder Utility</b> .....	<b>108</b>
<b>7.3</b>	<b>Using FTP with Windows Explorer</b> .....	<b>110</b>



<b>7.4</b>	<b>Using FTP with a Windows Command Prompt.....</b>	<b>112</b>
<b>7.5</b>	<b>Using FTP with Core FTP LE .....</b>	<b>114</b>
<b>7.6</b>	<b>Loading New Application Firmware.....</b>	<b>116</b>
<b>8</b>	<b>PROTOCOL-SPECIFIC INFORMATION</b>	<b>118</b>
<hr/> <hr/>		
<b>8.1</b>	<b>Modbus/TCP .....</b>	<b>118</b>
8.1.1	Overview .....	118
8.1.2	Coil & Discrete Input Mappings .....	120
<b>8.2</b>	<b>EtherNet/IP.....</b>	<b>122</b>
8.2.1	Overview .....	122
8.2.2	ODVA AC/DC Drive Profile .....	124
8.2.3	ControlLogix Examples: Setup .....	132
8.2.4	ControlLogix Example: I/O Messaging .....	134
8.2.5	ControlLogix Example: Generic Default I/O Add-On Instruction .....	139
8.2.6	ControlLogix Example: AC/DC Drive Profile Add-On Instruction .....	143
8.2.7	Explicit Messaging Tag Reference .....	147
8.2.8	ControlLogix Example: Read a Register Block .....	150
8.2.9	ControlLogix Example: Read a Single Register.....	158
8.2.10	ControlLogix Example: Multiple MSG Instructions.....	158
8.2.11	ControlLogix Example: Reading and Writing .....	160
<b>8.3</b>	<b>Allen Bradley CSP .....</b>	<b>162</b>
8.3.1	Overview .....	162

8.3.2	Tag Reference .....	162
8.3.3	SLC-5/05 Example: Read a Register Block.....	165
8.3.4	SLC-5/05 Example: Read a Single Register .....	172
8.3.5	SLC-5/05 Example: Multiple MSG Instructions .....	173
8.3.6	SLC-5/05 Example: Reading and Writing.....	174
<b>8.4</b>	<b>BACnet/IP .....</b>	<b>176</b>
8.4.1	Protocol Implementation Conformance Statement .....	176
8.4.2	Supported Objects .....	183
8.4.3	Supported Object Details .....	187
<b>8.5</b>	<b>MELSEC .....</b>	<b>192</b>
8.5.1	Overview.....	192
<b>8.6</b>	<b>PROFINET IO.....</b>	<b>195</b>
8.6.1	Overview.....	195
8.6.2	PROFIdrive Profile.....	197
8.6.2.1	PROFIdrive standard telegram 1 .....	197
8.6.2.2	PROFIdrive control and status words .....	198
8.6.2.3	PROFIdrive reference speed setpoint and actual speed.....	201
8.6.2.4	PROFIdrive state diagram .....	203
8.6.2.5	PROFIdrive-Specific Parameters.....	205
<b>9</b>	<b>TROUBLESHOOTING .....</b>	<b>206</b>

---



---

# 1 PRE-OPERATION INSTRUCTIONS

## 1.1 Product Overview

The FR-A7N-ETH Ethernet multiprotocol communication interface allows information to be transferred seamlessly between a 700-series inverter and several different Ethernet-based fieldbus networks with minimal configuration requirements. The interface installs directly onto the inverter's control board, and presents a standard 10/100BaseT Ethernet port for connection to the Ethernet network. In addition to the supported fieldbus protocols, the interface also hosts an embedded web server, which provides access to inverter information via a standard web browser for remote monitoring, configuration and control.

The option card communicates to the inverter via its built-in RS-485 communication port, located in the upper-left hand corner of the inverter's control board. Note that because the inverter's RS-485 port is used by the FR-A7N-ETH card, it is therefore unavailable for use by any other network when the FR-A7N-ETH interface is installed. **The RS-485 communication settings must first be configured before any Ethernet communication is possible with the card.**

Before using the interface, please familiarize yourself with the product and be sure to thoroughly read the instructions and precautions contained in this manual. In addition, please make sure that this instruction manual is delivered to the end user of the interface, and keep this instruction manual in a safe place for future reference or unit inspection.



Note that different interface firmware versions may provide varying levels of support for the various protocols. When using this manual, therefore, always keep in mind that the firmware version running on your interface must match this manual's respective revision in order for all documented aspects to apply.

The primary features of the FR-A7N-ETH are as follows:

### **Ethernet Port**

IEEE 802.3 10BASE-T/100BASE-TX Ethernet compliant. Shielded RJ45 connector accepts standard CAT5-type 8-conductor unshielded twisted-pair (UTP) patch cables. The single Ethernet port supports multiple simultaneous protocols. The port is set for auto-negotiation to automatically select the network speed and duplex.

### **Supported Protocols**

The interface currently provides server support for the following fieldbus protocols:

- Modbus/TCP Server
- EtherNet/IP Server
- Allen Bradley CSP Server (also known as “PCCC” and “AB Ethernet”)
- BACnet/IP Server
- Mitsubishi MELSEC Server (also known as “MC protocol”)
- PROFINET IO

Note that use of PROFINET IO is mutually exclusive of the other supported protocols. In order to use PROFINET IO, a separate application firmware file must be loaded into the interface (refer to section 7.6).

### **Adobe® Flash-Enabled Embedded Web Server**

Interface configuration and real-time inverter parameter monitoring & control are provided via an embedded web server. The interface's web server feature provides direct data access and control via standard web browsers such as Microsoft Internet Explorer and Mozilla Firefox. The latest version of Adobe Flash Player browser plug-in is required. Refer to section 5.

### **XML Configuration File Upload/Download**

All interface configuration files are stored in the unit's internal filesystem in XML format. These files can be transferred to/from a PC via the FTP protocol, which provides the capability for PC-based file backup and easy configuration copying to multiple units. Configuration files can also be viewed and edited via standard text editors, XML editors and web browsers. Refer to section 7.

### **Email-Based Alarm Notifications**

Up to 20 configurable alarm conditions can be programmed into the interface. Value, logical comparison and time-based conditions can be provided for the interface to autonomously monitor any available inverter register. When an alarm condition is triggered, a notification email can be sent to up to four destination email addresses. Refer to section 5.9.

### **Dashboard GUI**

A dashboard tab on the embedded web server provides 10 gauge windows, each of which can be configured to display any available scanned inverter register in a variety of meter, graph and gauge formats. Refer to section 5.11.



### **Network Timeout Action**

A configurable network timeout action can be programmed that allows parameters to have their own unique "fail-safe" conditions in the event of a network interruption. Refer to section 5.7.5.

### **Field-Upgradeable**

As new firmware becomes available, the interface can be upgraded in the field by the end-user. Refer to section 7.6 for more information.

### **EtherNet/IP Data Access Options**

The EtherNet/IP protocol provides access to inverter data via explicit messaging, user-defined I/O assembly instances, and the ODVA AC/DC drive profile. Refer to section 8.2 for more information.





### **PROFINET Data Access Options**

The PROFINET protocol provides access to inverter data via acyclic services, user-defined cyclic I/O modules, and the PROFIdrive profile. Refer to section 8.5 for more information.

## 1.2 Unpacking and Product Confirmation

### 1.2.1 Shipment Confirmation

Check the enclosed items. Confirm that the correct quantity of each item was received, and that no damage occurred during shipment.

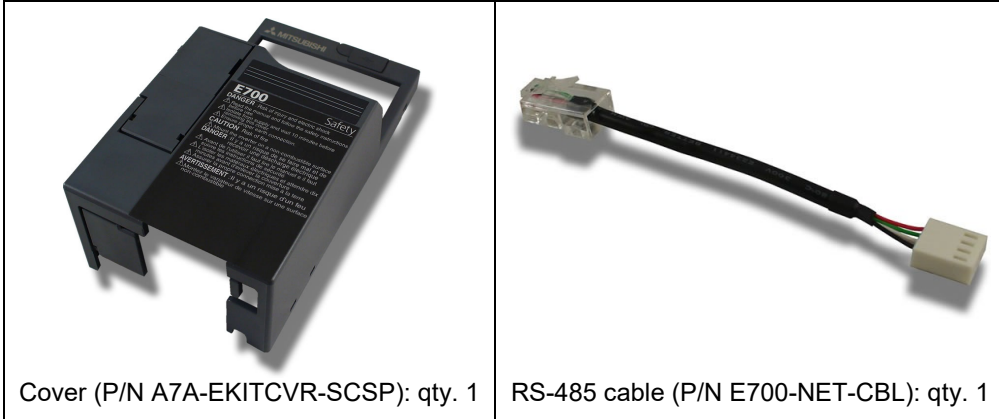
 <p>ICE FR-47N-ETH Industrial Control Communications, Inc. MAC 98405D 28C3A0D 4401ms US Patent 6,444,192</p>	 <p>RS-485 cable (P/N 10621): qty. 1</p>
 <p>Hex-head standoff (5.5mm): qty. 1</p>	 <p>Mounting screws (M3 x 6mm): qty. 2</p>

1

## PRE-OPERATION INSTRUCTIONS

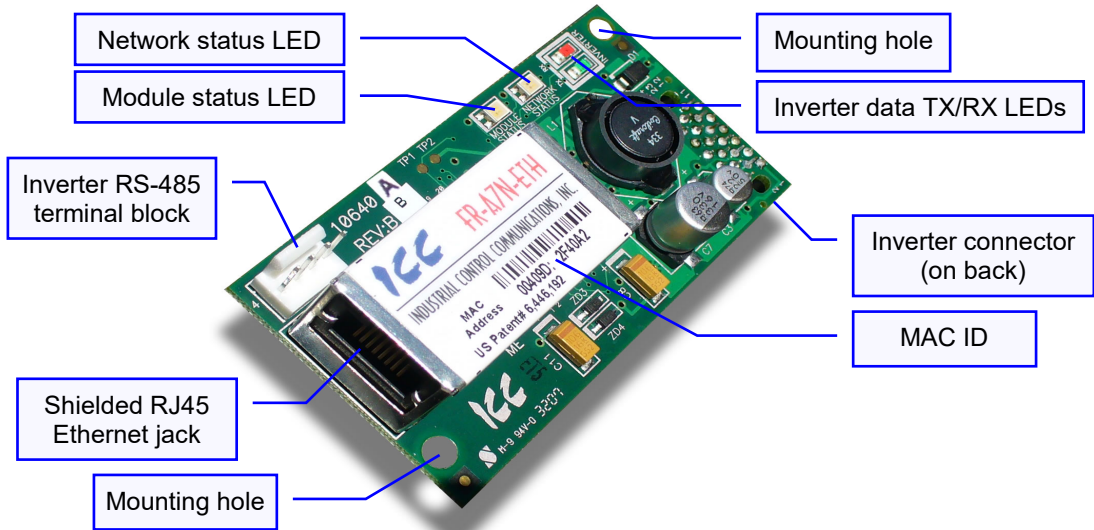


Installation on an E700 series inverter requires an optional cover and cable, which are available from Mitsubishi.





## 1.2.2 Component Overview



1



## 1.3 LED Indicators

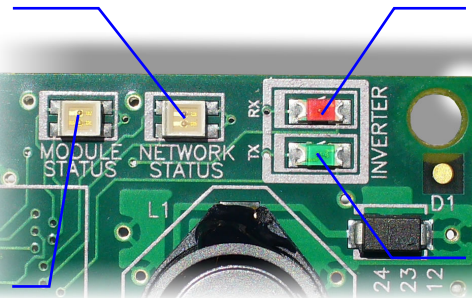
The upper right-hand corner of the option board contains several LEDs that provide a visual indication of the unit's overall status and communications activity with the inverter.

### Network Status LED

- When the multi-protocol firmware image (with EtherNet/IP support) is loaded, this LED conforms to the prescribed "network status LED" behavior as dictated in the EtherNet/IP specification, Volume 2, Chapter 9.
- When the PROFINET IO firmware image is loaded, this LED is on solid green when the controller has established a link with the interface board and is communicating with it.

### Module Status LED

- When the multi-protocol firmware image (with EtherNet/IP support) is loaded, this LED conforms to the prescribed "module status LED" behavior as dictated in the EtherNet/IP specification, Volume 2, Chapter 9.
- When the PROFINET IO firmware image is loaded, this LED is always off.
- If a blinking red error code is observed, refer to TROUBLESHOOTING section.



### Inverter Data RX

- Blinks when the interface board receives data from the inverter.
- Blinks rapidly when the interface board is running normally.

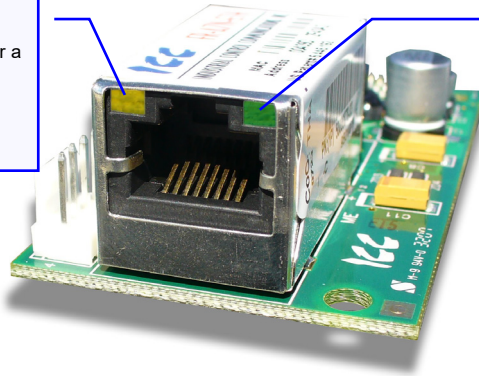
### Inverter Data TX

- Blinks when the interface board transmits data to the inverter.
- Blinks rapidly when the interface board is running normally.

The Ethernet jack also contains two embedded LEDs that provide insight into the Ethernet network.

### Ethernet Link LED

This amber LED is lit whenever a viable Ethernet network is connected to the port.



### Ethernet Activity LED

This green LED blinks briefly when network packets are sent or received. The LED may appear solid green if there is a large amount of network traffic.



## 1.4 Environmental Specifications

Item	Specification
Operating Environment	Indoors, less than 1000m above sea level, do not expose to direct sunlight or corrosive / explosive gasses
Operating Temperature	-10 ~ +50°C (+14 ~ +122°F)
Storage Temperature	-40 ~ +85°C (-40 ~ +185°F)
Relative Humidity	20% ~ 90% (without condensation)
Vibration	5.9m/s <sup>2</sup> (0.6G) or less (10 ~ 55Hz)
Grounding	Referenced to inverter's 24V power supply / isolated from inverter control power common
Power supply	Supplied from inverter
Cooling Method	Self-cooled
Communication Speed	10/100BaseT auto sensing

The FR-A7N-ETH interface is lead-free / RoHS-compliant.



Lead Free

## 2 INSTALLATION

### 2.1 Pre-Installation Instructions

**Make sure that the inverter's input power is off.**



#### **CAUTION**



**To avoid damage to the inverter or plug-in option card, never install or remove a plug-in option card while the inverter's input power is on.**

**2**

#### **A700 and F700 Inverters**

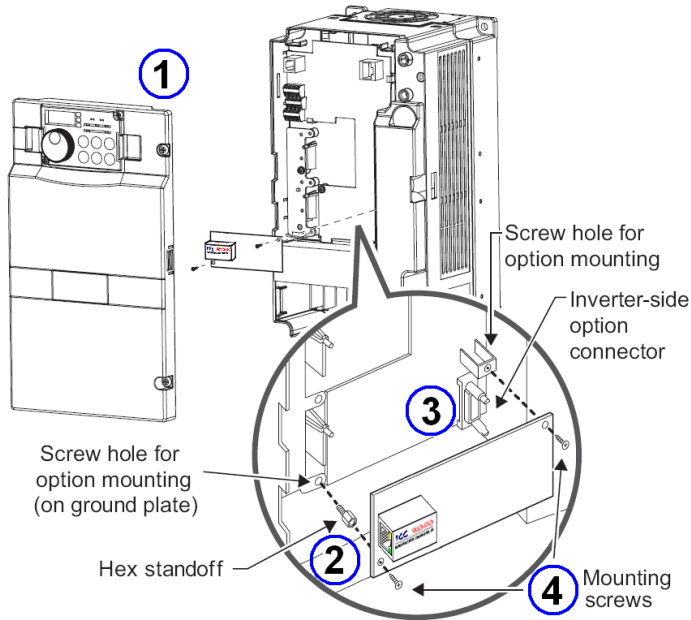
Physical installation of the option card is a two-step process. First, the card will be mounted onto an available option connector on the inverter's control board. Second, the card will be connected to the inverter's RS-485 communication port via the included #10621 RS-485 cable.

#### **E700 Inverters**

Physical installation of the option card is a two-step process. First, the card will be mounted onto the option connector on the inverter's control board. Second, the card will be connected to the inverter's RS-485 communication port (a.k.a. PU port) via the optional E700-NET-CBL RS-485 cable.



## 2.2 A700 & F700 Installation Procedure



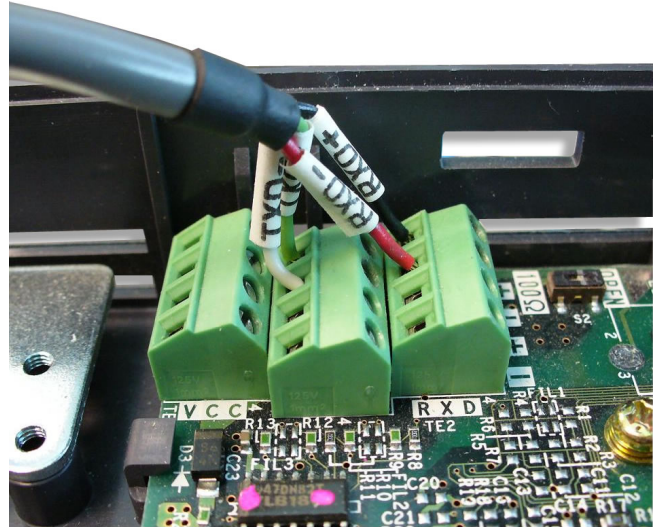
1) Remove the inverter's front cover.

2) Locate an open option connector and screw the included 5.5mm hex standoff into the corresponding ground plate screw hole (rated torque 0.56N·m to 0.75N·m).

3) Securely attach the option card to the inverter's option connector. Ensure that the option card is fully seated on the inverter's option connector and the hex standoff.

4) Secure the upper-right and lower-left corners of the option card with the included M3x6mm mounting screws. If the screw holes do not line up, the option card connector may not be fully seated on the inverter's option connector and the hex standoff.

5) Connect the stripped-wire end of the #10621 RS-485 cable to the inverter's RS-485 terminal blocks, and set the termination switch located on the inverter's control board to the "OPEN" position. Each of the wires is individually labeled with the name of the terminal signal to which they must be connected, and are pre-stripped for ease of installation. Connect the TXD+ (green) and TXD- (white) wires to the "TXD" terminal block, positions 1 & 2, respectively. Then connect the RXD+ (black) and RXD- (red) wires to the "RXD" terminal block, positions 1 & 2, respectively. Refer to the following picture for a post-connection view of the wiring connections.

**2**

## INSTALLATION

6) Connect the 4-position plug end of the #10621 RS-485 cable to connector CN1 in the upper-left corner of the option board. Note that the connector is keyed to prevent the possibility of reverse installation. Refer to the following picture for a post-connection view.

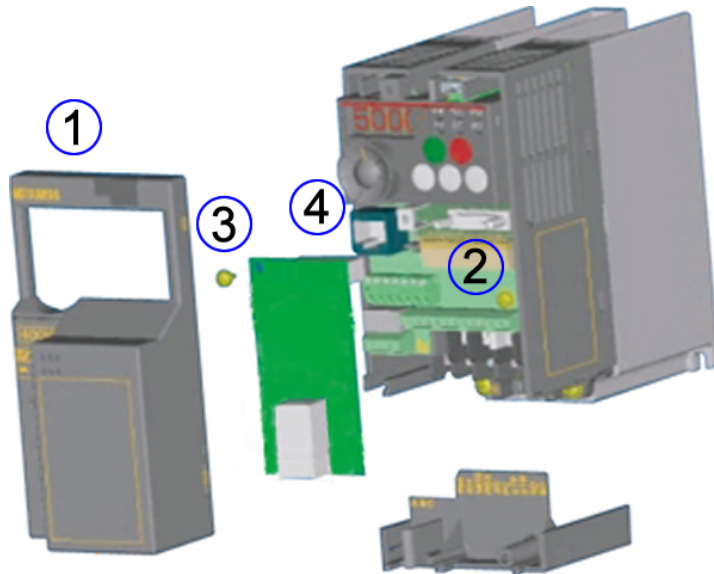


## REMOVAL

First disconnect the #10621 RS-485 cable from connector CN1 on the option board. Next, remove the two M3x6mm mounting screws. Lastly, remove the option board by grasping it on its left and right side and pulling it straight away from the inverter. Note that the removal process may be easier by first removing the inverter's control circuit terminal block.



## 2.3 E700 Installation Procedure



1) Remove the inverter's front cover.

2) Securely attach the option card to the inverter's option connector. Ensure that the option card is fully seated on the option connector.

3) Secure the upper-left corner of the option card with the included M3x6mm mounting screw.

4) Connect the RJ45 end of the E700-NET-CBL RS-485 cable to the inverter's RJ45 port (a.k.a. PU port).

5) Connect the 4-position plug end of the E700-NET-CBL RS-485 cable to connector CN1 in the bottom-left corner of the option board. Note that the connector is keyed to prevent the possibility of reverse installation.

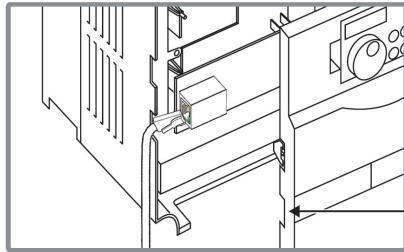
**2**

**REMOVAL**

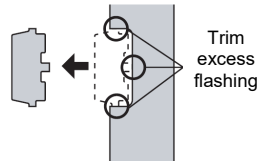
First disconnect the E700-NET-CBL RS-485 cable from connector CN1 on the option board. Next, remove the M3x6mm mounting screw. Lastly, remove the option board by grasping it on its left and right side and pulling it straight away from the inverter.

## 2.4 Wiring

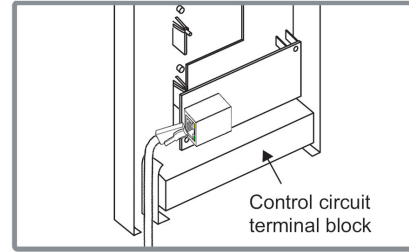
When installing the option card into an FR-A720-00900-NA (FR-A740-00440-NA) or smaller inverter, remove the wiring access knockout on the front cover and route the network cable through the opening. When installing the option card into an FR-A720-01150-NA (FR-A740-00570-NA) or larger inverter, route the network cable through the space adjacent to the control circuit terminal block.



FR-A720-00900-NA (FR-A740-00440-NA)  
and smaller




Remove front cover wiring access knockout and trim any excess flashing that may cause cable damage



FR-A720-01150-NA (FR-A740-00570-NA)  
and larger

**NOTE:** If the front cover wiring access knockout is removed, the protective structure (JEM1030) changes to open type (IP00).

### CAUTION

 Use caution during wiring to prevent any cable fragments and wire strands from falling into the inverter. Equipment damage may result if power is applied to the inverter in the presence of conductive debris.



### 3 INVERTER SETTINGS

The inverter parameters listed in Table 1 and Table 2 are critical for overall operation of the end-to-end communication system. Some of these parameters must be set to specific values, and some may have multiple allowable settings depending on the desired operation of the overall application. Although there may be many other inverter parameters that will require configuration for your specific application, it is important to understand the manner in which the following parameters will impact successful communications with, and control of the inverter.

**Table 1: A700 & F700 Inverter Settings**

<b>Parameter Number</b>	<b>Name</b>	<b>Refer to Page</b>
79	Operation mode selection	26
331	RS-485 communication station	20
332	RS-485 communication speed	21
334	RS-485 communication parity check selection	21
338	Communication operation command source	30
339	Communication speed command source	30
340	Communication startup mode selection	26
342	Communication EEPROM write selection	34
549	Protocol selection	24
550	NET mode control source selection	30

**Table 2: E700 Inverter Settings**

<b>Parameter Number</b>	<b>Name</b>	<b>Refer to Page</b>
79	Operation mode selection	26
117	PU communication station number	22
118	PU communication speed	22
120	PU communication parity check	23
122	PU communication check time interval	23
123	PU communication wait time setting	24
338	Communication operation command source	30
339	Communication speed command source	30
340	Communication startup mode selection	26
342	Communication EEPROM write selection	34
549	Protocol selection	24
550	NET mode operation command source selection	30



## 3.1 RS-485 Communication Settings

Because the FR-A7N-ETH option card communicates with the inverter via the inverter's on-board RS-485 port, certain RS-485 -related inverter parameters must be set appropriately in order to allow the option card to successfully exchange data with the inverter. If any one of these parameters is not correctly configured, the FR-A7N-ETH card will not be able to communicate with the inverter. If initial communication to the inverter fails, the FR-A7N-ETH will fall into an error state. The MODULE STATUS LED will blink a 7-1-1 red error code. Note that Ethernet communication will also be disabled.

### REMARKS

Whenever any of the RS-485 communication setting parameters is changed, power to the inverter must be cycled to validate the changes and allow the inverter to begin communicating at the new settings.

### 3.1.1 RS-485 Communication Station (Pr. 331)

**A700 & F700 only.** Typically, *Pr. 331* would set the inverter's station number on an RS-485 network. When using the FR-A7N-ETH option card, however, *Pr. 331* must be set to a specific value so that the option card can communicate with the inverter.

Parameter Number	Name	Default Value	Setting Range	Description
331	RS-485 communication station	0	0 to 247	Must be set to a value of "1" in order to allow the option card to communicate with the inverter.

### 3.1.2 RS-485 Communication Speed (Pr. 332)

**A700 & F700 only.** Pr. 332 determines the data rate at which the option card will communicate with the inverter.

Parameter Number	Name	Default Value	Setting Range	Description
332	RS-485 communication speed	96	3, 6, 12, 24, 48, 96, 192, 384	Must be set to a value of "384" (38.4kbaud), which will allow the option card to communicate with the inverter at the fastest possible rate.

**3**

### 3.1.3 RS-485 Communication Parity Check Selection (Pr. 334)

**A700 & F700 only.** Pr. 334 determines the parity & stop bit(s) which the inverter and option card will use when communicating with each other.

Parameter Number	Name	Default Value	Setting Range	Description
334	RS-485 communication parity check selection	2	0 to 2	Must be set to a value of "2" (even parity, 1 stop bit)



### 3.1.4 PU Communication Station Number (Pr. 117)

**E700 only.** Typically, Pr. 117 would set the inverter's station number on an RS-485 network. When using the FR-A7N-ETH option card, however, Pr. 117 must be set to a specific value so that the option card can communicate with the inverter.

Parameter Number	Name	Default Value	Setting Range	Description
117	PU communication station number	0	0 to 247	Must be set to a value of "1" in order to allow the option card to communicate with the inverter.

### 3.1.5 PU Communication Speed (Pr. 118)

**E700 only.** Pr. 118 determines the data rate at which the option card will communicate with the inverter.

Parameter Number	Name	Default Value	Setting Range	Description
118	PU communication speed	192	48, 96, 192, 384	Must be set to a value of "384" (38.4kbaud), which will allow the option card to communicate with the inverter at the fastest possible rate.



### 3.1.6 PU Communication Parity Check (Pr. 120)

**E700 only.** Pr. 120 determines the parity & stop bit(s) which the inverter and option card will use when communicating with each other.

Parameter Number	Name	Default Value	Setting Range	Description
120	PU communication parity check	2	0 to 2	Must be set to a value of "2" (even parity, 1 stop bit)

### 3.1.7 PU Communication Check Time Interval (Pr. 122)

**E700 only.** Pr. 122 determines the time interval. If a no-communication state persists for longer than the permissible time, the inverter will trip.

**3**

Parameter Number	Name	Default Value	Setting Range	Description
122	PU communication check time interval	0	0, 0.1 to 999.8s, 9999	Recommended to be set to a value of 9999.



### 3.1.8 PU Communication Waiting Time Setting (Pr. 123)

**E700 only.** Pr. 123 determines the waiting time between data transmission to the inverter and response.

Parameter Number	Name	Default Value	Setting Range	Description
123	PU communication waiting time setting	9999	0 to 150ms, 9999	Recommended to be set to a value of 9999.

### 3.1.9 Protocol Selection (Pr. 549)

Pr. 549 determines whether the inverter's RS-485 port will communicate using the Mitsubishi computer-link protocol, or the Modbus RTU protocol (default).

Parameter Number	Name	Default Value	Setting Range	Description
549	Protocol selection	1	0, 1	Must be set to a value of "1" to enable the option card to communicate to the inverter via the Modbus RTU protocol.

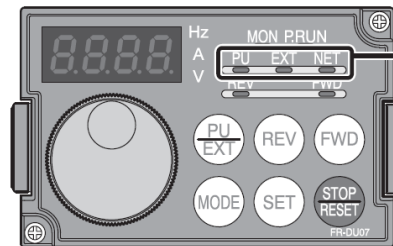
## 3.2 Operation Mode Setting

Three operation modes are available when a communication option card is installed into an inverter.

1. PU operation [PU] ..... The inverter is controlled by the operating panel (FR-DU07).
2. External operation [EXT] ... The inverter is controlled by the ON/OFF switching of external signals connected to the control circuit terminals (factory default.)
3. Network operation [NET] ... The inverter is controlled from the network via the communication option card (the operating commands and frequency command can be input via the control circuit terminals depending on the settings of Pr. 338 *Communication operation command source* and Pr. 339 *Communication speed command source*. Refer to page 31.)

### 3.2.1 Operation Mode Indication

FR-DU07



Operation mode indication (the inverter operates in accordance with the indicated LED.)

PU: PU operation mode

EXT: External operation mode

NET: Network operation mode



### **3.2.2 Operation Mode Switching & Comm. Startup Mode (Pr. 79, Pr. 340)**

#### **(1) Operation mode switching conditions**

Prior to switching the operation mode, confirm that:

- 1) The inverter is stopped
- 2) Both the STF and STR signals are off
- 3) The *Pr. 79 Operation mode selection* setting is correct. Refer to the appropriate inverter *user's manual (applied)* for further information regarding *Pr. 79*.

#### **(2) Operation mode selection at power on and after recovery from a momentary power failure**

The operation mode at power on and after recovery from a momentary power failure can be selected via *Pr. 340*. A value other than "0" will select network operation mode. After activating network operation mode, parameter writes from the network are enabled.

#### **REMARKS**


1. When *Pr. 340* is changed, the new setting is validated after powering on or resetting the inverter.
2. *Pr. 340* can be changed via the operation panel regardless of the operation mode.

<i>Pr. 340</i> Setting	<i>Pr. 79</i> Setting	Operation Mode at Power-On or Power Recovery	Operation Mode Switchover
0 (default)	0 (default)	External operation mode	Switching among external, PU, and NET operation modes is enabled <sup>*1</sup>
	1	PU operation mode	PU operation mode fixed
	2	External operation mode	Switching between external and NET operation modes is enabled, switching to PU operation mode is disallowed
	3, 4	External/PU combined operation mode	Operation mode switching is disallowed
	6	External operation mode	Switching among external, PU, and NET operation modes is enabled while running.
	7	X12 (MRS) signal ON.....external operation mode	Switching among external, PU, and NET operation modes is enabled <sup>*1</sup>
		X12 (MRS) signal OFF...external operation mode	External operation mode fixed (forcibly switched to external operation mode.)
1, 2 <sub>2</sub>	0	NET operation mode	Same as when <i>Pr. 340</i> = "0"
	1	PU operation mode	
	2	NET operation mode	
	3, 4	External/PU combined operation mode	
	6	NET operation mode	
	7	X12 (MRS) signal ON....NET operation mode	
		X12 (MRS) signal OFF...external operation mode	
10, 12 <sub>2</sub>	0	NET operation mode	Switching between PU and NET operation modes is enabled <sup>*3</sup>
	1	PU operation mode	Same as when <i>Pr. 340</i> = "0"
	2	NET operation mode	NET operation mode fixed
	3, 4	External/PU combined operation mode	Same as when <i>Pr. 340</i> = "0"
	6	NET operation mode	Switching between PU and NET operation modes is enabled while running <sup>*3</sup>
	7	External operation mode	Same as when <i>Pr. 340</i> = "0"

<sup>\*1</sup> The operation mode cannot be directly changed between PU mode and NET mode.

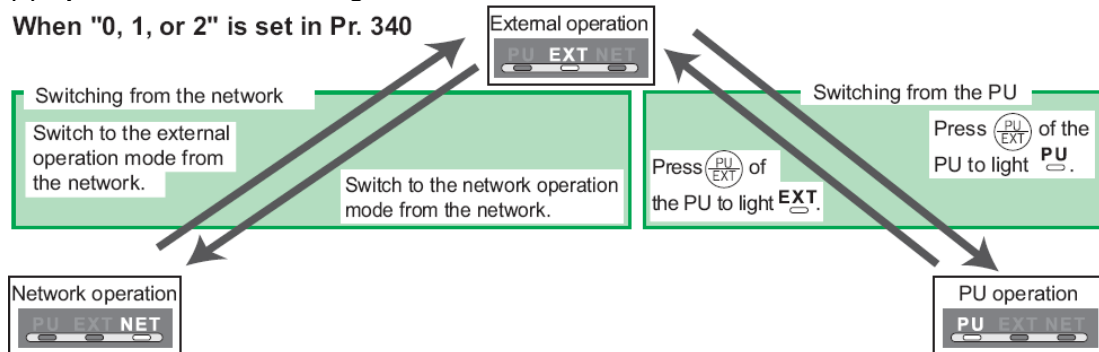


<sup>\*2</sup> *Pr. 340* settings "2" and "12" are mainly used for communication operation using the inverter's RS-485 port. When a value other than "9999" (automatic restart after momentary power failure) is set in *Pr. 57 Restart coasting time*, the inverter will resume the same operation state which it was in prior to a momentary power failure is such a failure occurs. When *Pr. 340* is set to "1" or "10" and a start command is active, then the start command will be deactivated if a momentary power failure occurs.

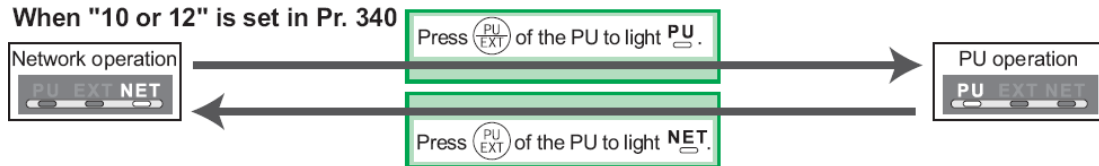
<sup>\*3</sup> The operation mode can be changed between PU mode and NET mode with the  key on the operating panel (FR-DU07) and X65 signal.

### (3) Operation mode switching method

When "0, 1, or 2" is set in Pr. 340



When "10 or 12" is set in Pr. 340



For a switching method via external terminal input signals, refer to the inverter's *user's manual (applied)*.



- When starting the inverter in NET mode upon powering-up or after an inverter reset, set a value other than 0 in Pr. 340. (Refer to page 26).
- When setting a value other than 0 in Pr. 340, make sure that the initial settings of the inverter are correct.



### 3.3 Operation & Speed Command Source (Pr.338, Pr.339, Pr.550)

#### (1) Select control source for NET mode (Pr. 550)

The control location for NET mode can be selected to be from either the inverter's RS-485 port or a plug-on communication option card. Although the option card physically plugs into the inverter's option card slot, it actually communicates to the inverter via the inverter's RS-485 port. Therefore, to control the inverter via the option card, Pr. 550 must be set to either "9999" (default) or "1" (RS-485 input valid).

Parameter Number	Name	Default Value	Setting Range	Description
550	NET mode operation command source selection	9999	0	Communication option card control is valid (FR-A7N-ETH control via the inverter's RS-485 port is invalid.)
			1	FR-A7N-ETH control via the inverter's RS-485 port is valid (communication option card control is invalid.)
			9999	Communication option automatic recognition. Normally, control via the inverter's RS-485 port is valid, which includes the situation when an FR-A7N-ETH card is installed. When a non-FR-A7N-ETH communication option card is installed, that communication option card's control is made valid instead of the inverter's RS-485 port.

Refer to the inverter's *user's manual (applied)* for further details.



## (2) Selection of control source for NET mode (Pr. 338, Pr. 339)

Control sources can be subdivided into two separate realms: 1) operation commands such as start/stop signals, etc. and 2) the speed command source that determines the inverter's frequency command. The various combinations of these realms that can be configured are summarized in the following table.

Control Location Selection	Pr. 338 Communication operation command source		0:NET			1:External			Remarks		
	Pr. 339 Communication speed command source		0:NET	1: External	2: External	0:NET	1: External	2: External			
Fixed functions (Functions equivalent to terminals)	Running frequency from communication		NET	—	NET	NET	—	NET			
	Terminal 2		—	External	—	—	External	—			
	Terminal 4		—	External		—	External				
	Terminal 1		Compensation								
Selective functions Pr. 178 to Pr. 189 settings	0	RL	Low-speed operation command/remote setting clear		NET	External		NET	External	Pr: 59 = "0" (multi-speed) Pr: 59 = "1, 2" (remote)	
	1	RM	Middle-speed operation command/remote setting deceleration		NET	External		NET	External		
	2	RH	High-speed operation command/remote setting acceleration		NET	External		NET	External		
	3	RT	Second function selection			NET		External			
	4	AU	Terminal 4 input selection			—	Combined		—	Combined	
	5	JOG	Jog operation selection			—		External			
	6	CS	Automatic restart after instantaneous power failure selection			External					
	7	OH	External thermal relay input			External					
	8	REX	15-speed selection		NET	External		NET	External	Pr: 59 = "0" (multi-speed)	
	9	X9	Third function			NET		External			
10	X10	Inverter operation enable signal			External						



Control Location Selection	Pr. 338 Communication operation command source		0:NET			1:External			Remarks		
	Pr. 339 Communication speed command source		0:NET	1: External	2: External	0:NET	1: External	2: External			
Selective functions Pr. 178 to Pr. 189 settings	11	X11	FR-HC connection, instantaneous power failure detection		External						
	12	X12	PU operation external interlock		External						
	13	X13	External DC injection brake operation is started		NET		External				
	14	X14	PID control valid terminal		NET	External		NET	External		
	15	BRI	Brake opening completion signal		NET		External				
	16	X16	PU operation-external operation switching		External						
	17	X17	Load pattern selection forward rotation reverse rotation boost		NET		External				
	18	X18	V/F swichover		NET		External				
	19	X19	Load torque high speed frequency		NET		External				
	20	X20	S-pattern acceleration/deceleration C switching terminal		NET		External				
	22	X22	Orientation command *1		NET		External				
	23	LX	Pre-excitation		NET		External				
			Output stop		Combined		External			Pr. 79 ≠ "7"	
	24	MRS	PU operation interlock		External					Pr. 79 = "7" When the X12 signal is not assigned	
	25	STOP	Start self-holding selection		—		External				
	26	MC	Control mode swichover		NET		External				
	27	TL	Torque limit selection		NET		External				
	28	X28	Start time tuning		NET		External				
	37	X37	Traverse function selection		NET		External				
	42	X42	Torque bias selection 1 *1		NET		External				
43	X43	Torque bias selection 2 *1		NET		External					
44	X44	P/PI control swichover		NET		External					

Control Location Selection	Pr. 338 Communication operation command source		0:NET			1:External			Remarks	
	Pr. 339 Communication speed command source		0:NET	1: External	2: External	0:NET	1: External	2: External		
Selective functions Pr. 178 to Pr. 189 settings	50	SQ	Sequence start	Combined			External			
	60	STF	Forward rotation command	NET			External			
	61	STR	Reverse rotation command	NET			External			
	62	RES	Reset	External						
	63	PTC	PTC thermistor selection	External						
	64	X64	PID forward rotation action switchover	NET	External		NET	External		
	65	X65	PU/NET operation switchover	External						
	66	X66	NET/external operation switchover	External						
	67	X67	Command source switchover	External						
	68	NP	Conditional position pulse train sign *1	External						
	69	CLR	Conditional position droop pulse clear *1	External						
	70	X70	DC feeding operation permission	NET			External			
71	X71	DC feeding cancel	NET			External				

\*1: Available only when used with the FR-A7AP.

[Table explanation]

External ..... Only external terminal input control is valid.

NET ..... Only network control is valid.

Combined ..... Either external terminal input control or network control is valid.

- ..... Both external terminal input control and network control are invalid.

Compensation ..... External terminal input control is only valid if Pr. 28 Multi-speed input compensation is set to "1".



### 3.4 Communication EEPROM Write Selection (Pr. 342)

When parameters are written via communications, by default both volatile RAM and nonvolatile EEPROM contents are modified. Due to the limited write cycle lifetime of EEPROM memory, however, it may be desirable to modify only the contents of RAM when frequent parameter writes via communications are necessary.

Parameter Number	Name	Default Value	Setting Range	Description
342	Communication EEPROM write selection	0	0	Parameter values modified via communications are written to both EEPROM and RAM.
			1	Parameter values modified via communications are written only to RAM.

When frequently modifying parameter values via communications, change the value of *Pr. 342* to a "1" in order to write them only to RAM. Performing frequent parameter writes to EEPROM will shorten the lifetime of the component.

#### REMARKS

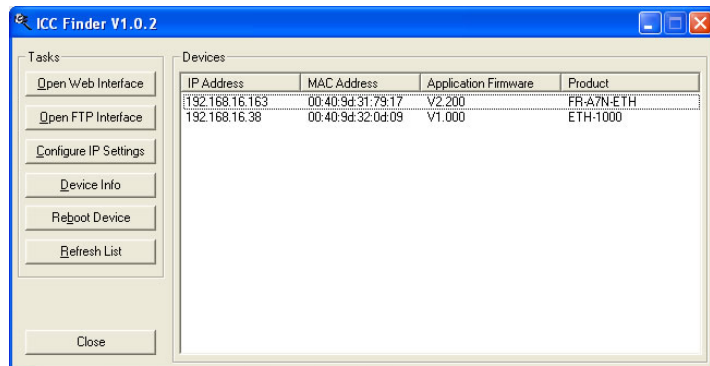
When *Pr. 342* is set to a value of "1" (write to RAM only), powering off the inverter will erase the changed parameter values. Therefore, the parameter values available when power is switched on again are those that were previously stored in EEPROM.

## 4 FINDER APPLICATION

### 4.1 Overview

The “ICC Finder” application is a simple Windows PC program (just a single .exe file, no installations, DLL’s etc.), which when executed discovers all ICC communication interfaces on the current Ethernet subnet, regardless of whether or not their network parameters are currently compatible with the subnet upon which they reside. Refer to Figure 1.

In order for the Finder application to discover devices, certain UDP Ethernet traffic must be allowed in and out of the computer, and firewall applications (such as Windows Firewall) are often configured to block such traffic by default. If the Finder is unable to discover any devices on the current subnet, be sure to check the computer’s firewall settings during troubleshooting, and add an exception to the firewall configuration if necessary.



**Figure 1: ICC Finder Discovery Utility**



All discovered devices can be organized in ascending or descending order by clicking on the desired sort header (IP Address, MAC Address, Application Firmware or Product). The buttons on the left side of the window perform the following actions:

**Open Web Interface**: Opens a web browser page of the selected device. Refer to section 5.

**Open FTP Interface**: Opens the computer's default FTP application, which could be either Windows Explorer, a web browser, or a 3<sup>rd</sup>-party FTP program (whatever the computer/operating system is configured for by default). This allows you to interact directly with the unit's on-board flash filesystem, enabling you to drag and drop files to/from the unit and upload new firmware. Refer to section 7.

**Configure IP Settings**: Allows configuration of whether the device will use static IP parameters or will obtain its IP parameters via DHCP. Refer to section 4.2 for more information.

**Device Info**: Opens a dialog box containing relevant device information.

**Reboot Device**: Opens a dialog box which prompts for a password to reboot the interface. Enter the case-sensitive system password (default is "icc"), then click **Reboot**. The reboot cycle has completed when the displayed status changes from "Rebooting" to "Ready" (note that this may require 30s or more to complete.) Clicking **Close** will then close the dialog box and cause the discovery utility to automatically rescan the network.

**Refresh List**: Causes the discovery utility to rescan the network.

**Close**: Closes the discovery utility.

## 4.2 Configuring the IP Address

---

Before you can access the interface from your web browser or begin using it as a part of your automation network, you must know its IP address. The interface comes from the factory configured to obtain an IP address dynamically (DHCP/BOOTP). You can determine the interface's current IP address using the *ICC Finder* application included on the CD provided with the interface, or available from the ICC website at <http://www.iccdesigns.com>.

### 4.2.1 Via the Finder Utility

It may be necessary to temporarily disable the PC's firewall and add the ICC Finder utility as a program exception to the firewall configuration.

To configure the interface to use a static IP address:

1. Connect the interface to your network and apply power to the inverter. When the interface boots up, it will attempt to obtain an IP address from a DHCP server or, failing that, will fall back to either the last static IP address assigned, or a default static IP address of 192.168.16.102 if no static IP address has yet been assigned.
2. To determine the initial IP address of your interface, start the `ICC FINDER.EXE` discovery utility.
3. The discovery utility scans the network for ICC devices and then lists each device's **IP Address**, **MAC Address**, **Firmware Version** and **Product ID**. Refer to Figure 1 on page 35. Identify your device by its unique MAC address (printed on a label on the top of the Ethernet network jack).



4. To change the IP address, select the device in the list of detected devices and click the **Configure IP Settings** button.
5. In the dialog that appears, select **Manually configure network settings**.
6. Enter the desired **IP Address**, **Subnet Mask** and **Default Gateway** in the appropriate boxes, and then click **Apply**.
7. Enter the case-sensitive system password (default is “icc”) in the Authentication dialog box.
8. A popup dialog box will prompt you to reboot. Click **Reboot Device**. Rebooting may require 30s or more to complete. When the device status indicates “Ready”, click **Close**.
9. The discovery utility will automatically rescan the network. Confirm that the new IP address has been accepted by the device.

### **4.2.2 Via the Web Page**

Once an initial IP address has been assigned to the device and the configuration web page can be accessed, the IP address-related parameters can also be modified via the web page. Refer to section 5.7.3.



## **5 EMBEDDED WEB SERVER**

### **5.1 Overview**

The interface contains an embedded web server (also known as an HTTP server), which allows users to access the inverter's internal data in a graphical manner with web browsers such as Microsoft Internet Explorer or Mozilla Firefox. In this way, the inverter can be monitored, configured and controlled from across the room or from across the globe.

In order to view the interface's web page, the free Adobe (formerly Macromedia) Flash Player browser plug-in is required. If the plug-in is not already installed on your computer, then your browser will automatically be redirected to the appropriate Adobe download web site when you initially attempt to access the interface's web page. Alternatively, the plug-in can be downloaded directly by going to <http://www.adobe.com>, and choosing the "get Adobe Flash Player" link. Always ensure that you have the latest version of the Flash Player installed: if some aspect of the web page does not appear to be displayed properly, installing the latest Flash Player update usually resolves the problem.

To access an interface's embedded web server, either use the finder application (refer to section 4) and select the "Open Web Interface" button when the target unit is highlighted, or just directly enter the target unit's IP address into the address (URL) field of your web browser. Refer to Figure 2 for a representative screenshot of the web server interface.

The screenshot shows the web interface for the FR-A7N-ETH Embedded Server. The browser title is "FR-A7N-ETH - Windows Internet Explorer" and the address bar shows "http://192.168.16.163". The page header includes the Mitsubishi Electric logo and the text "FR-A7N-ETH Embedded Server". Below the header is a navigation menu with tabs for "Monitor", "PROFInET", "BACnet", "Config", "EtherNet/IP", "Alarm", "Modbus", "MELSEC", and "Dashboard". The "Monitor" tab is selected.

Under the "Monitor" tab, there is a "Parameter Group" dropdown menu with options: "All", "Command registers", "Monitor registers", "Alarm history registers", and "Universal access registers". To the right of this menu, it displays "CPU firmware version: V2.310" and "Refreshing values...". Below this, there are "FILTER", "REFRESH", "DEC", and "HEX" buttons.

The main content area displays a table with the following data:

Parameter	Register	Description	Value (Decimal)
0	1000	Torque boost	0
1	1001	Maximum frequency	12000
2	1002	Minimum frequency	0
3	1003	Base frequency	8000
4	1004	Multi-speed setting (high speed)	1308
5	1005	Multi-speed setting (middle speed)	1309
6	1006	Multi-speed setting (low speed)	1310
7	1007	Acceleration time	50
8	1008	Deceleration time	50
9	1009	Electronic thermal O/L relay	770
10	1010	DC injection brake operation frequency	300
11	1011	DC injection brake operation time	5
12	1012	DC injection brake operation voltage	40
13	1013	Starting frequency	50
14	1014	Load pattern selection	1
15	1015	Jog frequency	500
16	1016	Jog acceleration/deceleration time	5
17	1017	MRS input selection	0

At the bottom right of the page, there is a logo for "icc INDUSTRIAL CONTROL CORPORATION, INC." and a status bar showing "Done" and "Internet".

Figure 2: Embedded Web Server

In order to access the web server and view the parameter values, destination TCP ports 80 and 843 must be accessible from the client computer. If an “XML socket connection failed” error message is displayed in the information window, and no parameter values are shown, this is typically indicative of port 843 being blocked by a firewall or Ethernet router situated between the client computer and the interface card.



## 5.2 Authentication

For security, the interface requires valid user authentication whenever the web page is accessed. The authentication request will appear as a browser popup box that will request entry of a user name and password. Refer to Figure 3.

The factory-default user name is “root”, and the password is “icc”. Note that the username and password are case-sensitive, and that once authenticated, the authentication will remain in effect from that point until all browser windows are closed. The authentication credentials can also be changed from their default settings (refer to section 5.7.2.)

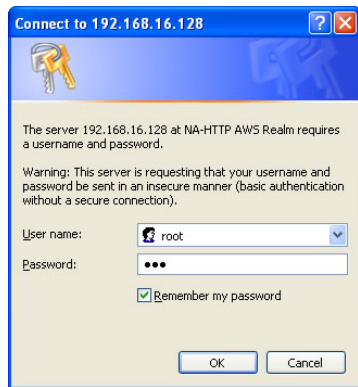
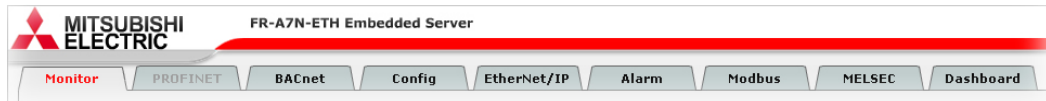


Figure 3: Web Server Authentication

## 5.3 Page Select Tabs

The web interface is subdivided into several different “tabs” of associated information, much the same as how folders in a filing cabinet are arranged. Refer to Figure 4. To change tabs, just click on the tab you wish to view. The title of the currently-selected tab is red. Note that because different protocols are supported by the interface with different firmware images, not all tabs may be accessible with the firmware image currently loaded. The titles of tabs that are not accessible are grayed-out, and clicking them has no effect.



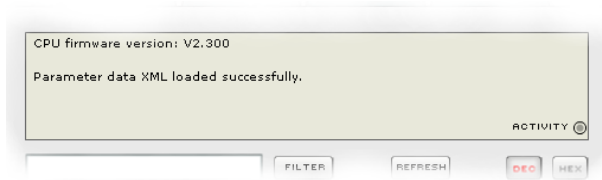
**Figure 4: Page Select Tabs**



## 5.4 Monitor Tab

### 5.4.1 Information Window

Figure 5 shows the Information Window, which is located in the upper-right hand corner of the monitor tab. This window displays various informational messages regarding the status of the interface card or web browser session. There is also an “activity” indicator located in the lower-right hand corner of the Information Window, which blinks periodically to show the status of data communication between the web browser and the interface card. If you do not observe the activity indicator blink at all for several seconds or more, it is possible that the web browser may have lost contact to the web server due to an inverter power cycle or a network problem: to reestablish communications, select “refresh” on your web browser.



**Figure 5: Monitor Tab Information Window**

### 5.4.2 Parameter Group Selection List

The Parameter Group Selection List is located in the upper-left hand corner of the Monitor Tab. Refer to Figure 6. Individual groups can be selected by clicking on the group name. Multiple groups may also be selected by holding down the CTRL key while clicking on the group names, or a range of groups can be selected by first selecting the starting group, and then holding down the SHIFT key while selecting the last group in the range. When a parameter group is selected, the parameters contained in that

parameter group are displayed in the Parameter List (refer to section 5.4.3). The following parameter groups are available:

**All:** All parameters/registers are available.

**Command registers:** Inverter command registers are available.

**Monitor registers:** Inverter monitor registers are available.

**Alarm history registers:** Inverter alarm history registers are available.

**Universal access registers:** The interface card's "universal access registers" are available (refer to section 6.5 for an explanation of the universal access registers).

**Parameters 0 to 99:** Inverter parameters *Pr. 0* to *Pr. 99* are available.

**Parameters 100 to 199:** Inverter parameters *Pr. 100* to *Pr. 199* are available.

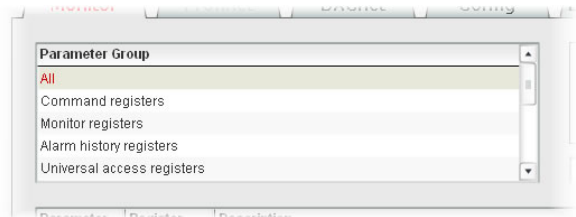
**Parameters 200 to 299:** Inverter parameters *Pr. 200* to *Pr. 299* are available.

**Parameters 300 to 399:** Inverter parameters *Pr. 300* to *Pr. 399* are available.

**Parameters 400 to 499:** Inverter parameters *Pr. 400* to *Pr. 499* are available.

**Parameters 500 to 599:** Inverter parameters *Pr. 500* to *Pr. 599* are available.

**Parameters 600 to 699:** Inverter parameters *Pr. 600* to *Pr. 699* are available.



**Figure 6: Parameter Group Selection List**



**Parameters 800 to 899:** Inverter parameters *Pr. 800* to *Pr. 899* are available.

Note that because parameters in the “Parameters X00 to X99” groups are non-scanned, their values are not continuously updated from the inverter. For more information regarding scanned vs. non-scanned parameters, refer to sections 5.4.5 and 6.2.

### **5.4.3 Parameter List**

The bottom half of the Monitor tab contains the parameter list (refer to Figure 7). The parameters that are displayed in the list at any given time depend on the parameter groups that are currently selected (refer to section 5.4.2), as well as whether or not any filters have been applied (refer to section 5.4.4).

The first column of the Parameter List shows the inverter parameter (*Pr. XXX*) number that is normally used when accessing a given register via the inverter’s keypad. Note that not all registers will have parameter number entries: command and monitor registers, for example, are not accessed via the inverter’s keypad by entering a “*Pr. XXX*” number, and their associated parameter column fields on the Monitor tab are therefore empty. Also note that this column is for user convenience and inverter user’s manual cross-reference only: parameters are not referenced through the interface card by their parameter numbers, but by their register numbers.

The second column of the Parameter List shows the register number that provides network access to each parameter (refer to section 6). The third column contains the parameter descriptions, which are used by the filter function. The last column performs two functions: it displays the current value of the parameter, and (for writable parameters) also allows changing the parameter’s value by clicking on the number in the value column and entering the new value.



Parameter	Register	Description	Value (Decimal)
0	1000	Torque boost	40
1	1001	Maximum frequency	12000
2	1002	Minimum frequency	0
3	1003	Base frequency	6000
4	1004	Multi-speed setting (high speed)	6000
5	1005	Multi-speed setting (middle speed)	3000
6	1006	Multi-speed setting (low speed)	1000
7	1007	Acceleration time	50
8	1008	Deceleration time	50
9	1009	Electronic thermal O/L relay	1100
10	1010	DC injection brake operation frequency	300
11	1011	DC injection brake operation time	5
12	1012	DC injection brake operation voltage	40
13	1013	Starting frequency	50
14	1014	Load pattern selection	0
15	1015	Jog frequency	500
16	1016	Jog acceleration/deceleration time	5
17	1017	MRS input selection	0

**Figure 7: Parameter List**

Some items to keep in mind when interacting with the Parameter List are:

- When entering new parameter values, be sure that the number being entered is appropriate for the currently-selected radix (refer to section 5.4.6): for example, an entered value of “1000” in hexadecimal is equal to 4096 in decimal.

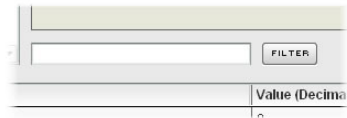


- If desired, the column widths can be changed by dragging the vertical bars that separate the header row's cells to a different position.
- If you begin changing a parameter value and then decide to abandon the change, pressing the ESC key on your keyboard will abandon the change and redisplay the current parameter value.
- When editing a parameter value, clicking someplace off the entry cell is equivalent to hitting the ENTER key.
- The values of all non-scanned (*Pr. XXX*) parameters are only read from the inverter by the web browser when the “refresh” button is clicked (refer to section 5.4.5). Values of non-scanned parameters are immediately written down to the inverter when they are changed, however.

#### **5.4.4 Parameter List Filter**

A filter function provides Parameter List search capabilities. To use the filter function, simply type a word or portion of a word into the filter entry box and then click the “filter” button. Refer to Figure 8.

The filter will then display only those parameters currently available in the Parameter List that satisfy the search criteria. For example, to find all monitor parameters that contain some derivative of the word “volt” (such as “voltage” or “volts”), select the “Monitor parameters” group, enter “volt” in the filter entry box, and then click the “filter” button.



**Figure 8: Parameter List Filter**

Once a filter has been entered, it will continue to be applied to all information normally displayed in the Parameter List for as long as the filter term is left in the filter entry box. Continuing the previous example where we filtered on the root term “volt” in the monitor parameters, we can then easily apply this filter to all available parameters simply by selecting the “All” parameter group. The Parameter List will now display all command, monitor, alarm history etc. parameters that contain the root term “volt”.

To remove the filter, delete any characters contained in the filter entry box and then click the “filter” button.

### **5.4.5 Non-Scanned Parameter Refresh**

The values of all non-scanned (*Pr. XXX*) parameters are only read from the inverter when the “refresh” button (shown in Figure 9) is clicked. Once clicked, the interface card will display the message “refreshing values...” in the Information Window (refer to section 5.4.1), and will begin to retrieve the values of all non-scanned parameters from the inverter. This process may require 5s-10s to complete. When completed, the current values of all non-scanned parameters will be updated in the Parameter List, and the Information Window will display the message “refresh completed.” For further details regarding scanned vs. non-scanned parameters, refer to section 6.2.



**Figure 9: Refresh Button**



### 5.4.6 Radix Selection

Figure 10 shows the radix selection buttons. These selection buttons allow changing the Parameter List “value” column data display and entry radix between decimal and hexadecimal formats.



Figure 10: Radix Selection

When “DEC” is selected, the “value” column heading will be “*Value (Decimal)*”, current parameter values will be displayed in decimal, and values to be written to parameters must be entered in decimal format. For example, to change the inverter’s frequency command to 40.00Hz, enter the decimal value 4000.

Similarly, when “HEX” is selected, the “value” column heading will be “*Value (Hexadecimal)*”, current parameter values will be displayed in hexadecimal, and values to be written to parameters must be entered in hexadecimal format. For example, to turn on bit #10 in the inverter’s command word (MRS), enter the hexadecimal number 0400.

## 5.5 PROFINET Tab

This section is only applicable when the PROFINET firmware is loaded onto the interface card. The PROFINET tab provides for the configuration of the device on a PROFINET network. Refer to Figure 11.

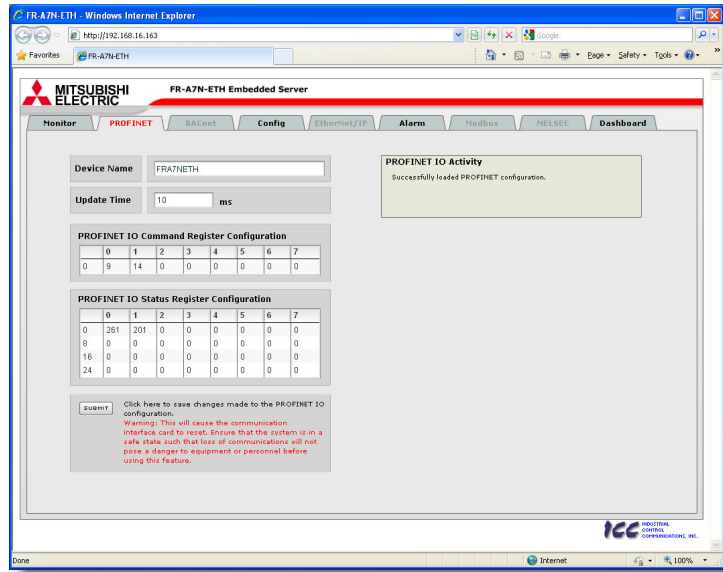


Figure 11: PROFINET Tab



### 5.5.1 Information Window

Figure 12 shows the Information Window, which is located in the upper-right hand corner of the PROFINET tab. This window displays various informational messages regarding the status of the PROFINET configuration (loading or submitting).



Figure 12: PROFINET Tab Information Window

### 5.5.2 Device Identification and Configuration

There are several identification and configuration items available for setting various characteristics of the PROFINET device. These items are shown in Figure 13 and are explained in further detail below.

A PROFINET device's name (station name) must be unique across the entire PROFINET network because it is used by controllers to uniquely identify PROFINET devices. The **Device Name** text entry box is used to configure this unique device identifier on every inverter.

The **Update Time** field is a configuration item which changes the frequency with which command and status data updates take place internally in the device. This setting is not related to the frequency with which data communications take place on the Ethernet network. This time setting is a 32-bit value adjustable in 1ms increments. Typically, this value should not need to be changed from its default value of 10ms. Note that this

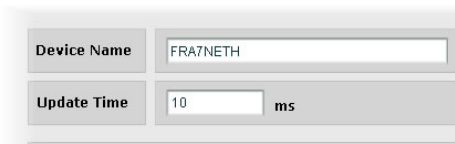


Figure 13: PROFINET Device Identification and Configuration

setting is not related to, and therefore should not be confused with, the I/O Cycle Update Time (in Siemens Step 7 or an equivalent hardware configuration tool).

### 5.5.3 I/O Data Configuration Arrays

The interface card supports two different types of PROFINET data transfer. One type is included with the implementation of the PROFIdrive profile, and requires no user configuration (refer to section 8.5 for more information). The other type, however, is entirely user-configurable, and is utilized when a standard “IN: XX, OUT:YY” I/O module is chosen during network configuration.

The I/O data configuration arrays consist of two separate elements (refer to Figure 14.) The command register configuration defines the structure of the command data sent from the PROFINET controller to the inverter, and the status register configuration defines the structure of the status data sent from the inverter back to the controller. These arrays allow the creation of custom-built I/O data. Up to 8 command registers can be sent to the inverter, and up to 32 status registers can be sent back to the controller. Each box in an array is capable of containing a register number. Because all inverter registers are 16-bit data elements, each box therefore represents two bytes of input or output data.

Profinet IO Command Register Configuration								
	0	1	2	3	4	5	6	7
0	9	14	0	0	0	0	0	0

Profinet IO Status Register Configuration								
	0	1	2	3	4	5	6	7
0	201	261	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0

**Figure 14: I/O Data Configuration**



The command register array locations are numbered 0-7, and traverse from left to right. The status register array locations are numbered 0-31, and traverse from left to right across each row, and then increment to the left-most position on the next row. Clicking on a box in an array allows the user to enter a register number that will be referenced at that location when data is either received from or sent to the controller. Note that only scanned registers may be entered into the register array locations: if an attempt is made to enter a non-scanned register number, an error dialog box will appear (refer to section 6.2 for an explanation of scanned vs. non-scanned registers.) A value of 0 indicates that no register is referenced at that location, which will cause corresponding command data to be ignored and status data to be a default value of 0.

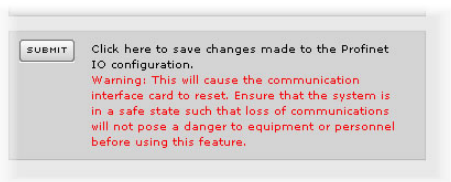
As an example, looking at the default configuration shown in Figure 14, we can see that each array contains two defined registers. Therefore, up to 4 “meaningful” bytes of data can be both received and sent (the qualifier “meaningful” is used here because the module currently selected by the controller may require larger input and/or output data sizes, but all unreferenced command data will be ignored, and all unreferenced status data will contain dummy “0” values). The first word (two bytes) of command data will be written to register 9 (command register) and the second word will be written to register 14 (frequency command). Similarly, the first word of status data will contain the value of register 201 (output frequency) and the second word will contain the value of register 261 (status register).



### 5.5.4 Submitting Changes

Whenever any of the PROFINET configuration elements (I/O array configuration, Device Name, etc.) have been changed, the “submit” button located in the lower left-hand portion of the web page must be clicked in order to write these settings to the interface card’s filesystem.

Note that because these configuration elements are read from the filesystem only when the interface card boots up, the act of submitting configuration changes will also reset the interface card. Please allow 30 seconds for the interface card to reboot, at which time it will then be operating with the recently-submitted configuration. Refer to Figure 15.



**Figure 15: Submit PROFINET Changes**

## 5.6 BACnet Tab

The BACnet tab provides for the configuration of the device on a BACnet/IP network. Refer to Figure 16.

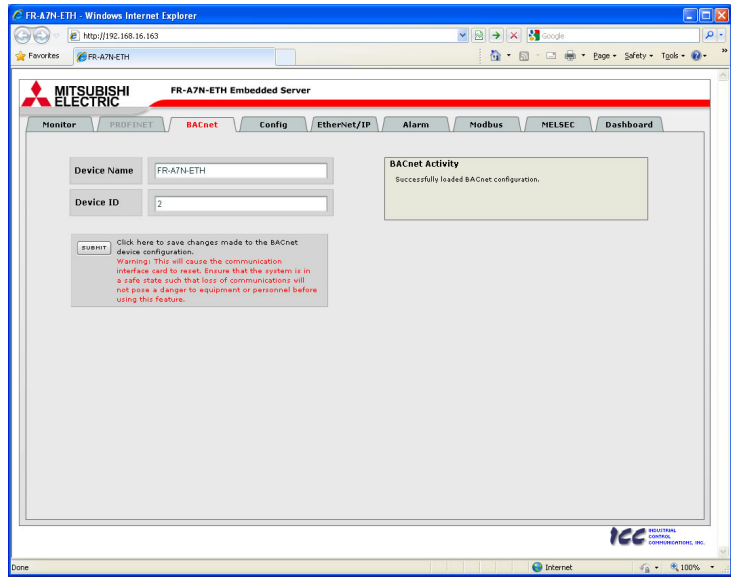


Figure 16: BACnet Tab

### 5.6.1 Information Window

Figure 17 shows the Information Window, which is located in the upper-right hand corner of the BACnet tab. This window displays various informational messages regarding the status of the BACnet configuration (loading or submitting).

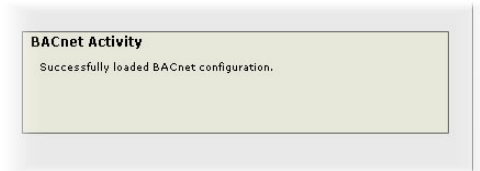


Figure 17: BACnet Tab Information Window

### 5.6.2 Device Identifiers

A BACnet device's name and ID (the Object\_Name and Object\_Identifier properties, respectively, of the Device Object) must be unique across the entire BACnet network because they are used to uniquely identify BACnet devices. The text entry boxes shown in Figure 18 are used to configure these unique device identifiers on every inverter.

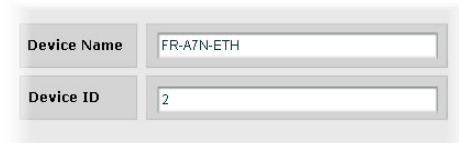


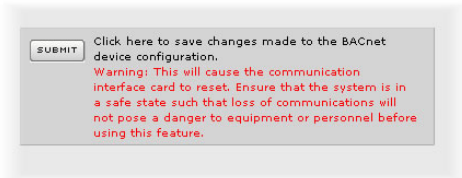
Figure 18: BACnet Device Identifiers



### 5.6.3 Submitting Changes

Whenever either of the BACnet configuration elements (Device Name or Device ID) has been changed, the “submit” button located in the left-hand portion of the web page must be clicked in order to write these settings to the interface card’s filesystem.

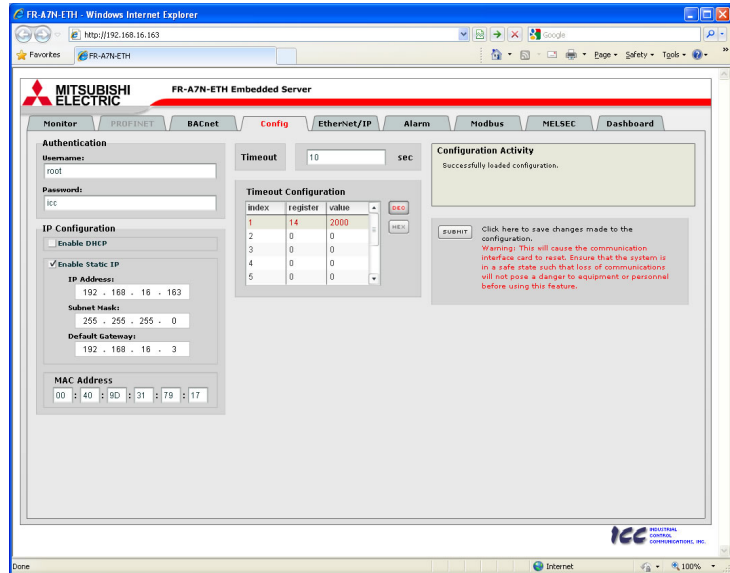
Note that because these configuration elements are read from the filesystem only when the interface card boots up, the act of submitting configuration changes will also reset the interface card. Please allow 30 seconds for the interface card to reboot, at which time it will then be operating with the recently-submitted configuration. Refer to Figure 19.



**Figure 19: Submit BACnet Changes**

## 5.7 Config Tab

The Config tab provides access to various configuration items. Refer to Figure 20.



5

Figure 20: Config Tab



### 5.7.1 Information Window

Figure 21 shows the Information Window, which is located in the upper-right hand corner of the Config tab. This window displays various informational messages regarding the status of the configuration parameters (loading or submitting).

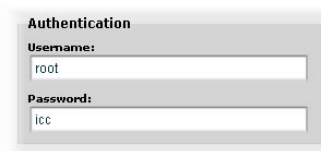


**Figure 21: Config Tab Information Window**

### 5.7.2 Authentication Configuration

Figure 22 shows the entry boxes used to modify the authentication credentials. The case-sensitive username and password can contain letters (“a”..”z” and “A”..”Z”) and numbers (“0”..”9”), and can each be up to 80 characters in length.

Be sure to make a note of the new settings whenever these credentials are changed, as they must be entered whenever the web page is accessed, an FTP session is initiated, or when a configuration change is performed via the Finder utility. Contact ICC for assistance if you have forgotten your customized credentials.



**Figure 22: Authentication Configuration**

### 5.7.3 IP Address Configuration

Figure 23 shows the configuration items used to modify the IP address-related parameters. Modification of these settings is consistent with the technique used with the Finder utility (refer to section 4.1).

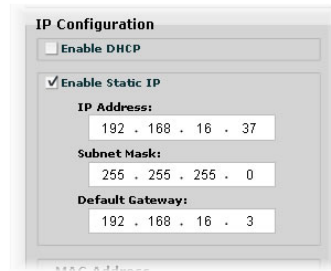


Figure 23: IP Address Configuration

### 5.7.4 MAC Address Configuration

Figure 24 shows the entry boxes that are used to view and/or modify the unique MAC address of the interface. The MAC address should not be changed without first consulting ICC Technical Support.



Figure 24: MAC Address Configuration



## 5.7.5 Timeout Configuration

The interface can be configured to perform a specific set of actions when network communications are lost. Support for this feature varies depending on the protocol: refer to the protocol-specific section of this manual for further information.

There are two separate elements that comprise the timeout configuration (refer to Figure 25):

- The timeout time
- The timeout configuration array

The **timeout time** is a floating-point number which allows adjustment down to 1/100<sup>th</sup> of a second (0.01 second increments). This time setting is used by certain protocols in order to determine abnormal loss-of-communications conditions and, optionally, to trigger a timeout processing event. The default timeout time is 10s.

The **timeout configuration array** allows up to 10 register/value pairs to be designated by the user. When a timeout event is triggered by a protocol, the timeout configuration array indexes are parsed. If the “register” field for an index is set to 0, then this index is “disabled” and therefore ignored. If, on the other hand, the “register” field is non-zero, then the value contained in the “value” field is automatically written to the designated register. This flexible mechanism allows up to 10 designated inverter registers to have their own unique “fail-safe” conditions in the event of a network interruption.

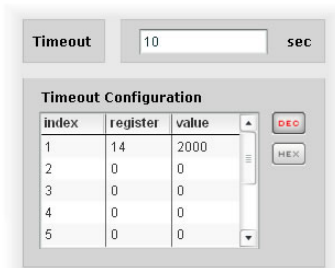


Figure 25: Timeout Configuration



For example, Figure 25 shows a timeout time of 10s, and one timeout entry assignment. If a protocol that makes use of timeout processing triggers a timeout event, then a value of 2000 will automatically be written to inverter register 14 (the frequency command). Provided the inverter has a valid “run” command and is currently configured to use the network frequency command as its master frequency command, it will ramp to 20.00Hz.

If timeout/failsafe processing is not desired, just set the “register” fields for all indexes to 0 (disabled). This is the default condition.

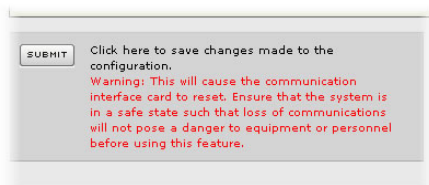
“DEC” and “HEX” selection buttons are also available, and allow changing the “value” column data display and entry radix between decimal and hexadecimal formats, respectively. These buttons provide the ability to interact with the various inverter registers in their most natural radix (e.g. a hexadecimal command word vs. a decimal frequency command value).



## 5.7.6 Submitting Changes

Whenever any of the configuration elements has been changed, the “submit” button located in the right-hand portion of the web page must be clicked in order to write these settings to the interface card’s filesystem.

Note that because these configuration elements are read from the filesystem only when the interface card boots up, the act of submitting configuration changes will also reset the interface card. Please allow 30 seconds for the interface card to reboot, at which time it will then be operating with the recently-submitted configuration. Refer to Figure 26.



**Figure 26: Submit Configuration Changes**

## 5.8 EtherNet/IP Tab

The EtherNet/IP tab provides access to configuration items related to communication on an EtherNet/IP network. Refer to Figure 27.

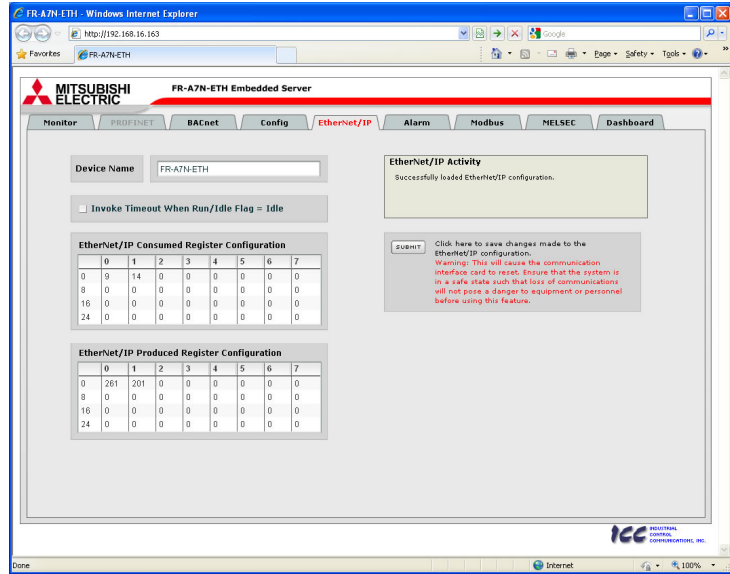


Figure 27: EtherNet/IP Tab



### 5.8.1 Information Window

Figure 28 shows the Information Window, which is located in the upper-right hand corner of the EtherNet/IP tab. This window displays various informational messages regarding the status of the EtherNet/IP configuration parameters (loading or submitting).

#### Ethernet/IP Activity

Successfully loaded Ethernet/IP configuration.

**Figure 28: EtherNet/IP Tab Information Window**

### 5.8.2 Device Identification

A text entry box is available which allows customization of the device's name for identification on the EtherNet/IP network. This string is accessible as the “product name” attribute of the identity object. Refer to Figure 29.

Device Name

FR-A7N-ETH

**Figure 29: EtherNet/IP Device Identification**

### 5.8.3 Run/Idle Flag Behavior

EtherNet/IP clients (such as PLCs) have the option of adding a 32-bit “run/idle” header to all class 1 (I/O) data packets sent to devices. Bit 0 of this header is called the “run/idle flag” by the EtherNet/IP specification, and is intended to signify when the client is in a “running” state or an “idle” state. A

Invoke Timeout When Run/Idle Flag = Idle

**Figure 30: Run/Idle Flag Behavior Selection**

running state (run/idle flag = Run) is indicated whenever the client is performing its normal processing (e.g. scanning its ladder logic). An idle state (run/idle flag = Idle) is indicated otherwise. For example, Allen Bradley ControlLogix PLCs will set their run/idle flag to Idle whenever their processor keyswitch is placed in the “PROG” position, presumably in preparation to receive a new application program from RSLogix.

The behavior of EtherNet/IP devices when they receive I/O data from a controller with the run/idle flag set to Idle is not specified in the EtherNet/IP specification. The interface card allows the option of two different behavioral responses when a run/idle flag = Idle condition is received, depending on the state of the checkbox indicated in Figure 30.

- If the checkbox is cleared (default setting), then the interface card will maintain the last I/O data values received from the client. For example, if the inverter was being commanded to run prior to the run/idle flag being set to Idle, then it will continue to run.
- If the checkbox is checked, then the interface card will invoke its user-configured timeout processing (refer to section 5.7.5). This setting allows the user to determine any inverter behavior they may desire (stop the inverter, fault the inverter, ramp to a preset speed, etc.)

### **5.8.4 Class 1 (I/O) Data Configuration Arrays**

The interface card supports two different types of EtherNet/IP class 1 (I/O) data transfer. One type is included with the implementation of the AC/DC drive profile, and requires no user configuration. The other type, however, is entirely user-configurable, and is utilized when the client opens a connection to the interface using assembly instances 100 and 150.



The user-configurable data arrays consist of two separate elements (refer to Figure 31.) The consumed register configuration defines the structure of the command data sent from the EtherNet/IP controller (for example, a ControlLogix PLC) to the inverter, and the produced register configuration defines the structure of the status data sent from the inverter back to the controller. These arrays allow the creation of custom-built I/O data. Up to 32 command registers can be sent to the inverter, and up to 32 status registers can be sent back to the controller. Each box in an array is capable of containing a register number. Because all inverter registers are 16-bit data elements, each box therefore represents two bytes of consumed or produced data.

Each of the register array locations are numbered 0-31, and traverse from left to right across each row, and then increment to the left-most position on the next row. Clicking on a box in an array allows the user to enter a register number that will be referenced at that location when data is either consumed from the controller or produced to the network. Note that only scanned registers may be entered into the register array locations: if an attempt is made to enter a non-scanned register number, an error dialog box will appear (refer to section 6.2 for an explanation of scanned vs. non-scanned registers.) A value of 0 indicates that no register is referenced at that location, which will cause the corresponding consumed data to be ignored and produced data to be a default value of 0.

**Ethernet/IP Consumed Register Configuration**

	0	1	2	3	4	5	6	7
0	9	14	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0

**Ethernet/IP Produced Register Configuration**

	0	1	2	3	4	5	6	7
0	261	201	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0

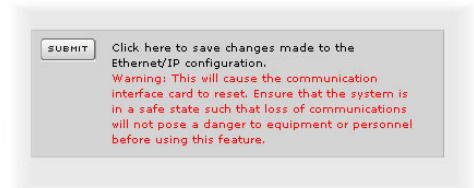
**Figure 31: EtherNet/IP Class 1 (I/O) Data Configuration**

As an example, looking at the default configuration shown in Figure 31, we can see that each array contains two defined registers. Therefore, up to 4 “meaningful” bytes of data can be both received and sent (the qualifier “meaningful” is used here because the connection sizes configured in the controller may request larger consumed and/or produced data sizes, but all unreferenced consumed data will be ignored, and all unreferenced produced data will contain dummy “0” values). The first word (two bytes) of consumed data will be written to register 9 (command register) and the second word will be written to register 14 (frequency command). Similarly, the first word of produced data will contain the value of register 261 (status register) and the second word will contain the value of register 201 (output frequency).

### 5.8.5 Submitting Changes

Whenever any of the EtherNet/IP configuration elements (Device Name or I/O array configurations) have been changed, the “submit” button located in the right-hand portion of the web page must be clicked in order to write these settings to the interface card’s filesystem.

Note that because these configuration elements are read from the filesystem only when the interface card boots up, the act of submitting configuration changes will also reset the interface card. Please allow 30 seconds for the interface card to reboot, at which time it will then be operating with the recently-submitted configuration. Refer to Figure 32.



**Figure 32: Submit Configuration Changes**

## 5.9 Alarm Tab

The Alarm tab provides a configurable mechanism by which the interface card can autonomously monitor any available inverter register and send emails to up to four recipients when a certain condition is detected. The alarm conditions have both value and time constraints, and can be configured to retrigger at a fixed interval as long as the alarm condition continues to be satisfied. Twenty individually-configurable alarms are available. Refer to Figure 33.

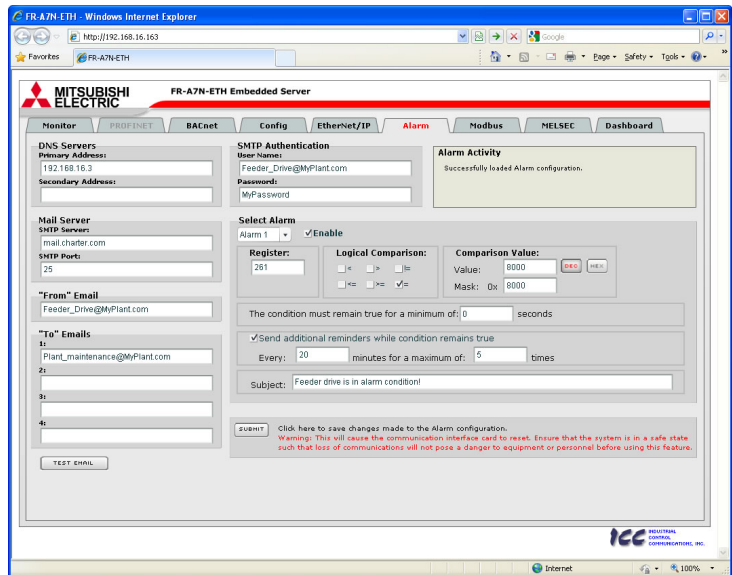
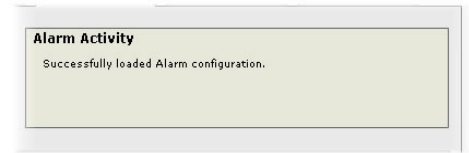


Figure 33: Alarm Tab



### 5.9.1 Information Window

Figure 34 shows the Information Window, which is located in the upper-right hand corner of the Alarm tab. This window displays various informational messages regarding the status of the Alarm configuration parameters (loading or submitting) and test emails.



**Figure 34: Alarm Tab Information Window**



## 5.9.2 Email Configuration

In order for an alarm trigger to successfully send a notification email, some network settings must first be configured properly (refer to Figure 35 and Figure 36.)

**DNS Servers:** Enter the dotted-decimal IP addresses of the primary and secondary DNS servers which will be used to resolve the configured SMTP server name. Only the primary DNS server is required, but if a secondary DNS server is entered, then it will be used if the primary server is inaccessible.

**Mail Server:** Enter the SMTP server address as a name or as a dotted-decimal IP address, and the SMTP port (default=25) that the SMTP server listens for incoming emails on.

**“From” Email:** Enter the email address that will appear as the sender’s email address in the email headers.

**“To” Emails:** Up to four recipients can be designated to receive alarm emails. Blank entries will not be processed by the interface.

**“Test Email” Button:** When the “Test Email” button is pressed, the interface card will use the information currently entered in the above-mentioned fields to send a test email. Note that you do not have to first “submit” the settings to the interface card’s filesystem (refer to section 5.9.4) in order to test them:

The screenshot shows a web interface for email configuration. It is divided into several sections:

- DNS Servers:** Contains two input fields. The "Primary Address" field contains the IP address "192.168.16.3". The "Secondary Address" field is empty.
- Mail Server:** Contains two input fields. The "SMTP Server" field contains "mail.charter.com". The "SMTP Port" field contains "25".
- “From” Email:** Contains one input field with the text "Feeder\_Drive@MyPlant.com".
- “To” Emails:** Contains four input fields labeled "1:", "2:", "3:", and "4:". The first field contains "Plant\_Maintenance@MyPlant.com", while the others are empty.
- At the bottom, there is a button labeled "TEST EMAIL".

Figure 35: Email Configuration

fields can be changed and retested on-the-fly without affecting the operation of the interface card's control protocols. When the correct settings have been confirmed with a successfully-sent test email, "submit" the changes at that time to commit them to the interface card's filesystem: any changes made prior to submitting as described in section 5.9.4 are temporary only and will be lost if a different configuration tab is selected or if the web browser is closed.

**SMTP Authentication:** Some email servers require that clients wishing to send emails first authenticate themselves. If the email server in use requires authentication, then enter the user name and password as indicated in Figure 36. If the email server in use does not require authentication, then these entries can be disregarded.

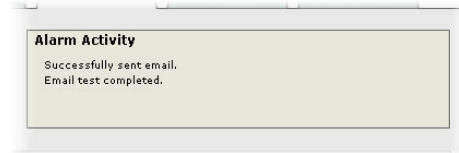


**Figure 36: SMTP AUTH Configuration**

When a test email transmission is initiated, completes successfully, or fails due to an error, the information window (refer to section 5.9.1) will display appropriate messages such as those shown in Figure 37 and Figure 38. Although the test email is sent



**Figure 37: Information Window at Test Email Initiation**



**Figure 38: Information Window at Test Email Successful Completion**



immediately, note that due to internet and/or email server delays, it may take several minutes to receive test emails.

### 5.9.3 Alarm Configuration

The interface supports twenty independently-configurable alarms. As shown in Figure 39, each alarm has a variety of configuration elements, which will be explained further below.

**Alarm Selection:** This drop-down box allows the selection of one of the twenty available alarms. When an alarm is selected, that alarm's current configuration parameters will be populated in the alarm configuration box.

**“Enable” Check Box:** If checked, this alarm is active and will be evaluated every second. If unchecked, this alarm is inactive and will therefore not be evaluated.

**Register:** Enter the inverter register that this alarm will continuously monitor. For example, the alarm displayed in Figure 39 is configured to monitor register 261, which is the inverter status register. Note that only scanned registers may be used for alarms (refer to section 6.2).

Figure 39: Alarm Configuration Box

**Logical Comparison:** Choose a comparison operator which will be used to compare the current value of the indicated “Register” with the reference “Comparison Value”. Available selections are “less than” (<), “less than or equal to” (<=), “greater than” (>), “greater than or equal to” (>=), “not equal to” (!=), and “equal to” (=).

**Comparison Value:** The reference comparison value is comprised of two subcomponents: a “Mask” field and a “Value” field. Each time the alarm is evaluated, the current value of the indicated “Register” is first bit-wise “AND”ed with the “Mask” field. The resulting derived value is then compared with the “Value” field by way of the “Logical Comparison” operator. While the “Mask” field is always a hexadecimal number, the display and entry radix of the “Value” field can be changed between decimal and hexadecimal with the associated “DEC” and “HEX” buttons.

Registers that correspond to “analog” process variables (e.g. frequencies, voltages, etc.) should typically have their “Mask” fields set to 0xFFFF, which causes all data bits to be retained for the “Value” field comparison. For registers that correspond to “enumerated” process variables (e.g. status words where each bit of the register indicates a different item), however, the “Mask” can be chosen to single out one or more specific data bits of the register. For example, the “Mask” value of 0x8000 displayed in Figure 39 isolates bit #15 of the inverter status register, which indicates whether or not an inverter “alarm occurrence” state exists. The “Value” field is also set to a hexadecimal value of 0x8000, so the alarm condition will be evaluated as “true” when bit #15 of the inverter status register equals 1.

**The Condition Must Remain True For A Minimum Of:** Alarm analysis processing is performed by the interface card once per second. Enter the number of seconds that the condition must be continuously



evaluated as “true” for the alarm to be triggered. A time of 0 seconds means that just a single evaluation of “true” will immediately trigger the alarm.

**Send Additional Reminders While The Condition Remains True:** If this check box is unchecked, then only one email transmission event will occur when an alarm condition is triggered: further email transmissions will not be attempted for this alarm unless the alarm condition is first evaluated as “false” (which resets the alarm), and then once again is triggered by a subsequent event.

If this check box is checked, then as long as the alarm condition continues to be evaluated as “true”, subsequent email transmissions will be automatically retriggered every indicated number of minutes for a maximum of the indicated number of times. If at any time during the subsequent transmissions the alarm condition is evaluated as “false”, then the alarm will be reset and email transmissions for this alarm will stop (until the next time the alarm is triggered, of course).

**Subject:** Enter a string of up to 128 characters in length which will appear in the “subject” line of the alarm email. The body of the alarm email is empty.

## **5.9.4 Submitting Changes**

Whenever any of the Alarm configuration elements (alarm settings or email configuration parameters) have been changed, the “submit” button located in the lower right-hand portion of the web page must be clicked in order to write these settings to the interface card’s filesystem.

Note that because these configuration elements are read from the filesystem only when the interface card boots up, the act of submitting configuration changes will also reset the interface card. Please

allow 30 seconds for the interface card to reboot, at which time it will then be operating with the recently-submitted configuration. Refer to Figure 40.



**Figure 40: Submit Configuration Changes**

## 5.10 Modbus Tab

The Modbus tab provides access to configuration items related to communication on a Modbus/TCP network. Refer to Figure 41.

The screenshot shows the Modbus configuration page for the Mitsubishi Electric FR-A7N-ETH Embedded Server. The browser window is titled 'FR-A7N-ETH - Windows Internet Explorer' and the address bar shows 'http://192.168.16.163'. The page has a navigation menu with tabs for Monitor, PROFIBET, BACnet, Config, Ethernet/IP, Alarm, Modbus (selected), MELSEC, and Dashboard. The Modbus tab is active, displaying several configuration options and a table.

**Enable Supervisory Timer**

**Enable Connection Timer**

**Modbus Register Remap Configuration**

Register	Remaps To	Register	Remaps To	Register	Remaps To	Register	Remaps To	Register	Remaps To
2001	0	2011	0	2021	0	2031	0	2041	0
2002	0	2012	0	2022	0	2032	0	2042	0
2003	0	2013	0	2023	0	2033	0	2043	0
2004	0	2014	0	2024	0	2034	0	2044	0
2005	0	2015	0	2025	0	2035	0	2045	0
2006	0	2016	0	2026	0	2036	0	2046	0
2007	0	2017	0	2027	0	2037	0	2047	0
2008	0	2018	0	2028	0	2038	0	2048	0
2009	0	2019	0	2029	0	2039	0	2049	0
2010	0	2020	0	2030	0	2040	0	2050	0

**Modbus Activity**  
 Successfully loaded Modbus configuration.

**Warning!** Click here to save changes made to the Modbus configuration. Warning! This will cause the communication interface card to reset. Ensure that the system is in a safe state such that loss of communications will not pose a danger to equipment or personnel before using this feature.

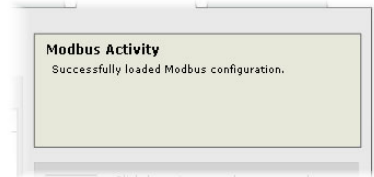
Done Internet 100%

Figure 41: Modbus Tab



### 5.10.1 Information Window

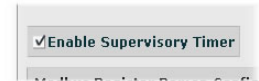
Figure 42 shows the Information Window, which is located in the upper-right hand corner of the Modbus tab. This window displays various informational messages regarding the status of the Modbus configuration parameters (loading or submitting).



**Figure 42: Modbus Tab Information Window**

### 5.10.2 Supervisory Timer Selection

Figure 43 shows the checkbox which enables the interface card's Modbus "supervisory timer" function. This timer provides the ability for the interface card to monitor timeout occurrences between successive Modbus/TCP socket connections, as opposed to the standard timeout functionality (refer to section 5.7.5), which monitors timeout occurrences only within the scope of each client socket connection. While this feature provides an additional level of fail-safe functionality for those applications that require it, there are several ramifications that must be understood prior to enabling this capability. Before enabling this timer, therefore, it is suggested that users read the ICC whitepaper titled "*A Discussion of Modbus/TCP Server-Side Timeout Processing*", which can be found in the documents section at <http://www.iccdesigns.com>.

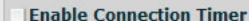


**Figure 43: Supervisory Timer Selection**



### 5.10.3 Connection Timer Selection

Figure 44 shows the checkbox which enables the interface card's Modbus "connection timer" function. If enabled, the socket timeout determination is performed on a per-socket basis.



Enable Connection Timer

**Figure 44: Connection Timer Selection**

### 5.10.4 Register Remap Configuration

At times, it may be convenient to access inverter registers in bulk Modbus transactions. This may be especially true in situations where it is desired to access certain registers that are natively non-contiguous. For example, if it were desired to read the inverter's operating frequency (register 201), inverter status (register 261) and converter output voltage (register 208), this could be accomplished in two different ways:

1. Implement three separate Modbus read transactions, each one reading one register only, or
2. Implement one single Modbus read transaction, starting at register 201 for a quantity of 61 registers. Then, pick out the registers of interest and ignore the rest of the response data.

While both of these methods will certainly work, neither one of them is optimized for the task at hand, which is to access three specific register values. A fully optimized solution can be realized, however, by making use of the interface card's Modbus register remapping capabilities. This mechanism operates by allocating a block of 50 user-configurable registers (2001..2050) that remap to other inverter registers. In this way, non-contiguous inverter registers can be grouped together in any order and accessed efficiently via the Modbus/TCP "read multiple registers" and "write multiple registers" function codes. The net effect is one of being able to transfer larger blocks of registers using fewer Modbus

transactions, which results in improved network utilization and simpler data manipulation code on the Modbus master device.

Enable Supervisory Timer

**Modbus Register Remap Configuration**

Register	Remaps To	Register	Remaps To	Register	Remaps To	Register	Remaps To	Register	Remaps To
2001	9	2011	0	2021	0	2031	0	2041	0
2002	14	2012	0	2022	0	2032	0	2042	0
2003	201	2013	0	2023	0	2033	0	2043	0
2004	261	2014	0	2024	0	2034	0	2044	0
2005	208	2015	0	2025	0	2035	0	2045	0
2006	0	2016	0	2026	0	2036	0	2046	0
2007	0	2017	0	2027	0	2037	0	2047	0
2008	0	2018	0	2028	0	2038	0	2048	0
2009	0	2019	0	2029	0	2039	0	2049	0
2010	0	2020	0	2030	0	2040	0	2050	0

**Figure 45: Modbus/TCP Register Remap Configuration**

Figure 45 shows the register remap configuration array. Clicking on an entry field in the “Remaps To” column allows the user to enter an inverter register number that will then be accessible at the register indicated in the adjacent “Register” column. An assignment of 0 in the “Remaps To” column indicates that no inverter register is remapped at that location, which results in written values being ignored and read values returned as a default value of 0. Note that remapped inverter registers are still accessible at their original locations: remapping simply provides an additional means of accessing the original register’s value.



As an example, the configuration shown in Figure 45 reveals that a total of five inverter registers have been remapped: register 9 (command word) has been remapped to register 2001, register 14 (frequency command) has been remapped to register 2002, register 201 (output frequency) has been remapped to register 2003, register 261 (inverter status) has been remapped to register 2004, and register 208 (converter output voltage) has been remapped to register 2005. With this configuration, it is now possible to efficiently interact with these five non-contiguous inverter registers via just two Modbus “read/write multiple registers” transactions. Writing to the command word and frequency command can be accomplished with a single “write multiple registers” transaction by writing a quantity of two registers starting at register 2001. Similarly, reading the output frequency, inverter status word and converter output voltage (in that order) can be accomplished with a single “read multiple registers” transaction by reading a quantity of three registers starting at register 2003.

### 5.10.5 Submitting Changes

Whenever the Modbus configuration has been changed, the “submit” button located on the right-hand portion of the web page must be clicked in order to write these settings to the interface card’s filesystem. Refer to Figure 46.

Note that because these configuration elements are read from the filesystem only when the interface card boots up, the act of submitting configuration changes will also reset the interface card. Please allow 30 seconds for the interface card to reboot, at which time it will then be operating with the recently-submitted configuration.



**Figure 46: Submit Configuration Changes**

## 5.11 MELSEC Tab

The MELSEC tab provides for the configuration of the device on an Ethernet network. Refer to Figure 47.

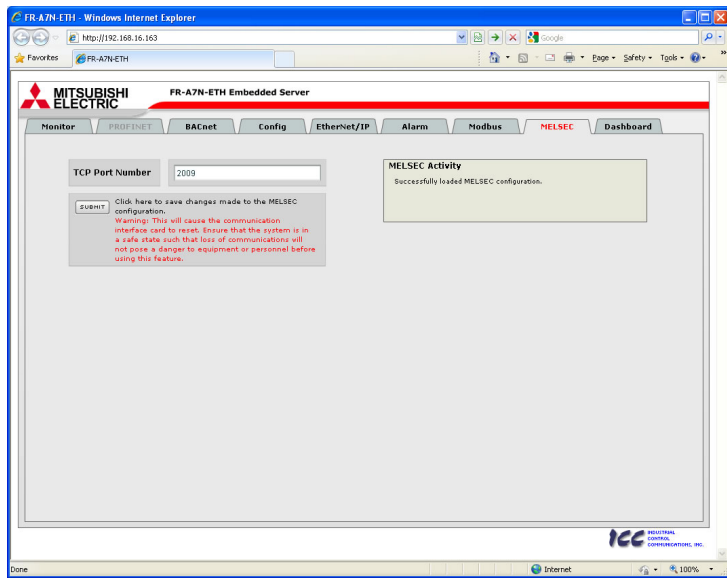
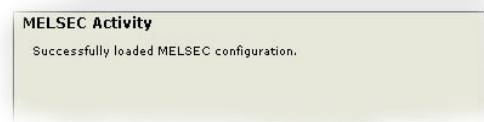


Figure 47: MELSEC Tab

### 5.11.1 Information Window

Figure 48 shows the Information Window, which is located in the upper-right hand corner of the MELSEC tab. This window displays various informational messages regarding the status of the MELSEC configuration (loading or submitting).



**Figure 48: MELSEC Tab Information Window**

### 5.11.2 TCP Port

The MELSEC Server TCP port must be specified before establishing any connections. The text entry box shown in Figure 49 is used to configure the TCP port number.



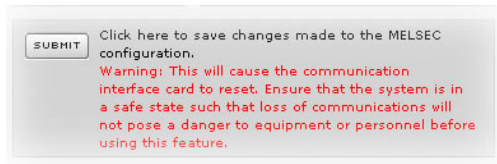
**Figure 49: MELSEC TCP Port Number**



### 5.11.3 Submitting Changes

Whenever the MELSEC TCP Port Number is changed, the “submit” button located in the left-hand portion of the web page must be clicked in order to write the setting to the interface card’s filesystem.

Note that because these configuration elements are read from the filesystem only when the interface card boots up, the act of submitting configuration changes will also reset the interface card. Please allow 30 seconds for the interface card to reboot, at which time it will then be operating with the recently-submitted configuration. Refer to Figure 50.



**Figure 50: Submit MELSEC Changes**



## 5.12 Dashboard Tab

The Dashboard Tab provides access to a variety of gauges, meters and graphs that can be configured to provide an at-a-glance graphical overview of critical application variables in real-time. A total of 10 gauge windows are available (four at a time), and each gauge window can be configured to display any scanned register's value via one of six different gauge types. User-defined engineering units, scaling and range limits are also configurable. Refer to Figure 51.

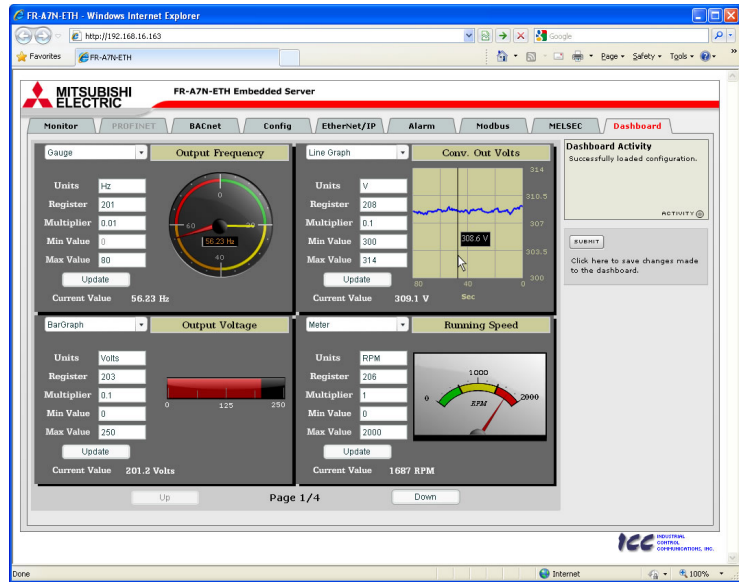


Figure 51: Dashboard Tab



### **5.12.1 Information Window**

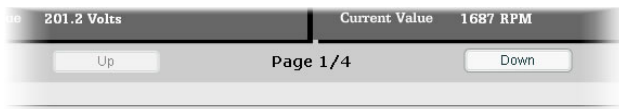
Figure 52 shows the Information Window, which is located in the upper-right hand corner of the Dashboard Tab. This window displays various informational messages regarding the status of the Dashboard configuration parameters (loading or submitting).



**Figure 52: Dashboard Tab Information Window**

### **5.12.2 Gauge Window Navigation**

Figure 53 shows the two buttons that provide for navigation of the gauge windows. Gauge windows are displayed four at a time in the Dashboard Tab, and by clicking the “Up” or “Down” buttons, the gauge windows will scroll in the corresponding direction, two at a time.



**Figure 53: Gauge Window Navigation**

### **5.12.3 Gauge Window Configuration**

Each of the gauge windows can be independently configured to display a user-defined register with a variety of flexible configuration options. While the behavior and presentation may vary slightly depending on the specific gauge chosen, all of the gauges share the following common elements (refer to Figure 54 for an example):

**Gauge Selector:** A drop-down selection box in the upper left-hand corner of the gauge window, which allows the user to select the type of gauge that will be displayed.

**Title:** A text entry box located above the gauge, in which the user can enter a descriptive gauge title comprised of up to 16 characters.

**Units:** A text entry box in which the user can enter an engineering units string comprised of up to 8 characters. This units string will be appended to all locations in the gauge window that display the designated register's current value.

**Register:** The designated register whose value is to be reflected on the gauge. Note that only scanned registers may be displayed in Dashboard gauges (refer to section 6.2 for a discussion of scanned registers).

**Multiplier:** The multiplier value is a floating-point number that is used to scale the raw value of a register. As its name suggests, the multiplier value is multiplied by the designated register's current raw value in order to calculate the gauge's indicated value. Negative values can also be used if desired.

**Min Value:** The gauge's minimum indicated value. Negative values can be used if desired (e.g. if a negative Multiplier attribute is used to generate a negative indicated value). Not all gauges allow adjustment of the min value.



**Max Value:** The gauge's maximum indicated value. Similar to the Min Value attribute, negative values can be used if desired. Indicated value characteristics can even be inverted by setting the Max Value attribute to a value less than the Min Value attribute.

**Update Button:** Clicking the update button will apply the current configuration attribute settings to the gauge. Note, however, that simply updating the gauge's current display properties does not write these settings to the interface card's filesystem. To save the current configuration of all the gauge windows to the filesystem, the Dashboard tab's "submit" button must be selected (refer to section 5.12.4).

**Current Value:** The current indicated value of the designated register is numerically displayed with the configured Units string at the bottom of each gauge window.

The following is a summary of the different available gauge types:

**Gauge:** Refer to Figure 54. This type of meter implements a rotary dial-type display format. The indicated value and units are shown numerically on the face of the gauge, and via the red indicator needle. The yellow needle shows the previous indicated value, thereby providing a simple historical reference. The "Min Value" attribute is not configurable; this gauge always starts at 0.

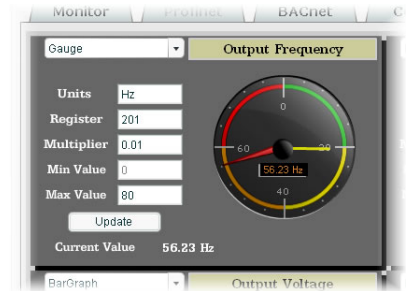
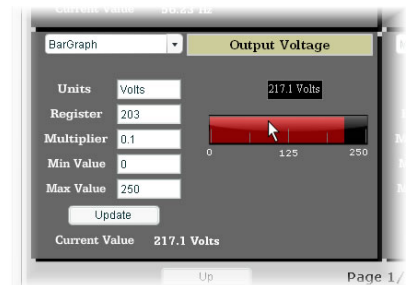


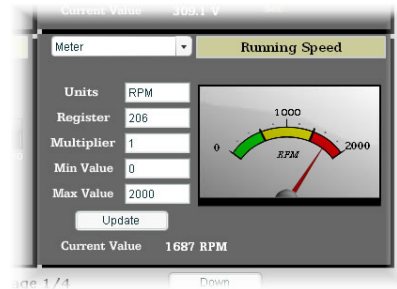
Figure 54: Gauge

**BarGraph:** Refer to Figure 55. This type of meter implements a linear bar graph display format. Hovering the mouse pointer over the red portion of the graph pops up a tooltip which displays the current indicated value and units.



**Figure 55: BarGraph**

**Meter:** Refer to Figure 56. This type of meter implements a common panel meter-type display format. The units string is shown on the face of the meter. All raw register values are interpreted as positive numbers (i.e. 0..0xFFFF equates to 0..65535<sub>10</sub>.)



**Figure 56: Meter**



**Pos/Neg Meter:** Refer to Figure 57. Similar to the “meter” gauge, this type of meter also implements a common panel meter-type display format, but in this instance the indicated value can be positive or negative (two’s complement interpretation). In other words, raw register values of  $0..0x7FFF$  equate to  $0..32767_{10}$ , and values of  $0x8000..0xFFFF$  equate to  $-32768..-1$ . Because the meter placard is always centered around zero, the “Min Value” attribute is not configurable, and the “Max Value” attribute is used for both the maximum positive indicated value as well as the maximum negative indicated value.

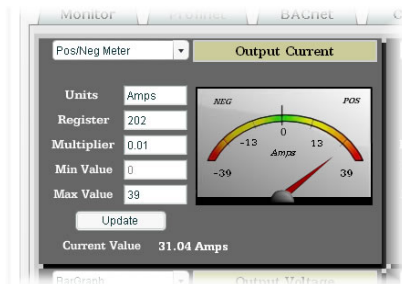


Figure 57: Pos/Neg Meter

**Thermometer:** Refer to Figure 58. This type of meter implements the universally-identifiable thermometer display format. Hovering the mouse pointer over the red “mercury” portion of the graph pops up a tooltip which displays the current indicated value and units.

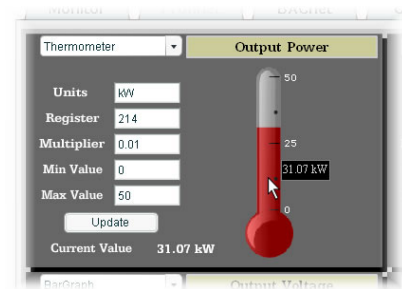
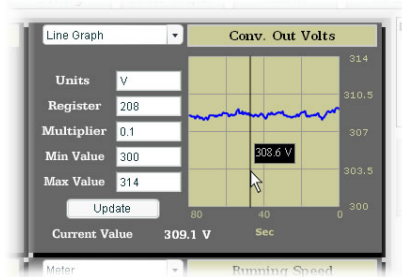
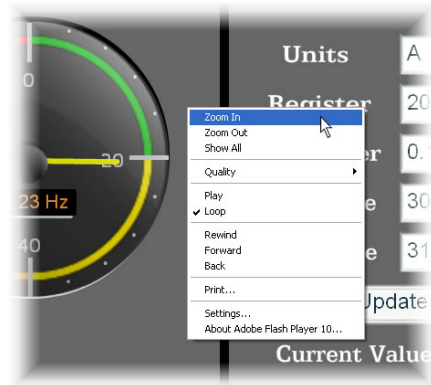


Figure 58: Thermometer

**Line Graph:** Refer to Figure 59. This type of graph implements a continuously-scrolling historical data logging line graph. Up to 80 seconds worth of historical data is available. Hovering the mouse pointer anywhere on the graph displays a vertical reference line at the corresponding time, and pops up a tooltip which displays the current indicated value at that time.



**Figure 59: Line Graph**



**Figure 60: Zooming**



**Gauge  
Usage  
Tip**

At times, it may be convenient to zoom in on a particular gauge or meter in order to more clearly see the indicator, or to fill the computer screen with a particular gauge's image. This can be easily accomplished with the web browser's Flash Player plug-in by right-clicking on the gauge and selecting the desired zoom level (refer to Figure 60).



## **5.12.4 Submitting Changes**

Whenever any of the gauge window configuration items in the Dashboard Tab have been changed, the “submit” button located on the right-hand portion of the web page must be selected in order to write these settings to the interface card’s filesystem. Refer to Figure 61. Note that submitting the Dashboard Tab configuration does not require rebooting of the interface card: the changes take effect immediately, and the interface card continues its operation without interruption.



**Figure 61: Submit Dashboard Changes**



## **6 PARAMETER NUMBERING AND BEHAVIOR**

### **6.1 Register Numbers**

All accessible inverter parameters are referenced by their Modbus register indexes, as defined in the appropriate Mitsubishi inverter user's manual. Refer to the applicable inverter user's manual for further details pertaining to each register/parameter. These same register numbers are used when accessing and configuring parameters via an Ethernet protocol.

Note that not all of the available registers that exist in the interface card's register map have corresponding parameters that exist in the inverter. In other words, if a read from or write to a register that does not correspond to an existing inverter parameter takes place, the read/write may be successful (depending on the specific register accessed; refer to section 6.2), but the data will have no meaning. This feature is beneficial in situations where the accessing of non-contiguous registers can be made more efficient by accessing an all-inclusive block of registers (some of which correspond to inverter parameters and some of which do not), while only manipulating those in your local programming that are known to exist.

### **6.2 Scanned and Non-Scanned Registers**

The interface card provides network access to inverter registers as two different classes. The first class, called "**scanned registers**" is comprised all inverter command and monitor registers, which exist in the inverter as register numbers 2 (inverter reset) through 508 (alarm history 8). These registers are



constantly being “scanned” by the interface card, which is to say that they are constantly being read and/or written (as applicable), and their current values are therefore mirrored in the interface card’s internal memory. Because these inverter registers are mirrored by the interface card, they have the following advantages:

- Access speed: when a network protocol accesses a scanned register (read or write), the value is available immediately. The interface card does not have to make a corresponding request to the inverter to resolve the network access.
- In the web browser’s monitor tab, the values of scanned registers are continuously updated. An explicit on-demand “refresh” does not have to be initiated in order to view the current inverter values (refer to section 5.4.5).
- Only scanned registers may be used when configuring alarm processing (section 5.9), dashboard gauges (section 5.11) and I/O protocols (such as EtherNet/IP consumed & produced data configuration arrays).

The principle disadvantage of scanned registers is that write data checking is not available. This means that when the value of a scanned register is modified via a network protocol or via the web browser’s monitor tab, the interface card itself is not able to determine if the new value will be accepted by the inverter (the value may be out-of-range, or the inverter may be in a state in which it will not accept new values being written via communications, etc.) For example, if a write is performed to a scanned command register with a data value that is out-of-range, the interface card will not generate a corresponding error. However, if end-to-end confirmation of such data writes is required, then the register can be read over the network at a later time to confirm that the written value “took hold” in the inverter. If the value was not accepted by the inverter, then the unsuccessful write can be observed by

reading the current (unchanged) value of the register during a subsequent network transaction. If the unsuccessful write was initiated via the web browser's monitor tab, then the displayed register will revert back to its original value automatically.

Visually, scanned registers are easily identifiable on the web server monitor tab by virtue of the fact that they do not have any entries in the "Parameter" column. This is because these registers are not accessed from the inverter's keypad as configuration parameters (i.e. by navigating to a *Pr. XXX* reference).

Accesses to any register in the scanned register range (2..508) will always be successful. Even if an inverter parameter corresponding to a given scanned register does not exist, the interface card still maintains a location in its internal mirroring memory for that register. This feature allows for the block access of non-contiguous registers as described in section 6.1.

The second class of registers is called "**non-scanned registers**", and is comprised of all inverter configuration registers (those which are accessible as *Pr. XXX* parameters via the inverter's keypad). The inverter itself exposes all configuration parameters externally as registers by adding 1000 to the parameter number. For example, the acceleration time parameter (*Pr. 7*) is made available by the inverter as register 1007. The non-scanned registers, therefore, exist as register numbers 1000 (torque boost) through 1899 (operation time rate). Non-scanned registers are not continuously accessed by the interface card. What this means is that for those protocols that allow accesses to non-scanned registers, such accesses will incur additional response latency, as the interface card must first forward the request to the inverter for it to process.



While the principle disadvantage of accessing non-scanned registers is the additional delay that is incurred, the unique advantage that non-scanned register accesses provide is that write data checking is available. Because all non-scanned register accesses must first be processed by the inverter itself, issues such as out-of-range data values will always be checked for, and any applicable errors will be immediately returned to the originating network protocol.

For those protocols (such as Modbus/TCP) that allow access to non-scanned registers, note that the inverter applies the following rules to these accesses:

- If only one non-scanned register is accessed, then that register must exist in the inverter, or an error will be returned.
- If more than one non-scanned register is accessed, then at least one register within the requested range must exist in the inverter, or an error will be returned.

### **6.3 Inverter Command and Status Register**

---

Inspection of the 700-series inverter user's manual reveals that the inverter has a combined command/status register located at register 9 (Modbus holding register 9). When writing to inverter register 9, the inverter uses the written value as its command word. Refer to the "Control input instruction" column of Table 3 for the bit-mapping of register 9. When reading from register 9, however, the returned data actually represents the inverter's status. To more clearly differentiate the command word from the status word, the interface card separates this combined command/status register into two separate registers. Register 9 is still the inverter's command word, but the inverter status word must be

accessed by requesting read-only register 261 from the interface card. Refer to the “Inverter status” column of Table 3 for the bit-mapping of register 261.

**Table 3: Representative Inverter Command/Status Word**

Bit	Definition	
	Control input instruction	Inverter status
0	Stop command	RUN (inverter running) *2
1	Forward rotation command	Forward rotation
2	Reverse rotation command	Reverse rotation
3	RH (high speed operation command) *1	SU (up to frequency) *2
4	RM (middle speed operation command) *1	OL (overload) *2
5	RL (low speed operation command) *1	IPF (instantaneous power failure) *2
6	JOG (Jog operation) *1	FU (frequency detection) *2
7	RT (second function selection) *1	ABC1 (alarm) *2
8	AU (current input selection) *1	ABC2 (—) *2
9	CS (selection of automatic restart after instantaneous power failure) *1	0
10	MRS (output stop) *1	0
11	STOP (start self-holding) *1	0
12	RES (reset) *1	0
13	0	0
14	0	0
15	0	Alarm occurrence



One side-effect of this combined inverter command/status word is that because the current value of the inverter's command word cannot be read by the interface card, there are no guarantees that the interface card's local value matches the current command word value in the inverter. Resetting a faulted inverter, for example, will clear its internal command word, but the command word residing in the interface card will be unchanged from the last written value in such a scenario. It is important to remember, therefore, that the interface card only has knowledge of the last written command word value, which may or may not equal the inverter's current internal command word value.

## **6.4 Inverter Operation Mode Setting and Status Register**

---

Inspection of the 700-series inverter user's manual reveals that the inverter has a combined operation mode/inverter setting register located at register index 40010 (Modbus holding register 10). When writing to inverter register 10, the inverter uses the written value as a command to change the current operation mode. When reading from register 10, however, the returned data actually represents the inverter's current operation mode. To more clearly differentiate the operation mode (commanded) from the inverter's current setting, the interface card separates this combined command/status register into two separate registers. Register 10 is still the inverter's operation mode (commanded), but the inverter's current mode setting must be accessed by requesting read-only register 262 from the interface card.

One side-effect of this combined inverter mode command/setting status word is that because the current value of the inverter's operation mode (commanded) cannot be read by the interface card, there are no guarantees that the interface card's local value matches the current mode command value in the inverter. Resetting a faulted inverter, for example, may change its internal operation mode

(commanded), but the operation mode (commanded) residing in the interface card will be unchanged from the last written value in such a scenario. It is important to remember, therefore, that the interface card only has knowledge of the last written operation mode (commanded), which may or may not equal the inverter's current internal operation mode (commanded).

## **6.5 Universal Access Registers**

---

### **6.5.1 Overview**

As detailed in section 6.2, the interface card provides network access to inverter registers as scanned and non-scanned classes. This section will introduce a mechanism called “universal access registers”, which essentially provides access to non-scanned registers within the scanned register class. There are two principle applications for universal access registers:

1. *To provide access to inverter configuration parameters via I/O protocol data sets.* One of the previously-mentioned limitations of non-scanned registers (which map to inverter configuration parameters) is that they cannot be included in I/O protocol data sets (such as EtherNet/IP consumed and produced assembly membership lists). If it becomes necessary to access inverter configuration parameters via I/O protocol data sets, then this can be accomplished via the universal access register mechanism.
2. *To provide access to non-standard inverter parameters.* Depending on the inverter, there may be non-standard parameters/registers available that lie outside the typical “Pr. XXX” accesses made available through the inverter's keypad. For example, there may be a variety of configuration and/or bias & gain setting registers that are available above register number 1899. The quantity and



allowable ranges of these registers may vary from inverter family to inverter family, and the universal access register mechanism is therefore the most convenient method of accessing these non-standard registers. Refer to the appropriate inverter user's manual for detailed information on available inverter registers, their allowable adjustment ranges, and any scaling factors that may apply.

The universal access register set is comprised of the following three interface card registers, all of which exist in the scanned register class:

- An address register accessible via interface card register 100 (Universal Access Register Address, or **UARA** for short),
- A value register accessible via interface card register 101 (Universal Access Register Value, or **UARV** for short), and
- A status register accessible via interface card register 102 (Universal Access Register Status, or **UARS** for short.)

The essential concept of using this powerful feature is to just set the UARA to the inverter register number that you wish to access (e.g. acceleration time is located at inverter register 1007), and then perform the desired operation ("read" to read the designated parameter and "write" to change it) on the UARV. The operation on the UARV will be directly performed on the inverter parameter indicated by the UARA. In this manner, any available inverter parameter can be read from or written to.

The UARS exists to support the universal register architecture. This 16-bit status word is divided into two 8-bit status codes: the lower byte represents the read status, and the upper byte represents the write status.



## 6.5.2 Universal Access Reads

When a new register number is entered into the UARA, it takes a certain amount of time for the UARV to reflect the value of this newly-entered register number. To indicate when the UARV has been updated with the value of the desired parameter, the read status (UARS low byte) is initialized to 0xFF whenever the UARA is changed. This code represents “not ready”, and is used to indicate that the value currently contained in the UARV is not yet the value of the parameter indicated in the UARA. Once the value of the parameter indicated in the UARA has been retrieved from the inverter, this “not ready” status will be replaced by one of the read codes indicated in Table 4. As long as the UARA is not changed, the indicated parameter will then be continuously read from the inverter, and the read code updated if a status change occurs.

**Table 4: UARS Read Codes**

Code	Meaning
0x00	Read successful (value in UARV)
0x02	Invalid parameter targeted
0x07	Communication to inverter lost
0x0A	Other / unclassified error
0xFF	Not ready (ignore UARV)

In summary, the procedure for reading an inverter parameter via the universal access registers is as follows:

1. Write the register value of the parameter to be accessed to the UARA. The UARV will immediately be set to 0, and the read status will immediately be set to 0xFF (not ready).
2. Poll the read status until it is no longer 0xFF.



3. If the read status transitioned to a value of 0, the UARV now contains the value of the desired parameter. This value will be continuously read & updated as long as the UARA is not changed.
4. If the read status transitioned to a nonzero value, then this value represents an error code obtained during the transaction with the inverter. The read attempt from this parameter will be continually repeated as long as the UARA is not changed

### 6.5.3 Universal Access Writes

When a new value is written to the UARV, it takes a certain amount of time for the write transaction with the inverter to be completed and the status to be known. To indicate when the write transaction has been completed, the write status (UARS high byte) is initialized to 0xFF whenever a write operation is performed on the UARV. This code represents “write pending”, and is used to indicate that the write transaction has not yet been completed. Once the write transaction completes and the

success/failure outcome is known by the interface board, this “write pending” status will be replaced by one of the write codes indicate in Table 5. From that point on, as long as the UARA is not changed, the parameter indicated in the UARA will then be continuously read from the inverter, and the read code

**Table 5: UARS Write Codes**

<b>Code</b>	<b>Meaning</b>
0x00	Write successful
0x02	Invalid parameter targeted
0x03	Invalid data value / write to read-only parameter
0x07	Communication to inverter lost
0x0A	Other / unclassified error
0xFF	Write pending

updated if a status change occurs. The write code, however, will not change unless another value is written to the UARV, or unless the UARA is changed.

In summary, the procedure for writing an inverter parameter via the universal access registers is as follows:

1. With the UARA already set to the register value of the parameter to be accessed, write the new parameter value to the UARV. The write status will immediately be set to 0xFF (write pending.)
2. Poll the write status until it is no longer 0xFF.
3. If the write status transitioned to a value of 0, the write was successful.
4. If the write status transitioned to a nonzero value, then this value represents an error code obtained during the transaction with the inverter. The write transaction will not be automatically retried by the interface card.
5. The interface card will return to continuously reading the parameter indicated by the UARA.

### **6.5.4 Disabling**

Disabling universal access reads & writes can be accomplished by setting the UARA to 0. Whenever the UARA is 0, the UARV will also be 0, and the UARS will contain 0x00FF, which indicates “not ready”. This is the default initial condition of the universal access registers.



## 7 FILESYSTEM & FIRMWARE

### 7.1 Overview

The interface card's on-board filesystem is used to store files for use by the application firmware. Currently, the application firmware's main use of the filesystem is to store XML-encoded configuration files that dictate the characteristics of the various protocols. Each protocol that requires configuration will have its own XML file stored on the filesystem. For easy identification, the filename will begin with the corresponding protocol which it configures. For example, a BACnet configuration file's filename will begin with "bacnet", and an Ethernet/IP file will begin with "eip".

Whenever the configuration for a specific protocol is completed, it is suggested that a backup copy of the configuration file be downloaded from the unit to a PC. One reason for this is in case it becomes necessary to restore a previous configuration at a later time. Another reason is that it may be desirable to load multiple units with the same configuration, as a downloaded configuration file can be uploaded again to any compatible unit, allowing the user to easily clone multiple units with the same configuration.

Each time the interface card boots up, it will search the filesystem for the configuration files required by the protocols currently operating in the unit. If it does not find a required file, it will create one and initialize it with factory-default values. Therefore, if it is ever desired to reset a protocol's configuration to factory-default values, this can be easily accomplished by simply deleting the appropriate configuration file from the filesystem and rebooting the unit.

Note that the application firmware uses specific filenames for the configuration files. This means that if a file with a different filename is loaded onto the unit, it will be stored correctly, but will not be used by the application firmware. Similarly, if an existing configuration file's filename is changed, then the unit will again create a default configuration file at next boot-up, which will be stored in the filesystem alongside the file with the changed name.

Configuration files are only read by the protocol drivers at unit boot-up. Therefore, if a new configuration file is loaded onto a unit's filesystem, that unit must be rebooted for the configuration file's settings to take effect. Rebooting a unit can be performed by power-cycling the inverter in which the card is installed, or by selecting the "Reboot Device" button in the Finder application.

Interacting with the filesystem is performed by use of the File Transfer Protocol (FTP). Using FTP allows the user to interact with the files on the interface card's filesystem in the same manner as though they were traditional files stored on a local or remote PC. While there are many different FTP applications available, the following sections will provide general examples of using some of the most commonly-available ones.

Note that certain routers or firewall applications (such as Windows Firewall) can block FTP traffic. If an FTP connection to the interface card cannot be established, be sure to check the computer's firewall settings during troubleshooting, and add an exception to the firewall configuration if necessary.



## 7.2 Initiating FTP via the Finder Utility

After discovering all interface cards on the current subnet as described in section 4, select the target interface card and then click on the “Open FTP Interface” button. This will open the computer’s default FTP application, which could be Windows Explorer, a web browser, or a 3<sup>rd</sup>-party FTP program (whatever the computer/operating system is configured for by default). This example will assume that a web browser (Microsoft Internet Explorer) is configured as the default FTP application.

An authentication dialog will appear (refer to Figure 62.) Enter the currently-configured user name and case-sensitive password (defaults are “root” and “icc”, respectively), then click “Log On.”

The web browser will then display the filesystem’s contents (refer to Figure 63.) FTP access via a web browser allows viewing and downloading files to a computer, but does not allow advanced file manipulation such as cut, paste, drag-and-drop, etc. For advanced file manipulation abilities, use of a different FTP application is required.

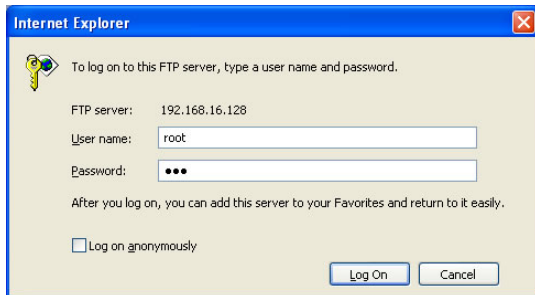


Figure 62: FTP Authentication

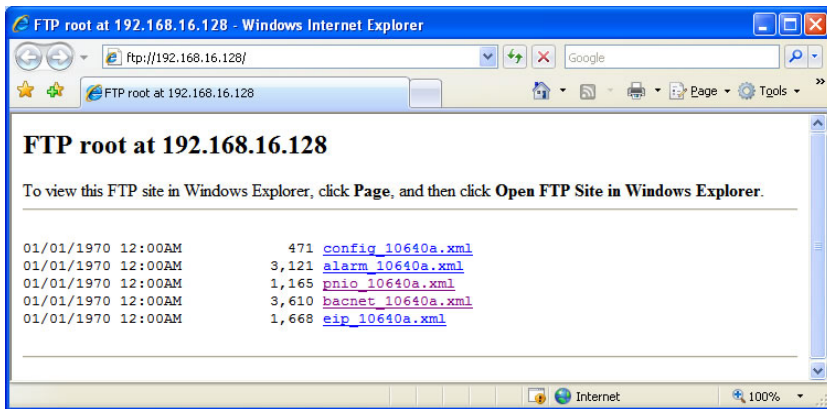


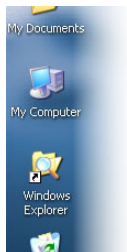
Figure 63: FTP Navigation with Internet Explorer



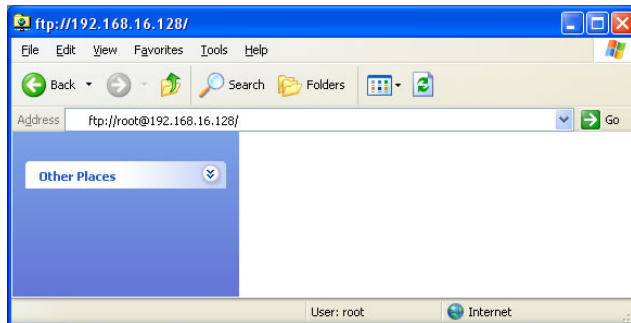
## 7.3 Using FTP with Windows Explorer

To use FTP with Microsoft Windows Explorer, first open either “Windows Explorer” or “My Computer”. Refer to Figure 64. Please note that the indicated procedure, prompts and capabilities outlined here can vary depending on such factors as the installed operating system, firewalls and service packs.

In the “Address” field, type in “ftp://root@” and then the IP address of the target interface card (if the user name has been changed from its default, then replace “root” in “ftp://root@” with the new user name.) Refer to Figure 65.



**Figure 64: Accessing Windows Explorer**

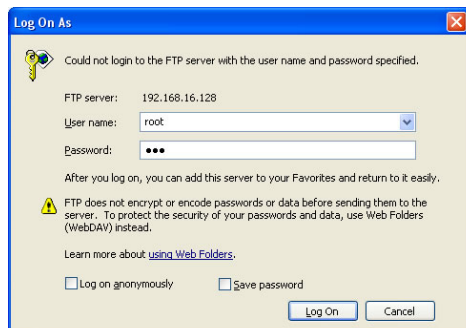


**Figure 65: FTP Navigation with Windows Explorer**

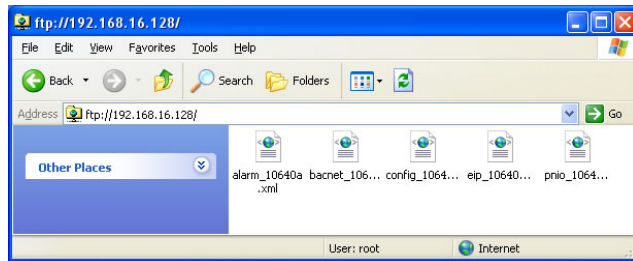


You will then be presented with an authentication dialog (refer to Figure 66.) The user name will already be filled-in. Enter the case-sensitive password (default is “icc”) and click “Log On.”

Windows Explorer will then display the filesystem’s contents (refer to Figure 67.) You can now perform normal file manipulation actions on the available files (cut, copy, paste, open, rename, drag-and-drop transfers etc.) in the same manner as though you were manipulating any traditional file stored on your computer’s hard drive.



**Figure 66: FTP Authentication**



**Figure 67: File Access via Windows Explorer**




## 7.4 Using FTP with a Windows Command Prompt

To use FTP with a Windows command (DOS) prompt, first open a command prompt by either selecting *Start ...All Programs ...Accessories ...Command Prompt*, or by selecting *Start ...Run* and typing “cmd” in the “Run” dialog.

Once the command prompt opens, type “ftp” and the IP address of the target interface card. The FTP client will connect to the unit and then prompt for the username and case-sensitive password (defaults are “root” and “icc”, respectively). Upon successful entry of the authentication information, you will be presented with an “ftp>” prompt. Refer to Figure 68.

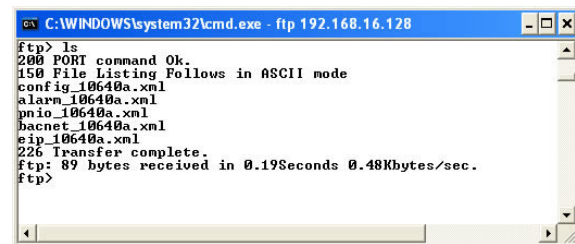
At this point, you can use standard Unix-style file and directory manipulation commands to perform such actions as listing files (Figure 69), copying files to your computer (Figure 70), and copying files to the unit (Figure 71).



```
C:\WINDOWS\system32\cmd.exe - ftp 192.168.16.128

Z:\>ftp 192.168.16.128
Connected to 192.168.16.128.
220 NET+OS 7.3 FTP server ready.
User (192.168.16.128:(none)): root
331 User root OK, send password.
Password:
230 Password OK.
ftp>
```

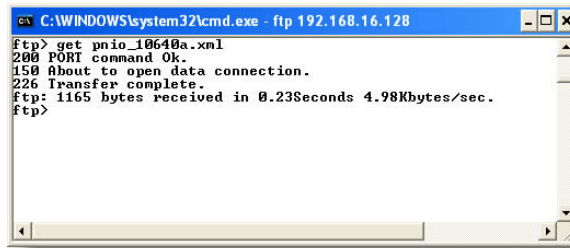
Figure 68: FTP Initiation and Authentication



```
C:\WINDOWS\system32\cmd.exe - ftp 192.168.16.128

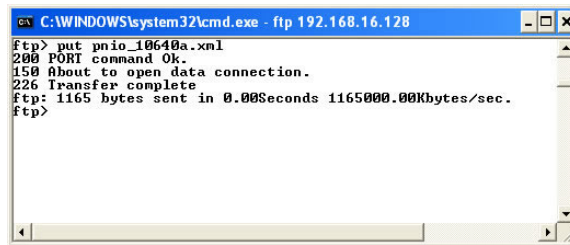
ftp> ls
200 PORT command Ok.
150 File Listing Follows in ASCII mode
config_10640a.xml
alarm_10640a.xml
pnio_10640a.xml
bacnet_10640a.xml
eip_10640a.xml
226 Transfer complete.
ftp: 89 bytes received in 0.19Seconds 0.48Kbytes/sec.
ftp>
```

Figure 69: Listing Files with "ls" Command



```
C:\WINDOWS\system32\cmd.exe - ftp 192.168.16.128
ftp> get prio_10640a.xml
200 PORT command Ok.
150 About to open data connection.
226 Transfer complete.
ftp: 1165 bytes received in 0.23Seconds 4.98Kbytes/sec.
ftp>
```

**Figure 70: Copying a File from the Unit With "get" Command**



```
C:\WINDOWS\system32\cmd.exe - ftp 192.168.16.128
ftp> put prio_10640a.xml
200 PORT command Ok.
150 About to open data connection.
226 Transfer complete
ftp: 1165 bytes sent in 0.00Seconds 1165000.00Kbytes/sec.
ftp>
```

**Figure 71: Copying a File to the Unit With "put" Command**



## 7.5 Using FTP with Core FTP LE

Core FTP LE (Lite) is a 3<sup>rd</sup>-party FTP application that can be downloaded for free from <http://www.coreftp.com>. Core FTP is just one example of the various commercial and freeware FTP client applications available on the internet.

After installing Core FTP LE, run the program. If the “Site Manager” window (Figure 72) does not automatically open, open it by choosing “File...connect”.

Click on the “New Site” button, then enter a Site Name, IP Address, username (default is “root”) and case-sensitive password (default is “icc”). The “Port”, “Timeout”, and “Retries” fields should already contain the default values. Click the “Connect” button when done.

Core FTP LE will then try to connect and authenticate to the FTP server, and if successful, will populate the right-hand side of the main page with the unit’s filesystem contents. Refer to Figure 73.

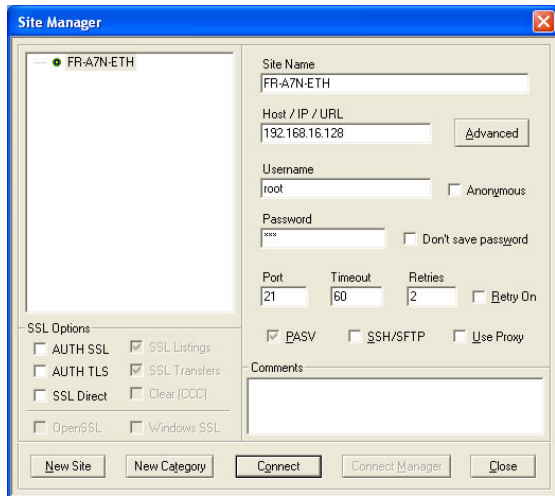


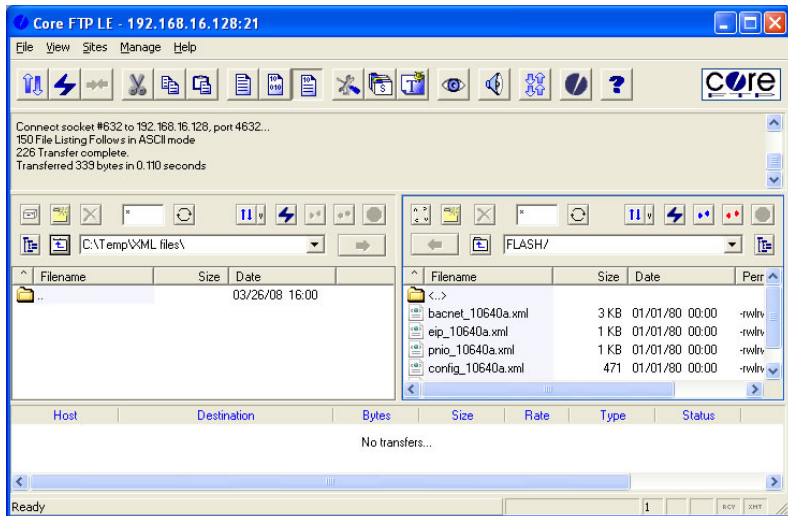


Figure 72: Core FTP Site Manager

Files can be easily downloaded from the unit by choosing the appropriate destination folder on your computer in the left-hand side of the main page, choosing the file to download, and then clicking the “download”  button in the right-hand (source) side. Similarly, files can be easily uploaded to the unit by choosing the file to upload and then clicking the “upload”  button in the left-hand (source) side of the main page.

Like most 3<sup>rd</sup>-party FTP client applications, Core FTP LE has a wide array of configuration and file management capabilities, which are beyond the scope of this manual. Refer to the program’s Help file for more detailed instructions.



**Figure 73: Core FTP in "Connected" State**



## 7.6 Loading New Application Firmware

---

The interface card's embedded firmware resides in flash memory that can be updated in the field. Firmware updates may be released for a variety of reasons, such as custom firmware implementations, firmware improvements and added functionality as a result of user requests. Additionally, it may be necessary to load different firmware onto the unit in order to support various protocols (such as PROFINET I/O).

ICC is continually striving to enhance the functionality and flexibility of our products, and we therefore periodically release new embedded firmware to achieve these goals and meet customer requests. Flash firmware files and all related documentation (such as updated user manuals) can be downloaded from <http://www.iccdesigns.com>. It is suggested that users check this Internet site prior to installation, and then periodically afterwards to determine if new firmware has been released and is available to upgrade their units.

Besides the new firmware file, firmware updates require only a PC with the same FTP client capabilities as described earlier in this section. The new firmware is loaded on the unit via the FTP protocol in the same manner as uploading a configuration (.XML) file. Some notes on uploading new firmware via FTP are:

- Please be sure to read the firmware release notes and updated user's manual for any important notices, behavior precautions or configuration requirements prior to updating your firmware. For example, upgrading to a new firmware version may affect user-defined configuration files; prior to starting an update procedure always back up your configuration file to a PC for later recovery if necessary.

- Because the FTP application firmware in the unit distinguishes application firmware files from XML configuration files by virtue of the filename, don't change the default name of the firmware file to be uploaded to the unit.
- Although the firmware file is uploaded from your PC to the unit in the same manner as configuration files are uploaded, the firmware cannot be downloaded from the unit, because the firmware does not reside in the unit's filesystem like configuration files do.
- After the firmware upload process has been completed (typically requiring 30-45 seconds), the unit will reset automatically 5s after the FTP connection is closed. When the unit boots up again, it will be running the new application firmware, which can be confirmed by observing the version displayed in the web server's information window (refer to section 5.4.1).
- Because the transferred firmware file is not written to the unit's flash memory until the FTP connection is closed, be sure to close the connection in order to trigger the write process. The method of closing an FTP connection differs depending on the FTP transfer mechanism used. For example, if using Windows Explorer, close the window. If using the command prompt method, type "quit" to close the connection. If using Core FTP, click on the "disconnect" button, etc.
- Because the interface's web pages may have changed with the new firmware version, it is recommended to clear your web browser's cache memory before accessing the web server after the firmware update. This will ensure that your web browser will load the updated web server pages directly from the interface, instead of just loading the local (old) copy stored in your computer's cache memory. To accomplish this in Internet Explorer, go to *Tools...Internet Options...Browsing History...Delete*, and delete all temporary internet files.



## **8 PROTOCOL-SPECIFIC INFORMATION**

This section will discuss topics that are specific to each of the supported protocols.

### **8.1 Modbus/TCP**

#### **8.1.1 Overview**

The interface card supports Schneider Electric's Modbus/TCP protocol, release 1.0. The interface is conformance class 0 and partial class 1 and class 2 compliant, and allows up to 8 simultaneous Modbus/TCP client connections (sockets). Other notes of interest are:

- Supported Modbus TCP functions are indicated in Table 6.
- Inverter registers can be addressed as either holding registers (4X references) or input registers (3X references). For example, accessing the output frequency involves accessing holding register 40201 or input register 30201 (i.e.

**Table 6: Supported Modbus/TCP Functions**

<b>Function Code</b>	<b>Function</b>	<b>Modbus/TCP Class</b>
1	Read coils	1
2	Read input status	1
3	Read multiple registers	0
4	Read input registers	1
5	Write coil	1
6	Write single register	1
15	Force multiple coils	2
16	Write multiple registers	0



offset 201).

- Specific bits within inverter registers can be accessed as either coils (0X references) or discrete inputs (1X references).
- Because the transaction is handled locally within the interface card, write data checking is not available for scanned registers (refer to section 6.2.) For example, if a write is performed to a register with a data value that is out-of-range of the corresponding parameter object, no Modbus exception will be immediately returned.
- Both scanned and non-scanned registers can be freely accessed. The response time for scanned register accesses will be faster than that for non-scanned registers.
- The “unit identifier” (UI) field of the request packets is ignored.
- The socket timeout time is determined by the “timeout” setting on the web server’s “Config” tab (refer to section 5.7.5). Depending on the timeout options (refer to sections 5.10.2 and 5.10.3), if a particular open socket experiences no activity for more than the timeout time setting, then the interface assumes that the client or network has experienced some sort of unexpected problem, and will close that socket.
- Because the socket timeout determination is performed on a per-socket basis, note that a certain degree of caution must be exercised when using the network timeout feature to avoid “nuisance” timeouts from occurring. Specifically, do not perform inadvisable behavior such as sending a request from the master device to the interface, and then closing the socket prior to successfully receiving the unit’s response. The reason for this is because the interface will then experience an error when attempting to respond via the now-closed socket, which will immediately trigger the



timeout action. Always be sure to manage socket life cycles “gracefully”, and do not abandon outstanding requests.

- If a socket timeout occurs (regardless of whether it was due to a communication lapse or abnormal socket error), the driver will trigger a timeout event as described in section 5.7.5.

### **8.1.2 Coil & Discrete Input Mappings**

The Modbus TCP driver provides read/write support for coils (0X references) and read-only support for discrete inputs (1X references). These will collectively be referred to from here on out as simply “discretes”. Accessing discretes does not reference any new physical data: discretes are simply indexes into various bits of existing registers. What this means is that when a discrete is accessed, that discrete is resolved by the interface into a specific register, and a specific bit within that register. The pattern of discrete-to-register/bit relationships can be described as follows:

Discrete 1...16 map to register #1, bit0...bit15 (bit0=LSB, bit15=MSB)

Discrete 17...32 map to register #2, bit0...bit15, and so on.

Arithmetically, the discrete-to-register/bit relationship can be described as follows: For any given discrete, the register in which that discrete resides can be determined by:

$$register = \left\lfloor \frac{discrete + 15}{16} \right\rfloor$$

**Equation 1**

Where the bracket symbols “ $\lfloor \ ]$ ” indicate the “floor” function, which means that any fractional result (or “remainder”) is to be discarded, with only the integer value being retained.

Also, for any given discrete, the targeted bit in the register in which that discrete resides can be determined by:

$$bit = (discrete - 1) \% 16 \quad \text{Equation 2}$$

Where “discrete”  $\in [1 \dots 65535]$ , “bit”  $\in [0 \dots 15]$ , and “%” is the modulus operator, which means that any fractional result (or “remainder”) is to be retained, with the integer value being discarded (i.e. it is the opposite of the “floor” function).

For clarity, let’s use Equation 1 and Equation 2 in a calculation example. Say, for instance, that we are going to read coil #34. Using Equation 1, we can determine that coil #34 resides in register #3, as  $\lfloor 3.0625 \rfloor = \lfloor 3 \ r1 \rfloor = 3$ . Then, using Equation 2, we can determine that the bit within register #3 that coil #34 targets is  $(34-1)\%16 = 1$ , as  $33\%16 = \text{mod}(2 \ r1) = 1$ . Therefore, reading coil #34 will return the value of register #3, bit #1.



## **8.2 EtherNet/IP**

---

### **8.2.1 Overview**

The EtherNet/IP protocol is an application-level protocol implemented on top of the Ethernet TCP/IP and UDP/IP layers. It shares its object model with ControlNet and DeviceNet through the Common Industrial Protocol (CIP). This protocol allows the transfer of data and I/O over Ethernet.

EtherNet/IP incorporates both the TCP and UDP layers of Ethernet in the transmission of data. Because TCP/IP is a point-to-point topology, EtherNet/IP uses this layer only for explicit messaging; i.e. those messages in which the data field carries both protocol information and instructions for service performance. With explicit messaging, nodes must interpret each message, execute the requested task and generate responses. These types of messages can be used to transmit configuration, control and monitor data.

The UDP/IP protocol layer, which has the ability to multi-cast, is used for implicit (I/O or cyclic) messaging. With I/O messaging, the data field contains only real-time I/O data; no protocol information is sent because the meaning of the data is pre-defined at the time the connection is established, which in turn minimizes the processing time of the node during run-time. I/O messages are short and have low overhead, which allows for the time-critical performance needed by controllers.

The interface card supports both explicit and I/O messaging. Further, two different types of I/O messaging are supported. One type (invoked when the client opens a connection to the interface using assembly instances 20 & 70 or 21 & 71) is included with the implementation of the AC/DC drive profile,

and requires no user configuration. The other type, however, is entirely user-configurable, and is invoked when the client opens a connection to the interface using assembly instances 100 and 150.

The following sections demonstrate specific examples of how to use EtherNet/IP to transfer data between the inverter and Allen-Bradley Logix-brand PLCs.

Some other notes of interest are:

- The interface card supports the EtherNet/IP protocol, as administered by the Open DeviceNet Vendor Association (ODVA).
- This product has been self-tested by ICC, Inc. and found to comply with ODVA EtherNet/IP Conformance Test Software Version A-5.
- I/O connection sizes for assembly instances 100 and 150 are adjustable between 0 and 64 bytes (32 registers max @ 2 bytes per register = 64 bytes). Because registers are 16-bit elements, however, connection sizes cannot be odd (i.e. 1, 3, 5 etc.)
- The interface card's product type code is 2 (AC Drive.)
- Supports unconnected messages (UCMM), and up to 16 simultaneous class 1 (I/O) or class 3 (explicit) connections.
- Assembly instances 100 and 150: if a register entry in the consumed data configuration array is 0, then any consumed data that corresponds to that location will be ignored. Conversely, if a register entry in the produced data configuration array is 0, then any produced data that corresponds to that location will be a default value of 0. Refer to section 5.8.4 for further information on the data configuration arrays.



- Class 1 implicit I/O supports both multicast and point-to-point (unicast) when producing data in the T→O direction.
- Point-to-point class 1 connected messages will be produced targeting the IP address of the device that instantiated the connection, UDP port 0x08AE (UDP port 2222).
- If a class 1 point-to-point connection is established in the (T→O) direction, no more class 1 connections can be established.
- If a class 1 connection's consuming half (O→T) times out, then the producing half (T→O) will also time-out and will stop producing.
- If a class 1 or class 3 connection timeout occurs, the driver will trigger a timeout event as described in section 5.7.5. For class 1 connections, the timeout value is dictated by the scanner/client and is at least four times the RPI (Requested Packet Interval). For class 3 connections, the timeout value is also dictated by the scanner/client, but is typically a much larger value than for class 1 connections.

### **8.2.2 ODVA AC/DC Drive Profile**

The interface card supports the ODVA AC/DC drive profile. No special EtherNet/IP configuration of the interface card is required when using the AC/DC drive profile: all that is needed is that the controller must target either assembly instances 20 & 70 or 21 & 71 in its connection parameters.



The AC/DC drive profile implementation provides support for several required CIP objects, which are specified in Table 7. While the various supported attributes of all of these objects are accessible via explicit messaging, the main intent of using the AC/DC drive profile is to interact with the predefined input and output assembly instances via an I/O connection. The structure of these assembly instances is defined by the EtherNet/IP specification in order to engender interoperability among different vendor's products. This section will focus primarily on the format of the AC/DC drive profile I/O assemblies supported by the interface card, and the inverter data which their various constituent elements map to.

**Table 7: AC/DC Drive Profile-Related Objects**

<b>Class Code</b>	<b>Object Name</b>
0x04	Assembly Object
0x28	Motor Data Object
0x29	Control Supervisor Object
0x2A	AC Drive Object

**Output Instances 20 and 21 Detail**

Instance	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
20	0						Fault Reset		Run Fwd
	1								
	2	Speed Reference (Low Byte)							
	3	Speed Reference (High Byte)							
21	0		NetRef	NetCtrl			Fault Reset	Run Rev	Run Fwd
	1								
	2	Speed Reference (Low Byte)							
	3	Speed Reference (High Byte)							

**Mapping Detail**

**Run Fwd:** forward rotation command (0=forward rotation off, 1=forward rotation on). Maps to inverter register 9, bit 1 (forward rotation command).

**Run Rev:** reverse rotation command (0=reverse rotation off, 1=reverse rotation on). Maps to inverter register 9, bit 2 (reverse rotation command).

**Fault Reset:** Inverter reset command (0=no action, 0→1 rising edge=reset). Maps to inverter register 2 (inverter reset).



**NetCtrl:** Run/stop control source selection (0=local control, 1=network control). Maps to inverter register 10 (operation mode).

**NetRef:** Speed reference source selection (0=local control, 1=network control). Maps to inverter register 10 (operation mode).

**NetCtrl and NetRef explanation:** Although the intent of these bits is to be able to independently separate the inverter’s run/stop command source and speed reference source, the inverter only has one externally-accessible “operation mode” command register, which controls both the run/stop source and speed reference source simultaneously. Therefore, Table 8 can be used to determine the resultant value of the “operation mode” register based on the various combinations of NetCtrl and NetRef.

**Table 8: NetCtrl/NetRef Behavior**

NetCtrl	NetRef	“Operation Mode” Register Value
0	0	0x0010 (EXT mode)
0	1	No change from current value
1	0	
1	1	0x0014 (NET mode)

**Speed Reference:** Inverter speed reference in RPM. Maps to inverter register 14 (frequency command). The frequency command value written to the inverter depends on the settings of parameters *Pr. 37* and *Pr. 144* (refer to Table 9). The speed reference component of the AC/DC drive profile output instances is always in units of RPM. With certain combinations of *Pr. 37* and *Pr. 144*, however, the inverter interprets its incoming frequency command either as an actual frequency command (with units of Hz) or as a speed reference (with units of RPM). Therefore, the interface card will apply an RPM-to-Hz conversion equation when necessary, or will pass the consumed speed reference value directly on to the inverter as-is when the inverter is already expecting the frequency command to be in units of RPM.

The RPM-to-Hz conversion equation is  $[RPM \times \text{number of motor poles} / 120]$ . Table 9 indicates the scenarios under which the conversion equation is and is not applied, and the value used for the “number of motor poles” component in the numerator of the conversion equation. For more information on the manners in which *Pr. 37* and *Pr. 144* affect the inverter’s operation, please refer to the appropriate inverter user’s manual. Note that the values of *Pr. 37* and *Pr. 144* are read by the interface card only at boot-up, so if these parameter values are changed, then the interface card must be rebooted in order for it to recognize the new values and once again determine the required conversion scenarios.

**Table 9: Inverter Speed Reference and Speed Actual Conversion Scenarios**

<i>Pr. 37</i>	<i>Pr. 144</i>	Conversion Applied?	Number of Motor Poles Used in Conversion Equation
0 (default value)	0	Yes	4
	2..10	Yes	2..10
	102..110	No	N/A
1..9998	0	Yes	4
	2..10	No	N/A
	102..110	Yes	2..10

### Input Instances 70 and 71 Detail

Instance	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
70	0						Running1		Failed
	1								
	2	Speed Actual (Low Byte)							
	3	Speed Actual (High Byte)							
71	0	At Reference	Ref From Net	Ctrl From Net	Ready	Running2 (REV)	Running1 (FWD)	Warning	Failed
	1	Drive State							
	2	Speed Actual (Low Byte)							
	3	Speed Actual (High Byte)							

### Mapping Detail

**Failed:** Inverter fault signal (0=not faulted, 1=faulted). Maps to inverter register 260 (status word), bit 15 (alarm occurrence).

**Warning:** This bit is not used (it is always 0).

**Running1 (FWD):** Running forward status signal (0=not running forward, 1=running forward). Maps to inverter register 260 (status word), bits 0 and 1. The Running1 bit will be 1 whenever bit 0 (inverter running) and bit 1 (forward rotation) of the status word are both 1, and will be 0 otherwise. Note that if



the setting of inverter parameter *Pr. 190* (RUN terminal function selection) is changed from its factory-default value, the indicated status of the Running1 bit will no longer be accurate.

Running2 (REV): Running reverse status signal (0=not running reverse, 1=running reverse). Maps to inverter register 260 (status word), bits 0 and 2. The Running2 bit will be 1 whenever bit 0 (inverter running) and bit 2 (reverse rotation) of the status word are both 1, and will be 0 otherwise. Note that if the setting of inverter parameter *Pr. 190* (RUN terminal function selection) is changed from its factory-default value, the indicated status of the Running2 bit will no longer be accurate.

Ready: Inverter ready signal (0=not ready, 1=ready). The Ready bit will be 1 whenever the Drive State attribute (see below) is in the Ready, Enabled or Stopping state.

CtrlFromNet: Inverter command reference status (0=command reference is not from network, 1=command reference is from network). Maps to inverter register 261 (operation mode status). CtrlFromNet will be 1 whenever the operation mode status is a value of 0x0004 (NET mode), and will be 0 otherwise.

RefFromNet: Inverter speed reference status (0=speed reference is not from network, 1=speed reference is from network). Maps to inverter register 261 (operation mode status). RefFromNet will be 1 whenever the operation mode status is a value of 0x0004 (NET mode), and will be 0 otherwise.

AtReference: Up-to-speed signal (0=not up-to-speed, 1=up-to-speed). Maps to inverter register 260 (status word), bit 3 (SU/up to frequency). Note that if the setting of inverter parameter *Pr. 191* (SU terminal function selection) is changed from its factory-default value, the indicated status of the AtReference bit will no longer be accurate.

**Drive State:** Indicates the current state of the Control Supervisor Object state machine. Refer to the ODVA EtherNet/IP specification (object library) for detailed information on the Control Supervisor Object state machine.

**Speed Actual:** Inverter operating speed in RPM. Maps to inverter register 201 (output frequency). The output frequency value read from the inverter depends on the settings of parameters *Pr. 37* and *Pr. 144* (refer to Table 9). The speed actual component of the AC/DC drive profile input instances is always in units of RPM. With certain combinations of *Pr. 37* and *Pr. 144*, however, the inverter provides its output frequency either as an actual frequency (with units of Hz) or as a speed (with units of RPM). Therefore, the interface card will apply a Hz-to-RPM conversion equation when necessary, or will pass the inverter's output frequency value directly on to the network as-is when the inverter is already providing the output frequency in units of RPM.

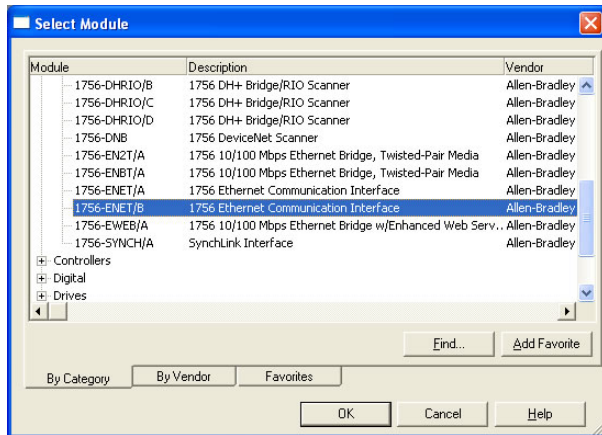
The Hz-to-RPM conversion equation is  $[output\ frequency \times 120 / number\ of\ motor\ poles]$ . Table 9 indicates the scenarios under which the conversion equation is and is not applied, and the value used for the “number of motor poles” component in the denominator of the conversion equation. For more information on the manners in which *Pr. 37* and *Pr. 144* affect the inverter's operation, please refer to the appropriate inverter user's manual. Note that the values of *Pr. 37* and *Pr. 144* are read by the interface card only at boot-up, so if these parameter values are changed, then the interface card must be rebooted in order for it to recognize the new values and once again determine the required conversion scenarios.



### 8.2.3 ControlLogix Examples: Setup

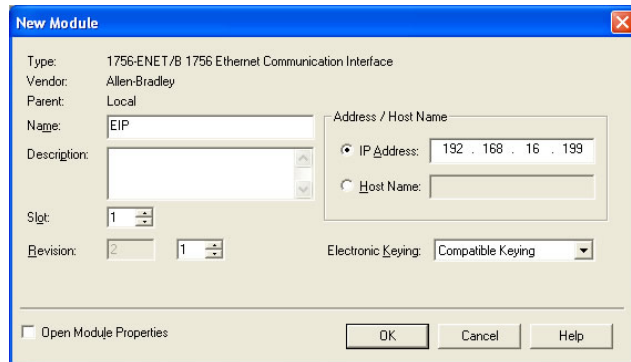
This section will demonstrate how to initially setup a ControlLogix PLC (such as a 1756-L61) coupled with a 1756-ENET/B communication interface (adjust this procedure according to your specific equipment). Later sections will provide specific read/write examples using this configuration with I/O or explicit messaging.

- 1) Run RSLogix 5000, and create a new configuration.
- 2) To add a 1756-ENET/B to your I/O configuration, first switch to offline mode.
- 3) Right click on the I/O Configuration node in the controller organizer view and choose “New Module...”
- 4) The “Select Module” window will open.
- 5) Under “Communications”, select “1756-ENET/B”, and click OK. Refer to Figure 74.
- 6) The “New Module” window will open. Refer to Figure 75.
- 7) Assign the Ethernet module a name (we will use “EIP”) and an IP address,

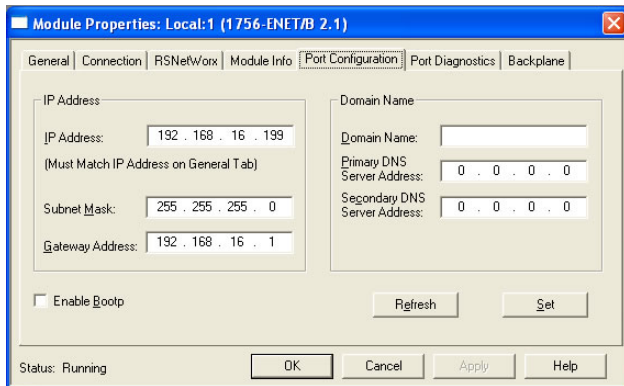


**Figure 74: Adding a New Module**

- deselect “Open Module Properties”, and click OK.
- 8) Download the configuration.
  - 9) Switch to online mode. Right click on the 1756-ENET/B module in the I/O Configuration and choose “Properties”.
  - 10) Select the Port Configuration tab from the Module Properties dialog box.
  - 11) Confirm that the IP Address, Subnet Mask and Gateway Address fields are configured correctly. The IP Address must match the IP Address entered when the new module was first created. Refer to Figure 76.
  - 12) Apply the settings using the “Set” button.
  - 13) You should now be able to confirm that the 1756-ENET/B module is configured properly by (for example) opening the module’s web interface in a web browser.



**Figure 75: Identifying the New Module**



**Figure 76: Confirming the Module's Properties**

### **8.2.4 ControlLogix Example: I/O Messaging**

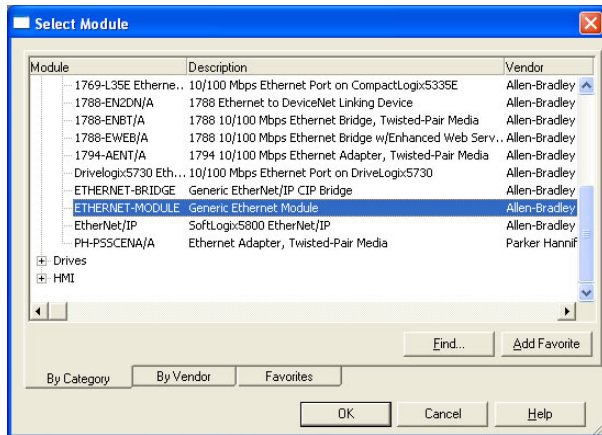
This section will demonstrate how to setup and use an EtherNet/IP I/O connection via vendor-specific assembly instances 100 & 150. EtherNet/IP I/O messaging allows the inverter's registers to be directly mapped into tags in the ControlLogix PLC. Once an I/O connection is established, it is automatically synchronized at an interval defined by the Requested Packet Interval (RPI).



- 1) Switch to offline mode.
- 2) Right click on the 1756-ENET/B node under the I/O Configuration in the controller organizer view and choose “New Module...”
- 3) Choose “Generic Ethernet Module” in the Select Module dialog box and click “OK”. Refer to Figure 77.
- 4) The module properties dialog box will open (refer to Figure 78). Enter a Name and Description which will allow easy identification of the inverter on the network (the tags created in RSLogix 5000 will be derived from this Name). Because all inverter data is stored as 16-bit registers, change the “Comm Format” selection to “Data-INT”. Enter the IP address of the targeted interface card.

In the “Connection Parameters” portion of the dialog box, enter the following information:

**Input:** The Input Assembly is the collection of monitor data that is produced by the interface card and is received as an input to the PLC. Its



**Figure 77: Adding a New Generic Ethernet Module**



structure is defined by the Produced Register Configuration as described in section 5.8.4. The Input Assembly Instance must be set to 150 when connecting to the vendor-specific I/O assembly instances (or 70/71 when using the ODVA AC/DC drive profile), and the size must be set to the number of 16-bit registers that we wish to receive from the interface card. For the purposes of this example, we are assuming that the produced configuration array is defined as shown in Figure 31, with two relevant registers (261 and 201). We therefore set the Input Size to 2.

**Output:** The Output Assembly is the collection of command & configuration data that is sent as an output from the PLC and consumed by the interface card. Its structure is defined by the Consumed Register Configuration as described in section 5.8.4. The Output Assembly Instance must be set to 100 when connecting to the vendor-specific I/O assembly instances (or 20/21 when using the ODVA AC/DC drive profile), and the size must be set to the number of 16-bit registers that we wish to send to the interface card. For the purposes of this example, we are assuming that the consumed configuration array is defined as shown in Figure 31, with two relevant registers (9 and 14). We therefore set the Output Size to 2.

**Configuration:** The Configuration Assembly Instance is unused, and its

The screenshot shows the 'New Module' dialog box with the following details:

- Type:** ETHERNET-MODULE Generic Ethernet Module
- Vendor:** Allen-Bradley
- Parent:** EIP
- Name:** FR\_A7N\_ETH
- Description:** Conveyor #6 first roller
- Comm Format:** Data - INT
- Address / Host Name:**
  - IP Address: 192 . 168 . 16 . 22
  - Host Name:
- Connection Parameters:**

	Assembly Instance:	Size:	
Input:	150	2	(16-bit)
Output:	100	2	(16-bit)
Configuration:	1	0	(8-bit)
Status Input:			
Status Output:			

Buttons at the bottom:  Open Module Properties, OK, Cancel, Help.

Figure 78: Interface Card Module Properties

instance number and size are therefore irrelevant (you can just enter “1” and “0”, respectively).

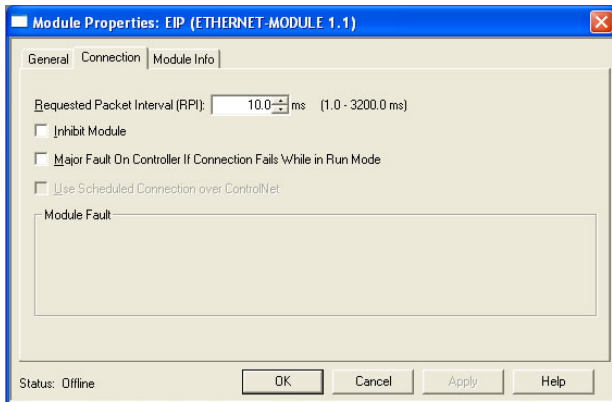
When done, click “OK”.

- 5) You should now see the new module (named “ETHERNET-MODULE FR\_A7N\_ETH”) in the 1756-ENET/B branch under the I/O Configuration in the controller organizer view. Right click on this new module, choose “Properties”, and select the Connection tab. Refer to Figure 79.

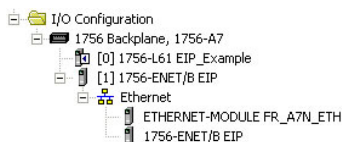
Confirm the setting of the Requested Packet Interval (RPI). The RPI defines the amount of time (in milliseconds) between data exchanges across an I/O connection. The smallest RPI supported by the interface card is 10ms.

Click OK when done.

- 6) After adding the I/O Module to the configuration, the full I/O Configuration tree should appear similar to Figure 80.



**Figure 79: Module Properties Connection Tab**



**Figure 80: I/O Configuration Tree**

7) Switch to online mode and download the project to the PLC. Verify that the newly-added inverter is available and operating correctly by observing any indications shown on the inverter's icon. When the inverter's icon is selected, its status and any available error messages will be displayed in the area below the project tree. Refer to Figure 81. Also confirm that the interface card's "Network Status" LED should be solid green, indicating an "online/connected" state.

8) By double-clicking "Controller Tags" in the project tree, it is possible to view the newly-added tags. Refer to Figure 82. The FR\_A7N\_ETH:C configuration tag is unused, the FR\_A7N\_ETH:I tag allows viewing of the input data, and the FR\_A7N\_ETH:O tag allows modification of the output data. These tags will be synchronized with the inverter at whatever rate was established for the module's RPI.

We can directly interact with these

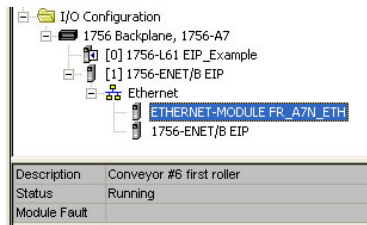


Figure 81: Online Module Status

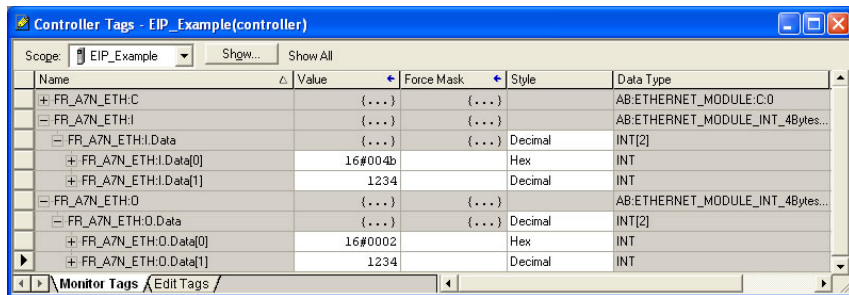


Figure 82: Controller Tags for I/O Access

tags in order to control and monitor the inverter. In Figure 82, for example, we can see that the first 16-bit word of output data (FR\_A7N\_ETH:O.Data[0]) has been set to a hexadecimal value of 0x0002. Referring back to Figure 31, we can see that the first element of the consumed register configuration references register 9, which is the inverter's command register. A value of 0x0002, therefore, means that the run forward bit has been turned ON.

Similarly, we can see that the second 16-bit word of output data (FR\_A7N\_ETH:O.Data[1]) has been set to a decimal value of 1234. Once again referring back to Figure 31, we can see that the second element of the consumed register configuration references register 14, which is the inverter's frequency command register. A value of 1234, therefore, equates to a frequency command of 12.34Hz.

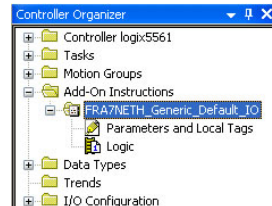
The input data from the inverter shows similar expected results. Values of 0x004B and 1234 corresponding to registers 261 (status register) and 201 (output frequency), respectively, are consistent with the inverter running at the parameters commanded by the output tag.

### **8.2.5 ControlLogix Example: Generic Default I/O Add-On Instruction**

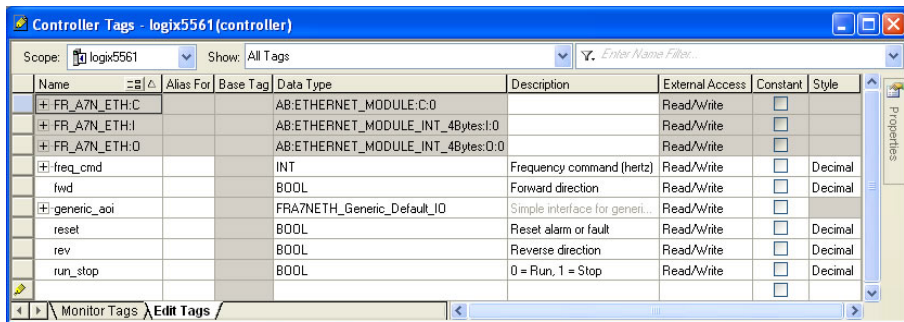
The generic default I/O add-on instruction is a simple interface to command and monitor the inverter. It is based on the vendor-specific assembly instances 100 & 150 and the default produce and consume data configuration arrays (refer to section 5.8.4).

- 1) Complete all the steps in section 8.2.4.

- 2) Right click on “Add-On Instructions” in the controller organizer view and select “Import Add-On Instruction”. Browse and import the generic default I/O add-on instruction. Refer to Figure 83.
- 3) Double click “Controller Tags” in the controller organizer view and select the “Edit Tags” tab at the bottom.
- 4) Create the tags in Figure 84.

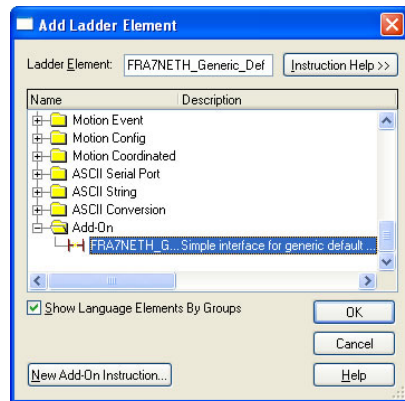


**Figure 83: Generic Default IO Add-On Instruction**

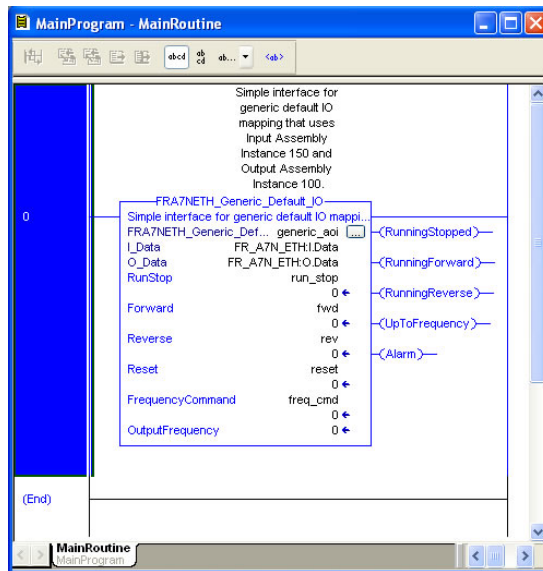


**Figure 84: Create Generic Default AOI Tags**

- 5) Double click “MainRoutine” under Tasks ...MainTask ...MainProgram in the controller organizer view.
- 6) Right click on the first ladder logic rung in the MainRoutine window and select “Add Ladder Element...”
- 7) The “Add Ladder Element” window appears.
- 8) Select the generic default I/O add-on instruction in the Add-On folder. Refer to Figure 85.
- 9) Click OK.
- 10) Edit the add-on instruction according to Figure 86.
- 11) The program is now complete.
- 12) Save, download and run the program.



**Figure 85: Add Generic Default Add-On Instruction**



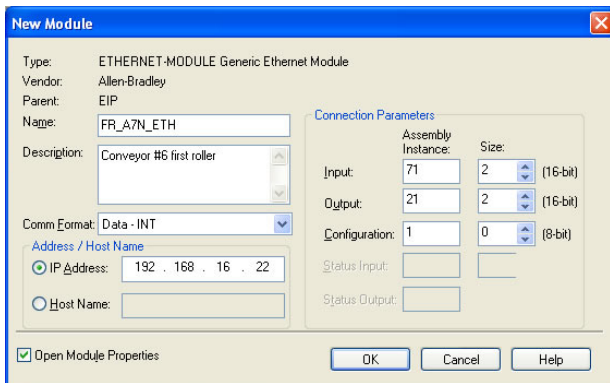
**Figure 86: Configure Generic Default AOI**



## 8.2.6 ControlLogix Example: AC/DC Drive Profile Add-On Instruction

The AC/DC drive profile add-on instruction is a simple interface to command and monitor the inverter. It is based on the assembly instances 21 & 71.

- 1) Complete all the steps in section 8.2.4. Please note that the Assembly Input Instance must be changed to 71 and the Assembly Output Instance must be changed to 21. Refer to Figure 87.



**Figure 87: AC/DC Drive Profile Generic Ethernet Module Configuration**

- 2) Right click on “Add-On Instructions” in the controller organizer view and select “Import Add-On Instruction”. Browse and import the AC/DC drive profile add-on instruction. Refer to Figure 88.

- 3) Double click “Controller Tags” in the controller organizer view and select the “Edit Tags” tab at the bottom.
- 4) Create the tags in Figure 89.

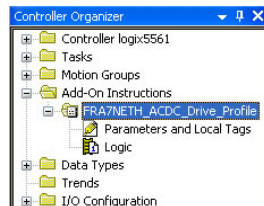


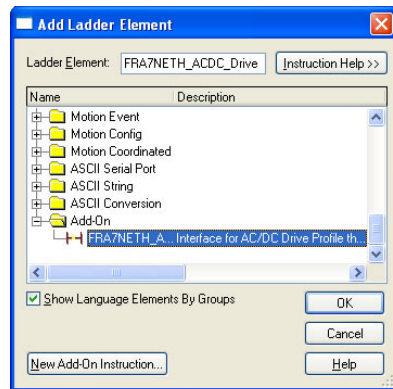
Figure 88: AC/DC Drive Profile Add-On Instruction

The screenshot shows the 'Controller Tags - logix5561 (controller)' window. The 'Scope' is set to 'logix5561' and 'Show' is set to 'All Tags'. The table below lists the tags created for the AC/DC Drive Profile AOI.

Name	Alias For	Base Tag	Data Type	Description	External Access	Constant	Style
+ acdc_aoi			FRA7NETH_ACDC_Drive_Profile	Interface for AC/DC Drive Pr...	Read/Write	<input type="checkbox"/>	
+ FR_A7N_ETH.C			AB:ETHERNET_MODULE:C:0		Read/Write	<input type="checkbox"/>	
+ FR_A7N_ETH.I			AB:ETHERNET_MODULE_INT_4Bytes:I:0		Read/Write	<input type="checkbox"/>	
+ FR_A7N_ETH.O			AB:ETHERNET_MODULE_INT_4Bytes:O:0		Read/Write	<input type="checkbox"/>	
net_ctrl			BOOL	Control from network	Read/Write	<input type="checkbox"/>	Decimal
net_ref			BOOL	Speed reference from network	Read/Write	<input type="checkbox"/>	Decimal
reset			BOOL	Fault reset	Read/Write	<input type="checkbox"/>	Decimal
run_fwd			BOOL	Run forward direction	Read/Write	<input type="checkbox"/>	Decimal
run_rev			BOOL	Run reverse direction	Read/Write	<input type="checkbox"/>	Decimal
+ speed_ref			INT	Speed reference (RPM)	Read/Write	<input type="checkbox"/>	Decimal

Figure 89: Create AC/DC Drive Profile AOI Tags

- 5) Double click “MainRoutine” under Tasks ...MainTask ...MainProgram in the controller organizer view.
- 6) Right click on the first ladder logic rung in the MainRoutine window and select “Add Ladder Element...”
- 7) The “Add Ladder Element” window appears.
- 8) Select the AC/DC drive profile add-on instruction in the Add-On folder. Refer to Figure 90.
- 9) Click OK.
- 10) Edit the add-on instruction according to Figure 91.
- 11) The program is now complete.
- 12) Save, download and run the program.



**Figure 90: Add AC/DC Drive Profile Add-On Instruction**

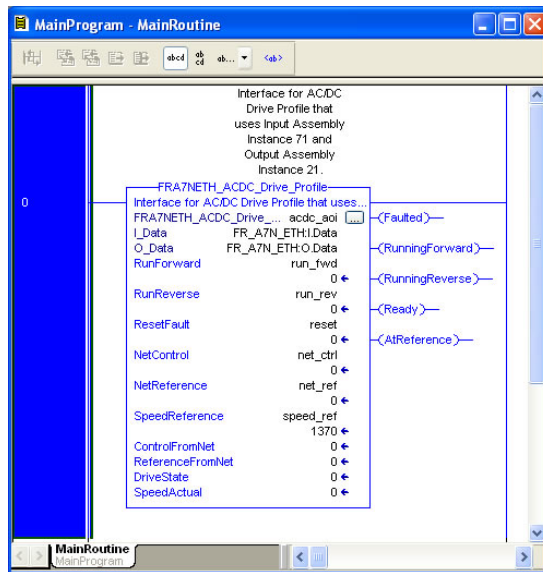


Figure 91: Configure AC/DC Drive Profile AOI



### **8.2.7 Explicit Messaging Tag Reference**

When class 3 (explicit messaging) connections are used, register contents are read from and written to the interface card via EtherNet/IP by reference to “tag names”. Tags are read via the EtherNet/IP “data table read” service, and tags are written via the EtherNet/IP “data table write” service. Different tags exist for reading vs. writing.

Any given scanned or non-scanned register can be accessed with its own unique tag name, or an array tag can be used to access a group of registers with one PLC instruction. Tag names are generated according to the following structure:

**[action prefix]\_reg\_[register number]**

Where

**[action prefix]** is a 2-character field, and is equal to either “rd” for read tags, or “wr” for write tags. Although commonly followed for naming clarity, this “read vs. write” naming convention is not strictly enforced by the interface card, however: it is perfectly acceptable to write to a tag that starts with “rd” and read from a tag that starts with “wr”.

**\_reg\_** is just the 5-character sequence “\_reg\_”.

**[register number]** is a 1- to 4-character field (“1”, “2”...“507”, “508”, “1000”, “1001”... etc.) corresponding to the referenced register number.

Examples

Read “output voltage” (register #203).....rd\_reg\_203

Write “frequency command” (register #14).....wr\_reg\_14



Read "inverter status" (register #261).....rd\_reg\_261  
 Write "acceleration time" (*Pr. 7* / register #1007) .....wr\_reg\_1007

Additionally, a few special tags exist which provide alternate names for commonly-used registers. These are specified in Table 10.

**Table 10: Special Tag Reference**

Service	Tag Name	Register Start	Same As...
Data table read	rd_uar_addr	100	rd_reg_100
Data table read	rd_uar_val	101	rd_reg_101
Data table read	rd_uar_stat	102	rd_reg_102
Data table read	rd_reg_monitor	201	rd_reg_201
Data table read	rd_freq_out	201	rd_reg_201
Data table read	rd_inv_stat	261	rd_reg_261
Data table read	rd_op_mode_stat	262	rd_reg_262
Data table read	rd_reg_alarm_history	501	rd_reg_501
Data table write	wr_inv_rst	2	wr_reg_2
Data table write	wr_param_clr1	3	wr_reg_3
Data table write	wr_all_param_clr1	4	wr_reg_4
Data table write	wr_param_clr2	6	wr_reg_6
Data table write	wr_all_param_clr2	7	wr_reg_7
Data table write	wr_cmd_reg	9	wr_reg_9
Data table write	wr_op_mode	10	wr_reg_10

Service	Tag Name	Register Start	Same As...
Data table write	wr_freq_cmd	14	wr_reg_14
Data table write	wr_freq_cmd_eeprom	15	wr_reg_15
Data table write	wr_uar_addr	100	wr_reg_100
Data table write	wr_uar_val	101	wr_reg_101
Data table write	wr_reg_alarm_history1	501	wr_reg_501

To read data from the interface card, the application PLC program must reference a “source element” from which to start reading and the “number of elements” to read. The “source element” will be a tag name constructed according to the naming convention shown above, or a special tag as shown in Table 10. The “source element” can be either a base tag (such as “rd\_reg\_201”, which starts at register 201), or an offset from a base tag (such as “rd\_reg\_201[4]”, which starts at register 201+4 = register 205, the inverter’s frequency setting monitor register).

In a similar manner, to write data to the interface card, the application PLC program must reference a “destination element” to which to start writing and the “number of elements” to write. Again, the “destination element” will be a tag name constructed according to the naming convention shown above, or a special tag as shown in Table 10.

Whether reading or writing, the “number of elements” can be any quantity of registers from 1 to the maximum allowable length.

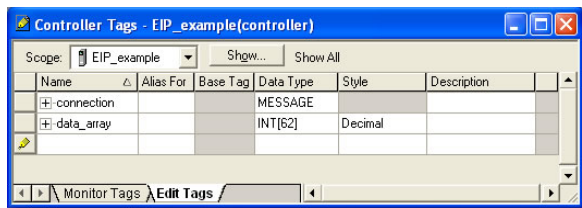


## 8.2.8 ControlLogix Example: Read a Register Block

This example program will show how to continuously read a block of registers from the inverter with a single MSG instruction. Only one read request is outstanding at any given time. Note that the steps for configuring the MSG instruction for the PLC5E may vary slightly. For more information about the MSG instruction, please refer to the PLC's Instruction Set Reference Manual.

### 1) Create new Tags.

- a) Double click "Controller Tags" in the controller organizer view.
- b) The "Controller Tags" window appears. Refer to Figure 92.
- c) Select the "Edit Tags" tab at the bottom.
- d) Create a new tag by entering "connection" in the first blank Name field, and change its Data Type to "MESSAGE". This tag will contain configuration information for the MSG instruction.
- e) Select the "Monitor Tags" tab. Expand the "connection" tag by clicking on the "+" sign next to the tag name. Scroll down to the connection.UnconnectedTimeout field and change its value from the default 30000000 (30s in 1uS increments) to 1000000 (1s). This value determines how long to wait before timing out and retransmitting a connection request if a connection failure occurs. Refer to Figure 93.

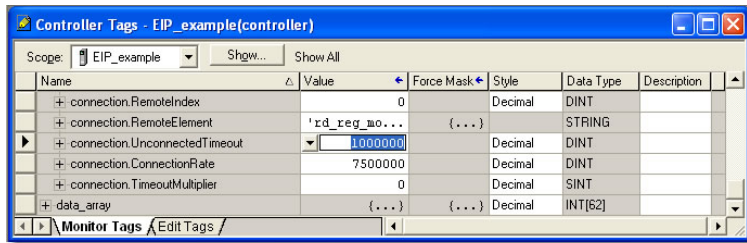


**Figure 92: Create New Tags**



f) Collapse the “connection” tag again by clicking on the “-” sign next to the tag name.

g) Select the “Edit Tags” tab again. Create another new tag by entering “data\_array” in the next blank Name field, and change its Data Type by typing in “INT[62]” in the Data Type field. This tag is an array of INTs that will be able to hold up to 62 16-bit registers from the inverter. Always make sure that the destination tag size is large enough to hold all elements to be read.

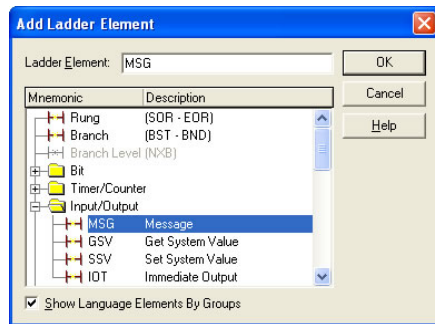


**Figure 93: Reduce the UnconnectedTimeout Value**

## 2) Add a MSG instruction to the main program.

a) Double click “MainRoutine” under Tasks ...MainTask ...MainProgram in the controller organizer view.

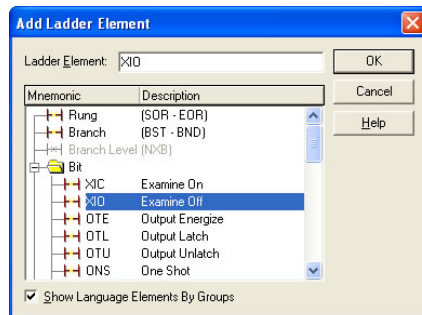
b) Right click on the first ladder logic rung in the MainRoutine window and select “Add Ladder Element...”



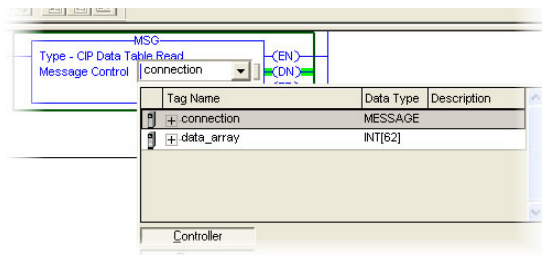
**Figure 94: Adding a MSG Instruction**



- c) The “Add Ladder Element” window appears.
  - d) Select the “MSG” instruction in the Input/Output folder. Refer to Figure 94.
  - e) Click OK.
- 3) Add an XIO element to the main program.**
- a) Right click on the ladder logic rung containing the MSG instruction in the MainRoutine window and select “Add Ladder Element...” again.
  - b) The “Add Ladder Element” window appears.
  - c) Select the “XIO” element in the Bit folder. Refer to Figure 95.
  - d) Click OK.
- 4) Configure the MSG instruction.**
- a) Edit the “Message Control” field on the MSG instruction to use the previously-created “connection” tag. Refer to Figure 96.
  - b) Click the message configuration button (“...”) in the MSG instruction. The



**Figure 95: Adding an XIO Element**



**Figure 96: MSG Instruction Tag Assignment**

“Message Configuration” window will open. Refer to Figure 97.

c) “Configuration” tab settings:

i) Change the “Message Type” to “CIP Data Table Read”.

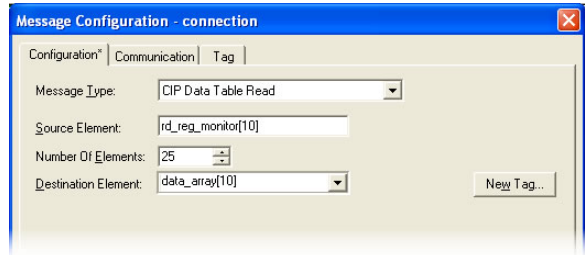
ii) In the “Source Element” field, enter the read tag you wish to access (refer to section 8.2.5.) In this example, we will be reading a total of 25 registers beginning at rd\_reg\_monitor[10].

Offset 10 in the interface card’s rd\_reg\_monitor root tag (which starts at register 201) refers to  $201+10 =$  register 211 (output current peak value). If we wish, we could also use the tag name which references output current peak value directly (rd\_reg\_211) to achieve the same results.

iii) Enter the Number Of Elements to read. In this example, we will read 25 registers.

iv) For the Destination Element, either directly type in “data\_array[10]”, or select element #10 in the data\_array tag via the drop-down box (refer to Figure 98). The destination could be any offset in the data\_array tag, as long as the offset plus the Number Of Elements (25) does not exceed the tag’s defined size (62).

d) “Communication” tab settings (refer to Figure 99):



**Figure 97: MSG Instruction Configuration**

- i) Enter the Path to the interface card. A typical path is formatted as “*Local\_ENB,2,target\_IP\_address*”, where:
- *Local\_ENB* is the name of the 1756-ENBx module in the local chassis (we named ours “EIP” in section 8.2.3),
  - 2 is the Ethernet port of the 1756-ENBx module in the local chassis, and
  - *target\_IP\_address* is the IP address of the target node.

In our example, this path would be entered as “EIP,2,192.168.16.163”.

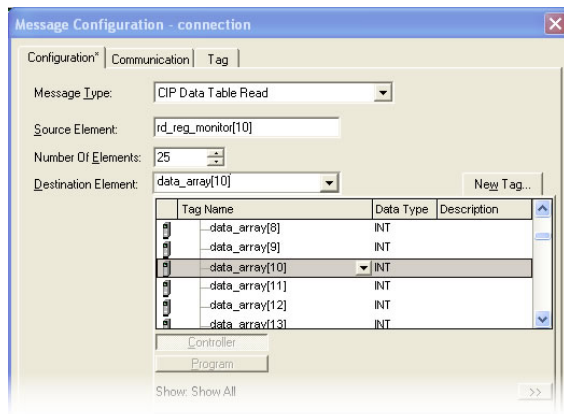


Figure 98: Selecting the Destination Element

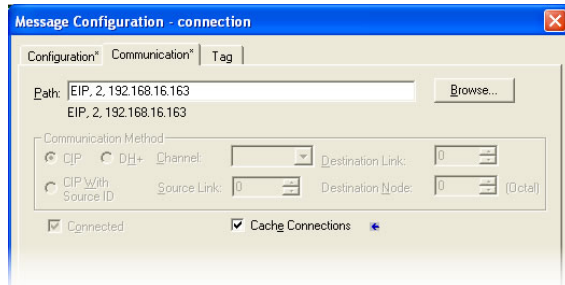
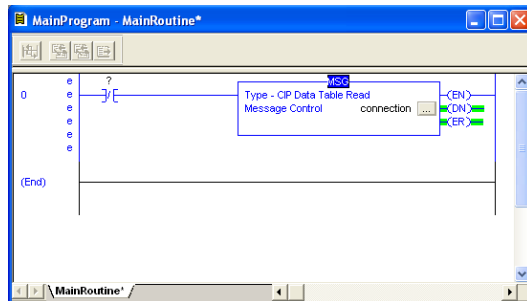


Figure 99: Setting the Communication Path

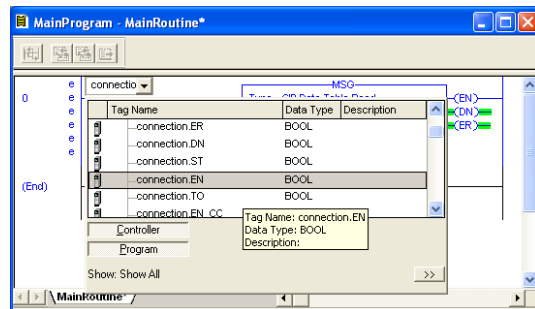
- ii) If “Cache Connections” is enabled (checked), the connection remains open after transmission. If disabled (unchecked), the connection is opened before and closed after every transmission. For efficiency, it is recommended to enable “Cache Connections”.
- e) Click “OK” to close the MSG Configuration dialog. At this stage, MainRoutine should look like Figure 100.



**Figure 100: MainRoutine**

**5) Assign a tag to the XIO element.**

- a) Double-click on the XIO element located to the left of the MSG block. In the drop-down box, double-click on the “connection.EN” field. Refer to Figure 101. This configuration causes the MSG instruction to automatically retrigger itself when it completes. While this is acceptable for the purposes of this example, it can produce high network utilization. In actual practice, it may be desirable to incorporate additional logic elements to allow triggering the MSG



**Figure 101: Configure XIO Element**

instruction at a specific rate or under specific conditions.

6) The program is now complete. Refer to Figure 102.

7) Save, download and run the program.

- a) To view the values of the registers being read from the interface card, double-click “Controller Tags” in the controller organizer view.
- b) Select the “Monitor Tags” tab.
- c) Expand the data\_array tag. Refer to Figure 103.
- d) 25 register values starting at register #211 are being continuously read from the interface card and placed in the 25 sequential offsets of data\_array starting at the 11<sup>th</sup> offset (data\_array[10]). In Figure 103, we can see that data\_array[11] (register 212 / converter output voltage peak value) has a value of 3190 (319.0V), data\_array[15] (register 216 / output terminal status) has a value of 19, etc.

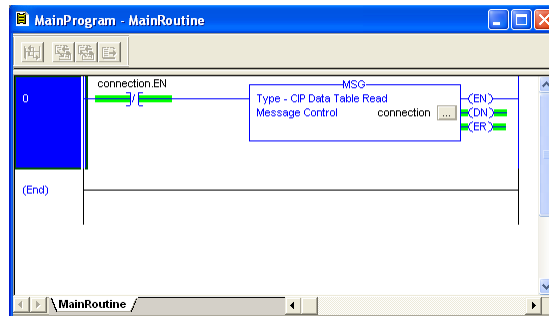


Figure 102: Complete Program

Controller Tags - EIP\_example(controller)

Scope: EIP\_example Show... Show All

Name	Value	Force Mask	Style	Data Type	Description
+ data_array[9]	0		Decimal	INT	
+ data_array[10]	0		Decimal	INT	
+ data_array[11]	3190		Decimal	INT	
+ data_array[12]	0		Decimal	INT	
+ data_array[13]	0		Decimal	INT	
+ data_array[14]	0		Decimal	INT	
+ data_array[15]	19		Decimal	INT	
+ data_array[16]	0		Decimal	INT	
+ data_array[17]	0		Decimal	INT	
+ data_array[18]	0		Decimal	INT	
+ data_array[19]	4375		Decimal	INT	
+ data_array[20]	0		Decimal	INT	
+ data_array[21]	0		Decimal	INT	
+ data_array[22]	1266		Decimal	INT	
+ data_array[23]	0		Decimal	INT	
+ data_array[24]	0		Decimal	INT	

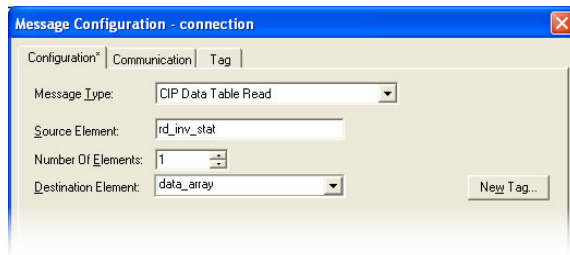
Monitor Tags / Edit Tags

**Figure 103: Viewing the Register Values**



## 8.2.9 ControlLogix Example: Read a Single Register

The configuration and execution for reading a single register is in general identical to that required for reading a block of registers as detailed in section 8.2.8. The only difference is in the configuration of the MSG instruction. Figure 104 shows an example MSG instruction's Configuration tab, which will read a single tag (rd\_inv\_stat / rd\_reg\_261, the inverter's status register) and place it in the first element (offset 0) of data\_array.



**Figure 104: Read the Inverter's Status Register**

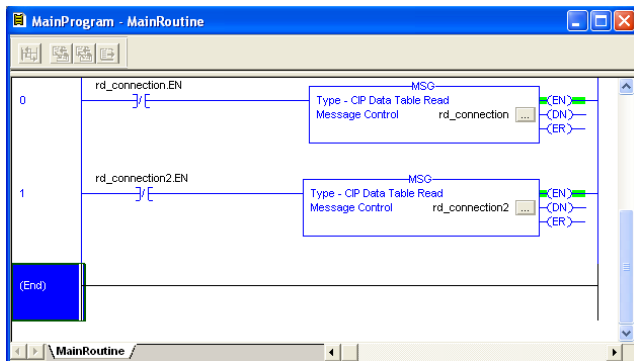
## 8.2.10 ControlLogix Example: Multiple MSG Instructions

At times, reading from different groups of registers may be necessary. For example, a specific application may require access to the universal access registers and some other monitor registers. To accomplish this task, multiple MSG instructions will need to be implemented in the PLC program. This example can be adapted for the MicroLogix and PLC5E PLCs.

The configuration and execution for implementing multiple MSG instructions is in general identical to that required for implementing just one MSG instruction. Each MSG instruction will require its own message controller tag. In the case of read MSG instructions, more than one instruction may use the same Destination Element tag, but the storage locations must not overlap. Figure 105 shows an example of two MSG instructions, each accessing different read tags. It is evident from this logic that



“rd\_connection” and “rd\_connection2” are the two independent message controller tags created for these instructions.

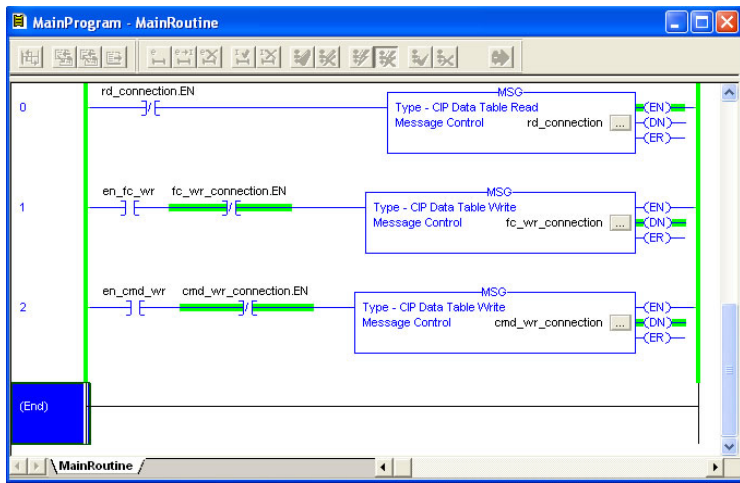


**Figure 105: Reading Via Multiple MSG Instructions**



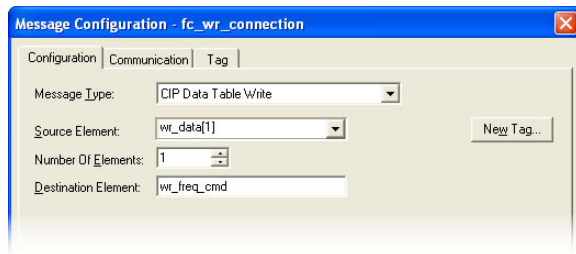
## 8.2.11 ControlLogix Example: Reading and Writing

Often times, applications may need to both read data from and write data to the inverter. At a minimum, this will require two MSG instructions and two message controller tags. Figure 106 shows an example of three MSG instructions, one for reading and two for writing (the inverter's frequency command and command word). The only item of note that differentiates this example from the multiple-read example in section 8.2.10 is the addition of the en\_xx\_wr XIC elements. The reason for the addition of these elements is that while reading from a remote device is often continuously performed (monitoring), data is typically written to the remote device only when necessary (i.e. when the value to write has changed). This conserves both network bandwidth and potentially EEPROM lifespans on the target device. The en\_xx\_wr elements in this example, therefore, would typically be replaced in an actual application program by user-provided logic that controls the conditions under which write operations would be performed.



**Figure 106: Reading and Writing via MSG Instructions**

Figure 107 shows the configuration details of the example `fc_wr_connection` MSG instruction. Note that the chosen “Message Type” is “CIP Data Table Write”, and that this instruction will only be writing to one inverter register: namely, the frequency command (Destination Element is `wr_freq_cmd` / `wr_reg_14`). The Source Element in this case is the 2<sup>nd</sup> element (starting from index 0) of an INT array tag named “`wr_data`”.



**Figure 107: MSG Configuration for Writing**

Note that when writing data via explicit messaging, use caution to ensure that the commanded registers are not also simultaneously being commanded in the background via I/O messaging. Indeterminate behavior can occur if MSG instructions and background I/O data transfers are both writing to the same registers. In other words, if the I/O messaging example procedure detailed in section 8.2.4 has already been implemented, and the same program is now being modified to implement explicit messaging, then it is recommended to inhibit the target module by selecting the “Inhibit Module” checkbox in the Connection tab of the Module Properties dialog. This example can be adapted for the MicroLogix and PLC5E PLCs.



## **8.3 Allen Bradley CSP**

---

### **8.3.1 Overview**

Ethernet-enabled Allen-Bradley legacy PLCs (such as the PLC5E and SLC-5/05 series) use a protocol called CSP (Client Server Protocol) to communicate over the Ethernet network. The flavor of CSP used by these PLCs is also known as “PCCC” (Programmable Controller Communication Commands) and “AB Ethernet”. The interface card supports CSP for direct connectivity to these PLCs.

If a connection timeout or socket-level error occurs, the driver will trigger a timeout event as described in section 5.7.5.

### **8.3.2 Tag Reference**

Register contents are read from and written to the interface card via CSP by reference to an integer “file/section number” and an “offset/element” within that file. Reading is performed via the CSP “PLC5 Read” (DF1 protocol typed read) service, and writing is performed via the CSP “PLC5 Write” (DF1 protocol typed write) service.

The formula to calculate which register is targeted in the interface card is provided in Equation 3.

$$\text{target register} = (\text{file number} - 10) \times 100 + \text{offset} \qquad \text{Equation 3}$$

In Equation 3, “target register”  $\in [1 \dots 1899]$ , “file number”  $\in [10 \dots 28]$  (which means N10...N28), and “offset” is restricted only by the limitations of the programming software (but is a value of 1899 max).

Table 11 provides some examples of various combinations of file/section numbers and offsets/elements which can be used to access inverter registers. Note that there are multiple different combinations of file/section numbers and offsets/elements that will result in the same inverter register being accessed.

**Table 11: CSP Target Register Examples**

File/Section Number	Offset/Element	Start Target Register
N10	2	2
N12	62	262
N11	162	262
N15	1	501
N15	8	508
N10	508	508
N10	1899	1899
N28	99	1899

In addition to providing access to the inverter registers in their “standard” numerical locations as mentioned above, the registers can also be accessed in a special “assembly object” type format by targeting integer file N50. What this means is that when N50 is targeted for reading, what is actually returned by the interface card is the user-defined register data as ordered by the EtherNet/IP produced register configuration array (refer to section 5.8.4). Similarly, when N50 is targeted for writing, the written data is disseminated to the inverter’s registers according to the definition contained in the EtherNet/IP consumed register configuration array. By appropriate configuration of the EtherNet/IP consumed and produced register configuration arrays, therefore, bulk access to non-contiguous but



frequently-used inverter registers can be conveniently provided by performing only one read and/or write instruction targeting file N50.

Because both the EtherNet/IP consumed and produced register configuration arrays are comprised of 32 register definitions, the targeted “offset/element” must be within the range of 0 to 31 inclusive. Refer to Table 12 for some examples of N50 accesses.

**Table 12: Examples of EtherNet/IP-Style Bulk Access via File N50**

<b>Offset/Element</b>	<b>Start Target Register of Configuration Array</b>	<b>Max Number of Accessible Elements</b>
0	1st	32
:	:	:
15	16th	16
:	:	:
31	32nd	1

The application PLC program uses a MSG instruction that is configured with a “Data Table Address” from which to start the access and a “Size in Elements” which determines the number of items to access (read or write). The “Data Table Address” is constructed by selecting a “File/Section Number” and an “Offset/Element” according to Equation 3. For example, a “File/Section Number” of N12 and “Offset/Element” of 61 = N12:61, which corresponds to register 261 (the inverter’s status register).

### 8.3.3 SLC-5/05 Example: Read a Register Block

This example program will show how to continuously read a block of registers from the inverter with a single MSG instruction. Only one read request is outstanding at any given time.

1) Run RSLogix 500, and create a new configuration.

2) Create a control and a data file.

- a) Right click Data Files and select New... The “Create Data File” dialog box appears (refer to Figure 108).
- b) To create a control file, enter a file number (e.g. 20), set the type to “Integer”, enter a descriptive name (e.g. “CONTROL”), and enter a number of elements (e.g. 100). Click OK to create the file. The control file is used to store configuration information pertaining to the functionality of the MSG instruction which will perform the data read.
- c) Follow the same procedure to create a data file. This file will be used to store the incoming data read from the interface card. Enter a file number (e.g. 18), set the type to “Integer”, enter a descriptive name (e.g. “DATA”), and enter a

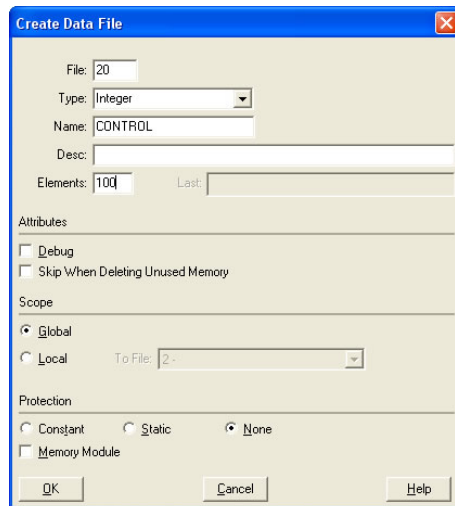


Figure 108: Creating a Control File



number of elements (e.g. 200). Refer to Figure 109. Click OK to create the file.

### 3) Add a MSG instruction to the program.

- If not already visible, double-click “LAD2” under Project...Program Files in the controller organizer view to bring up the ladder logic program.
- Right click on the default rung number on the left-hand side of the LAD2 window and select “Insert Rung”.
- Right click on the rung number of the new editable rung and select “Append Instruction”.
- Select the “MSG” instruction from the “Input/Output” classification, then click OK. Refer to Figure 110.

### 4) Add an XIO element to the program.

- Right click on the rung number of the rung currently being edited and select “Append Instruction” again.
- Select the “XIO” instruction from the “Bit” classification, then click OK. Refer to Figure 111.

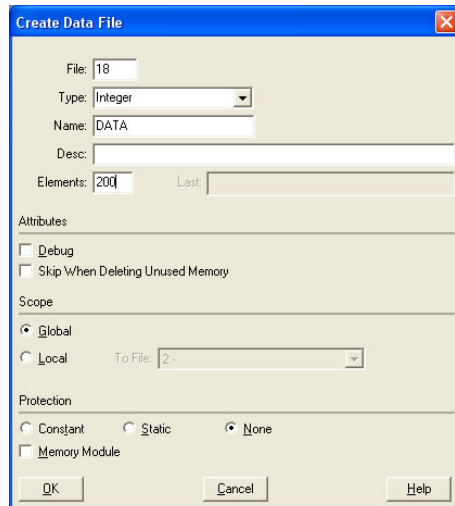
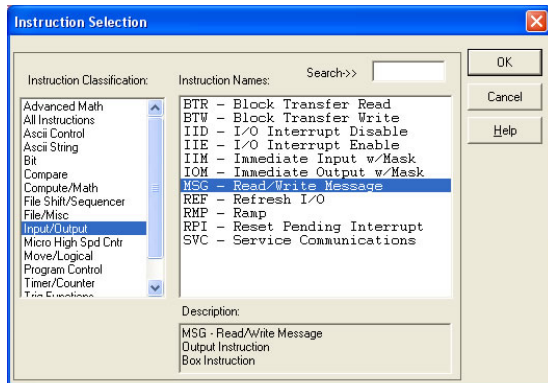
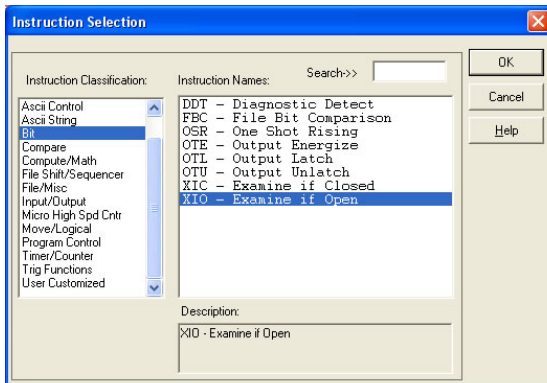


Figure 109: Creating a Data File





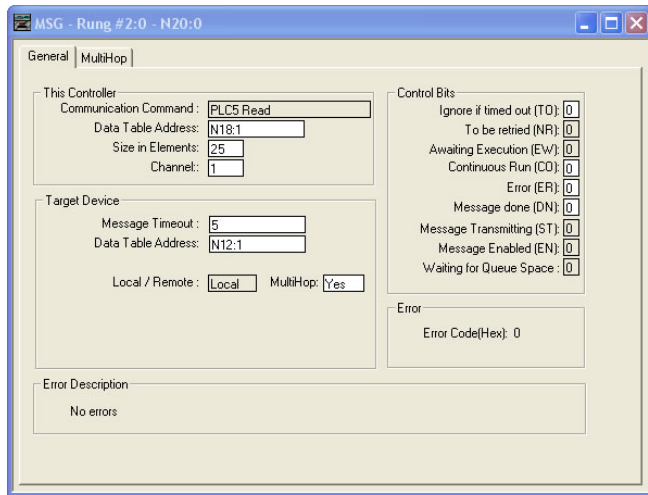
**Figure 110: MSG Instruction Selection**



**Figure 111: XIO Instruction Selection**

## 5) Configure the MSG instruction.

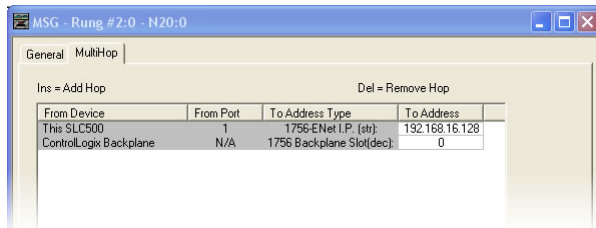
- Set the “Read/Write” field to “Read”, “Target Device” field to “PLC5”, “Local/Remote” field to “Local”, and “Control Block” to “N20:0”.
- Upon hitting the <ENTER> key while in the “Control Block” entry box, the MSG Properties dialog box should appear (or it can be opened by clicking on the “Setup Screen” button at the bottom of the MSG instruction). Refer to Figure 112.



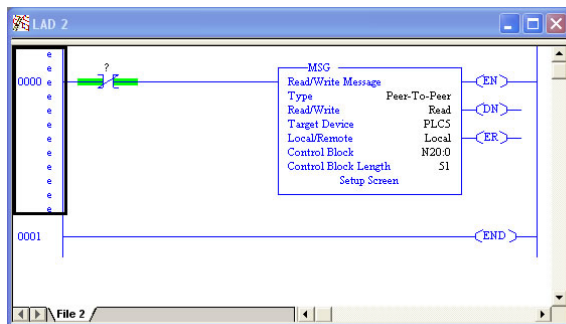
**Figure 112: MSG Configuration, "General" Tab**

- c) In this example, we will be reading a total of 25 registers beginning at N12:1 (register 201, the inverter's "output frequency" register). To configure this, under "This Controller" set the "Data Table Address" field to N18:1, set the "Size in Elements field" to 25, and set the "Channel" field to 1 (Ethernet).

- d) Under “Target Device”, set the “Data Table Address” field to N12:1 (starting target register=201) and set the “MultiHop” field to Yes to cause the “MultiHop” tab to appear.
  - e) Under the “MultiHop” tab settings, set the “To Address” in the first row to the inverter’s IP address, and the “To Address” in the second row to 0. Refer to Figure 113.
  - f) Close the dialog box. At this point, the program should appear as shown in Figure 114.
- 6) Assign a tag to the XIO element.**
- a) Double-click on the XIO element located to the left of the MSG block. Type in N20:0/15 (MSG instruction’s enable bit). This configuration causes the MSG instruction to automatically retrigger itself when it completes. While this is acceptable for the purposes of this example, it can produce high network



**Figure 113: MSG Configuration, “MultiHop” Tab**



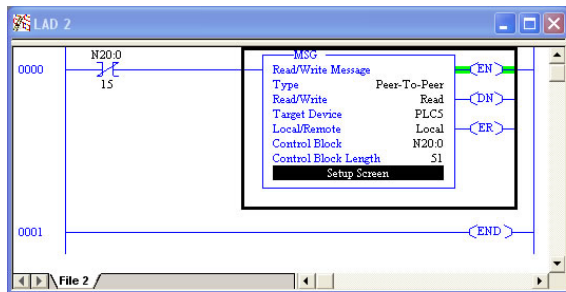
**Figure 114: PLC Program after MSG Instruction Configuration**

utilization. In actual practice, it may be desirable to incorporate additional logic elements to allow triggering the MSG instruction at a specific rate or under specific conditions.

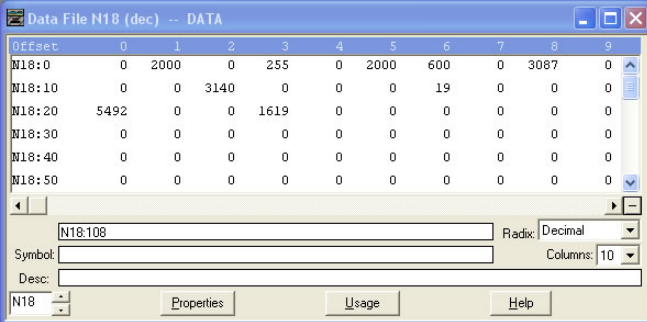
7) **The program is now complete. Refer to Figure 115.**

8) **Save, download, and run the program.**

- a) To view the registers being read from the interface card, double-click the data file N18 under “Data Files” in the controller organizer view. 25 register values starting at register #201 are being continuously read from the interface card and placed in the 25 sequential offsets of N18 starting at N18:1. Refer to Figure 116. We can see that N18:1 (output frequency) has a value of 2000 (20.00Hz), N18:3 (output voltage) has a value of 255 (25.5V), etc.



**Figure 115: Completed PLC Program**



Offset	0	1	2	3	4	5	6	7	8	9
N18:0	0	2000	0	255	0	2000	600	0	3087	0
N18:10	0	0	3140	0	0	0	19	0	0	0
N18:20	5492	0	0	1619	0	0	0	0	0	0
N18:30	0	0	0	0	0	0	0	0	0	0
N18:40	0	0	0	0	0	0	0	0	0	0
N18:50	0	0	0	0	0	0	0	0	0	0

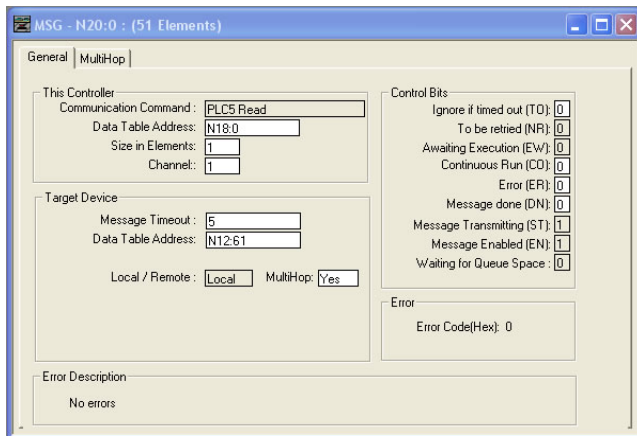
Symbol:  Radix:   
 Columns:   
 Desc:

**Figure 116: Monitoring the Data Being Read from the Inverter**



### 8.3.4 SLC-5/05 Example: Read a Single Register

The configuration and execution for reading a single register is in general identical to that required for reading a block of registers as detailed in section 8.3.3. The only difference is in the configuration of the MSG instruction. Figure 117 shows an example MSG instruction's General tab, which will read a single element (N12:61, which corresponds to the inverter's status register) and place it in the first element (offset 0) of N18.



**Figure 117: Read the Inverter's Status Register**

### 8.3.5 SLC-5/05 Example: Multiple MSG Instructions

At times, reading from different groups of registers may be necessary. For example, a specific application may require some registers located in various disjoint locations in the register map. To accomplish this task efficiently, multiple MSG instructions can be implemented in the PLC program.

The configuration and execution for implementing multiple MSG instructions is in general identical to that required for implementing just one MSG instruction. Each MSG instruction will require its own message control file. In the case of read MSG instructions, more than one instruction may use the same data file to store the received register values, but the storage locations must not overlap. Figure 118 shows an example of two MSG instructions, each accessing different target integer files. It is evident from this logic that N20 and N21 are the two independent message control files created for these instructions.

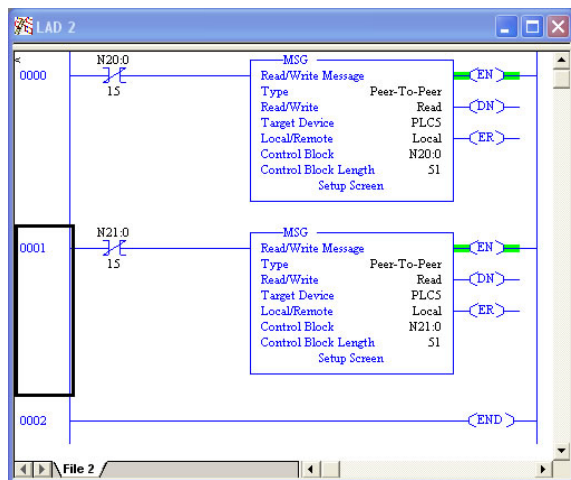


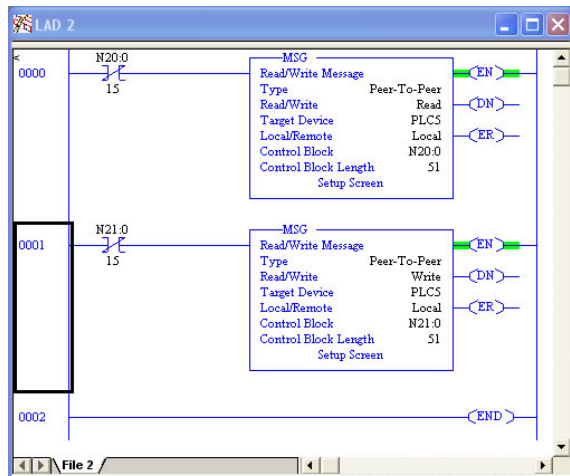
Figure 118: Multiple MSG Instructions



### 8.3.6 SLC-5/05 Example: Reading and Writing

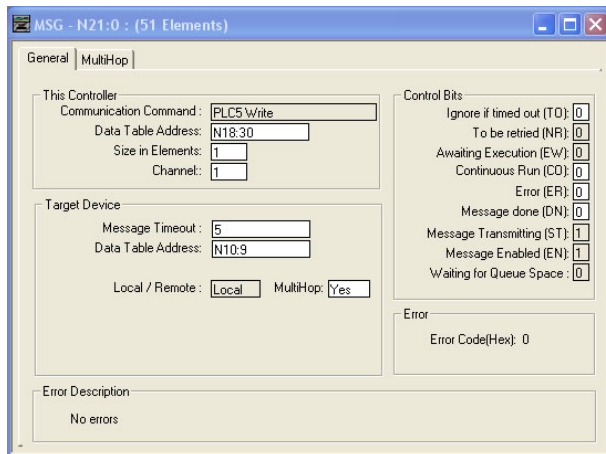
Often times, applications may need to both read data from and write data to the inverter. At a minimum, this will require two MSG instructions and two message control files. Figure 119 shows an example of two MSG instructions, one for reading and one for writing. Note that the “Read/Write” field of each of the MSG instructions is set according to their function.

Figure 120 shows the configuration details of the “write” MSG instruction. Note that this instruction will only be writing to one inverter register: namely, the command word (Target Data Table Address is N10:9, which equates to inverter register 9). The source Data Table Address in this case is N18:30.



**Figure 119: Reading and Writing via MSG Instructions**





**Figure 120: MSG Configuration for Writing**



## **8.4 BACnet/IP**

---

- The interface card supports the BACnet/IP (Annex J) protocol over Ethernet via UDP port 47808.
- The BACnet driver does not trigger timeout events (section 5.7.5).

### **8.4.1 Protocol Implementation Conformance Statement**

#### **BACnet Protocol**

Date:	July 21, 2015
Vendor Name:	ICC, Inc.
Product Name:	Mitsubishi Inverter FR-700
Product Model Number:	FR-A7N-ETH
Applications Software Version:	V2.333
Firmware Revision:	V2.333
BACnet Protocol Revision:	2
Product Description:	

The Mitsubishi 700-series inverter family represents inverters featuring reduced high-frequency noise, reduced harmonics, and high-precision and high-speed torque control with or without sensors.



**BACnet Standard Device Profile (Annex L):**

- BACnet Operator Workstation (B-OWS)
- BACnet Building Controller (B-BC)
- BACnet Advanced Application Controller (B-AAC)
- BACnet Application Specific Controller (B-ASC)
- BACnet Smart Sensor (B-SS)
- BACnet Smart Actuator (B-SA)

**BACnet Interoperability Building Blocks Supported (Annex K):**

- Data Sharing – ReadProperty-B (DS-RP-B)
- Data Sharing – ReadPropertyMultiple-B (DS-RPM-B)
- Data Sharing – WriteProperty-B (DS-WP-B)
- Device Management – Dynamic Device Binding-B (DM-DDB-B)
- Device Management – Dynamic Object Binding-B (DM-DOB-B)

**Segmentation Capability:**

None

- |                                                        |                   |
|--------------------------------------------------------|-------------------|
| <input type="checkbox"/> Segmented requests supported  | Window Size _____ |
| <input type="checkbox"/> Segmented responses supported | Window Size _____ |

**Standard Object Types Supported:**

See "Object Types/Property Support Table".

**Data Link Layer Options:**

- BACnet IP, (Annex J)
- BACnet IP, (Annex J), Foreign Device
- ISO 8802-3, Ethernet (Clause 7)
- ANSI/ATA 878.1, 2.5 Mb. ARCNET (Clause 8)
- ANSI/ATA 878.1, RS-485 ARCNET (Clause 8), baud rate(s) \_\_\_\_\_
- MS/TP master (Clause 9), baud rate(s): 9600, 19200, 38400, 76800
- MS/TP slave (Clause 9), baud rate(s): \_\_\_\_\_
- Point-To-Point, EIA 232 (Clause 10), baud rate(s): \_\_\_\_\_
- Point-To-Point, modem, (Clause 10), baud rate(s): \_\_\_\_\_
- LonTalk, (Clause 11), medium: \_\_\_\_\_
- Other: \_\_\_\_\_

**Device Address Binding:**

Is static device binding supported? (This is currently for two-way communication with MS/TP slaves and certain other devices.)     Yes     No

**Networking Options:**

- Router, Clause 6 - List all routing configurations
- Annex H, BACnet Tunneling Router over IP
- BACnet/IP Broadcast Management Device (BBMD)
  - Does the BBMD support registrations by Foreign Devices?  Yes  No

**Character Sets Supported:**

Indicating support for multiple character sets does not imply that they can all be supported simultaneously.

- ANSI X3.4
- IBM™/Microsoft™ DBCS
- ISO 8859-1
- ISO 10646 (UCS-2)
- ISO 10646 (UCS-4)
- JIS C 6226

If this product is a communication gateway, describe the types of non-BACnet equipment/networks(s) that the gateway supports: N/A

**Datatypes Supported:**

The following table summarizes the datatypes that are accepted (in the case of a write property service) and returned (in the case of a read property service) when targeting the present value property of each supported object type.



Object Type	Service	
	Read Property	Write Property
<b>Analog Output</b>	Real	Real, Unsigned, Integer, Null
<b>Analog Input</b>	Real	N/A
<b>Binary Output</b>	Enumerated	Enumerated, Boolean, Real, Unsigned, Integer, Null
<b>Binary Input</b>	Enumerated	N/A

Notes:

- The Null data type is used to relinquish a previously-commanded entry at the targeted priority in the priority array.
- When writing to Binary Output objects, all non-zero values are interpreted as a “1”.

**Object Types/Property Support Table**
**Table 13: BACnet Object Types /Properties Supported**

Property	Object Type				
	Device	Binary Input	Binary Output	Analog Input	Analog Output
<b>Object Identifier</b>	R	R	R	R	R
<b>Object Name</b>	R	R	R	R	R
<b>Object Type</b>	R	R	R	R	R
<b>System Status</b>	R				
<b>Vendor Name</b>	R				
<b>Vendor Identifier</b>	R				
<b>Model Name</b>	R				
<b>Firmware Revision</b>	R				
<b>Appl Software Revision</b>	R				
<b>Protocol Version</b>	R				
<b>Protocol Revision</b>	R				
<b>Services Supported</b>	R				
<b>Object Types Supported</b>	R				
<b>Object List</b>	R				
<b>Max APDU Length</b>	R				
<b>Segmentation Support</b>	R				



Property	Object Type				
	Device	Binary Input	Binary Output	Analog Input	Analog Output
<b>APDU Timeout</b>	R				
<b>Number APDU Retries</b>	R				
<b>Max Master</b>					
<b>Max Info Frames</b>					
<b>Device Address Binding</b>	R				
<b>Database Revision</b>	R				
<b>Present Value</b>		R	W	R	W
<b>Status Flags</b>		R	R	R	R
<b>Event State</b>		R	R	R	R
<b>Reliability</b>		R	R	R	R
<b>Out-of-Service</b>		R	R	R	R
<b>Units</b>				R	R
<b>Priority Array</b>			R		R
<b>Relinquish Default</b>			R		R
<b>Polarity</b>		R	R		
<b>Active Text</b>		R	R		
<b>Inactive Text</b>		R	R		

R – readable using BACnet services

W – readable and writable using BACnet services



## 8.4.2 Supported Objects

Table 14: Binary Input Object Instance Summary

Instance ID	Object Name	Description	Active/ Inactive Text
BI1	RUN_STOP_STATUS	Run/stop status	running/stopped
BI2	FOR_ROT_STATUS	Forward rotation status	forward/off
BI3	REV_ROT_STATUS	Reverse rotation status	reverse/off
BI4	SU_STATUS	Up to frequency	on/off
BI5	OVERLOAD_STATUS	Overload status	on/off
BI6	IPF_STATUS	Instantaneous power failure	on/off
BI7	FU_STATUS	Frequency detection	on/off
BI8	ABC1_STATUS	Terminal function selection status	on/off
BI9	ABC2_STATUS	Terminal function selection status	on/off
BI10	ALARM_OCC_STATUS	Alarm occurrence status	on/off


**Table 15: Binary Output Object Instance Summary**

<b>Instance ID</b>	<b>Object Name</b>	<b>Description</b>	<b>Active/ Inactive Text</b>
BO1	STOP_CMD	Stop command	stop/no action
BO2	FOR_ROT_CMD	Forward rotation command	forward/off
BO3	REV_ROT_CMD	Reverse rotation command	reverse/off
BO4	RH_CMD	High speed operation command	on/off
BO5	RM_CMD	Medium speed operation command	on/off
BO6	RL_CMD	Low speed cooperation command	on/off
BO7	JOG_OP	Jog	on/off
BO8	RT_SEL	Second function selection	on/off
BO9	AU_SEL	Current input selection	on/off
BO10	CS_SEL	Auto restart selection after IPF	on/off
BO11	MRS	Output stop	on/off
BO12	STOP_HOLDING	Start self-holding	on/off
BO13	RESET	Reset	on/off

**Table 16: Analog Input Object Instance Summary**

Instance ID	Object Name	Description	Units
AI1	OUTPUT_FREQ	Output frequency	Hz
AI2	OUTPUT_CURRENT	Output current	Amps
AI3	OUTPUT_VOLTAGE	Output voltage	Voltage
AI4	INPUT_POWER	Input power	kW
AI5	OUTPUT_POWER	Output power	kW
AI6	CUMULATIVE_POWER	Energy consumption	kWh
AI7	OP_MODE_STATUS	Op mode status	None
AI8	UAR_STATUS	Universal access register status	None
AI9	PID_SET_POINT	PID set point	%
AI10	PID_MEASURED_VALUE	PID measured value	%
AI11	PID_DEVIATION_VALUE	PID deviation value	%


**Table 17: Analog Output Object Instance Summary**

Instance ID	Object Name	Description	Units
AO1	INVERTER_RESET	Reset	None
AO2	PARAM_CLEAR1	Clear parameters	None
AO3	ALL_PARAM_CLEAR1	Clear all parameters	None
AO4	PARAM_CLEAR2	Clear parameters except communication parameters	None
AO5	ALL_PARAM_CLEAR2	Clear all parameters except communication parameters	None
AO6	OPERATION_MODE	Operation mode register	None
AO7	FREQ_CMD_REG	Frequency command register	Hz
AO8	FREQ_CMD_EEPROM_REG	EEPROM Frequency command register	Hz
AO9	UAR_ADDRESS	Universal address register address	None
AO10	UAR_VALUE	Universal address register value	None

### 8.4.3 Supported Object Details

#### Binary Input Objects

- B11 .....Indicates the status of the RUN output terminal function configured by *Pr. 190 RUN terminal function selection*. Corresponds to register 261, bit 0.
- B12 .....Indicates whether the inverter is running forward, or stopped. Corresponds to register 261, bit 1.
- B13 .....Indicates whether the inverter is running reverse, or stopped. Corresponds to register 261, bit 2.
- B14 .....Indicates the status of the SU output terminal function configured by *Pr. 191 SU terminal function selection*. Corresponds to register 261, bit 3.
- B15 .....Indicates the status of the OL output terminal function configured by *Pr. 193 OL terminal function selection*. Corresponds to register 261, bit 4.
- B16 .....Indicates the status of the IPF output terminal function configured by *Pr. 192 IPF terminal function selection*. Corresponds to register 261, bit 5.
- B17 .....Indicates the status of the FU output terminal function configured by *Pr. 194 FU terminal function selection*. Corresponds to register 261, bit 6.
- B18 .....Indicates the status of the ABC1 output terminal function configured by *Pr. 195 ABC1 terminal function selection*. Corresponds to register 261, bit 7.
- B19 .....Indicates the status of the ABC2 output terminal function configured by *Pr. 196 ABC2 terminal function selection*. Corresponds to register 261, bit 8.



BI10 .....Indicates whether or not the inverter is in an alarm condition. Corresponds to register 261, bit 15.

### **Binary Output Objects**

BO1.....Stop command. Corresponds to register 9, bit 0.

BO2.....Forward command. Corresponds to register 9, bit 1.

BO3.....Reverse command. Corresponds to register 9, bit 2.

BO4.....Activates the RH input terminal function configured by *Pr. 182 RH terminal function selection*.  
Corresponds to register 9, bit 3.

BO5.....Activates the RM input terminal function configured by *Pr. 181 RM terminal function selection*.  
Corresponds to register 9, bit 4.

BO6.....Activates the RL input terminal function configured by *Pr. 180 RL terminal function selection*.  
Corresponds to register 9, bit 5.

BO7.....Activates the JOG input terminal function configured by *Pr. 185 JOG terminal function selection*.  
Corresponds to register 9, bit 6.

BO8.....Activates the RT input terminal function configured by *Pr. 183 RT terminal function selection*.  
Corresponds to register 9, bit 7.

BO9.....Activates the AU input terminal function configured by *Pr. 184 AU terminal function selection*.  
Corresponds to register 9, bit 8.

- BO10....Activates the CS input terminal function configured by *Pr. 186 CS terminal function selection*.  
Corresponds to register 9, bit 9.
- BO11....Activates the MRS input terminal function configured by *Pr. 187 MRS terminal function selection*.  
Corresponds to register 9, bit 10.
- BO12....Activates the STOP input terminal function configured by *Pr. 188 STOP terminal function selection*.  
Corresponds to register 9, bit 11.
- BO13....Activates the RES input terminal function configured by *Pr. 189 RES terminal function selection*.  
Corresponds to register 9, bit 12.

### **Analog Input Objects**

- AI1 ..... The output frequency of the inverter in 0.01 Hertz units (6000=60.00Hz). Corresponds to register 201.
- AI2 ..... The output current of the inverter in 0.1 or 0.01 Amp units (depends on inverter capacity).  
Corresponds to register 202.
- AI3 ..... The output voltage of the inverter in 0.1 Volt units (1000=100.0V). Corresponds to register 203.
- AI4 ..... Input power of the inverter in 0.1 or 0.01 kW units (depends on inverter capacity). Corresponds to register 213.
- AI5 ..... Output power of the inverter in 0.1 or 0.01 kW units (depends on inverter capacity).  
Corresponds to register 214.



- AI6 .....Energy consumption in kWh. Corresponds to register 225.
- AI7 .....Inverter operation mode status. Corresponds to register 262.
- AI8 .....Universal Access Register Status (UARS). Corresponds to register 102.
- AI9 .....PID set point monitor in 0.1% units. Corresponds to register 252.
- AI10 .....PID measurement value monitor in 0.1% units. Corresponds to register 253.
- AI11 .....PID deviation value monitor in 0.1% units. Corresponds to register 254.

### **Analog Output Objects**

- AO1.....Inverter fault reset command. Corresponds to register 2.
- AO2.....Parameter clear command. Corresponds to register 3.
- AO3.....All parameter clear command. Corresponds to register 4.
- AO4.....Parameter clear command (communication parameters are not cleared). Corresponds to register 6.
- AO5.....All parameter clear command (communication parameters are not cleared). Corresponds to register 7.
- AO6.....Selects the operation mode of the inverter. Corresponds to register 10.
- AO7.....The frequency command (RAM) of the inverter in 0.01 Hertz units. Corresponds to register 14.





AO8.....The frequency command (EEPROM) of the inverter in 0.01 Hertz units. Corresponds to register 15.

AO9.....Universal Access Register Address (UARA). Corresponds to register 100.

AO10....Universal Access Register Value (UARV). Corresponds to register 101.



## **8.5 MELSEC**

---

### **8.5.1 Overview**

The Mitsubishi MELSEC communication protocol (also known as MC protocol) is an application-level protocol implemented on top of the Ethernet TCP/IP and UDP/IP layers. It is typically used to read and write data from/to devices supporting compatible 3E and 1E frames.

The interface card acts as a server and supports Device Memory Batch Read (0x0401), Batch Write (0x1401), Random Read (0x0403), and Random Write (0x1402) commands for word access for 3E frames. For 1E frames, both Device Memory Batch Read Word (0x01) and Batch Write Word (0x03) commands are supported. All commands only support word units. All MELSEC communication is conducted over only TCP/IP.

The only supported device type is the File Register. The 3E hex code for the File Register device type is 0xB0 and the 1E hex code is 0x5A52. The device point is the register number (refer to 6). If applicable, the network number is 0, the PC number is 0xFF, the module I/O number is 0x3FF, and the module station number is 0. For more information regarding the MELSEC protocol, refer to the Mitsubishi MELSEC Communication Protocol reference manual.

A MELSEC client device is required to initiate communications with the interface card. If a Mitsubishi PLC is commissioned as the MELSEC client device, please consult Mitsubishi Electric for GX Developer and/or GX Works programs implementing the MELSEC client driver. For additional information regarding the MELSEC protocol, refer to the Mitsubishi MELSEC Communication Protocol reference manual.

Other notes of interest are:

- Supports up to 8 simultaneous connections.



- The TCP port is user-configurable.
- The max MELSEC frame size is 1460 bytes. Refer to Table 18.
- The 3E Random Write command will attempt to write to all requested device points even if an error is encountered. Ensure that all requested device points are valid before using Random Write.
- Both scanned and non-scanned registers can be freely accessed. The response time for scanned register accesses will be faster than that for non-scanned registers.
- The socket timeout time is determined by the “timeout” setting on the web server’s “Config” tab (refer to section 5.7.5). This means that if a particular open socket experiences no activity for more than the timeout time setting, then the interface assumes that the client or network has experienced some sort of unexpected problem, and will close that socket.
- Because the socket timeout determination is performed on a per-socket basis, note that a certain degree of caution must be exercised when using the network timeout feature to avoid “nuisance” timeouts from occurring. Specifically, do not perform inadvisable behavior such as sending a request from the master device to the interface, and then closing the socket prior to successfully receiving the unit’s response. The reason for this is because the interface will then experience an error when attempting to respond via the now-closed socket, which will immediately trigger the timeout action. Always be sure to manage socket life cycles “gracefully”, and do not abandon outstanding requests.
- If a socket timeout occurs (regardless of whether it was due to a communication lapse or abnormal socket error), the driver will trigger a timeout event as described in section 5.7.5.

**Table 18: MELSEC Commands**

<b>Frame Type</b>	<b>Command Name</b>	<b>Command Code</b>	<b>Subcommand Code</b>	<b>Max Points</b>
3E	Batch Read	0x0401	0x0000	724
3E	Batch Write	0x1401	0x0000	719
3E	Random Read	0x0403	0x0000	192
3E	Random Write	0x1402	0x0000	192
1E	Batch Read	0x01	NA	256
1E	Batch Write	0x03	NA	256

## 8.6 PROFINET IO

---

### 8.6.1 Overview

The interface card supports two different types of PROFINET data transfer. One type is entirely user-configurable, and is utilized when a standard I/O module is chosen during network configuration. The other type is a module that is compliant with the PROFIdrive profile.

Other notes of interest are:

- Allows access to only 1 PROFINET controller.
- Supports conformance class A and real time (RT) communication.
- Supports DCP (Discovery Control Protocol).
- Supports alarms.
- Supports I&M.
- A total of 85 modules are available for selection by the controller (84 standard I/O modules and one PROFIdrive module). The standard I/O modules follow the naming conventions of “IN:XX, OUT:YY”, where XX is the number of input (status) registers to be sent to the controller, and YY is the number of output (command) registers to be sent to the inverter. The single PROFIdrive profile-compliant module is named “Standard Telegram 1”. Refer to the GSDML file for specific module information.
- When using standard I/O modules, up to 8 command registers can be sent to the inverter, and up to 32 status registers can be retrieved from the inverter.



- No explicit module selection is required on the interface card: the module will be selected automatically according to the controller's configuration.
- The lowest recommended I/O Cycle Update Time (in Step 7 or an equivalent hardware configuration tool) is 8ms. Lower update times are possible but determinism may vary depending on the interface's processing load. Note that this setting is not related to, and should therefore not be confused with, the Update Time setting on the embedded web server's PROFINET tab.
- If the RT connection timeout occurs, the driver will trigger a timeout event as described in section 5.7.5. The timeout value is dictated by the PROFINET controller and is at least three times the IO Cycle update time. The timeout value is also known as the "IO Cycle Watchdog" time.
- The interface card supports PROFIdrive profile V4.1. For more information on PROFIdrive communications, refer to the *PROFIdrive Profile Drive Technology Technical Specification for PROFIBUS and PROFINET* (PROFIBUS International order #3.172.)
- Any available inverter parameter can be accessed via PROFINET acyclic services. Set the API to 0, Slot to 1, and SubSlot to 1. The record index (unless reserved) directly corresponds to a register number (section 6).
- Standard I/O module configuration: if a register entry in the command data configuration array is 0, then any commanded data that corresponds to that location will be ignored. Conversely, if a register entry in the status data configuration array is 0, then any status data that corresponds to that location will be a default value of 0. Refer to section 5.5.3 for further information on the data configuration arrays.

## 8.6.2 PROFdrive Profile

For optimal interoperability, the interface card supports the PROFdrive profile version 4.1. No special configuration of the interface card is required when using the PROFdrive profile. The controller **must** support the PROFdrive profile and **must** be configured to use the “Standard Telegram 1” module on the interface card. If the controller does not support the PROFdrive profile, use the configurable I/O “IN: XX WORDS, OUT: YY WORDS” module. The PROFdrive profile is only partially described in this manual due to its complexity. The complete PROFdrive profile specifications can be obtained from <http://www.profibus.com/>.

- Implements Application Class 1 (standard drive)
- Supports only Standard Telegram 1 (ST1, PZD-2/2) on slot 1 (similar to PROFIBUS PPO type 3)
- Supports only Speed Control Mode

### 8.6.2.1 PROFdrive standard telegram 1

The standard telegram 1 mapping is described in Table 19.

**Table 19: Standard Telegram 1**

IO Data Word Offset	Setpoint (PLC to Inverter)		Actual Value (Inverter to PLC)	
	Significance	Description	Significance	Description
0	STW1	Control word 1	ZSW1	Status word 1
1	NSOLL_A	Reference speed setpoint	NIST_A	Speed actual



### 8.6.2.2 *PROFdrive control and status words*

The control word, STW1, is the principal means for controlling the drive. It is sent by the controller (PLC) to the device (inverter). The bitmapping for the control word is described in Table 20. The status word, ZSW1, returns status information from the inverter to the controller. The bitmapping for the status word is described in Table 21.

**Table 20: STW1 Control Word Mapping**

Bit	Value	Significance	Description
0	1	ON	Run command ON
	0	OFF	Run command OFF
1	1	ON2	No coast stop
	0	OFF2	Coast to a stop
2	1	ON3	No quick stop
	0	OFF3	Quick stop
3	1	Enable Operation	Enable inverter operation
	0	Disable Operation	Disable inverter operation
4	1	Enable Ramp Generator	Enable the ramp frequency generator (RFG)
	0	Disable Ramp Generator	Hold the output frequency to 0 Hz
5	1	Unfreeze Ramp Generator	Unfreeze the RFG



Bit	Value	Significance	Description
	0	Freeze Ramp Generator	Freeze the RFG with the current output frequency
6	1	Enable Setpoint	Enable command
	0	Disable Setpoint	Disable command
7	1	Fault Acknowledge	Reset the alarm on a positive edge (0->1 transition)
	0	No significance	Do not reset the alarm
8 - 9	Not used	---	---
10	1	Control By PLC	Enable remote control. The IO process data is valid.
	0	No Control By PLC	Disable remote control. The IO process data is not valid.
11 - 15	Not used	---	---

**Table 21: ZSW1 Status Word Mapping**

Bit	Value	Significance	Description
0	1	Ready To Switch ON	Ready to run command ON
	0	Not Ready To Switch ON	Not ready to run command ON
1	1	Ready to Operate	Ready to run



Bit	Value	Significance	Description
	0	Not Ready To Operate	Not ready to run
2	1	Operation Enabled	Running
	0	Operation Disabled	Running disabled
3	1	Fault Present	Inverter tripped as indicated by register 261 bit 15
	0	No Fault	No trip present as indicated by register 261 bit 15
4	1	Coast Stop Not Activated	Follows STW1 bit 1, ON2 active
	0	Coast Stop Activated	Follows STW1 bit 1, OFF2 active
5	1	Quick Stop Not Activated	Follows STW1 bit 2, ON3 active
	0	Quick Stop Activated	Follows STW1 bit 2, OFF3 active
6	1	Switch ON Inhibited	Not ready to run command ON
	0	Switch ON Not Inhibited	Ready to run command ON
7	Not Used	---	---
8	1	Speed Within Tolerance	Actual value equals the reference value and is within the tolerance as indicated by SU. Refer to register 261 bit 3.
	0	Speed Out Of Tolerance	Actual value differs from the reference value or is outside of the tolerance as indicated by SU. Refer to register 261 bit 3.

Bit	Value	Significance	Description
9	1	Control Requested	Control by PLC is possible when the operation mode status (register 262) is reporting network operation.
	0	No Control Requested	Control is not possible by the controller when the operation mode status (register 262) is not reporting network operation mode. The inverter must be explicitly set to network operation mode.
10	1	Frequency Reached Or Exceeded	The actual value $\geq$ max reference value. Refer to <i>Pr. 1</i> (maximum frequency).
	0	Frequency Not Reached	The actual value $<$ max reference value. Refer to <i>Pr. 1</i> (maximum frequency).
11 - 15	Not used	---	---

### 8.6.2.3 **PROFIdrive reference speed setpoint and actual speed**

The speed setpoint value, NSOLL\_A, is the commanded speed reference (normalized) sent from the controller to the inverter. Similarly, the speed actual value, NIST\_A, is the actual operating speed (normalized) of the inverter sent back to the controller. As the inverter natively operates in units of Hz or RPM depending on certain combinations of *Pr. 37* and *Pr. 144*, the interface card must apply conversion equations to and from the normalized values. Refer to Table 22 to determine the appropriate conversion equations. For more information on the manners in which *Pr. 37* and *Pr. 144* affect the inverter's operation, please refer to the appropriate inverter user's manual.



**Table 22: Inverter Speed Setpoint and Speed Actual Conversion Scenarios**

<i>Pr. 37</i>	<i>Pr. 144</i>	Inverter Unit	Number of Motor Poles	NSOLL_A Conversion	NIST_A Conversion
0 (default value)	0, 2..10	Hz	N/A	Equation 5	Equation 6
	102..110	RPM	2..10	Equation 7	Equation 8
1..9998	0, 102..110	Hz	N/A	Equation 5	Equation 6
	2..10	RPM	2..10	Equation 7	Equation 8

$$\text{Max RPM} = \frac{\text{Max Frequency} \times 120}{\text{Number of Motor Poles}} \quad \text{Equation 4}$$

$$\text{Hz} = \frac{\text{NSOLL\_A} \times \text{Max Frequency}}{0x4000} \quad \text{Equation 5}$$

$$\text{NIST\_A} = \frac{\text{Hz} \times 0x4000}{\text{Max Frequency}} \quad \text{Equation 6}$$

$$\text{RPM} = \frac{\text{NSOLL\_A} \times \text{Max RPM}}{0x4000} \quad \text{Equation 7}$$

$$\text{NIST\_A} = \frac{\text{RPM} \times 0x4000}{\text{Max RPM}} \quad \text{Equation 8}$$

Note: “Max Frequency” equates to the value of *Pr. 1* (register 1001, maximum frequency).

NSOLL\_A: The reference speed setpoint is a normalized value. The interface card applies the Normalize-to-Hz or Normalize-to-RPM conversion indicated by Equation 5 or Equation 7 respectively. The resulting value is written to inverter register 14 (frequency command).

NIST\_A: The inverter operating actual speed is a normalized value that is calculated from inverter register 201 (output frequency). The interface card applies the Hz-to-Normalize or RPM-to-Normalize conversion indicated by Equation 6 or Equation 8 respectively.

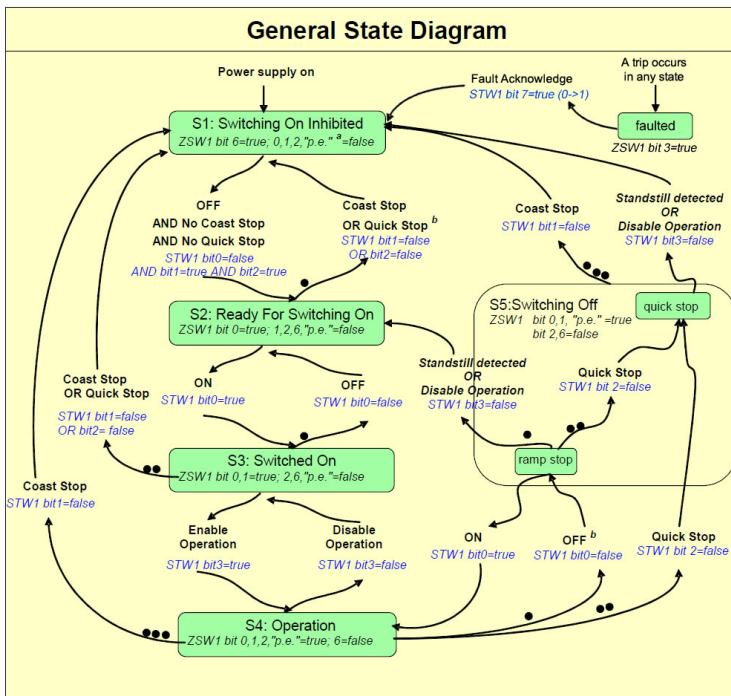
A normalized value of 0x4000 corresponds to 100% of the maximum frequency. A positive normalized value indicates forward rotation and a negative normalized value indicates reverse rotation.

Note that the values of *Pr. 1*, *Pr. 37*, and *Pr. 144* are read by the interface card only at boot-up. If the values of these function codes are changed, then the interface card must be rebooted in order for it to read the new values and once again determine the required conversion scenarios.

#### **8.6.2.4      *PROFIdrive state diagram***

The state diagram is displayed in Figure 121.

Figure 121: PROFIdrive State Diagram



### 8.6.2.5 PROFdrive-Specific Parameters

The PROFdrive-specific parameters are shown in Table 23. The parameters are read-only.

**Table 23: PROFdrive Specific Parameters**

PNU	Index	Description
711	None	NSOLL_A – Speed setpoint A
712	None	NIST_A – Speed actual A
833	None	STW1 – Control word 1
834	None	ZSW1 – Status word 1
922	None	Telegram selection = 1 (Standard telegram 1)
923	1,2,5,6	List of all parameters for signals
944	None	Fault message counter
947	0..7	Fault number (Alarm history registers 501..508)
964	0..6	Drive Unit identification
965	None	Profile identification number = Profile 3, Version 4.1
975	0..7	DO identification
980	0..5	Number list of defined parameter
1401	None	DO IO Data reference parameter



## 9 TROUBLESHOOTING

Although by no means exhaustive, the following table provides possible causes behind some of the most common errors experienced when using the FR-A7N-ETH interface.

Problem	Symptom	Solution
No communications between FR-A7N-ETH and the inverter	The FR-A7N-ETH's "INVERTER" TX and RX LEDs are blinking slowly, sporadically, or not at all	<ul style="list-style-type: none"> <li>• Check the inverter's RS-485 Communication Settings (refer to section 3.1)</li> <li>• Check the connections and orientation of the RS-485 cable between the FR-A7N-ETH and the inverter. For A700 &amp; F700 inverters, refer to section 2.2. For E700 inverters, refer to section 2.3.</li> </ul>
Firmware-generated error	"MODULE STATUS" LED is flashing red. The number of times the LED flashes indicates an error code.	<ul style="list-style-type: none"> <li>• If the blinking error code is 7-1-1, refer to the solution for "No communications between FR-A7N-ETH and the inverter".</li> <li>• For all other error codes, please contact ICC for further assistance.</li> </ul>
ICC Finder program cannot discover the FR-A7N-ETH	The ICC Finder displays "No Devices Found" or the expected FR-A7N-ETH does not appear in the Device list	<ul style="list-style-type: none"> <li>• Confirm that the FR-A7N-ETH is running normally. Refer to the solution for "No communications between FR-A7N-ETH and the inverter".</li> <li>• Add the ICC Finder as an exception to the computer's firewall</li> <li>• Temporarily disable the computer's firewall</li> </ul>



Problem	Symptom	Solution
<p>No communications between the network and the FR-A7N-ETH</p>	<p>Communications cannot be established, the Ethernet “link” LED is off, or the Ethernet “activity” LED flashes only infrequently or not at all</p>	<ul style="list-style-type: none"> <li>• Confirm that the FR-A7N-ETH is running normally. Refer to the solution for “No communications between FR-A7N-ETH and the inverter”.</li> <li>• Ensure that the appropriate (straight-through or cross-over) Ethernet cable is used. Try a known working cable and switch.</li> <li>• Confirm that the destination IP address programmed into the controller equipment or computer matches that of the interface card, as displayed by the finder utility.</li> <li>• Confirm that intermediate firewalls or routers have been configured to allow access to the interface via the applicable TCP/UDP ports.</li> <li>• If attempting to access the web server on a computer whose web browser is configured to use a proxy server, ensure that the proxy server is accessible to the computer, and that the interface card is accessible to the proxy server</li> </ul>
<p>XML socket connection failed</p>	<p>Message on web server tab information window</p>	<p>TCP port 843 is blocked by a firewall, router or some other intermediate network equipment</p>
<p>Unable to control the inverter via network communications</p>	<p>Cannot write to command parameters via network communications, or writing to these parameters has no apparent effect</p>	<p>Set the inverter to NET mode. The inverter will reject all command and parameter write requests from the network if it is not in NET mode. Refer to section 3.2.</p>

## REVISIONS

Date	Manual Number	Details
September 2007	10653-1.000-000	Initial release
April 2008	10653-2.000-000	Updates for V2.000 firmware release
July 2008	10653-2.100-000	Updates for V2.100 firmware release
August 2008	10653-2.110-000	Updates for V2.110 firmware release
February 2009	10653-2.200-000	Updates for V2.200 firmware release
April 2009	10653-2.210-000	Updates for V2.210 firmware release
October 2010	10653-2.300-000	Updates for V2.300 firmware release
May 2011	10653-2.310-000	Updates for V2.310 multiprotocol firmware release
August 2011	10653-2.320-000	Updates for V2.320 firmware and add-on instructions
April 2013	10653-2.320-001	Add E700 installation section, updates to LED descriptions and TROUBLESHOOTING sections
July 2015	10653-2.333-000	Updates for V2.333 firmware release



## **INDUSTRIAL CONTROL COMMUNICATIONS, INC.**

1600 Aspen Commons, Suite 210  
Middleton, WI USA 53562-4720  
Tel: [608] 831-1255 Fax: [608] 831-2045

<http://www.iccdesigns.com>

Printed in U.S.A