



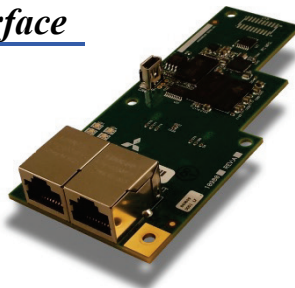
INVERTER

Plug-in option

A8NETH-2P

INSTRUCTION MANUAL

*Ethernet multiprotocol
communication interface*



PRE-OPERATION INSTRUCTIONS	1
INSTALLATION	2
INVERTER SETTINGS	3
REGISTER NUMBERING AND BEHAVIOR	4
MITSUBISHI CONFIGURATION STUDIO	5
EMBEDDED WEB SERVER	6
FILESYSTEM	7
FIRMWARE	8

Thank you for choosing this Mitsubishi inverter plug-in option. 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.

Safety instructions

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".




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  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.

◆ Electrical Shock Prevention

Warning

- 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.

◆ 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.

◆ Additional Instructions

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



Caution

Transportation and mounting

- 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.
- If halogen-based materials (fluorine, chlorine, bromine, iodine, etc.) infiltrate into a Mitsubishi product, the product will be damaged. Halogen-based materials are often included in fumigants which are used to sterilize or disinfest wooden packaging. When packaging, prevent residual fumigant components from being infiltrated into Mitsubishi products, or use an alternative sterilization or disinfection method for packaging, such as heat disinfection, etc. Whenever possible, sterilization or disinfection of wooden packaging should be performed prior to packaging the product.

Trial run

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



Warning

Usage

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



Caution

Usage

- 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.

Maintenance, inspection and parts replacement

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

Disposal

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

General instructions

- For clarity purposes, illustrations in this manual may be drawn with covers or safety guards removed. Never operate the inverter in this manner. Ensure all covers and safety guards are properly installed prior to starting operation.

– CONTENTS –

1	PRE-OPERATION INSTRUCTIONS	11
1.1	Product Overview	11
1.2	Features and Specifications	12
1.3	Unpacking and Product Confirmation.....	22
1.3.1	Shipment Confirmation.....	22
1.3.2	Component Overview	23
1.4	LED Indicators	24
1.4.1	Port Status LED Description.....	24
1.4.2	Standard LED Description.....	25
1.4.3	EtherCAT LED Description.....	26
2	INSTALLATION	27
2.1	Pre-Installation Instructions	27
2.2	Installation Procedure	28
3	INVERTER SETTINGS	32
3.1	Network Setting	33
3.1.1	IP Address (Pr. 1305 ~ Pr. 1308)	33
3.1.2	Subnet Mask (Pr. 1309 ~ Pr. 1312).....	33
3.1.3	Default Gateway Address (Pr. 1313 ~ Pr. 1316).....	33
3.2	Operation Mode Setting	34
3.2.1	Operation Mode Indication	34
3.2.2	Operation Mode Switching & Comm. Startup Mode (Pr. 79, Pr. 340).....	35
3.3	Operation & Speed Command Source (Pr. 338, Pr. 339, Pr. 550)	38

3.4	Communication EEPROM Write Selection (<i>Pr. 342</i>)	39
4	REGISTER NUMBERING AND BEHAVIOR	40
<hr/>		
4.1	Register Numbers	40
4.2	Scanned Registers	46
4.3	Inverter Command Register	47
4.4	Inverter Reset Register	48
4.5	Alarm History Clear Register	48
4.6	All Parameter Clear Register	48
4.7	Inverter State Register	49
4.8	DriveStatus Register	50
4.9	SetpointSpeed Register	51
4.10	ActualSpeed Register	51
4.11	Fault History Codes	51
5	MITSUBISHI CONFIGURATION STUDIO	52
<hr/>		
5.1	Overview	52
5.2	General Object Editing Activities	56
5.3	Ethernet Settings	58
5.3.1	Authentication	58
5.3.2	Network Configuration	58
5.4	Internal Logic Settings	59
5.4.1	Fail-safe Values	59
5.4.1.1	Overview	59
5.4.2	Timeout Time	59
5.4.2.1	Timeout Object Configuration	59
5.5	Discovery over Ethernet	61

5.6	Manage Device Parameters	62
5.7	Backup and Restore Parameters	63
5.8	Restore Factory Settings	65
5.9	Help	65

6 EMBEDDED WEB SERVER 66

6.1	Overview	66
6.2	Monitor Tab	68
6.2.1	Information Window.....	68
6.2.2	Parameter Group Selection List	68
6.2.3	Parameter List	68
6.2.4	Parameter List Filter	70
6.2.5	Radix Selection.....	70
6.3	Dashboard Tab.....	71
6.3.1	Information Window.....	72
6.3.2	Gauge Window Navigation.....	72
6.3.3	Gauge Window Configuration	72
6.3.4	Submitting Changes	77
6.4	Customizing the Embedded Web Server	78
6.4.1	Customization Overview.....	78
6.4.2	XTPro Overview	78
6.4.3	XTPro Web Browser-Based Implementation	79
6.4.4	XTPro HMI-Based Implementation	81
6.4.5	XTPro Supported Commands	82

7 FILESYSTEM 83

7.1	Overview	83
-----	----------------	----

7.2	USB with Windows Explorer	84
7.3	FTP With Windows Explorer.....	86
7.4	Loading New Web Server Content.....	87

8 FIRMWARE 88

8.1	Overview	88
8.2	Update Procedure.....	88

9 PROTOCOL-SPECIFIC INFORMATION 90

9.1	Modbus/TCP.....	90
9.1.1	Overview.....	90
9.1.2	Holding & Input Registers.....	91
9.1.3	Coil & Discrete Input Mappings.....	91
9.1.4	Connection Timeout Options.....	92
9.1.5	Node Settings.....	93
9.1.6	Holding/Input Register Remap Settings.....	93
9.2	EtherNet/IP	95
9.2.1	Overview.....	95
9.2.2	Server Settings.....	96
9.2.3	Connection Timeout Options.....	97
9.2.4	Generic Class 1 I/O Produced and Consumed Data Settings.....	98
9.2.5	Generic Class 1 (I/O) Connection Access.....	99
9.2.6	AC/DC Drive Profile Class 1 (I/O) Connection Access.....	100
9.2.7	Explicit Messaging Via Get/Set Attribute Single Services.....	103
9.2.8	Explicit Messaging Via Data Table Read/Write Services.....	103
9.2.9	Inverter Register Access Tag Format.....	104
9.2.10	ControlLogix Examples: Setup.....	105

9.2.11	ControlLogix Example: EDS Add-On Profile (AOP)	107
9.2.11.1	<i>ControlLogix Example: EDS Add-On Profile (AOP) Generic I/O Messaging</i>	112
9.2.11.2	<i>ControlLogix Example: EDS Add-On Profile (AOP) AC/DC Drive Profile</i>	114
9.2.12	ControlLogix Example: I/O Messaging	117
9.2.12.1	<i>ControlLogix Example: Generic Default I/O Add-On Instruction</i>	120
9.2.12.2	<i>ControlLogix Example: AC/DC Drive Profile Add-On Instruction</i>	123
9.2.13	ControlLogix Example: Read Registers	126
9.2.14	ControlLogix Example: Reading and Writing Multiple MSG Instructions	131
9.3	Allen Bradley CSP (PCCC)	133
9.3.1	Overview	133
9.3.2	Explicit Messaging Via Typed Read/Write Services	133
9.3.3	Inverter Register File Number Offset Format	134
9.3.4	SLC-5/05 Example: Read Registers	136
9.3.5	SLC-5/05 Example: Reading and Writing Multiple MSG Instructions	141
9.4	BACnet/IP	143
9.4.1	Protocol Implementation Conformance Statement (PICS)	143
9.4.2	Default Supported Objects	151
9.4.4	Server Settings	154
9.4.5	Node Settings	154
9.4.6	Device Object Settings	154
9.4.7	BACnet Object Settings	154
9.4.7.1	<i>Analog Input Object Settings</i>	155
9.4.7.2	<i>Analog Output Object Settings</i>	155
9.4.7.3	<i>Analog Value Object Settings</i>	156
9.4.7.4	<i>Binary Input Object Settings</i>	157
9.4.7.5	<i>Binary Output Object Settings</i>	158
9.4.7.6	<i>Binary Value Object Settings</i>	160
9.4.7.7	<i>Multi-state Input Object Settings</i>	161

9.4.7.8	<i>Multi-state Output Object Settings</i>	161
9.4.7.9	<i>Multi-state Value Object Settings</i>	162
9.5	MELSEC SERVER	163
9.5.1	Overview	163
9.5.2	Supported Commands	164
9.5.3	Server Settings	164
9.5.4	Connection Timeout Options	164
9.6	PROFINET IO	166
9.6.1	Overview	166
9.6.2	Device Settings	167
9.6.3	Connection Timeout Options	167
9.6.4	Cyclic I/O Produced and Consumed Data Access Settings	167
9.6.5	PROFIdrive Profile	169
9.6.5.1	<i>PROFIdrive Standard Telegram 1</i>	170
9.6.5.2	<i>PROFIdrive Control and Status Words</i>	170
9.6.5.3	<i>PROFIdrive Reference Speed Setpoint and Actual Speed</i>	173
9.6.5.4	<i>PROFIdrive-Specific Parameters</i>	174
9.6.5.5	<i>PROFIdrive State Diagram</i>	175
9.6.6	Acyclic Data Access	176
9.6.7	TIA Portal (STEP 7) Hardware Configuration Example	176
9.6.7.1	<i>Register the GSDML File</i>	176
9.6.7.2	<i>Add the Device to the Configuration</i>	178
9.6.7.3	<i>Select the IO Controller</i>	178
9.6.7.4	<i>Assign the I/O Module</i>	179
9.6.7.5	<i>Configure the Device Properties</i>	180
9.6.7.6	<i>Online Device Discovery and Configuration</i>	182
9.6.7.7	<i>Save the Configuration</i>	183
9.6.8	GE Proficy Configuration Example	184
9.6.8.1	<i>Register the GSDML File</i>	184

9.6.8.2	Add the Device to the Configuration	185
9.6.8.3	Configure the Device Properties	187
9.6.8.4	Save the Configuration.....	187
9.7	EtherCAT	188
9.7.1	Overview.....	188
9.7.2	Device Settings.....	188
9.7.3	Transmit and Receive Process Data Word Settings.....	188
10	TROUBLESHOOTING	191



PRE-OPERATION INSTRUCTIONS

1.1 Product Overview

The A8NETH-2P Ethernet multiprotocol communication interface allows information to be transferred seamlessly between an 800-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 two RJ-45 jacks with an embedded 10BASE-T/100BASE-TX Ethernet switch for connection to the Ethernet network. In addition to the supported fieldbus protocols, the interface also hosts a fully customizable embedded web server, which provides access to inverter information via a standard web browser for remote monitoring and control.

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 release date of the firmware version running on your interface as it must match this manual's respective release date in order for all documented aspects to apply.

The A8NETH-2P may be referred to throughout the remainder of the manual as the device, interface, card, and option (or any combination thereof.)

Supported Protocols

The interface currently provides support for the following fieldbus protocols:

- Modbus/TCP Server
- EtherNet/IP Server (DLR node)
- Allen Bradley CSP Server (also known as "PCCC" and "AB Ethernet")
- BACnet/IP Server
- Mitsubishi MELSEC Server (also known as "MC protocol")

- PROFINET IO Device (MRP client)
- EtherCAT Slave



- *Modbus is a registered trademark of Schneider Electric USA, Inc., licensed to the Modbus Organization, Inc.*
- *EtherNet/IP is a registered trademark of ODVA, Inc.*
- *BACnet is a registered trademark of ASHRAE*
- *MELSEC is a registered trademark of Mitsubishi Electric Corporation*
- *PROFINET is a registered trademark of PROFIBUS and PROFINET International (PI)*
- *EtherCAT is a registered trademark of Beckhoff Automation GmbH*

1.2 Features and Specifications

Table 1: Features

Item	Description
Simultaneous Protocols	Supports all standard unmodified Ethernet (SUE) protocols simultaneously
Mitsubishi Configuration Studio	Graphical user interface for discovery, configuration, and firmware update
WEB Server (HTTP)	Access all parameters, dashboard with gauges, customizable with XTPro
Communication Loss Detection	Configurable actions for "fail-safe" conditions
Field Upgradeable	Firmware updates automatically handled by the studio
Parameter Management	Advanced management of parameter access and scan priority
Parameter Backup and Restore	Drive cloning

Table 2: General Hardware Specifications

Item	Description
Power Supply	Directly powered by the inverter
Grounding	Referenced to inverter's 5V power supply / isolated from inverter control power common
LED Indicators	Module Status, Network Status, 2 x Ethernet Link/Activity
USB Port	USB 2.0, mini-B 5-pin

Table 3: Ethernet Hardware Specifications

Item	Description
Number of Ports	2 (internal switch)
Standard	IEEE 802.3 10BASE-T/100BASE-TX Ethernet compliant
Communication Speed and Duplex	10Mbps half/full, 100Mbps half/full (auto sense optimal speed and duplex)
Connector Type	RJ-45 Shielded
Auto MDI-X	Yes (supports all straight-through and cross-over cables)
Cable Type	CAT5-type 8-conductor UTP patch cables
Cable Length	100m per segment max
Topologies	Star/Tree, Linear/Bus/Daisy-chain, Ring (MRP / DLR)

Table 4: Modbus/TCP Specifications

Item	Description
Conformance Class	Class 0, Class 1 (partial), Class 2 (partial)
Read Function Codes	Read coils (1), Read input status (2), Read multiple registers (3), Read input registers (4), Diagnostics (8)
Write Function Codes	Write coil (5), Write single register (6), Force multiple coils (15), Write multiple registers (16)
Number of Connections	8
Max Read Register Size	125 registers
Max Write Register Size	123 registers
Register Data Type	16-bit integer
Unit (slave) ID	Ignored, echoed in response
TCP Port	502
Response Time	Min 160us, Typically less than 1ms

Table 5: EtherNet/IP Specifications

Item	Description
Conformance Tested	ODVA EtherNet/IP Conformance Test Software Version CT-13
Product Type Code	2 (AC Drive)
AC/DC Drive Profile	Yes
UCMM	Yes
Class 3 (Explicit) Messaging	Yes
Class 1 (Implicit I/O) Messaging	Yes
Class 1 Unicast T→O	Yes
Class 1 Multicast T→O	Yes
Number of Connections	16 (Total for both Class 1 and Class 3)
RPI	Min 1ms
I/O Input Size	Max 32 input words, user configurable
I/O Output Size	Max 32 output words, user configurable
Generic (User Configurable) Assembly Instances	100 (input) and 150 (output)
AC/DC Drive Profile Assembly Instances	20 (input) and 70 (output), 21 (input) and 71 (output)
Data Table Read/Write	Yes
DLR	Device Level Ring Node
Class 1 UDP Port	2222 (0x08AE)
Explicit Messaging Port	44818 (0xAF12)
Explicit Messaging Response Time	Min 160us, Typically less than 1ms

Table 6: Allen Bradley CSP (PCCC) Specifications

Item	Description
Read Services	PLC5 Read (DF1 protocol typed read, 0x68)
Write Services	PLC5 Write (DF1 protocol typed write, 0x67)
Data Type	16-bit Integer
File Type	N (Integer)
Logical ASCII Addressing	Yes
Logical Binary Addressing	Yes
Max Read Size	240 bytes (120 16-bit Integers)
Max Write Size	240 bytes (120 16-bit Integers)

Table 7: MELSEC (MC Protocol) Specifications

Item	Description
Frame Types	3E, 1E
Transport Types	TCP/IP only
3E Frame Read Function Codes	Device Memory Batch Read (0x0401, Word units), Device Memory Random Read (0x0403, Word units)
3E Frame Write Function Codes	Device Memory Batch Write (0x1401, Word units), Device Memory Random Write (0x1402, Word units)
1E Frame Read Function Codes	Device Memory Batch Read (0x01, Word units)
1E Frame Write Function Codes	Device Memory Batch Write (0x03, Word units)
Number of Connections	8
Max Read Points	724 points (varies with function code)
Max Write Points	719 points (varies with function code)
3E Device Types	File Register 0xB0
1E Device Types	File Register 0x5A52
PC Number	0xFF
Module I/O Number	0x03FF
Module Station Number	0
TCP Port	2009 (Configurable)
Response Time	Min 160us, Typically less than 1ms

Table 8: PROFINET Specifications

Item	Description
Protocol Level	RT (real-time)
RT Conformance Class	Class B
Netload Class	III
I/O Cycle Time	Min 1ms
I/O Input Size	Max 32 input words, user configurable
I/O Output Size	Max 32 output words, user configurable
MRP	Media Redundancy Protocol Client
DCP	Discovery, set station name, set IP address
LLDP	Yes
I&M	I&M0
Alarms	Plug, Pull
Number of Controllers	Allows access to only 1 controller

Table 9: BACnet/IP Specifications

Item	Description
BACnet IP	Annex J
Protocol Revision	2
Standard Device Profile (Annex L)	BACnet Application Specific Controller (B-ASC)
BACnet Interoperability Building Blocks (BIBB)	ReadProperty-B (DS-RP-B), ReadPropertyMultiple-B (DS-RPM-B), WriteProperty-B (DW-WP-B), Dynamic Device Binding-B (DM-DDB-B), Dynamic object Binding-B (DM-DOB-B)
Segmentation	Not supported
Max APDU Length	1444 bytes
Character Sets	ANSI X3.4
Object Types	Analog Output, Analog Input, Analog Value, Binary Output, Binary Input, Binary Value, Multi-state Output, Multi-state Input, Multi-state Value
Priority Array	Yes
UDP Port	47808 (0xBAC0, configurable)
Response Time	Min 160us, Typical less than 1ms

Table 10: EtherCAT Specifications

Item	Description
Conformance Tested	EtherCAT ET9400 Conformance Test Tool V1.20.80
Source Code	Beckhoff ET9300 EtherCAT Slave Source Code (SSC) V5.10
Slave Controller	Equivalent to Beckhoff ET1100
Process I/O Transmit Size	Max 32 transmit words, user configurable
Process I/O Receive Size	Max 32 receive words, user configurable
Operating Modes	FreeRun/SM-Synchron, DC-Synchron
FreeRun Cycle Time	Min 1ms
Distributed Clocks	SYNC0/1
Fieldbus Memory Management Unit (FMMU)	8 max
Sync Managers (SM)	8 max
Mailbox	CoE
CoE Complete Access Support	Yes
CoE SDO Info Support	Yes
Latency	700ns end-to-end

Table 11: 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 ² (0.6G) or less (10 ~ 55Hz)
Cooling Method	Self-cooled
RoHS (Lead free)	Yes

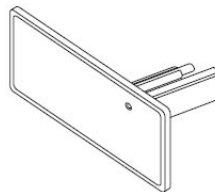
1.3 Unpacking and Product Confirmation

1.3.1 Shipment Confirmation

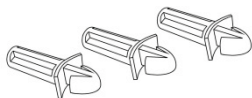
Check the enclosed items. Confirm that the correct quantity of each item was received, and that no damage occurred during shipment.



Plug-in option: qty. 1



Communication option LED display cover: qty. 1



5.5mm Spacer: qty. 3



M3 x 8mm Mounting screws: qty. 3

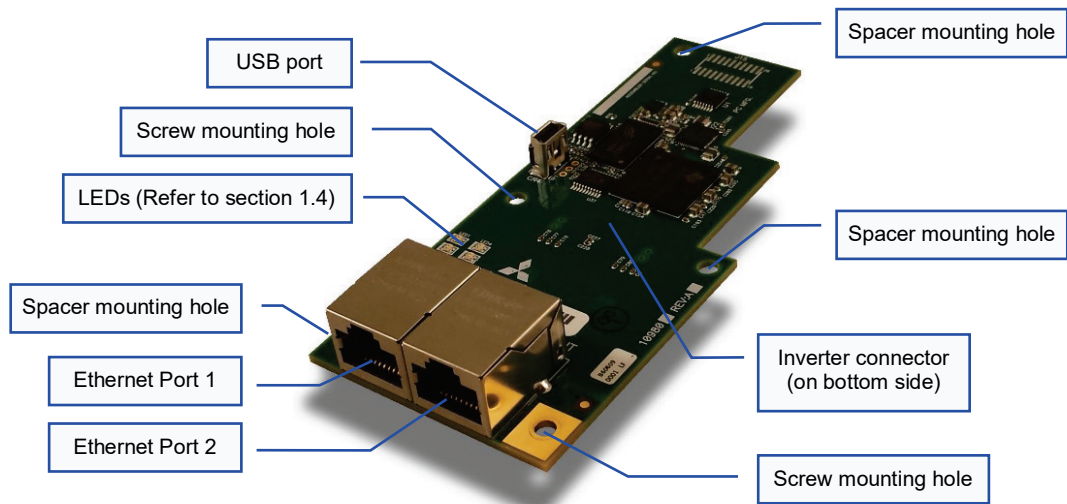


Grounding bar: qty. 1



USB interface cable: qty. 1

1.3.2 Component Overview



1.4 LED Indicators

The option board contains several bi-color LEDs (visible through the LED display cover after mounting) that provide a visual indication of the unit's overall status.



1.4.1 Port Status LED Description

Ethernet Port 1 (P1 LNK/ACT) and Ethernet Port 2 (P2 LNK/ACT)

LED Activity	Status	Note
Green On	Link	A valid Ethernet link exists: communication is possible on this port
Green Off	No Link	A valid Ethernet link does not exist: communication is not possible on this port
Red Blink	Activity	Indicates when a packet is transmitted or received on this port

1.4.2 Standard LED Description

Module Status (MS)

LED Activity	Status	Note
Off	Device Off	The inverter power is off
Green Blink, Red Blink	Startup	Startup blink sequence
Green On	Device On	Normal status
Green Blink	Discovery Identification	PROFINET discovery and identification (DCP)
Red Blink	Error Code	Refer to the TROUBLESHOOTING section

Network Status (NS)

LED Activity	Status	Note
Off	Device Off	The inverter power is off
Green Blink, Red Blink	Startup	Startup blink sequence
Green Blink	No Connection	EtherNet/IP connection is not established
Green Off	No Connection	PROFINET connection is not established
Green On	Connection Established	EtherNet/IP or PROFINET connection is established

1.4.3 EtherCAT LED Description

Run (LED3)

LED Activity	Slave State	Note
Off	Device Off or INITIALISATION	The inverter power is off or the device is in state INIT
Green Blink, Red Blink or Red Flickering	INITIALISATION	Startup blink sequence; the device is booting and has not yet entered the INIT state
Green Blinking	PRE-OPERATIONAL	The device is in state PRE-OPERATIONAL
Single Green Flash	SAFE-OPERATIONAL	The device is in state SAFE-OPERATIONAL
Green On	OPERATIONAL	The device is in state OPERATIONAL
Red Blink	Error Code	Refer to the TROUBLESHOOTING section

Error (LED4)

LED Activity	Status	Note
Off	Device Off or No Error	The inverter power is off or EtherCAT communication is normal
Green Blink, Red Blink	Startup	Startup blink sequence
Red On	Application Controller Failure	A critical communication or application controller error has occurred
Double Red Flash	Process Data Watchdog Timeout or EtherCAT Watchdog Timeout	An application watchdog timeout has occurred
Single Red Flash	Local Error	Due to local error, the slave device application has changed the EtherCAT state autonomously
Red Blink	Invalid Configuration	General configuration error

INSTALLATION

2.1 Pre-Installation Instructions

Check that the inverter's input power and the control circuit power are both OFF.

Caution

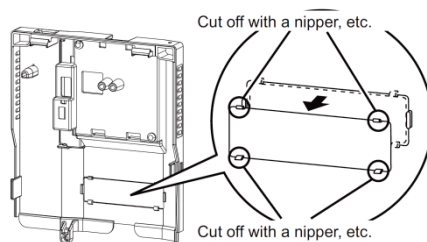
- To prevent explosions or similar damage, do not install or remove the plug-in option with input power ON.
- To prevent damage from electrostatic discharge, always touch a grounded piece of metal prior to touching any equipment.

2

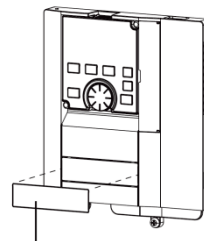
2.2 Installation Procedure

◆ Installing the communication option LED display cover

- 1) Remove the inverter's front cover (refer to chapter 2 of the inverter's *Instruction Manual (Detailed)* for information on how to remove the front cover.)
- 2) Cut off the tabs on the back side of the inverter's front cover with a nipper, etc. and open the knockout for the communication option LED display cover.



- 3) Fit the communication option LED display cover into the knockout on the front side of the inverter's front cover, orienting it with the LED positions on the plug-in option circuit board. Push the LED display cover firmly until the tabs lock into place.



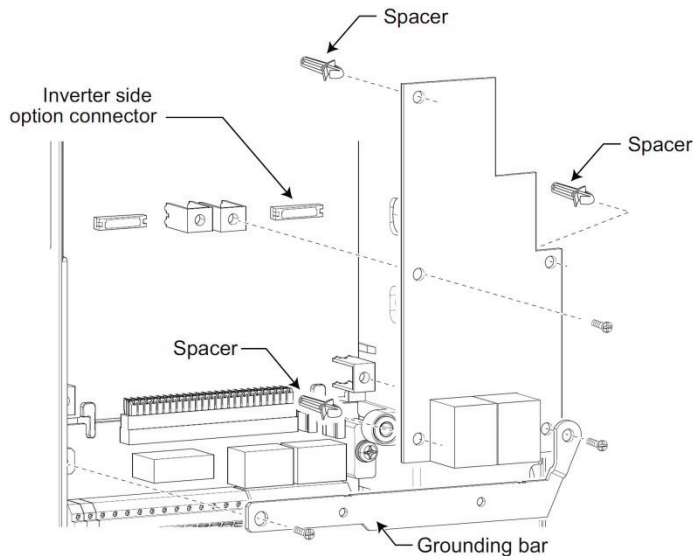
Communication option LED display cover

Caution

- To prevent hand injury, avoid contacting the tabs on the backside of the inverter's front cover.

◆ Installing the option

- 1) Insert spacers from the back side of the option card into the 3 mounting holes as indicated.
- 2) Securely attach the option card to the inverter's option Connector 1 (refer to Figure 1 on page 30 for the location of Connector 1 on the inverter's control board). Ensure that the option card is fully seated on the inverter's option connector.
- 3) Install the grounding bar as indicated, using a mounting screw to attach the left side of the grounding bar to the inverter's standoff, and another mounting screw to attach the right side of the grounding bar through the option card to the standoff below it.
- 4) Install the remaining mounting screw through the hole on the middle left side of the option card to the standoff below it.
- 5) Tighten all mounting screws to a torque setting of 0.33 N·m to 0.40 N·m.



◆ Completing the installation

- 1) Attach the network cable(s) to the plug-in option's Ethernet port(s) as required, and route the cable(s) out of the inverter along with other control-level wiring. Ensure that sufficient clearance exists between the communication cables and the inverter's input power and output motor wiring.
- 2) Reinstall the inverter's front cover (refer to chapter 2 of the inverter's *Instruction Manual (Detailed)* for additional information.)

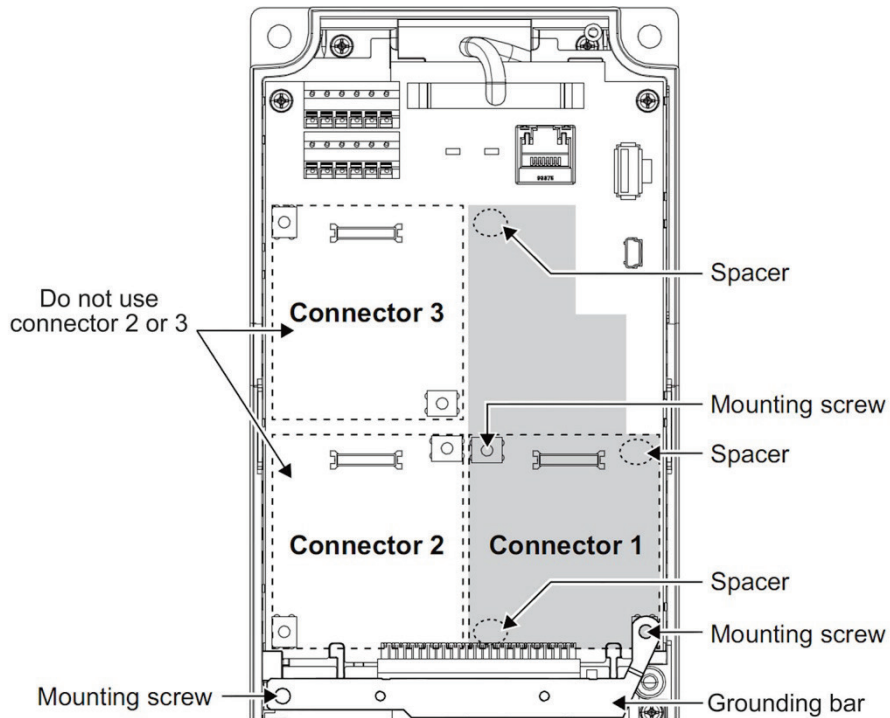


Figure 1: Installation Positions of Screws and Spacers

 **NOTE**

- To minimize the possibility of damage, hold only the sides of the circuit board when installing or uninstalling the plug-in option. Stress applied to circuit board components by directly pressing/pulling on them may cause component damage and subsequent failure.
- Use caution to avoid dropping mounting screws into the inverter enclosure when installing or uninstalling the plug-in option.
- To minimize the possibility of malfunction due to electrical noise, always install the included grounding bar.
- The plug-in option must only be installed on the inverter control board's option Connector 1 (refer to Figure 1 on page 30.) If installed onto the Connector 2 or Connector 3 positions, the inverter will fault with an E.2 or E.3 fault code.
- Even if the plug-in option is installed onto the correct Connector 1 position, the inverter may still fault with an E.1 fault code if the option card is not properly installed or fully seated onto the inverter control board's option connector.

Mounted position	Fault indication
Option Connector 1	E. 1
Option Connector 2	E. 2
Option Connector 3	E. 3

- When removing the plug-in option, first remove the three M3x8mm mounting screws and grounding bar. Lastly, remove the option card by grasping it on its left and right side and pulling it straight away from the inverter.

3

INVERTER SETTINGS

The inverter parameters listed in Table 12 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 12: Inverter Settings

Parameter Number	Name	Refer to Page
79	Operation mode selection	35
338	Communication operation command source	38
339	Communication speed command source	38
340	Communication startup mode selection	35
342	Communication EEPROM write selection	39
541	Frequency command sign selection	-
550	NET mode control source selection	38
1305	IP address most significant octet	33
1306	IP address high octet	33
1307	IP address low octet	33
1308	IP address least significant octet	33
1309	Subnet mask most significant octet	33
1310	Subnet mask address high octet	33
1311	Subnet mask address low octet	33
1312	Subnet mask address least significant octet	33
1313	Default gateway address most significant octet	33
1314	Default gateway address low octet	33
1315	Default gateway address high octet	33
1316	Default gateway address least significant octet	33

3.1 Network Setting

The network settings can **optionally** be set using the inverter parameters described in this section. However, to avoid invalid configuration, it is recommended to use the Configuration Studio to modify the network settings whenever possible. Please consult with your network administrator for the proper settings of these fields.

3.1.1 IP Address (Pr. 1305 ~ Pr. 1308)

The IP address is created from the octet values specified in *Pr. 1305 ~ Pr. 1308*. The byte order is defined according to the following example for an IP address of 192.168.16.100:

IP Address	192	168	16	100
Parameter Number	1305	1306	1307	1308

3.1.2 Subnet Mask (Pr. 1309 ~ Pr. 1312)

The subnet mask is created from the octet values specified in *Pr. 1309 ~ Pr. 1312*. The byte order is defined according to the following example for a subnet mask of 255.255.255.0:

Subnet Mask	255	255	255	0
Parameter Number	1309	1310	1311	1312

3.1.3 Default Gateway Address (Pr. 1313 ~ Pr. 1316)

The default gateway address is created from the octet values specified in *Pr. 1313 ~ Pr. 1316*. The default gateway address can be "disabled" by setting *Pr. 1313 ~ Pr. 1316* to a value of 0. The byte order is defined according to the following example for a default gateway address of 192.168.16.1:

Gateway Address	192	168	16	1
Parameter Number	1313	1314	1315	1316

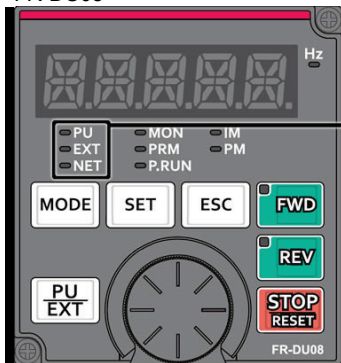
3.2 Operation Mode Setting

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-DU08).
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 38.)

3.2.1 Operation Mode Indication

FR-DU08



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)

◆ Operation mode switching conditions

Prior to switching the operation mode, confirm that:

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

◆ 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.




- When *Pr. 340* is changed, the new setting is validated after powering on or resetting the inverter.
- *Pr. 340* can be changed via the operation panel regardless of the operation mode.
- When setting a value other than 0 in *Pr. 340*, confirm that the initial settings of the inverter are correct.
- Refer to the inverter's *Instruction Manual (Detailed)* for additional information regarding *Pr. 79* and *Pr. 340*.

Pr. 340 Setting	Pr. 79 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 ^{*1}
	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 ^{*1}
		X12 (MRS) signal OFF...external operation mode	External operation mode fixed (forcibly switched to external operation mode)
1, 2 ^{*2}	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 ^{*2}	0	NET operation mode	Switching between PU and NET operation modes is enabled ^{*3}
	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 ^{*3}
	7	External operation mode	Same as when <i>Pr. 340</i> = "0"

*1 The operation mode cannot be directly changed between PU mode and NET mode.

*2 *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 if 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.

*3 The operation mode can be changed between PU mode and NET mode with the  key on the operating panel (FR-DU08) and X65 signal.

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

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

Parameter Number	Name	Default Value	Setting Range	Description
338	Communication operation command source	0	0	Communication option card is the run command source (A8NETH-2P run command is valid.)
			1	External input is the run command source. (A8NETH-2P run command is invalid.)
339	Communication speed command source	0	0	Communication option card is the speed command source (A8NETH-2P speed command is valid.)
			1	External input is the speed command source. (A8NETH-2P speed command is invalid.)
			2	External input is the speed command source. (If there is no external input, the frequency command via communication is valid, and the frequency command from terminal 2 is invalid.)
550	NET mode operation command source selection	9999	0	Communication option card control is valid (A8NETH-2P control is valid.)
			1	Control via the inverter's RS-485 port (A8NETH-2P control is invalid.)
			9999	Communication option automatic recognition. Normally, control via the inverter's RS-485 port is valid. When an A8NETH-2P communication option card is installed, that communication option card's control is made valid instead of the inverter's RS-485 port.

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.



- 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

REGISTER NUMBERING AND BEHAVIOR

4.1 Register Numbers

All accessible inverter parameters are referenced by their register number as defined in Table 13 and can be conveniently referenced in the Configuration Studio (section 5.6) and the embedded web server (section 6.2.3). Note that the register list is not exhaustive, the registers may not exist for all inverters, and the register data contents may vary depending on the inverter. The register numbers are used when accessing and configuring registers via an Ethernet protocol. Information regarding several commonly used registers is included in this manual for user convenience. Registers 5 to 7 are documented in the inverter user manual “Mitsubishi inverter protocol” section. Registers 201 to 449 are documented in the inverter user manual “List of inverter monitored items / command items”. For information regarding the remaining inverter parameters, refer to the inverter user manual “Parameter List”.

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 register/parameter takes place, the read/write may be successful (depending on the specific register accessed; refer to section 4.2), 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.

Table 13: Register Parameter List

Register	Description
1	Command register (refer to section 4.3)
5	Inverter reset (refer to section 4.4)
6	Alarm history clear (refer to section 4.5)
7	All parameter clear (refer to section 4.6)
102	Inverter state (refer to section 4.7)
201	Output frequency
202	Output current
203	Output voltage
205	Frequency setting value
206	Motor speed
207	Motor torque
208	Converter output voltage
209	Regenerative brake duty
210	Electronic thermal relay function load factor
211	Output current peak value
212	Converter output voltage peak value
213	Input power
214	Output power
215	Input terminal status
216	Output terminal status
217	Load meter
218	Motor excitation current
219	Position pulse
220	Cumulative energization time
222	Orientation status
223	Actual operation time
224	Motor load factor

Register	Description
225	Cumulative power
226	Position command (lower 16 bits)
227	Position command (upper 16 bits)
228	Current position (lower 16 bits)
229	Current position (upper 16 bits)
230	Drop pulse (lower 16 bits)
231	Drop pulse (upper 16 bits)
232	Torque order
233	Torque current order
234	Motor output
235	Feedback pulse
238	Trace status
240	PLC function user monitor 1
241	PLC function user monitor 2
243	PLC function user monitor 3
246	Motor temperature
250	Power saving effect
251	Cumulative saving power
252	PID set point
253	PID measured value
254	PID deviation value
258	Option input terminal status1
259	Option input terminal status2
260	Option output terminal status
261	Motor thermal load factor
262	Transistor thermal load factor
264	PTC thermistor resistance
265	Output power (with regenerative display)

Register	Description
266	Cumulative regenerative power
267	PID measured value 2
268	Second PID set point
269	Second PID measured value
270	Second PID deviation
271	Cumulative pulse
272	Cumulative pulse carrying-over times
273	Cumulative pulse (control terminal option)
274	Cumulative pulse carrying-over times (control terminal option)
280	Integrated power on time
281	Running time
282	Saving energy monitor
284	Fault code 1 – Most recent fault (refer to section 4.11)
285	Fault code 2 (refer to section 4.11)
286	Fault code 3 (refer to section 4.11)
287	Fault code 4 (refer to section 4.11)
288	Fault code 5 (refer to section 4.11)
289	Fault code 6 (refer to section 4.11)
290	Fault code 7 (refer to section 4.11)
291	Fault code 8 – Least recent fault (refer to section 4.11)
295	Second PID measured value 2
296	Second PID manipulated variable
300	Current position 2 (lower 16 bits)
301	Current position 2 (upper 16 bits)
302	PID manipulated variable
449	Run command
1224	DriveControlMaskWrite (lower 16 bits)
1225	DriveControlMaskWrite (upper 16 bits)

Register	Description
1226	DriveControl
1227	DriveStatus (lower 16 bits) (refer to section 4.8)
1228	DriveStatus (upper 16 bits)
1229	SetpointSpeed (refer to section 4.9)
1230	CommandSpeed
1231	ActualSpeed (refer to section 4.10)
1232	SpeedScaleNumerator (lower 16 bits)
1233	SpeedScaleNumerator (upper 16 bits)
1234	SpeedScaleDenominator (lower 16 bits)
1235	SpeedScaleDenominator (upper 16 bits)
1236	RatedSpeed
1237	PoleCount
1238	RatedCurrent (lower 16 bits)
1239	RatedCurrent (upper 16 bits)
1240	RatedVoltage
1241	MotorType
1242	DriveMode
1243	SupportedModes (lower 16 bits)
1244	SupportedModes (upper 16 bits)
1245	AccelerationDeltaSpeed (lower 16 bits)
1246	AccelerationDeltaSpeed (upper 16 bits)
1247	AccelerationDeltaTime (lower 16 bits)
1248	AccelerationDeltaTime (upper 16 bits)
1249	DecelerationDeltaSpeed (lower 16 bits)
1250	DecelerationDeltaSpeed (upper 16 bits)
1251	DecelerationDeltaTime (lower 16 bits)
1252	DecelerationDeltaTime (upper 16 bits)
1253	QuickDecelerationDeltaSpeed (lower 16 bits)

Register	Description
1254	QuickDecelerationDeltaSpeed (upper 16 bits)
1255	QuickDecelerationDeltaTime (lower 16 bits)
1256	QuickDecelerationDeltaTime (upper 16 bits)
1257	MaxSpeed (lower 16 bits)
1258	MaxSpeed (upper 16 bits)
1259	MinSpeed (lower 16 bits)
1260	MinSpeed (upper 16 bits)
1261	TargetTorque
1262	ActualTorque
1263	TorqueSlope (lower 16 bits)
1264	TorqueSlope (upper 16 bits)
1265	TorqueProfileType
1266	RatedTorque (lower 16 bits)
1267	RatedTorque (upper 16 bits)
1268	TorqueScaleNumerator (lower 16 bits)
1269	TorqueScaleNumerator (upper 16 bits)
1270	TorqueScaleDenominator (lower 16 bits)
1271	TorqueScaleDenominator (upper 16 bits)
1272	DisableOptionCode
1273	ShutdownOptionCode
2000 and higher	<i>Pr. 0</i> and higher. To calculate the register number, add 2000 to the parameter number. For example, <i>Pr. 123</i> is register 2123 (123 + 2000).

4.2 Scanned Registers

All 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.

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, the register can be read over the network at a later time to confirm whether or not that the written value “took hold” in the inverter.

Even if an inverter parameter corresponding to a given scanned register does not exist, the interface card still maintains a placeholder location in its internal mirroring memory for that register. This feature allows for the block access of non-contiguous registers as described in section 4.1.

4.3 Inverter Command Register

The command word is register 1 and the bit-mapping is described in Table 14. These bits are mapped to DriveControl (register 1226).

Table 14: Inverter Command Register Mapping

Bit	Significance	Value	Description
0	Reserved	Not used	-
1	EnableOp	1	Enable operation
		0	Disable operation
2	EnableRevOp	1	Enable reverse operation
		0	Disable reverse operation
3	QuickStop	1	Enable quick stop
		0	Disable quick stop
4 - 10	Reserved	Not used	-
11	ResetMalfunction	1	Reset fault condition on a positive edge (0→1 transition)
		0	-
12 - 13	Reserved	Not used	-
14	NetRef	1	Request torque/speed reference from network
		0	Request local reference
15	NetCtrl	1	Request drive control from network
		0	Request local drive control

4.4 Inverter Reset Register

The inverter reset register is register 5. A value of 0x9696 or 0x9966 will reset the inverter.

4.5 Alarm History Clear Register

The alarm history clear is register 6. A value of 0x9696 will clear the alarm history.

4.6 All Parameter Clear Register

The all parameter clear is register 7. Refer to Table 15 for the appropriate value.

Table 15: All Parameter Clear Register

Pr. Data	Communi- cation Pr. *1	Calibration Pr. *2	Other Pr. *3	HEC HF3 HFF
H9696	○	×	○	○
H9966	○	○	○	○
H5A5A	×	×	○	○
H55AA	×	○	○	○

*1 Refer to communication related parameters.

*2 Refer to the list of calibration parameters.

*3 Pr. 75 is not cleared.

4.7 Inverter State Register

The inverter state is register 102 and the states are listed in Table 16.

Table 16: Inverter State

Value	Description
0	NOT_READY_TO_SWITCH_ON
1	SWITCH_ON_DISABLED
2	READY_TO_SWITCH_ON
3	SWITCHED_ON
4	OPERATION_ENABLED
5	DISABLE
6	SHUTDOWN
7	QUICK_STOP
8	FAULT_REACTION_ACTIVE
9	FAULTED

4.8 DriveStatus Register

The DriveStatus is register 1227 and the bit-mapping is described in Table 17.

Table 17: DriveStatus Register Mapping

Bit	Significance	Value	Description
0	ReadyToSwitchOn	1	Power supply is switched on, electronics initialized, main contact, if available, has dropped out, pulses are inhibited
		0	-
1	SwitchedOn	1	Voltage at the power converter, i.e. the main contact is closed (if present)
		0	-
2	OpEnabled	1	Drive function enabled
		0	-
3	Faulted	1	Unacknowledged faults present
		0	-
4	VoltageEnabled	1	Power enabled
		0	Power disabled
5	QuickStopDisabled	1	Quick stop disabled
		0	Executing quick stop request
6	SwitchOnDisabled	1	Drive in SWITCH_ON_DISABLED
		0	-
7	Reserved	Not used	-
8	RefFromNet	1	Network torque/speed reference
		0	Local torque/speed reference
9	CtrlFromNet	1	Control is from network
		0	Control is local
10	SetpointReached	1	Actual output frequency equals frequency set point
		0	-

Bit	Significance	Value	Description
11	LimitActive	1	Internal (manufacturer specific) limitation is active (e.g. current, power or speed limit)
		0	-
12 - 13	Reserved	Not used	-
14	RevOpEnabled	1	Drive function enabled in reverse mode
		0	-
15	Stopping	1	Drive is stopping
		0	-

4.9 SetpointSpeed Register

The SetpointSpeed is register 1229 and specifies the target speed in RPM. A positive value indicates forward motion, while a negative value indicates reverse motion. The value must be scaled based on the SpeedScale factor. By default, the SetpointSpeed is specified in 0.1 RPM units.

4.10 ActualSpeed Register

The ActualSpeed is register 1231 and reflects the actual speed in RPM. A positive value indicates forward motion, while a negative value indicates reverse motion. The value is scaled based on the SpeedScale factor. By default, the ActualSpeed is specified in 0.1 RPM units.

4.11 Fault History Codes

The fault history codes are reflected in registers 284 (most recent fault) to 291 (least recent fault). The fault code is described in the inverter user manual "List of fault display" section.

5 MITSUBISHI CONFIGURATION STUDIO

5.1 Overview

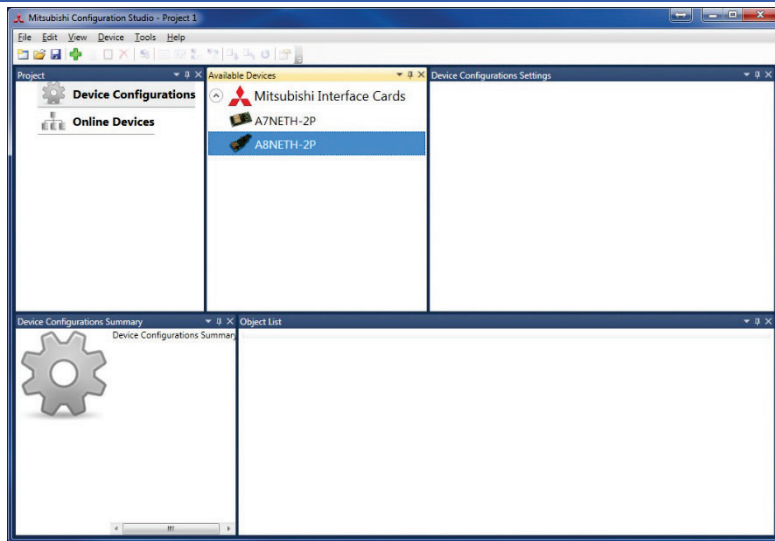


Figure 2: Mitsubishi Configuration Studio

The card is discovered, configured and updated by the Mitsubishi Configuration Studio PC application. The studio must be installed prior to connecting a card to ensure that the appropriate USB drivers are installed. The studio will typically require a

USB connection for reading/writing a configuration and updating the firmware. Depending on the currently-active drivers, remote discovery, network setting, and firmware updates are also possible via Ethernet. The latest release of the Configuration Studio can be downloaded from the [product web page](#). The remainder of this section will provide only a brief introduction to the configuration concepts. For protocol-specific configuration, refer to the relevant protocol section.

Creating a Device Configuration

A device can be added to the **Project** panel for configuration by first selecting the **Device Configurations** list heading and then:

- Double-clicking on the device in the **Available Devices** panel.
- Right-clicking on the device in the **Available Devices** panel and choosing **Add** from the context-sensitive menu.
- Hitting the <ENTER> key on the keyboard when the device is selected in the **Available Devices** panel.
- Dragging the device from the **Available Devices** panel into the **Project** panel.
- Selecting it and selecting **Add Selected Device** from the **Edit** menu.
- Selecting it and clicking the **Add** button in the toolbar.

The device will then be added to the list of **Device Configurations**.

Going Online with a Device

All connected devices are automatically added to the **Discovered Devices** panel. This panel is shown by selecting the **Online Devices** list heading in the **Project** panel. To go online with a device:

- Double-click on it in the **Discovered Devices** panel.
- Right-click on it in the **Discovered Devices** panel and choose **Go Online** from the context-sensitive menu.
- Hit the <ENTER> key on the keyboard when the device is selected in the **Discovered Devices** panel.
- Drag it from the **Discovered Devices** panel into the **Project** panel.
- Select it and select **Go Online with Device** from the **Edit** menu.
- Select it and click the **Go Online** button in the toolbar.

When the studio goes online with a device, its configuration is automatically read. While the studio is online with a device, it will appear in green text in the **Discovered Devices** panel. The studio may be online with multiple devices simultaneously.

Uploading a Device's Configuration into a Project

The current configuration of an online device can be uploaded into the **Project** panel by selecting a device under the **Online Devices** list heading and then:

- Right-clicking on it and choosing **Upload Configuration** from the context-sensitive menu.
- Dragging it from the **Online Devices** heading into the **Device Configurations** heading.
- Selecting it and selecting **Upload Configuration to Project** from the **Device** menu.
- Selecting it and clicking the **Upload Configuration** button in the toolbar.

The device's configuration will then be added to the list of **Device Configurations**. Once the configuration is uploaded into the project, it may be modified.

Removing a Device Configuration from a Project

A configuration can be removed from a project by:

- Selecting the device in the **Project** panel and dragging it. A trash can icon will appear at the bottom of the **Project** panel, and dragging and dropping the device in the trash will remove it from the project.
- Hitting the <DELETE> key on the keyboard when the device is selected in the **Project** panel.
- Right-clicking on the device in the **Project** panel and choosing **Remove** from the context-sensitive menu.
- Selecting **Remove Selected Item** from the **Edit** menu when the device is selected.
- Clicking on the **Remove** button in the toolbar when the device is selected.

Going Offline with a Device

To go offline with a device:

- Select the device in the **Project** panel and drag it. A trash can icon will appear at the bottom of the **Project** panel, and dragging and dropping the device in the trash will go offline with it.

- Hit the <DELETE> key on the keyboard when the device is selected in the **Project** panel.
- Right-click on the device in the **Project** panel and choose **Go Offline** from the context-sensitive menu.
- Select **Go Offline with Device** from the **Edit** menu when the device is selected.
- Click on the **Go Offline** button in the toolbar when the device is selected.

Downloading a Configuration to a Device

To download a configuration to an online device, first select the device under the **Device Configurations** heading in the **Project** panel, and then navigate to **Device...Download Configuration to Device**. If the studio is currently online with only one compatible device, then the configuration will be downloaded to the online device. Otherwise, a device selection prompt is displayed to select which device to download the configuration to. Do not power off the device or interrupt the connection once the download is in progress as this may corrupt the firmware and/or the configuration.

Updating Firmware

The studio automatically manages firmware updates when going online with a device and downloading a configuration to a device. Download the latest studio from the [product web page](#) to obtain the latest firmware. Do not power off the device or interrupt the connection once the update is in progress as this may corrupt the firmware and/or the configuration.

Resetting an Online Device

To reset an online device, first select the device in the **Project** panel and then navigate to **Device...Reset Device**.

General Configuration Process

To configure a device, add the desired protocol(s) and configure any objects associated with the respective protocol(s). Any changes will take effect once the configuration is downloaded to a device.

Note that numeric values can be entered not only in decimal but also in hexadecimal by including “0x” before the hexadecimal number.

5.2 General Object Editing Activities

The following editing activities apply for all types of configuration objects and project elements.

Adding an Object

To add an object, click on an item (protocol driver or Node, for example) in the **Project** panel. Any available objects for that item will be listed in the **Available Objects** panel (the panel title depends on the currently-selected item). An object can then be added to the item by:

- Double-clicking on it.
- Right-clicking on it and choosing **Add** from the context-sensitive menu.
- Hitting the <ENTER> key on the keyboard when the object is selected.
- Dragging it into the **Project** panel.
- Selecting it and selecting **Add Selected Device** from the **Edit** menu.
- Selecting it and clicking the **Add** button in the toolbar.

The object's configurable fields can then be populated with valid values (where applicable).

Viewing an Object

In the **Project** panel, select a parent object to display a summary of all its child objects. For example, selecting a protocol driver will display the driver's configuration in the **Summary** panel and list of current objects in the **Object List** panel.

Updating an Object

To update an object, select the object in the **Project** panel and make any required changes in the **Settings** panel.

Deleting an Object

An object can be deleted by performing one of the following actions:

- Selecting the object in the **Project** panel and dragging it. A trash can icon will appear at the bottom of the **Project** panel, and dragging the object to the trash will then delete it from the project.

- Hitting the <DELETE> key on the keyboard when the object is selected in the **Project** panel.
- Right-clicking on the object in the **Project** panel and choosing **Remove** from the context-sensitive menu.
- Selecting **Remove Selected Item** from the **Edit** menu when the object is selected.
- Clicking on the **Remove** button in the toolbar when the object is selected.

Note that this action cannot be undone. Deleting an object will also delete all of its child objects.

Copying and Pasting an Object

To copy an object, first click on an item in the **Project** panel. An object can then be copied by:

- Right-clicking on it and choosing **Copy** from the context-sensitive menu.
- Pressing the <CTRL+C> keys on the keyboard.
- Holding the <CTRL> key and dragging the item to the desired location in the **Project** panel.
- Dragging the item to a new location under a different parent object in the **Project** panel.
- Selecting **Copy Selected Item** from the **Edit** menu.
- Clicking on the **Copy** button in the toolbar.

To paste an object, first click on an item at the desired location in the **Project** panel. An object can then be pasted by:

- Right-clicking on it and choosing **Paste** from the context-sensitive menu.
- Pressing the <CTRL+V> keys on the keyboard.
- Dropping an item onto the desired location in the **Project** panel after holding the <CTRL> key and dragging the item.
- Dropping an item onto a new location under a different parent object in the **Project** panel after dragging the item.
- Selecting **Paste Item** from the **Edit** menu.
- Clicking on the **Paste** button in the toolbar.

After pasting an object, the object's configurable fields can then be modified with valid values (where applicable).

Note that the studio allows you to copy and paste items between different locations, including different devices. This is useful for copying partial configurations from one device to another.

Reordering Objects

Objects can be reordered in the **Project** panel by dragging the item to the desired location. If the item is dragged outside of the items in the project tree, it will be moved to the end.

5.3 Ethernet Settings

The **Ethernet Settings** panel contains Ethernet-related items that are not specific to any given protocol. These settings must be appropriately configured regardless of any Ethernet control protocols that may be enabled. The **Ethernet Settings** panel is then available whenever the **Ethernet** port is selected in the **Project** panel.

5.3.1 Authentication

Be sure to make a note of the new settings whenever authentication credentials are changed, as they must be entered whenever the web page is accessed or an FTP session is initiated.

User Name

The username is case-sensitive and can contain letters (“a...z” and “A...Z”) and numbers (“0...9”).

Password

The password is case-sensitive and can contain letters (“a...z” and “A...Z”) and numbers (“0...9”).

5.3.2 Network Configuration

The card supports a static IP address. The IP Address, Subnet Mask and Default Gateway fields must be configured. Please consult with your network administrator for the proper settings of these fields.

5.4 Internal Logic Settings

5.4.1 Fail-safe Values

5.4.1.1 Overview

The card can be configured to perform a specific set of actions when network communications are lost (timeout event). This allows each inverter parameter to have its own unique “fail-safe” condition in the event of network interruption. 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:

- The timeout time
- Timeout Object configuration

5.4.2 Timeout Time

The timeout time is the maximum number of milliseconds for a break in network communications before a timeout will be triggered. This timeout setting is configured at the protocol level as part of a driver’s configuration, and used by the protocol drivers themselves to determine abnormal loss-of-communications conditions. These conditions then trigger timeout processing events. If it is not desired to have a certain protocol trigger timeout processing events, then the protocol’s timeout time may be set to 0 (the default value) to disable this feature.

For some protocols, the timeout time is set by the master device (PLC, scanner, etc.), and a timeout time setting is therefore not provided in the Configuration Studio’s driver configuration. Additionally, not all protocols support timeout detection: refer to the protocol-specific sections of this manual for more information.

5.4.2.1 Timeout Object Configuration

A timeout object is used as part of the timeout processing to set certain parameters to “fail-safe” values. When a timeout event is triggered by a protocol, the timeout objects are parsed and written to the corresponding parameter(s). To add a timeout object, select the device in the **Project** panel, then add **Internal Logic...Fail-safe Values...Timeout Object**. The following paragraphs describe the configurable fields of a timeout object:

Description

This field is strictly for user reference: it is not used at any time by the device.

Register

Enter the register number (refer to section 4) corresponding to the inverter parameter.

Data Type

This is the size of valid values and is fixed to “16-Bit Unsigned” allows for a range of timeout values between 0 and 65535.

Value

Enter the “fail-safe” timeout value that the register encompassed by this timeout object will be automatically written with upon processing a timeout event triggered by a protocol.

5.5 Discovery over Ethernet

Depending on the currently-enabled driver, the Configuration Studio will automatically discover the device on the current Ethernet network, regardless of whether or not the card's network settings are compatible with the subnet upon which they reside. All connected devices are automatically added to the **Discovered Devices** panel. This panel is shown by selecting the **Online Devices** list heading in the **Project** panel. In the **Discovered Devices** panel, discovered Ethernet devices will be listed under **Ethernet** and will display the firmware version in brackets and the current IP address in parentheses to the right of the device name, similar to Figure 3.



Figure 3: Studio Discovery over Ethernet

In order for the studio to discover devices, certain UDP Ethernet traffic (port 4334) 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 studio is unable to discover any devices on the current subnet, be sure to check the computer's firewall settings during troubleshooting, and add the studio as a program exception to the firewall configuration if necessary. It may be necessary to restart your PC before the new firewall configuration can take effect.

The network settings of a discovered card can be configured remotely by:

- Right-clicking on the device in the **Project** panel and choosing **Configure Network Settings...** from the context-sensitive menu.
- Selecting the device in the **Project** panel and navigating to **Device...Configure Network Settings...**

The network settings pop-up should appear similar to Figure 4. Modify the network settings as necessary and click the OK button for the changes to take effect. Note that this will cause the device to become temporarily inaccessible and may fault the inverter.

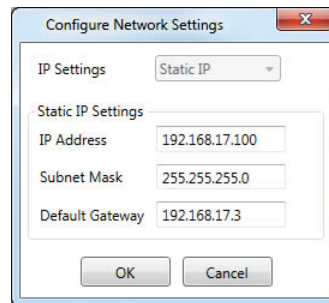


Figure 4: Remotely Configure Network Settings

5.6 Manage Device Parameters

The accessibility and scan priority of the inverter parameters can be adjusted (refer to Figure 5). This is an advanced feature and must only be used after consulting technical support to determine the appropriate settings for the target application. The **Manage Device Parameters** configuration window is found by:

- Right-clicking on the device in the **Project** panel and choosing **Manage Parameters...** from the context-sensitive menu.
- Selecting the device in the **Project** panel and navigating to **Device...Manage Device Parameters...**

A parameter is accessible and actively scanned (read from and written to the inverter) only if its corresponding checkbox is enabled. Likewise, a parameter is inaccessible if its checkbox is disabled.

Parameters that are accessed more frequently or require a faster update rate should be set to high priority. All other parameters should be set to low priority.

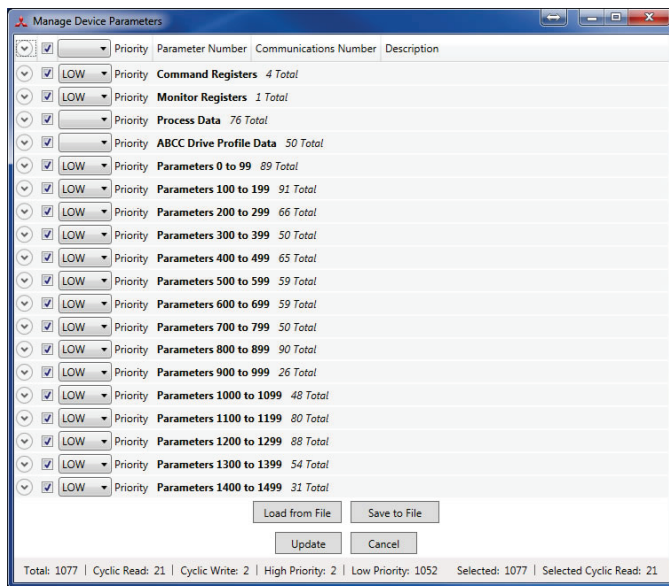


Figure 5: Manage Device Parameters

5.7 Backup and Restore Parameters

The parameter values can be backed up from the inverter and restored to the inverter (refer to Figure 6 and Figure 7). This allows for easy inverter cloning. The backup parameter values are stored as a CSV file. A parameter value can be excluded by disabling the corresponding checkbox. The parameter value can also be modified before the backup and restore is executed. Note that backup and restore does not modify the parameter list (refer to section 5.6). The backup and restore parameter configurations are found by:

- Right-clicking on the device in the **Project** panel and choosing **Backup Parameters...** or **Restore Parameters...** from the context-sensitive menu.
- Selecting the device in the **Project** panel and navigating to **Device...Backup Parameters from Device...** or **Restore Parameters to Device...**

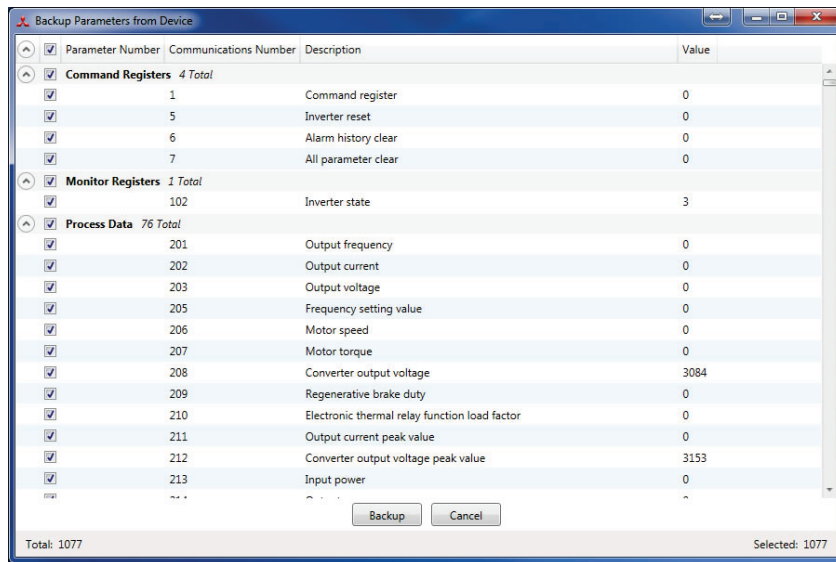


Figure 6: Backup Parameters

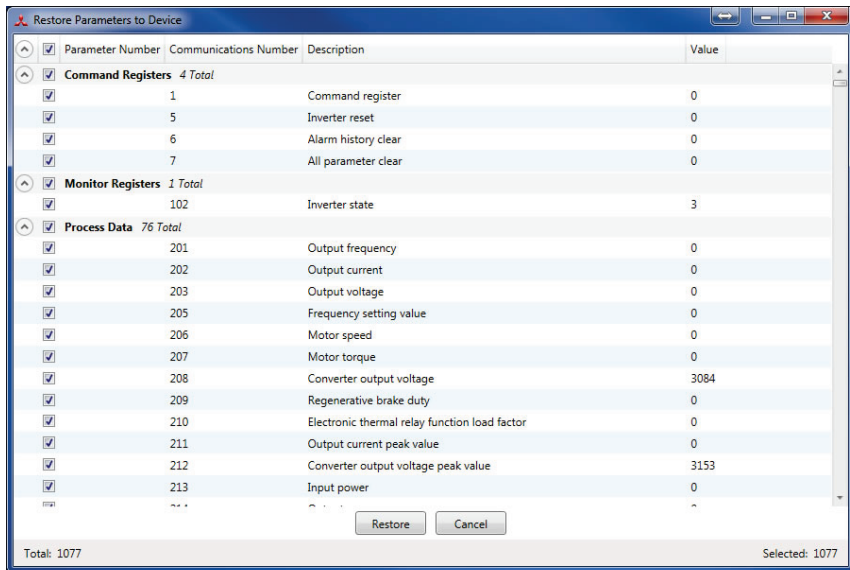


Figure 7: Restore Parameters

5.8 Restore Factory Settings

The interface card (connected via USB) can be restored to the factory settings. Note that the filesystem will be reformatted, which will destroy all custom modifications and configurations. Please backup the configuration before executing this feature. The factory settings can be restored by:

- Right-clicking on the device in the **Project** panel and choosing **Restore Factory Settings**.
- Selecting the device in the **Project** panel and navigating to **Restore Factory Settings**.

5.9 Help

For assistance in understanding configuration objects and fields, simply hover the mouse over the object or text to display a useful tooltip. Ensure that the **Help...Show Help Tooltips** option is checked. Links to videos and documents can be found in the **Help** menu. Please review the tooltips and links before contacting technical support for more in-depth assistance.

6

EMBEDDED WEB SERVER

6.1 Overview

The interface contains an embedded (HTTP) web 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 and controlled from across the room or from across the globe. To access an interface's embedded web server, directly enter the target unit's IP address into the address (URL) field of your web browser. Refer to Figure 8 for a representative screenshot of the web server interface. 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.

Note that in order to view the interface's web page, the free Adobe Flash Player browser plug-in is required. The plug-in can be downloaded from <http://www.adobe.com>.

The screenshot displays the Mitsubishi Electric ABNETH-2P Embedded Server web interface. The browser window shows the URL `http://192.168.17.102` and the page title `ABNETH-2P Embedded Server`. The interface features a navigation bar with the Mitsubishi Electric logo and the text `ABNETH-2P Embedded Server` and `V1`. Below the navigation bar, there are two tabs: `Monitor` (selected) and `Dashboard`. The main content area is divided into several sections:

- Parameter Group:** A dropdown menu with options: `All`, `Command registers`, `Monitor registers`, `Process data`, and `Configuration parameters`.
- Product Information:** A text box displaying `Product Name: ABNETH-2P` and `Ethernet CPU firmware version: V1.1.1`.
- Activity:** A section labeled `ACTIVITY` with a refresh icon.
- Table:** A table with four columns: `Parameter`, `Register`, `Description`, and `Value (Decimal)`. The table contains 15 rows of data.

Parameter	Register	Description	Value (Decimal)
	1	Command register	0
	5	Inverter reset	0
	6	Alarm history clear	0
	7	All parameter clear	0
	102	Inverter state	3
	201	Output frequency	0
	202	Output current	0
	203	Output voltage	0
	205	Frequency setting value	0
	206	Motor speed	0
	207	Motor torque	0
	208	Converter output voltage	3102
	209	Regenerative brake duty	0
	210	Electronic thermal relay function load factor	0
	211	Output current peak value	0
	212	Converter output voltage peak value	3153
	213	Input power	0
	214	Output power	0
	215	Input terminal status	0

Figure 8: Embedded Web Server

6.2 Monitor Tab

6.2.1 Information Window



Figure 9: Monitor Tab Information Window

Figure 9 shows the Information Window, which displays messages regarding the status of the interface card or web browser session. There is also an “ACTIVITY” indicator 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. To reestablish communications, “refresh” your web browser.

6.2.2 Parameter Group Selection List

The Parameter Group Selection List is shown in Figure 10. 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 6.2.3).

6.2.3 Parameter List

The parameter list is shown in Figure 11. The parameters that are displayed in the list at any given time depend on the parameter groups that are currently selected (refer to section 6.2.2) and the filter (refer to section 6.2.4).

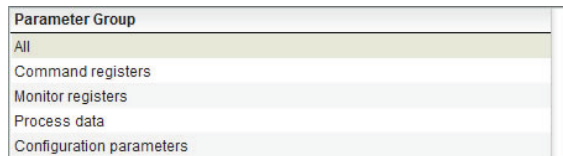


Figure 10: Parameter Group Selection List

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 4). 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	2000	Torque boost	60
1	2001	Maximum frequency	12000
2	2002	Minimum frequency	0
3	2003	Base frequency	6000
4	2004	Multi-speed setting (high speed)	6000
5	2005	Multi-speed setting (middle speed)	3000
6	2006	Multi-speed setting (low speed)	1000
7	2007	Acceleration time	50
8	2008	Deceleration time	50
9	2009	Electronic thermal O/L relay	255
10	2010	DC injection brake operation frequency	300
11	2011	DC injection brake operation time	5
12	2012	DC injection brake operation voltage	40
13	2013	Starting frequency	50
14	2014	Load pattern selection	0
15	2015	Jog frequency	500
16	2016	Jog acceleration/deceleration time	5
17	2017	MRS input selection	0
18	2018	High speed maximum frequency	12000

Figure 11: 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 6.2.5).
- The column widths can be changed by dragging the vertical bars that separate the header row's cells.
- 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.

6.2.4 Parameter List Filter

A filter function provides Parameter List search capabilities. To use the filter function, simply type a word into the filter entry box and then click the “filter” button. Refer to Figure 12. The filter will then display only those parameters currently available in the Parameter List that satisfy the search criteria. Once a filter has been entered, it will continue to be applied to all information normally displayed in the Parameter List. To remove the filter, delete all characters contained in the filter entry box and then click the “filter” button.

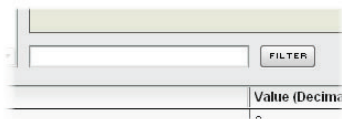


Figure 12: Parameter List Filter

6.2.5 Radix Selection

Figure 13 shows the radix selection buttons that allow changing the Parameter List “value” column data display and entry radix between decimal and hexadecimal formats.

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.



Figure 13: Radix Selection

6.3 Dashboard Tab

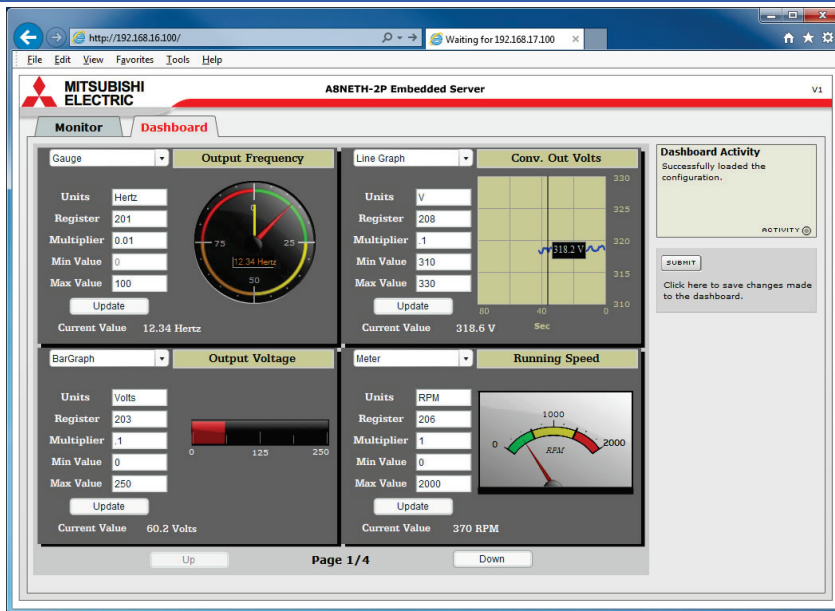


Figure 14: Dashboard Tab

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 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 register's value via one of six different gauge types. User-defined engineering units, scaling and range limits are also configurable. Refer to Figure 14.

6.3.1 Information Window

Figure 15 shows the Information Window, which displays messages regarding the status of the Dashboard configuration parameters (loading or submitting).



Figure 15: Dashboard Tab Information Window

6.3.2 Gauge Window Navigation

Figure 16 shows the two buttons that provide for navigation of the gauge windows. Clicking the "Up" or "Down" buttons will scroll the gauge windows.

6.3.3 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 17 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

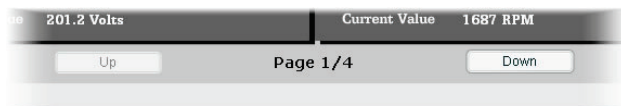


Figure 16: Gauge Window Navigation

be reflected on the gauge.

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 6.3.4).

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 17. 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.

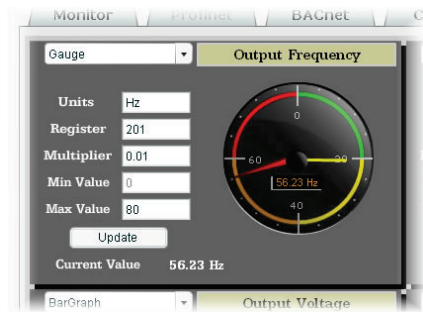


Figure 17: Gauge

BarGraph: Refer to Figure 18. 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.

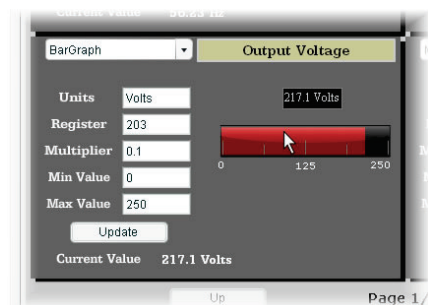


Figure 18: BarGraph

Meter: Refer to Figure 19. 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. $0 \dots 0xFFFF$ equates to $0 \dots 65535_{10}$.)

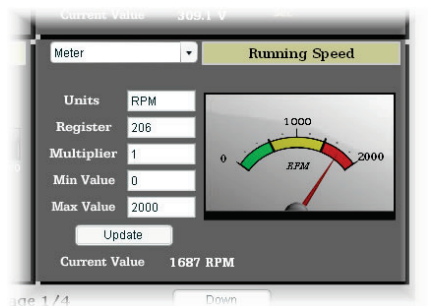


Figure 19: Meter

Pos/Neg Meter: Refer to Figure 20. 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...32767₁₀, 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.

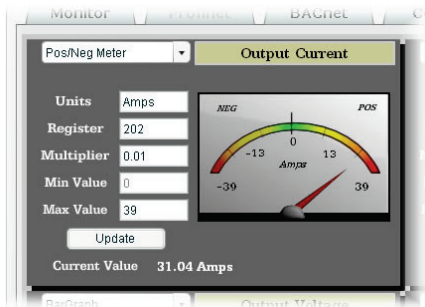


Figure 20: Pos/Neg Meter

Thermometer: Refer to Figure 21. 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.

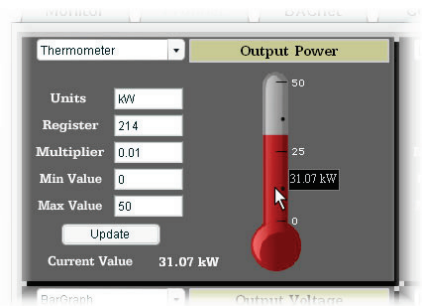


Figure 21: Thermometer

Line Graph: Refer to Figure 22. 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.

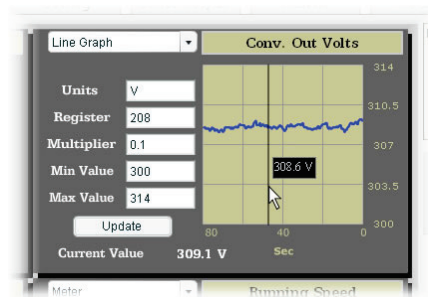


Figure 22: Line Graph

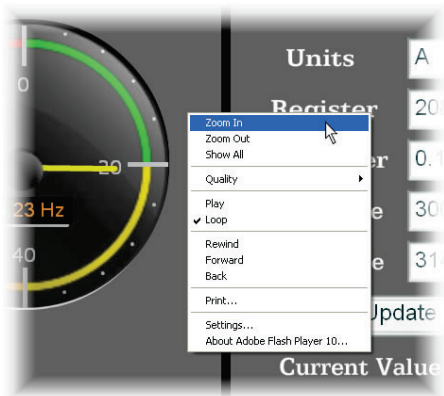


Figure 23: Zooming



**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 23).

6.3.4 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 24. 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 24: Submit Dashboard Changes

6.4 Customizing the Embedded Web Server

6.4.1 Customization Overview

It is possible for end-users to customize the embedded web server in order to create their own application-specific or corporate “look and feel”. Knowledge of authoring dynamic web content is required. Using windows explorer, it is possible to load customized web server content into the “WEB” folder on the interface card’s file system (refer to section 7.2). Usually, this web server content contains programming which implements the XML socket-based XTPro protocol (refer to section 6.4.2). Via XTPro, the embedded web server can gain access to any inverter register and the interface card file system resources, and manipulate them as required.

Notes

- All files in the “WEB” folder may be deleted or replaced if desired by the user.
- The default HTML file targeted by the web server is “index.htm”. Therefore, when customizing the web server content, ensure that initial file “index.htm” exists.
- All files accessed by the web server itself must reside in the “WEB” folder. Note that this does not restrict active web server content to using only the “WEB” folder, however, as XTPro “read_file” and “write_file” commands can access any existing location on the file system.
- If the factory-default “WEB” folder contents need to be recovered (if they are accidentally deleted, for example), they can be downloaded from the [product web page](#) on the internet.
- Two simultaneous web server sessions are supported. Note that the number of available simultaneous web server sessions is independent of the number of available simultaneous XTPro XML sockets.

6.4.2 XTPro Overview

XTPro is an acronym for **XML TCP/IP Protocol**. The XTPro specification is an application-layer (positioned at level 7 of the OSI model) messaging protocol that provides XML-based client/server communication via TCP port 843. Typically, XTPro is used for the implementation of graphical user interfaces (GUIs), such as advanced web servers or HMIs that have the ability to request information via XML sockets, and then manipulate and/or display the information in a rich application-specific manner.

XTPro is a request/response protocol that provides services specified by commands. For more information on XTPro, refer to the separate [XTPro Specification](#). This section will cover the device-specific implementation of the XTPro protocol.

6.4.3 XTPro Web Browser-Based Implementation

A representative implementation based upon using a web browser as the client is detailed in Figure 25. In this scenario, the client application is developed by using an active web server authoring tool (such as Adobe Flash®). The active content is then embedded into one or more HTML files and loaded onto the device's file system (refer to section 6.4.1 for detailed information regarding customization of the web server content). Accessing the device's web server via a standard web browser then loads the active content, which initiates communication with the server.

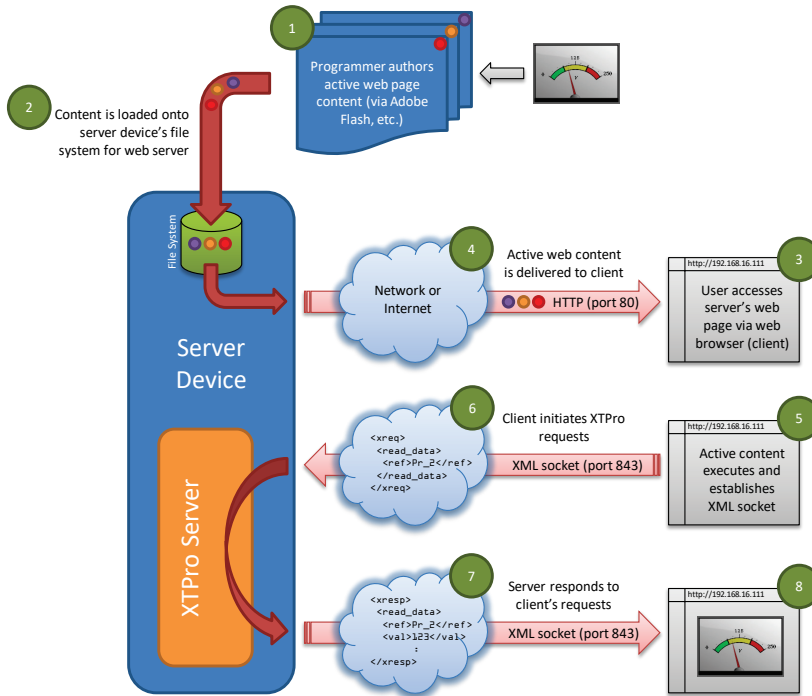


Figure 25: Web Browser-Based Implementation

6.4.4 XTPro HMI-Based Implementation

A representative implementation based upon a stand-alone HMI client is detailed in Figure 26. In this scenario, the client application is developed by using tools provided by the HMI manufacturer, and is hosted independently of the actual server device.

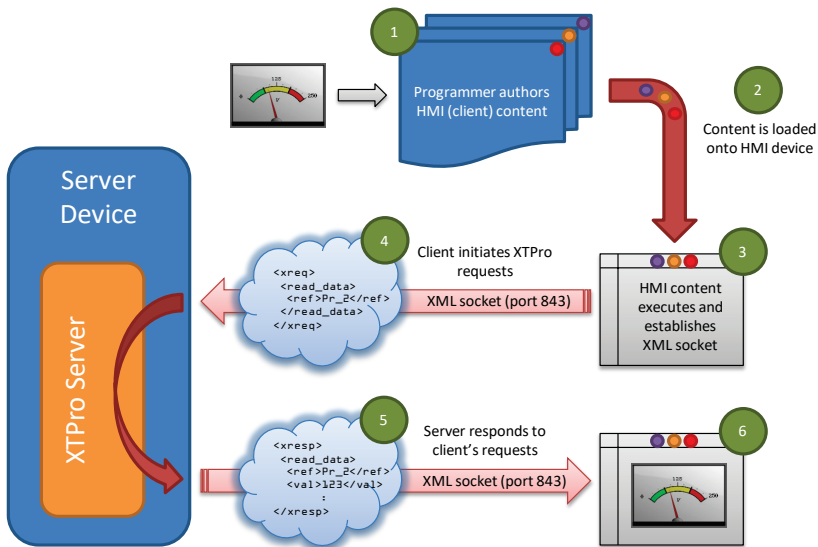


Figure 26: HMI-Based Implementation

6.4.5 XTPro Supported Commands

For a summary of the supported XTPro commands, refer to Table 18. Note that two simultaneous XTPro connections are available.

Table 18: Supported XTPro Commands

Command	Supported	Notes
noop	Yes	-
vzn	Yes	Supports XTPro specification version 1
id	Yes	-
read_data	Yes	"reference" is the inverter's register (e.g. "1227" for SetpointSpeed command), while "data_value" is a 16-bit hexadecimal value (e.g. "1F4" for a decimal value of 500)
write_data	Yes	
load_file	Yes	The absolute file path must start with a forward slash '/'
store_file	Yes	
reinit	No	Reinitializes only the configurable drivers and services (does not perform a complete device soft reboot)
auth	Yes	Authorization is not required
cov	Yes	COV notification messages are sent every 200ms

7.1 Overview

The interface card's on-board filesystem is used by the application firmware. Currently, the application firmware's main use of the filesystem is to store XML-encoded configuration files and the embedded web server. The studio must be used to manage the configuration via USB or FTP. Do not manually access the configuration files unless instructed by technical support.

The configuration is only read at unit boot-up. Therefore, if a new configuration file is loaded, that unit must be rebooted for the new configuration take effect. Rebooting a unit can be performed by power-cycling the inverter in which the card is installed.

The embedded web server is customizable and is located in the “*WEB*” folder. All web page related items should reside in the “*WEB*” folder.

Interacting with the filesystem can be performed via USB (using a mini-B USB cable) as the interface card enumerates as a standard USB mass storage device “flash drive”. The file system can also be accessed via FTP if the card has compatible network settings. Users can 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.

Note that the USB and FTP connection will prevent the file system from being accessed by other interfaces, such as the web server. Therefore, USB and FTP should only be connected when performing maintenance and configuration. USB and FTP should be disconnected while the card is running normally in a production environment.

7.2 USB with Windows Explorer

To use Microsoft Windows Explorer, first open either “Windows Explorer” or “My Computer”. Refer to Figure 27. Note that the indicated procedure, prompts and capabilities outlined here can vary depending on such factors as the installed operating system and service packs.

The interface card will typically be displayed as a removable medium such as a Removable Disk. Refer to Figure 28. Windows Explorer will then display the filesystem’s contents (refer to Figure 29.) You can now perform normal file manipulation actions on the available files and folders (cut, copy, paste, open, rename, drag-and-drop transfers etc.) in the same manner as though you were manipulating any traditional file and folder stored on your computer’s hard drive.

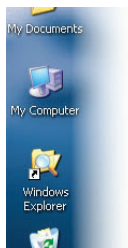


Figure 27: Accessing Windows Explorer

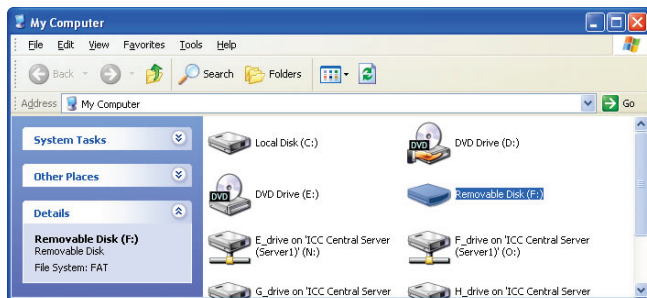


Figure 28: Removable Disk with Windows Explorer

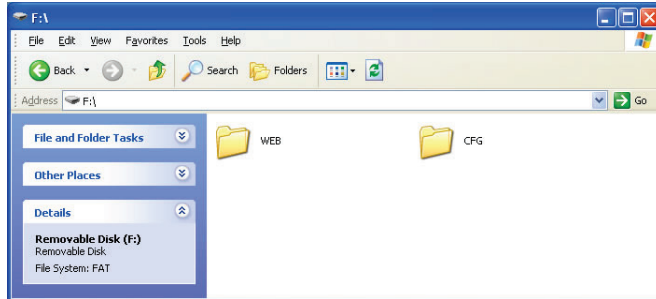


Figure 29: USB File Access via Windows Explorer

7.3 FTP With Windows Explorer

To use FTP with Microsoft Windows Explorer, first open either “Windows Explorer” or “My Computer”. 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://admin:admin@” and then the IP address of the target interface card (if the user name and password have been changed from its default, then replace the first “admin” with the new user name and the second “admin” with the password.) Refer to Figure 30.

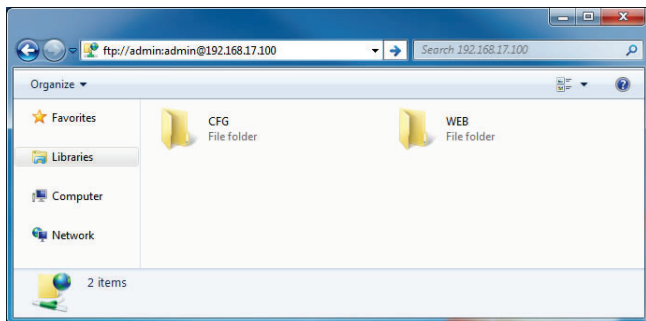


Figure 30: FTP via Windows Explorer

Note that the behavior of Windows Explorer FTP will vary from PC to PC. If you are having issues connecting FTP, there are other FTP client tools available such as Windows Command Prompt, Core FTP, FileZilla, SmartFTP etc. that can also be used to reliably access the card’s file system.

7.4 Loading New Web Server Content

The interface card's web server resides in the file system and can be updated in the field (refer to section 6.4). This section will discuss how to update the default web server. The update procedure similarly applies to a custom web server. Web server updates may be released for a variety of reasons, such as improvements and added functionality. When using the default web server, it is always recommended to use the latest release.

Treat web server updates independently of firmware updates since web server updates may or may not be related to firmware updates. The latest default web server can be downloaded from the [product web page](#). It is suggested that users first check this website during commissioning, and then periodically afterwards to determine if a new default web server has been released and is available to update their units.

Besides the new "WEB" folder containing the new web server, the update requires a USB or FTP connection as described earlier in this section. To update the web server, complete the following steps:

1. Navigate to the card's file system (see section 7.2 or section 7.3).
2. Backup the "WEB" folder if desired by copying it to the local computer.
3. Delete the "WEB" folder from the card's file system.
4. Copy the new "WEB" folder to the card's file system.
5. Power-cycle or reset the card.
6. Clear your internet browser's cache to ensure that the new web server content will be properly loaded from the interface card.

FIRMWARE

8.1 Overview

The interface card's embedded firmware 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. In order to ensure that the firmware update is successful, and in the interest of equipment and personnel safety, it is strongly recommended to stop all of the card's production activities prior to initiating the firmware update procedure. **Failure to follow the firmware update procedure could result in corrupt firmware!**

8.2 Update Procedure

Firmware update steps:

1. Always back up your configuration to a PC for later recovery if necessary.
2. Download and install the latest Configuration Studio, which can be obtained from the [product web page](#).
3. 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.
4. Ensure that the device is in a safe state prior to initiating the firmware update. The card may be temporarily inaccessible during the firmware update process.
5. Locally via USB: Connect a USB cable between the card and the PC and open the studio. If the studio contains newer firmware, it will automatically prompt you to update the firmware. Proceed with the firmware update.
6. Remotely via FTP: Connect an Ethernet cable and ensure that the card has compatible network settings.
7. Once the firmware update process has started, do not interrupt the card as this may corrupt the firmware. Do NOT manually power-cycle the inverter or reboot the card. Do NOT disturb the USB or Ethernet (FTP) connection.

8. After the firmware update has been completed, the card will reset automatically. When the card boots up again, it will be running the new application firmware, which can be confirmed by observing the version displayed in the **Device...Device Info** or the web server's information window (refer to section 6.2.1).
9. If new default web server content is available, load the new web server (refer to section 7.4).

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

9.1 Modbus/TCP

9.1.1 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 include:

- Supported Modbus/TCP functions are indicated in Table 19.
- The register mapping is provided in section 4.1.
- Inverter registers can be addressed as holding registers (4X references) and input registers (3X references).
- Specific bits within inverter registers can be accessed as either coils (0X references) or discrete inputs (1X references).
- Write data checking is not available (refer to section 4.2.) 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.
- The "unit identifier" (UI) field of the request packets is ignored and is echoed in the response.
- Modbus/TCP should not be confused with Modbus (serial) over TCP. Modbus over TCP is not compatible with Modbus/TCP and is not supported.

Table 19: Supported Modbus/TCP Functions

Function Code	Function	Modbus/TCP Class
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
8	Diagnostics (subfunction 0 only)	-
15	Force multiple coils	2
16	Write multiple registers	0

- The driver can be configured to detect a timeout (communication loss) and perform a timeout action.

9.1.2 Holding & Input Registers

The inverter registers by default are mapped as both holding registers (4X) and input registers (3X), and are accessed by using the inverter register numbers described in section 4.1. The 4X and 3X only serve as a naming convention for holding register and input register respectively, and should NOT be included as part of the actual on-the-wire register number. To further clarify, Modbus register 40201 is the same as Modbus holding register 201. The same description applies to input registers (3X).

For example, from a Modbus/TCP master's point of view, in order to access the output frequency (register 201) as a holding register, the Modbus/TCP master must execute the Read Multiple Registers function code and target register 201. This will similarly apply when accessing an inverter register as an Input Register.

9.1.3 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:

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

$$\text{register} = \left\lfloor \frac{\text{discrete} + 15}{16} \right\rfloor \quad \text{Equation 1}$$

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:

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

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 $\lfloor 3.0625 \rfloor = \lfloor 3 \text{ r}1 \rfloor = 3$. Then, using Equation 2, we can determine that the bit within register #3 that coil #34 targets is $(34-1)\%16 = 1$, as $33\%16 = \text{mod}(2 \text{ r}1) = 1$. Therefore, reading coil #34 will return the value of register #3, bit #1.

9.1.4 Connection Timeout Options

In the studio’s Project panel, navigate to **A8NETH-2P...Ethernet...Modbus/TCP Server**. The following configuration options will determine the actions to be taken if the connection is abnormally terminated or lost. 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. Note that a certain degree of caution must be exercised when using the timeout feature to avoid “nuisance” timeouts from occurring.

Enable Supervisory Timer

This timer provides the ability for the driver to monitor timeout occurrences on the overall receive activity for all connections.

- The timer will start after receiving the first request. Once the timer is started, it cannot be disabled.
- If the driver experiences no receive activity for more than the **Timeout** time setting, then the driver assumes that the client or network has experienced some sort of unexpected problem, and will perform the **Timeout Action**.

Enable Connection Timer

This timer provides the ability for the driver to monitor timeout occurrences and errors within the scope of each client connection.

- If a particular open socket experiences no activity for more than the **Timeout** time setting, then the driver assumes that the client or network has experienced some sort of unexpected problem, and will close that socket and perform the **Timeout Action**.
- If a socket error occurs (regardless of whether the error was due to a communication lapse or abnormal socket error), the driver will perform the **Timeout Action**. Specifically, do not perform inadvisable behavior such as sending a request from the client device, and then closing the socket prior to successfully receiving the server's response. The reason for this is because the server will experience an error when attempting to respond via the now-closed socket. Always be sure to manage socket life cycles "gracefully", and do not abandon outstanding requests.

Timeout

Defines the maximum number of milliseconds for a break in network communications before a timeout event will be triggered.

Timeout Action

Select an action from the drop down menu:

"None" No effect. The inverter will continue to operate with the last available settings.

"Apply Fail-safe Values" Apply the fail-safe values as described in section 5.4.1.

9.1.5 Node Settings

There are no node settings. A node is simply a container for objects.

9.1.6 Holding/Input Register Remap Settings

In the studio's **Project** panel, add **A8NETH-2P...Ethernet...Modbus/TCP Server...Holding/Input Register Remap**.

Holding/input register remap objects are **OPTIONAL**. By default, all inverter registers are already mapping as both holding (4X) and input (3X) registers (refer to section 9.1.2). For user convenience, register remap objects can be created to map any inverter register to holding/input register 5001 to 5050.

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 non-contiguous. For example, if it were desired to read the

inverter's output frequency (register 201), converter output voltage (register 208), and PID deviation value (register 254), 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 54 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 by making use of the register remap objects. 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.

Description

This 32-character (max) field is strictly for user reference: it is not used at any time by the driver.

Remap Register

Remap register that maps to the specified inverter register. Select from 5001 to 5050.

Register

Inverter register (refer to section 4) that is accessed by the **Remap Register**.

Data Type

Fixed to 16-Bit Unsigned. This is equivalent to two bytes.

9.2 EtherNet/IP

9.2.1 Overview

EtherNet/IP is a network adaptation of ODVA's Common Industrial Protocol (CIP). The card supports the EtherNet/IP server protocol, including the CSP server variant.

The interface card supports both implicit (class 1 I/O) and explicit (UCMM and class 3) messaging. Class 1 connections support two different types of I/O messaging. One type is the generic I/O assembly instances 100 and 150, which is entirely user-configurable (refer to section 9.2.5). The other type is the AC/DC drive profile assembly instances 20 & 70 or 21 & 71, which requires no user configuration (refer to section 9.2.6). With I/O messaging, the data field contains only real-time I/O data. The meaning of the data is pre-defined at the time the connection is established. I/O messages are short and have low overhead, and therefore minimize the processing time and allow for time-critical performance.

With explicit messaging (refer to section 9.2.7), 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 following sections demonstrate specific examples of how to use EtherNet/IP to transfer data between the inverter and Allen-Bradley Logix-brand PLCs.

Other notes of interest include:

- The interface card supports the EtherNet/IP protocol, as administered by the Open DeviceNet Vendor Association (ODVA).
- This product has been self-tested and found to comply with ODVA EtherNet/IP Conformance Test Software Version CT-13.
- The interface card's product type code is 2 (AC Drive).
- The EDS file can be obtained from the included CD-ROM or downloaded from the [product web page](#) on the internet.
- Supports DLR (Device Level Ring) node.
- Supports unconnected messages (UCMM), and up to 16 simultaneous class 1 (I/O) or class 3 (explicit) connections.
- 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 (communication loss) occurs, the driver can be configured to perform a timeout action. 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.

9.2.2 Server Settings

In the studio, navigate to **A8NETH-2P...Ethernet...EtherNet/IP Server**.

Device Name

The device name is used for identification of a device on the EtherNet/IP network. This string is accessible as the "product name" attribute of the identity object. Enter a string between 1 and 32 characters in length.

DLR

Device Level Ring is a ring redundancy protocol. All devices in a DLR ring must support DLR.

- If the checkbox is cleared (default setting), the card will not operate correctly in a DLR ring. By disabling this option, the card should not be installed in a DLR ring.
- If the checkbox is checked, the card can participate and will operate correctly in a DLR ring. By enabling this option, the card can be installed successfully in a DLR ring.

9.2.3 Connection Timeout Options

In the studio's Project panel, navigate to **A8NETH-2P...Ethernet...EtherNet/IP Server**. The following configuration options will determine the actions to be taken if the connection is abnormally terminated or lost. 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. Note that a certain degree of caution must be exercised when using the timeout feature to avoid "nuisance" timeouts from occurring.

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 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 5000.

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 defined in the EtherNet/IP specification. The driver allows the option of two different behavioral responses when a run/idle flag = Idle condition is received, depending on the state of the *Invoke Timeout When Run/Idle Flag = Idle* checkbox.

- If the checkbox is cleared (default setting), then the driver will maintain the last I/O data values received from the client. For example, if a device mapped to the database was being commanded to run prior to the run/idle flag being set to Idle by the client, then it will continue to run.
- If the checkbox is checked, then the driver will perform the **Timeout Action**.

Timeout Action

Select an action from the drop down menu:

"None" No effect. The inverter will continue to operate with the last available settings.

"Apply Fail-safe Values" Apply the fail-safe values as described in section 5.4.1.

9.2.4 Generic Class 1 I/O Produced and Consumed Data Settings

In the studio's **Project** panel, add **A8NETH-2P...Ethernet...EtherNet/IP Server...Produced I/O Data...Produced Data Word** and/or **Consumed I/O Data...Consumed Data Word**.

The Produced Data Word and Consumed Data Word objects are only applicable when connecting to assembly instances 100 and 150 (generic I/O), which is typically the case. The Produced Data Word defines the structure of status data sent from the inverter back to the controller. The Consumed Data Word objects will define the structure of the command data sent from the EtherNet/IP controller (for example, a ControlLogix PLC) to the inverter. These objects allow the creation of custom-built I/O data. Up to 32 “command” register values can be sent to the inverter, and up to 32 “status” register values can be sent back to the controller. Therefore, up to 32 Produced and 32 Consumed Data Word objects can be created. If a consumed word offset is not defined, that data will be ignored by the inverter. If a produce word offset is not defined, the value will default to 0. The size of the actual I/O produced and consumed data is determined by the client upon initial connection establishment. Since a data word utilizes 2 bytes, the size must be an even number of bytes. The I/O data format is summarized in Table 20.

Description

This 32-character (max) field is strictly for user reference: it is not used at any time by the driver.

Produced Data Word Offset

The value from the associated inverter register will populate this word offset of the produced data that is to be sent to the client. It is recommend to start at word offset 0.

Consumed Data Word Offset

The consumed data received from the client at this word offset will contain the value to be written to the associated inverter register. It is recommend to start at word offset 0.

Register

The inverter register (refer to section 4) associated with the word offset. For the Produced Data Word object, enter a “status” register to be monitored. For the Consumed Data Word object, enter a “command” register that can be written.

Data Type

Each data word is fixed to 16-Bit Unsigned. This is equivalent to two bytes.

Table 20: EtherNet/IP User-Configurable I/O Data Format

Consumed Data (PLC to Inverter)		Produced Data (Inverter to PLC)	
Word Offset	Register	Word Offset	Register
0	Any	0	Any
1	Any	1	Any
:	Any	:	Any
30	Any	30	Any
31	Any	31	Any

The default I/O configuration is described in Table 21. Always use the studio to confirm the configuration before commissioning the device.

Table 21: EtherNet/IP Default User-Configurable I/O Data Format

Consumed Data (PLC to Inverter)		Produced Data (Inverter to PLC)	
Word Offset	Register	Word Offset	Register
0	1	0	1227
1	1229	1	1231
:	None	:	None

9.2.5 Generic Class 1 (I/O) Connection Access

Clients may access the class 1 endpoint by opening a connection to assembly instances 100 and 150. The structure of I/O consumed and produced data for this assembly instance pair is entirely user-configurable (refer to section 9.2.4). The generic class 1 I/O connection is mutually exclusive of the AC/DC drive profile class 1 I/O connection. For a generic class 1 I/O application example, refer to sections 9.2.11.1 and 9.2.12.

9.2.6 AC/DC Drive Profile Class 1 (I/O) Connection Access

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 structure of I/O consumed and produced data for the AC/DC drive profile class 1 I/O is predefined and fixed (refer to Table 23 and Table 24) to 4 input bytes and 4 output bytes. It is highly recommended to complete the reading of this section to understand the data mapping and the implications of using the AC/DC drive profile. Note that when using the AC/DC drive profile class 1 I/O, the produced word and consumed word configuration do not apply (refer to section 9.2.4). For an AC/DC drive profile class 1 I/O application example, refer to sections 9.2.11.2 and 9.2.12.2.

The AC/DC drive profile implementation provides support for several required CIP objects, which are specified in Table 22. 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 22: AC/DC Drive Profile-Related Objects

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

Table 23: 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 1, bit 1 (EnableOp).

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

Fault Reset: Inverter reset command (0=no action, 0→1 rising edge=reset fault). Maps to inverter register 1, bit 11 (ResetMalfunction).

NetCtrl: Run/stop control source selection (0=local control, 1=network control). Maps to inverter register 1, bit 15 (NetCtrl).

NetRef: Speed reference source selection (0=local control, 1=network control). Maps to inverter register 1, bit 14 (NetRef).

Speed Reference: Inverter speed reference in RPM. Maps to inverter register 1229 (SetpointSpeed). The speed reference component of the AC/DC drive profile output instances is always in units of RPM. The interface card will pass the consumed speed reference value directly on to the inverter as-is.

Table 24: 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		Faulted
	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	Faulted
	1	Drive State							
	2	Speed Actual (Low Byte)							
	3	Speed Actual (High Byte)							

Mapping Detail

Faulted: Inverter fault signal (0=not faulted, 1=faulted). Maps to inverter register 1227 (DriveStatus), bit 3 (Faulted).

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 1227 (DriveStatus), bit 2 (OpEnabled).

Running2 (REV): Running reverse status signal (0=not running reverse, 1=running reverse). Maps to inverter register 1227 (DriveStatus), bit 14 (RevOpEnabled).

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 1227 (DriveStatus) bit 9 (CtrlFromNet).

RefFromNet: Inverter speed reference status (0=speed reference is not from network, 1=speed reference is from network). Maps to inverter register 1227 (DriveStatus) bit 8 (RefFromNet).

AtReference: Up-to-speed signal (0=not up-to-speed, 1=up-to-speed). Maps to inverter register 1227 (DriveStatus) bit 10 (SetpointReached).

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 1231 (ActualSpeed).

9.2.7 Explicit Messaging Via Get/Set Attribute Single Services

Get attribute single (0x0E) and set attribute single (0x10) are common services that can access the inverter registers by specifying the appropriate class code, instance number and attribute identifier. The class code is 0xA2. The instance number is the targeted register number (refer to section 4.1). The attribute identifier is 1, which is the 16-bit value of the register being accessed.

For example, a get attribute single request with a class code of 0xA2, instance number of 201, and attribute identifier of 1 will return the 16-bit value of inverter register output frequency.

9.2.8 Explicit Messaging Via Data Table Read/Write Services

Data table read (0x4C) and data table write (0x4D) services provide a direct method of accessing the inverter registers by reference to “tag names”. Tags are read via the EtherNet/IP “data table read” service, and written via the EtherNet/IP “data table write” service. Note that an underscore (“_”) is used to separate fields in the tag names.

To read data, the client must reference a starting “source element” and the “number of elements” to read. Similarly, to write data, the client must reference a starting “destination element” and the “number of elements” to write. The “number of elements” can be any quantity from 1 to the maximum allowable length, while the “source element” and “destination element” must be tag names constructed according to the naming conventions shown in section 9.2.9. The elements are 16-bit values.

9.2.9 Inverter Register Access Tag Format

Any inverter register (refer to section 4) 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:

[reg]_[register number]

Where

[reg] is just the 3-character sequence “reg”.

[register number] is a 1- to 4-character field (“1”...“201”...“1000”, “1001”... etc.) corresponding to the inverter register number.

Examples

“Output voltage” (register #203)	reg_203
“DriveStatus” (register #1227)	reg_1227
“SetpointSpeed” (register #1229)	reg_1229
“Acceleration time” (<i>Pr. 7</i> / register #2007)	reg_2007

For explicit messaging examples, refer to sections 9.2.13 and 9.2.14.

9.2.10 ControlLogix Examples: Setup

This section will demonstrate how to initially setup a ControlLogix PLC (such as a 1756-L61) coupled with a 1756-ENBT 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-ENBT 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) Select the “1756-ENBT”, and click “Create”. Refer to Figure 31.

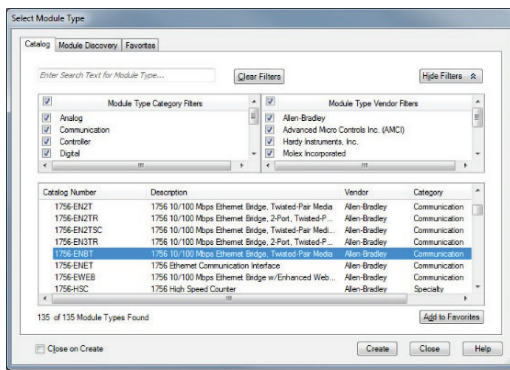


Figure 31: Adding a New Module

- 6) The “New Module” window will open. Refer to Figure 32.

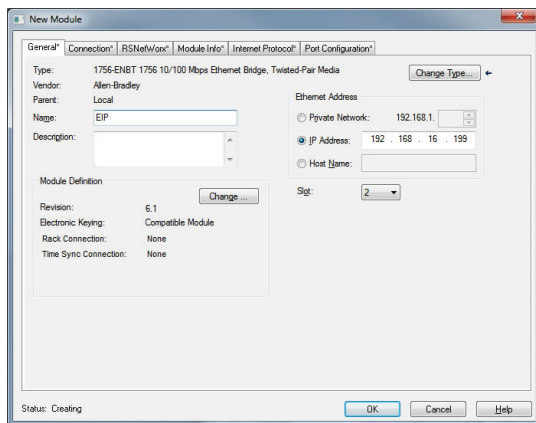


Figure 32: Identifying the New Module

- 7) Assign the Ethernet module a name (we will use “EIP”) and an IP address, deselect “Open Module Properties”, and click OK.
- 8) Download the configuration.
- 9) Switch to online mode. Right click on the 1756-ENBT module in the I/O Configuration and choose “Properties”.
- 10) Select the Internet Protocol tab from the Module Properties dialog box and confirm that the IP Settings are configured correctly.

9.2.11 ControlLogix Example: EDS Add-On Profile (AOP)

This section will demonstrate how to setup and use an EtherNet/IP I/O connection via EDS Add-On Profile. This example only applies to RSLogix5000 V20 (and later) that support EDS Add-On Profile. Otherwise, refer to the I/O examples in section 9.2.12. This section must be completed prior to attempting any of the following AOP example(s).

EtherNet/IP I/O messaging allows the inverter's parameter 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) Register the interface card's EDS file. In the menu bar, navigate to Tools...EDS Hardware Installation Tool. Refer to Figure 33.

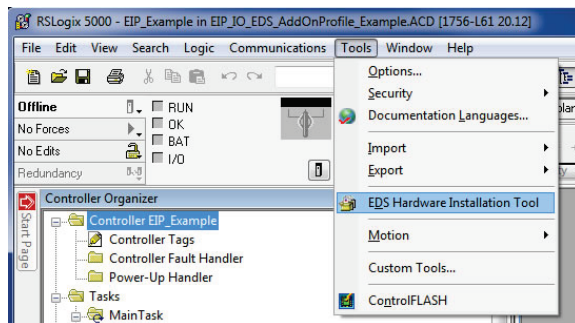


Figure 33: EDS Hardware Installation Tool Menu

- 2) This will start the "EDS Wizard". Click "Next".
- 3) Select "Register an EDS file(s)" and click "Next".
- 4) The "Registration" dialog will appear. Refer to Figure 34. Click "Browse", select the interface card's EDS file, and click "Next".

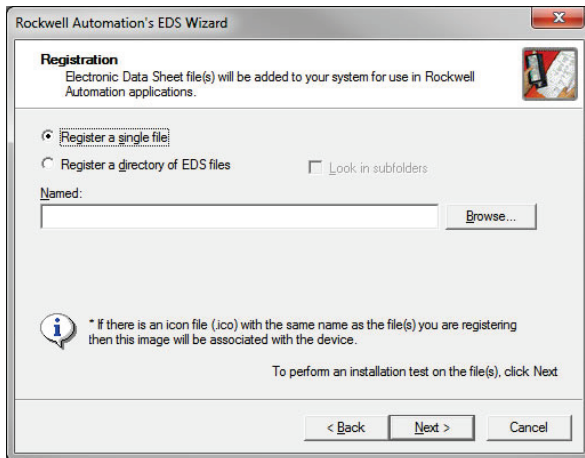


Figure 34: EDS Registration

- 5) Ensure that there are no errors in the test results. Click "Next".
- 6) A graphic image of the interface card is displayed. Click "Next".
- 7) The task summary will list the interface card as the device to register. Click "Next".
- 8) "You have successfully completed the EDS Wizard". Click "Finish".
- 9) The interface card is now available as a module.
- 10) Right click on the 1756-ENBT/A node under the "I/O Configuration" in the controller organizer view and choose "New Module..."

11) Find the interface card in the “Select Module” dialog box as shown in Figure 35.

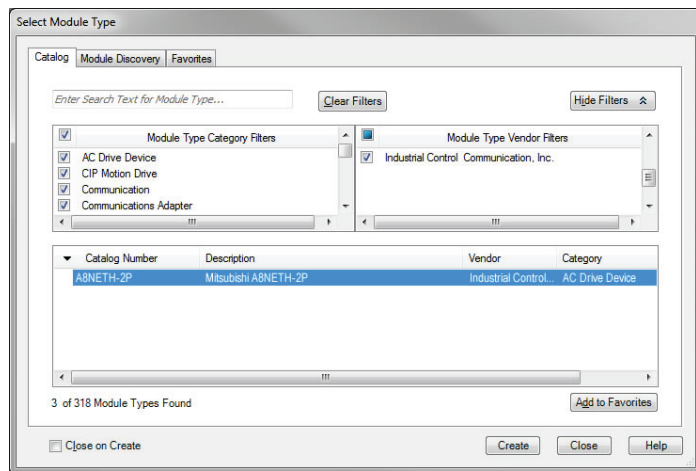


Figure 35: Adding a New Interface Card Module

12) The “New Module” properties dialog box will open as shown in Figure 36.

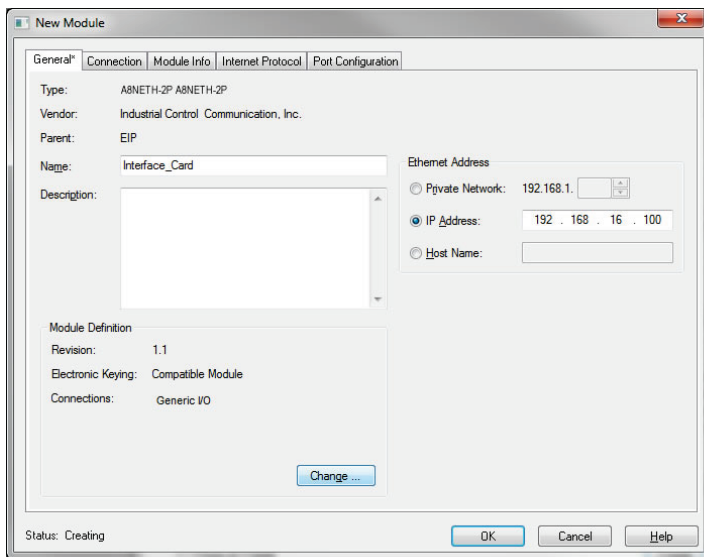


Figure 36: AOP Interface Card Module Properties

13) Click on the “Connection” tab. Refer to Figure 37.

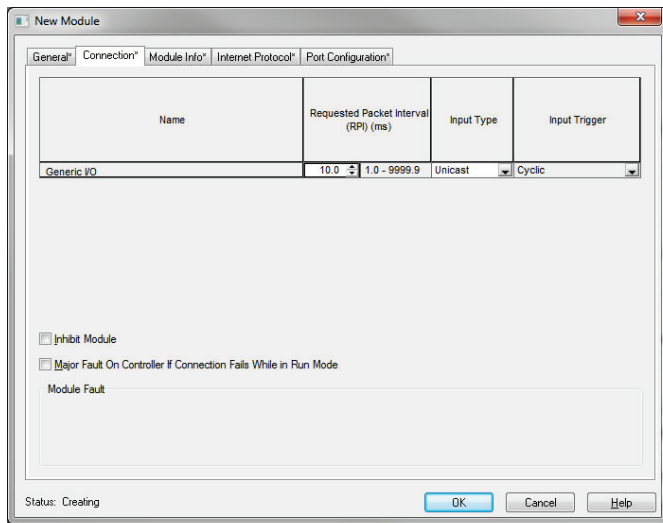


Figure 37: AOP New Module Properties Connection Tab

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 1ms. Click “OK” when done.

- 14) You should now see the interface card in the 1756-ENBT/A branch under the I/O Configuration in the controller organizer view. The full I/O Configuration tree should appear similar to Figure 38.

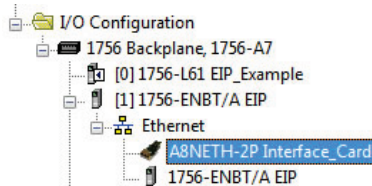


Figure 38: AOP Interface Card I/O Configuration

- 15) Continue with the AOP Generic I/O Messaging example in section 9.2.11.1 or AOP AC/DC Drive Profile example in section 9.2.11.2.

9.2.11.1 ControlLogix Example: EDS Add-On Profile (AOP) Generic I/O Messaging

This section will demonstrate how to configure the EtherNet/IP Generic I/O connection.

- 1) Complete all steps in section 9.2.11.
- 2) Locate the interface card in the 1756-ENBT/A branch under the “I/O Configuration” in the controller organizer view. Right click on the interface card, choose “Properties”, and select the “General” tab.
- 3) Configure the Generic I/O connection. Refer to Figure 39.

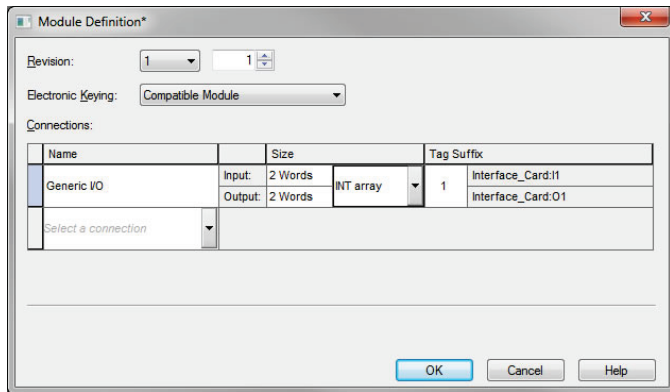


Figure 39: AOP Generic I/O Module Definition

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

Name: In this example, select Generic I/O.

Size: Because all inverter data is stored as 16-bit function codes, change the data type to “INT array”.

Input: The Input is the collection of monitor data that is produced by the interface card and is received as an input to the PLC. Its structure is defined by the Produced Data Configuration as described in section 9.2.4. The Input Size must be set to the number of 16-bit function codes that we wish to receive from the interface card. For the purposes of this example, we are assuming that the default produced data word configuration, with 2 relevant registers (1 and 1229). We therefore set the Input Size to 2 Words.

Output: The Output 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 Data Configuration as described in section 9.2.4. The Output Size must be set to the number of 16-bit function codes that we wish to send to the interface card.

For the purposes of this example, we are assuming that the default consumed data word configuration, with 2 relevant registers (1227 and 1231). We therefore set the Output Size to 2 Words.

When done, click “OK”.

- 4) 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. Also confirm that the interface card’s “Network Status” LED should be solid green, indicating an “online/connected” state.
- 5) By double-clicking “Controller Tags” in the project tree, it is possible to view the newly-added tags. The Interface_Card:I tag allows viewing of the input data, and the Interface_Card: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 tags in order to control and monitor the inverter.

9.2.11.2 ControlLogix Example: EDS Add-On Profile (AOP) AC/DC Drive Profile

This section will demonstrate how to configure the EtherNet/IP AC/DC drive profile I/O connection.

- 1) Complete all steps in section 9.2.11.
- 2) Locate the interface card in the 1756-ENBT/A branch under the “I/O Configuration” in the controller organizer view. Right click on the interface card, choose “Properties”, and select the “General” tab.
- 3) Configure the AC/DC Drive Profile connection. Refer to Figure 40.

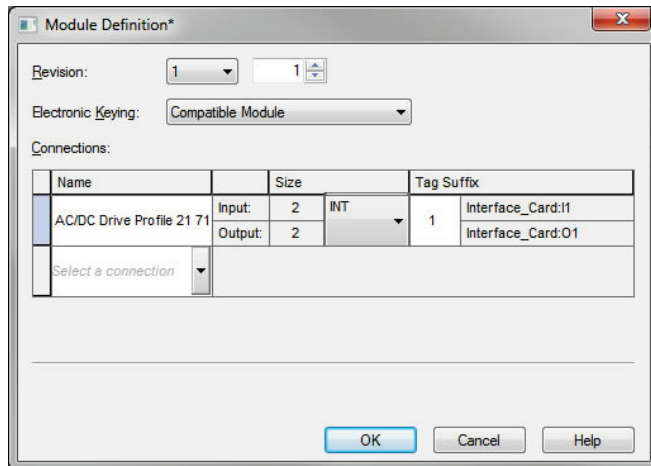


Figure 40: AOP AC/DC Drive Profile Module Definition

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

Name: In this example, select AC/DC Drive Profile 21 71.

Size: Because all inverter data is stored as 16-bit function codes, change the data type to “INT array”.

When done, click “OK”.

- 4) 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. Also confirm that the interface card's "Network Status" LED should be solid green, indicating an "online/connected" state.

- 5) By double-clicking "Controller Tags" in the project tree, it is possible to view the newly-added tags. The Interface_Card:I tag allows viewing of the input data, and the Interface_Card: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 tags in order to control and monitor the inverter. The AC/DC drive profile I/O data is described in section 9.2.6.

9.2.12 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 or 20 & 70 or 20 & 71. Do not complete this example if you are using EDS AOP, in which case, please refer to the AOP examples. EtherNet/IP I/O messaging allows the inverter's function codes 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-ENBT 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 “Create”. Refer to Figure 41.
- 4) The module properties dialog box will open (refer to Figure 42). Enter a Name 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 structure is defined by

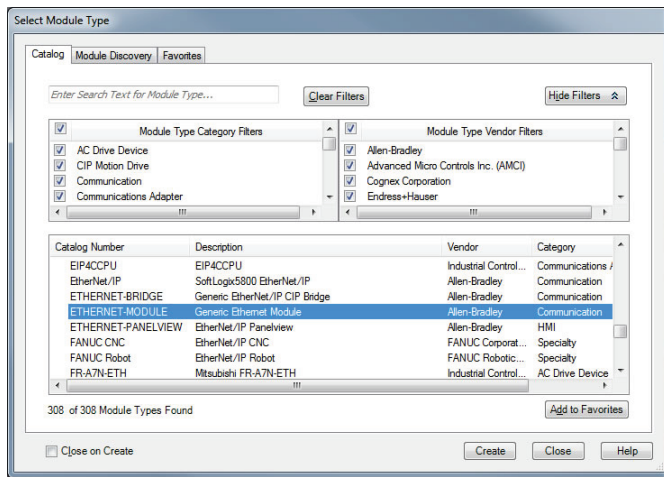


Figure 41: Adding a New Generic Ethernet Module

the Produced Register Configuration as described in section 9.2.4. The Input Assembly Instance must be set to 150 when connecting to the generic 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 default produced data word configuration, with two relevant registers (1227 and 1231). 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 9.2.4. The Output Assembly Instance must be set to 100 when connecting to the generic 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 default consumed data word configuration, with two relevant registers (1 and 1229). We therefore set the Output Size to 2.

Configuration: The Configuration Assembly Instance is unused, and its instance number and size are therefore irrelevant (you can just enter “1” and “0”, respectively).

When done, click “OK”.

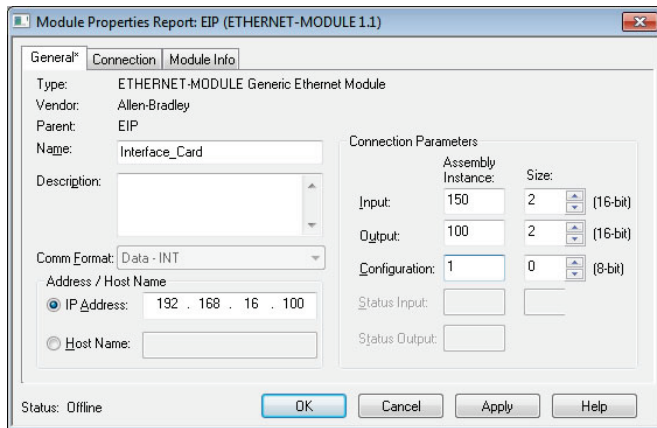


Figure 42: Interface Card Module Properties

- 5) You should now see the new module (named “ETHERNET-MODULE Interface_Card”) in the 1756-ENBT 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 43.

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 1ms. 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 44.

- 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 45. 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 46. The Interface_Card:C configuration tag is unused, the Interface_Card:I tag allows viewing of the input data, and the Interface_Card:O tag allows modification of the output data. These tags will be synchronized with the inverter at the rate that was established for the module’s RPI.

We can directly interact with these tags in order to control and monitor the inverter. In Figure 46, for example, we can see that the first 16-bit word of

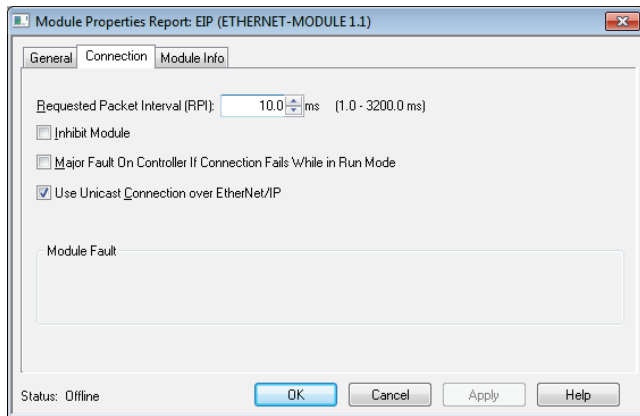


Figure 43: Module Properties Connection Tab

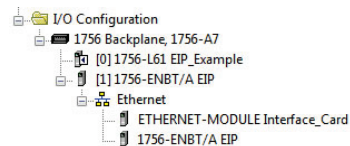


Figure 44: I/O Configuration Tree

output data (Interface_Card:O.Data[0]) has been set to a hexadecimal value of 0xC002. The default consumed data word configuration word offset 0 references register 1, which is the inverter's command register. A value of 0xC002, therefore, means that the NetCtrl, NetRef, and EnableOp bits have been turned ON.

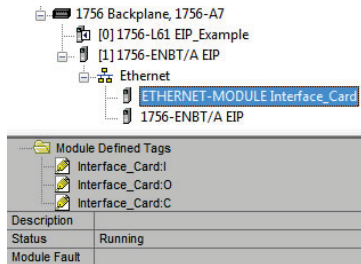


Figure 45: Online Module Status

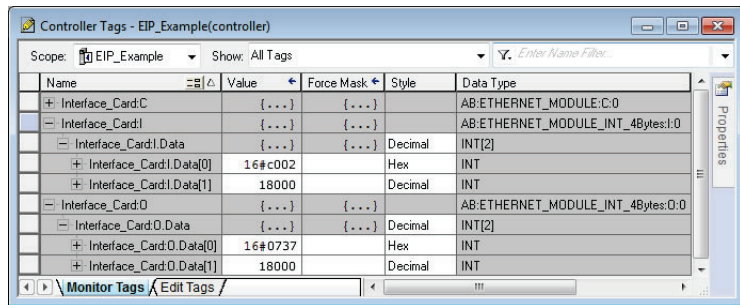


Figure 46: Controller Tags for I/O Access

Similarly, we can see that the second 16-bit word of output data (Interface_Card:O.Data[1]) has been set to a decimal value of 18000. The default consumed data word configuration word offset 1 references register 1229, which is the inverter's SetpointSpeed register. A value of 18000, therefore, equates to a speed command of 1800.0 RPM.

The input data from the inverter shows similar expected results. Values of 0x0737 and 18000 corresponding to registers 1227 (DriveStatus) and 1231 (ActualSpeed), respectively, are consistent with the inverter running at the parameter values commanded by the output tag.

9.2.12.1 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 (refer to section 9.2.4). The add-on instruction is optional and provided for user convenience.

- 1) Complete all the steps in section 9.2.12.
- 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 47.
- 3) Double click “Controller Tags” in the controller organizer view and select the “Edit Tags” tab at the bottom.
- 4) Create the tags shown in Figure 48.

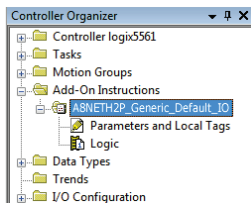


Figure 47: Generic Default IO Add-On Instruction

The screenshot shows the 'Controller Tags - logix5561(controller)' window. The 'Scope' is set to 'logix5561' and 'Show: All Tags'. The table below lists the tags created:

Name	Value	Force Mask	Style	Data Type	Description	Constant
enable	0		Decimal	BOOL	Enable operation	<input type="checkbox"/>
enable_rev	0		Decimal	BOOL	Enable reverse o...	<input type="checkbox"/>
generic_aoi	{...}	{...}		ASNETH2P_Generic_Default_IO	Simple interface f...	<input type="checkbox"/>
Interface_CardC	{...}	{...}		AB.ETHERNET_MODULE.C:0		<input type="checkbox"/>
Interface_CardI	{...}	{...}		AB.ETHERNET_MODULE_INT_4Bytes:I:0		<input type="checkbox"/>
Interface_CardO	{...}	{...}		AB.ETHERNET_MODULE_INT_4Bytes:O:0		<input type="checkbox"/>
net_ctl	0		Decimal	BOOL	Request drive co...	<input type="checkbox"/>
net_ref	0		Decimal	BOOL	Request speed re...	<input type="checkbox"/>
quick_stop	0		Decimal	BOOL	Quick stop	<input type="checkbox"/>
reset	0		Decimal	BOOL	Reset fault on po...	<input type="checkbox"/>
speed_cmd	0		Decimal	INT	Speed command ...	<input type="checkbox"/>

Figure 48: 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 49.
- 9) Click OK.
- 10) Edit the add-on instruction according to Figure 50.

- 11) The program is now complete.
- 12) Save, download and run the program.

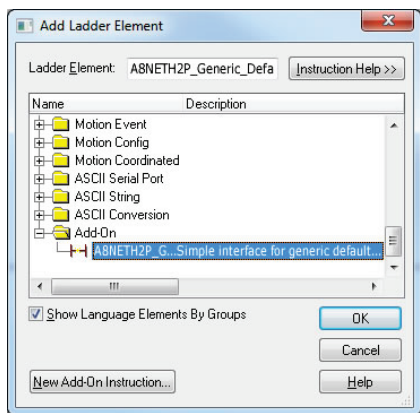


Figure 49: Add Generic Default Add-On Instruction

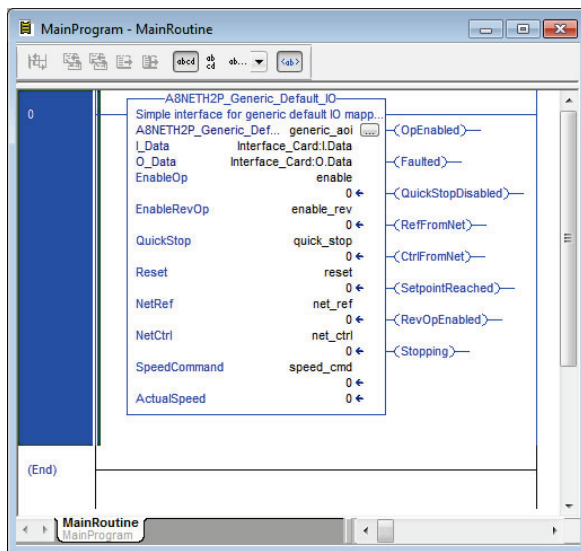


Figure 50: Configure Generic Default AOI

9.2.12.2 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. The add-on instruction is optional and provided for user convenience.

- 1) Complete all the steps in section 9.2.12. 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 51.

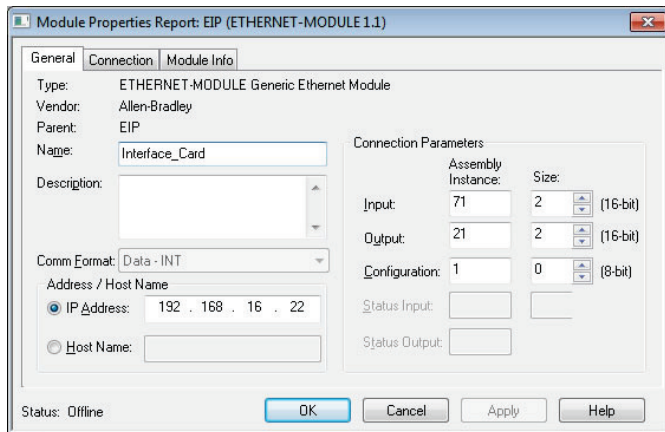


Figure 51: AC/DC Drive Profile Generic Ethernet Module Configuration

- 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 52.
- 3) Double click "Controller Tags" in the controller organizer view and select the "Edit Tags" tab at the bottom.

- 4) Create the tags shown in Figure 53.
- 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 54.
- 9) Click OK.

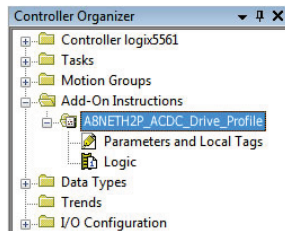


Figure 52: AC/DC Drive Profile Add-On Instruction

Name	Alias For	Base Tag	Data Type	Description	External Access	Constant
acdc_aoi			ABNETH2P_ACDC_Drive_Profile	Interface for AC/DC Drive Prof...	Read/Write	<input type="checkbox"/>
Interface_CardC			AB:ETHERNET_MODULE_C:0		Read/Write	<input checked="" type="checkbox"/>
Interface_CardI			AB:ETHERNET_MODULE_INT_4Bytes:1:0		Read/Write	<input checked="" type="checkbox"/>
Interface_CardO			AB:ETHERNET_MODULE_INT_4Bytes:0:0		Read/Write	<input checked="" type="checkbox"/>
net_ctrl			BOOL	Control from network	Read/Write	<input type="checkbox"/>
net_ref			BOOL	Speed reference from network	Read/Write	<input type="checkbox"/>
reset			BOOL	Fault reset	Read/Write	<input type="checkbox"/>
run_fwd			BOOL	Run forward direction	Read/Write	<input type="checkbox"/>
run_rev			BOOL	Run reverse direction	Read/Write	<input type="checkbox"/>
speed_ref			INT	Speed reference (RPM)	Read/Write	<input type="checkbox"/>

Figure 53: Create AC/DC Drive Profile AOI Tags

- 10) Edit the add-on instruction according to Figure 55.
- 11) The program is now complete.

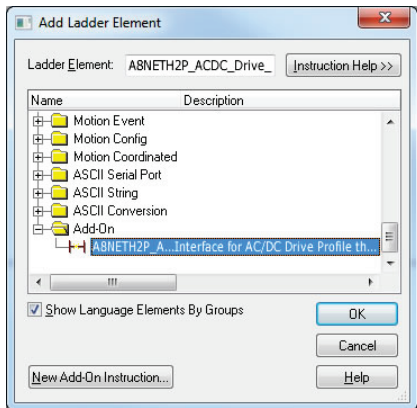


Figure 54: Add AC/DC Drive Profile Add-On Instruction

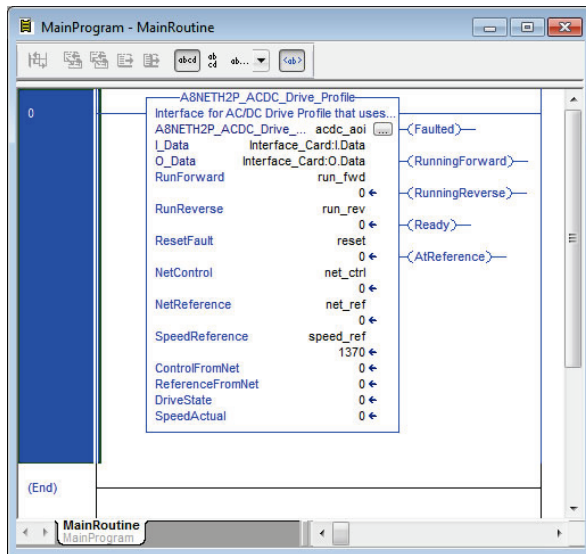


Figure 55: Configure AC/DC Drive Profile AOI

- 12) Save, download and run the program.

9.2.13 ControlLogix Example: Read Registers

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) Create new Tags.

- a) Double click “Controller Tags” in the controller organizer view.
- b) The “Controller Tags” window appears. Refer to Figure 56.
- 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.

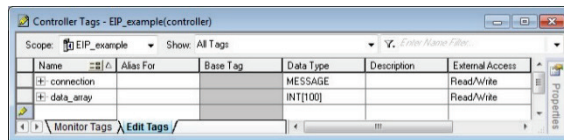


Figure 56: Create New Tags

- 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 3000000 (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.
- 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[100]” in the Data Type field. This tag is an array of INTs that will be able to hold up to 100 16-bit registers from the inverter. Always make sure that the destination tag size is large enough to hold all elements to be read.

2) Add a MSG instruction to the main program.

- Double click “MainRoutine” under Tasks ...MainTask ...MainProgram in the controller organizer view.
- Right click on the first ladder logic rung in the MainRoutine window and select “Add Ladder Element...”
- The “Add Ladder Element” window appears.
- Select the “MSG” instruction in the Input/Output folder. Refer to Figure 57.
- Click OK.

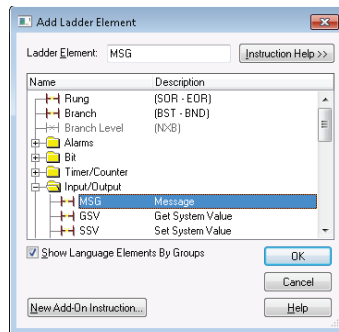


Figure 57: Adding a MSG Instruction

3) Add an XIO element to the main program.

- Right click on the ladder logic rung containing the MSG instruction in the MainRoutine window and select “Add Ladder Element...” again.
- The “Add Ladder Element” window appears.
- Select the “XIO” element in the Bit folder. Refer to Figure 58.
- Click OK.

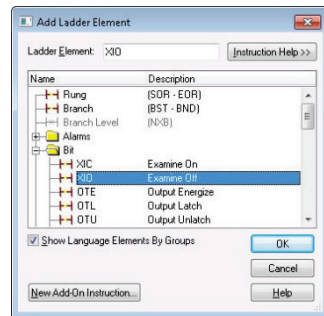


Figure 58: Adding an XIO Element

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 59.
- b) Click the message configuration button (“...”) in the MSG instruction. The “Message Configuration” window will open. Refer to Figure 60.
- 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 9.2.7). In this example, we will be reading a total of 25 registers beginning at reg_201 (output frequency).
 - iii) Enter the Number Of Elements to read. In this example, we will read 25 registers.
 - iv) For the Destination Element, select “data_array”.
- d) “Communication” tab settings (refer to Figure 61):

- 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 9.2.10),
 - 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.

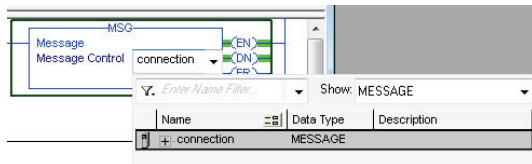


Figure 59: MSG Instruction Tag Assignment

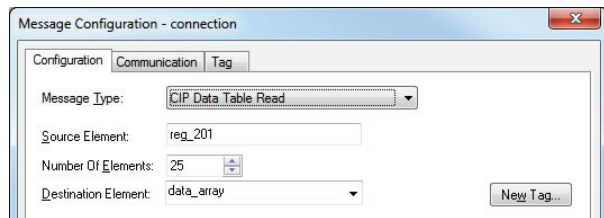


Figure 60: MSG Instruction Configuration

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

- 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 62.

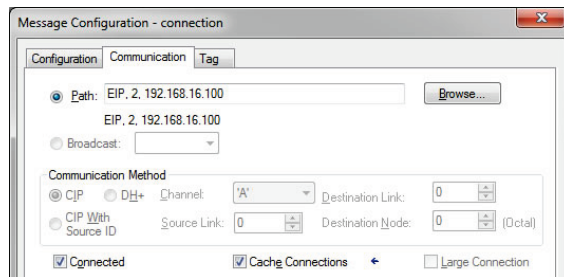


Figure 61: Setting the Communication Path

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 63. 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 instruction at a specific rate or under specific conditions.

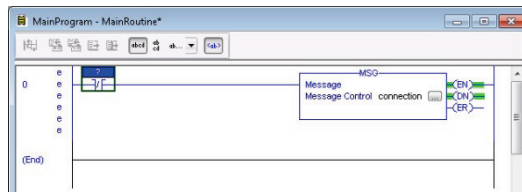


Figure 62: MainRoutine

6) The program is now complete. Refer to Figure 64.

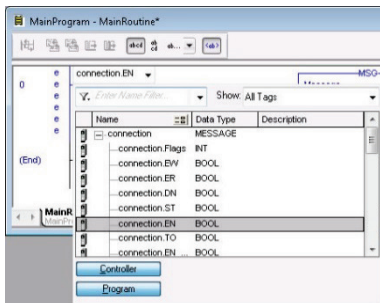


Figure 63: Configure XIO Element



Figure 64: Complete Program

7) Save, download and run the program.

- To view the values of the registers being read from the interface card, double-click “Controller Tags” in the controller organizer view.
- Select the “Monitor Tags” tab and expand the data_array tag.
- 25 register values starting at register #201 are being continuously read from the interface card and placed in the 25 sequential offsets of data_array starting at the 0th offset.

9.2.14 ControlLogix Example: Reading and Writing Multiple MSG Instructions

Often times, applications may need to both read data from and write data to the inverter. To accomplish this task, multiple MSG instructions will need to 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 controller tag.

Figure 65 shows an example of four MSG instructions, two for reading and two for writing. It is evident from this logic that “rd_connection_1” and “rd_connection_2” are the two independent message controller tags created for the read instructions. Note the addition of the en_wr_xx XIC elements for the write logic. 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_wr_xx 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.

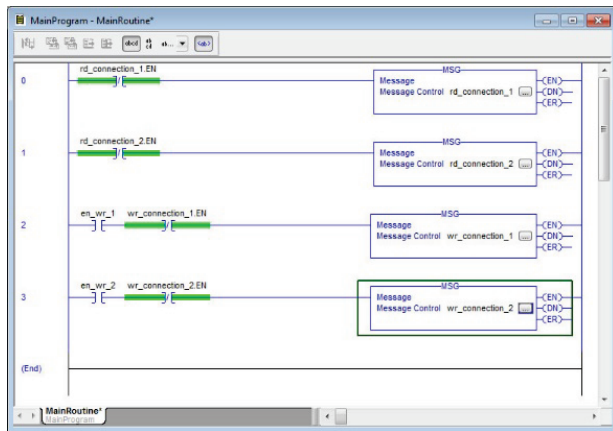


Figure 65: Reading and Writing Multiple MSG Instructions

Figure 66 shows the configuration details of the example wr_connection_xx 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. The value of the controller tag specified in the Source Element will be written to the inverter register specified in the Destination Element.

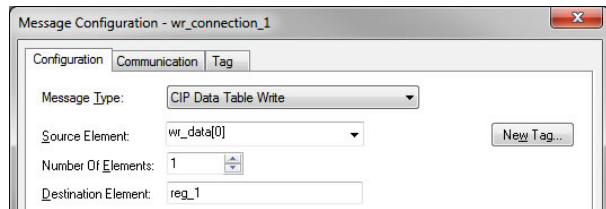


Figure 66: MSG Configuration for Writing

Note that when writing data via explicit messaging, use caution to ensure that the commanded function codes 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 function codes. In other words, if the I/O messaging example procedure detailed in section 9.2.12 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.

9.3 Allen Bradley CSP (PCCC)

9.3.1 Overview

Ethernet-enabled Allen-Bradley legacy PLCs (such as the PLC5E, SLC-5/05, and MicroLogix 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. Note that CSP runs under EtherNet/IP and is enabled by default when EtherNet/IP is added to the configuration.

If a connection timeout or socket-level error occurs, the driver can be configured to perform a timeout action as described in section 9.2.3.

9.3.2 Explicit Messaging Via Typed Read/Write Services

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, 0x68) service, and writing is performed via the CSP “PLC5 Write” (DF1 protocol typed write, 0x67) service. To read and write data, the client must reference a “target address” and the “size of elements”. The target address is constructed according to the conventions shown in section 9.3.3.

9.3.3 Inverter Register File Number Offset Format

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

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

In Equation 3, “target register” $\in [1 \dots 3299]$, “file number” $\in [10 \dots 42]$ (which means N10...N42), and “offset” is restricted only by the limitations of the programming software (but is a value of 3299 max). Refer to section 4.1 for the register list. Table 25 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 25: CSP Target Register Examples

Target Register	File/Section Number	Offset/Element	Address Format
1	N10	1	N10:1
7	N10	7	N10:7
201	N12	1	N12:1
254	N12	54	N12:54
1224	N22	24	N22:24
3299	N42	99	N42:99

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 data word configuration (refer to section 9.2.4). 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 data word configuration. By appropriate configuration of the EtherNet/IP consumed and produced data word configuration, 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 data word configurations are comprised of 32 register definitions, the targeted “offset/element” must be within the range of 0 to 31 inclusive. Refer to Table 26 for some examples of N50 accesses.

Table 26: Examples of EtherNet/IP-Style Bulk Access via File N50

File/Section Number	Offset/Element	Address Format	Start Target Register of Configuration Array	Max Number of Accessible Elements
N50	0	N50:0	1st	32
N50	:	:	:	:
N50	15	N50:15	16th	16
N50	:	:	:	:
N50	31	N50:31	32nd	1

9.3.4 SLC-5/05 Example: Read Registers

This example program will show how to continuously read a block of registers from the inverter with a single MSG instruction. This action is performed via the Typed Read (a.k.a. “PLC5 Read”) message type. Only one read request is outstanding at any given time. Note that the steps for the MicroLogix and PLC5E may vary slightly, but in general are similar.

- 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 67).
 - 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 number of elements (e.g. 200). Refer to Figure 68. Click OK to create the file.
- 3) **Add a MSG instruction to the program.**
 - a) If not already visible, double-click “LAD2” under Project...Program Files in the controller organizer view to bring up the ladder logic program.

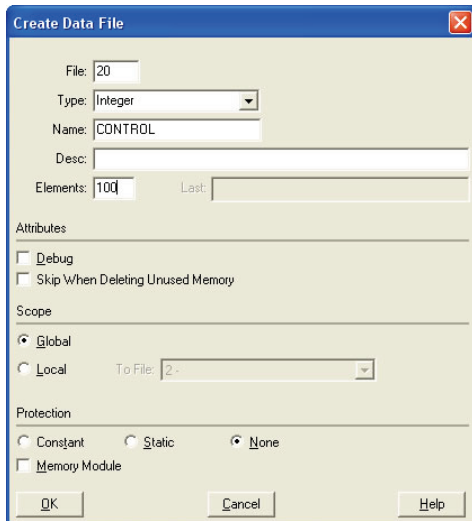


Figure 67: Creating a Control File

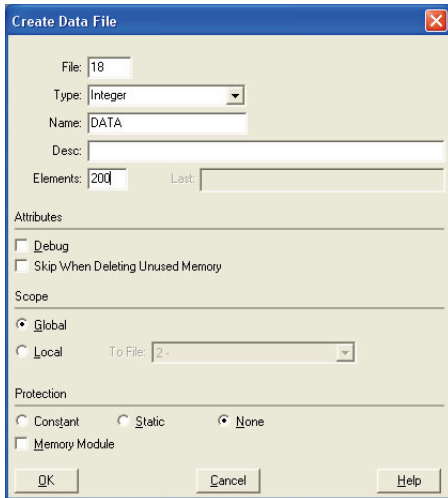


Figure 68: Creating a Data File

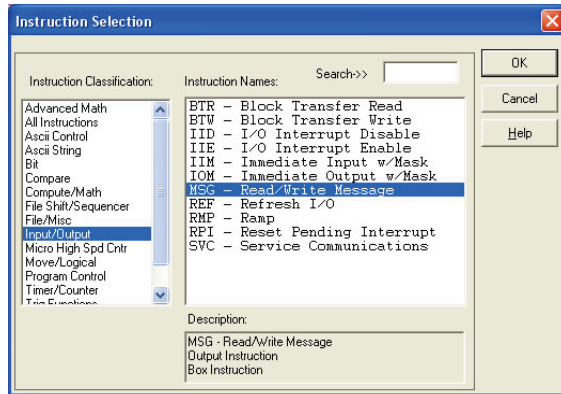


Figure 69: MSG Instruction Selection

- b) Right click on the default rung number on the left-hand side of the LAD2 window and select “Insert Rung”.
 - c) Right click on the rung number of the new editable rung and select “Append Instruction”.
 - d) Select the “MSG” instruction from the “Input/Output” classification, then click OK. Refer to Figure 69.
- 4) Add an XIO element to the program.**
- a) Right click on the rung number of the rung currently being edited and select “Append Instruction” again.

- b) Select the “XIO” instruction from the “Bit” classification, then click OK. Refer to Figure 70.

5) Configure the MSG instruction.

- a) Set the “Read/Write” field to “Read”, “Target Device” field to “PLC5”, “Local/Remote” field to “Local”, and “Control Block” to “N20:0”.
- b) 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 71.
- 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 72.
- f) Close the dialog box. At this point, the program should appear as shown in Figure 73.

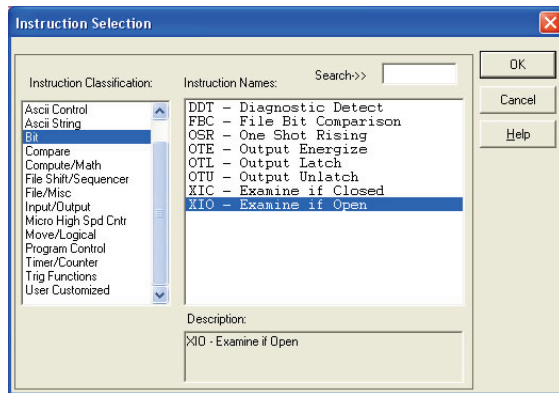


Figure 70: XIO Instruction Selection

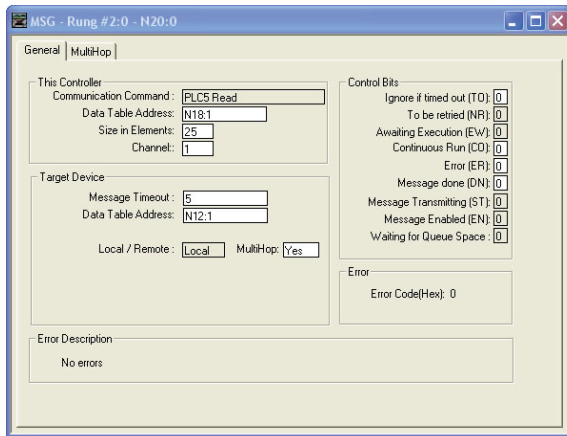


Figure 71: MSG Configuration, "General" Tab

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 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.

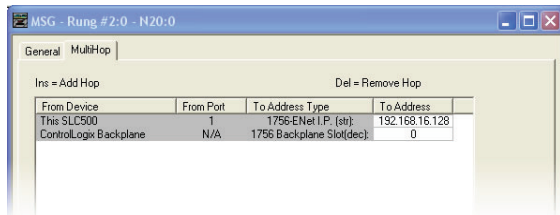


Figure 72: MSG Configuration, "MultiHop" Tab

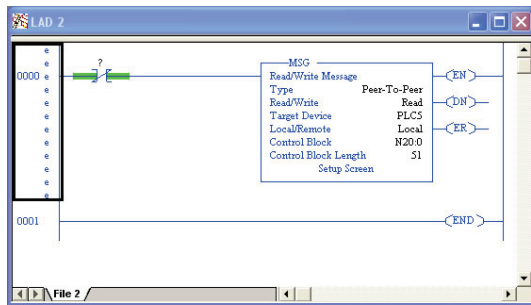


Figure 73: PLC Program after MSG Instruction Configuration

- 7) The program is now complete. Refer to Figure 74.
- 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 75. 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.

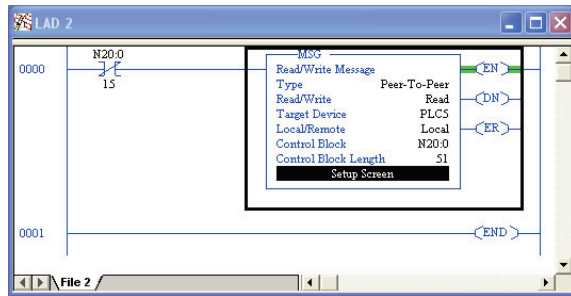


Figure 74: Completed PLC Program

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

Figure 75: Monitoring the Data Being Read from the Inverter

9.3.5 SLC-5/05 Example: Reading and Writing Multiple MSG Instructions

Often times, applications may need to both read data from and write data to the inverter. To accomplish this task, multiple MSG instructions will need to 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.

Figure 76 shows an example of two MSG instructions, one for reading and one for writing. It is evident from this logic that N20 and N21 are the two independent message control files created for these instructions. Note that the “Read/Write” field of each of the MSG instructions is set according to their function.

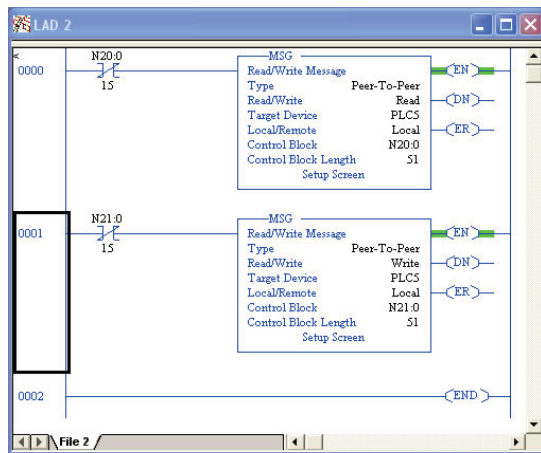


Figure 76: Reading and Writing via MSG Instructions

Figure 77 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 N10:1, which equates to inverter register 1). The source Data Table Address in this case is N18:30.

MSG - Rung #2:1 - N21:0

General | MultiHop

This Controller

Communication Command:

Data Table Address:

Size in Elements:

Channel:

Target Device

Message Timeout:

Data Table Address:

Local / Remote: Local Remote MultiHop: Yes No

Control Bits

Ignore if timed out (TO):

To be retried (NR):

Awaiting Execution (EW):

Continuous Run (CO):

Error (ER):

Message done (DN):

Message Transmitting (ST):

Message Enabled (EN):

Waiting for Queue Space:

Error

Error Code(Hex): 0

Error Description

No errors

Figure 77: MSG Configuration for Writing

9.4 BACnet/IP

- The interface card supports the BACnet/IP (Annex J) protocol over Ethernet via a configurable UDP port.
- The BACnet driver does not trigger timeout events (section 5.4.1).

9.4.1 Protocol Implementation Conformance Statement (PICS)

BACnet Protocol

Date:	May 26, 2016
Vendor Name:	ICC, Inc.
Product Name:	Mitsubishi Inverter FR-800
Product Model Number:	A8NETH-2P
Applications Software Version:	V1.1.1
Firmware Revision:	V1.1.1
BACnet Protocol Revision:	2
Product Description:	

The Mitsubishi 800-series inverter family delivers multi-use performance, accuracy and system flexibility in an ultra-reliable hardware package.

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

- Able to transmit segmented message Window Size _____
- Able to receive segmented message 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

Networking Security Options:

- Non-secure Device – is capable of operating without BACnet Network Security
- Secure Device – is capable of using BACnet Network Security (NS-SD BIBB)
 - Multiple Application-Specific Keys:
 - Supports encryption (NS-ED BIBB)
 - Key Server (NS-KS BIBB)

Character Sets Supported:

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

- ISO 10646 (UTF-8)
- IBM™/Microsoft™ DBCS
- ISO 8859-1
- ISO 10646 (UCS-2)
- ISO 10646 (UCS-4)
- JIS X 0208

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
Analog Output	Real	Real, Unsigned, Integer, Null
Analog Input	Real	N/A
Analog Value	Real	Real, Unsigned, Integer, Null
Binary Output	Enumerated	Enumerated, Boolean, Real, Unsigned, Integer, Null
Binary Input	Enumerated	N/A
Binary Value	Enumerated	Enumerated, Boolean, Real, Unsigned, Integer, Null
Multi-state Output	Unsigned	Enumerated, Real, Unsigned, Integer, Null
Multi-state Input	Unsigned	N/A
Multi-state Value	Unsigned	Enumerated, Real, Unsigned, Integer, Null

Notes:

- The Null data type is used to relinquish a previously-commanded entry at the targeted priority in the priority array.

Object Types/Property Support Tables:

Table 27: BACnet Device Object Types /Properties Supported

Property	Object Type
	Device
Object Identifier	R
Object Name	R
Object Type	R
System Status	R
Vendor Name	R
Vendor Identifier	R
Model Name	R
Firmware Revision	R
Appl Software Revision	R
Protocol Version	R
Protocol Revision	R
Services Supported	R
Object Types Supported	R
Object List	R
Max APDU Length	R
Segmentation Support	R
APDU Timeout	R
Number APDU Retries	R
Device Address Binding	R
Database Revision	R

R – readable using BACnet services

W – readable and writable using BACnet services

Table 28: BACnet Binary Object Types /Properties Supported

Property	Object Type		
	Binary Input	Binary Output	Binary Value
Object Identifier	R	R	R
Object Name	R	R	R
Object Type	R	R	R
Present Value	R	W	W
Status Flags	R	R	R
Event State	R	R	R
Out-of-Service	R	R	R
Priority Array		R	R
Relinquish Default		R	R
Polarity	R	R	
Active Text	R	R	
Inactive Text	R	R	

R – readable using BACnet services

W – readable and writable using BACnet services

Table 29: BACnet Analog Object Types /Properties Supported

Property	Object Type		
	Analog Input	Analog Output	Analog Value
Object Identifier	R	R	R
Object Name	R	R	R
Object Type	R	R	R
Present Value	R	W	W
Status Flags	R	R	R
Event State	R	R	R
Out-of-Service	R	R	R
Units	R	R	R
Priority Array		R	R
Relinquish Default		R	R

R – readable using BACnet services

W – readable and writable using BACnet services

Table 30: BACnet Multi-state Object Types /Properties Supported

Property	Object Type		
	Multi-state Input	Multi-state Output	Multi-state Value
Object Identifier	R	R	R
Object Name	R	R	R
Object Type	R	R	R
Present Value	R	W	W
Status Flags	R	R	R
Event State	R	R	R
Out-of-Service	R	R	R
Number of States	R	R	R
Priority Array		R	R
Relinquish Default		R	R

R – readable using BACnet services

W – readable and writable using BACnet services

9.4.2 Default Supported Objects

This section will describe the default objects. Since the objects are configurable, the system integrator is responsible for managing, maintaining, and documenting the actual configuration. Always use the studio to confirm the configuration before commissioning the device.

Table 31: Binary Input Object Instance Summary

Instance ID	Object Name	Description	Register	Bit (Mask)	Active Text
					Inactive Text
B11	OP_ENABLED	Drive function enabled	1227	2 (0x0004)	enabled
					disabled
B12	FAULTED	Unacknowledged faults present	1227	3 (0x0008)	faulted
					no fault
B13	QUICK_STOP_DISABLED	Quick stop disabled	1227	5 (0x0020)	stop
					disabled
B14	REF_FROM_NET	Speed reference from network	1227	8 (0x0100)	on
					off
B15	CTRL_FROM_NET	Control reference from network	1227	9 (0x0200)	on
					off
B16	SETPOINT_REACHED	Reached commanded speed	1227	10 (0x0400)	on
					off
B17	REV_OP_ENABLED	Drive function enabled in reverse mode	1227	14 (0x4000)	enabled
					disabled
B18	STOPPING	Drive is stopping	1227	15 (0x8000)	stopping
					off

Table 32: Binary Output Object Instance Summary

Instance ID	Object Name	Description	Register	Bit (Mask)	Active Text
					Inactive Text
BO1	OP_CMD	Enable operation	1	1 (0x0002)	operate
					off
BO2	REV_OP_CMD	Enable reverse operation	1	2 (0x0004)	reverse
					off
BO3	QUICK_STOP	Quick stop	1	3 (0x0008)	stop
					off
BO4	FAULT_RESET	Reset fault	1	11 (0x0800)	reset
					off
BO5	NET_REF_CMD	Request reference from network	1	14 (0x4000)	on
					off
BO6	NET_CTRL_CMD	Request control from network	1	15 (0x8000)	on
					off

Table 33: Analog Output Object Instance Summary

Instance ID	Object Name	Description	Register	Units
AO1	SETPOINT_SPEED	Setpoint speed	1229	Hz
AO2	INVERTER_RESET	Fault reset	5	None
AO3	ALARM_HISTORY_CLEAR	Clear alarm (fault) history	6	None
AO4	ALL_PARAM_CLEAR	Clear all parameters	7	None

Table 34: Analog Input Object Instance Summary

Instance ID	Object Name	Description	Register	Units
AI1	ACTUAL_SPEED	Actual speed	1231	RPM
AI2	OUTPUT_FREQ	Output frequency in 0.01 Hertz (6000=60.00Hz)	201	Hz
AI3	OUTPUT_CURRENT	Output current in 0.1 or 0.01 Amp (depends on inverter capacity)	202	Amps
AI4	OUTPUT_VOLTAGE	Output voltage in 0.1 Volt (1000=100.0V)	203	Voltage
AI5	INPUT_POWER	Input power in 0.1 or 0.01 kW (depends on inverter capacity)	213	kW
AI6	OUTPUT_POWER	Output power in 0.1 or 0.01 kW (depends on inverter capacity)	214	kW
AI7	CUMULATIVE_POWER	Energy consumption	225	kWh
AI8	PID_SET_POINT	PID set point in 0.1%	252	%
AI9	PID_MEASURED_VALUE	PID measured value in 0.1%	253	%
AI10	PID_DEVIATION_VALUE	PID deviation value in 0.1%	254	%
AI11	FAULT_CODE_1	Fault code 1 (most recent)	284	None
AI12	FAULT_CODE_2	Fault code 2	285	None
AI13	FAULT_CODE_3	Fault code 3	286	None
AI14	FAULT_CODE_4	Fault code 4	287	None
AI15	FAULT_CODE_5	Fault code 5	288	None
AI16	FAULT_CODE_6	Fault code 6	289	None
AI17	FAULT_CODE_7	Fault code 7	290	None
AI18	FAULT_CODE_8	Fault code 8 (least recent)	291	None

9.4.4 Server Settings

In the studio's **Project** panel, navigate to **A8NETH-2P...Ethernet...BACnet/IP Server**.

UDP Port

This is the UDP port on which to transmit and receive BACnet/IP packets on the local subnet. The default value is 47808 (0xBAC0). To ensure successful communications, use caution when using a port setting other than the default value.

9.4.5 Node Settings

There are no node settings. A node is simply a container for objects.

9.4.6 Device Object Settings

In the studio's **Project** panel, navigate to **A8NETH-2P...Ethernet...BACnet/IP Server...Node...Device Object**.

A Device Object is automatically added to every node, and cannot be removed. The Device Object contains several configurable fields that must be appropriately set for each device residing on a BACnet network.

Device Name

Defines the node's name. The device name must be unique across the entire BACnet network. Enter a string of between 1 and 32 characters in length.

Instance Number

Defines the node's instance number. The instance number must be unique across the entire BACnet network. Enter a value between 0...4194302 (0x0...0x3FFFFE).

9.4.7 BACnet Object Settings

In the studio's **Project** panel, navigate to **A8NETH-2P...Ethernet...BACnet/IP Server...Node** and add an object from the **Available Objects** panel.

The BACnet server hosts BACnet objects which contain many different properties for any BACnet client on the network to access. The driver supports a variety of different BACnet objects. All supported properties of these objects are readable, while only the present value property is writable (for Outputs and Values only).

9.4.7.1 Analog Input Object Settings

Object Name

The name of the BACnet object. Enter a string of between 1 and 32 characters in length. All object names must be unique within a node.

Instance

The BACnet object's instance number. Enter a value between 0...4194302 (0x0...0x3FFFFE).

Register

The inverter register number (refer to section 4) that the BACnet object's present value will access.

Units

Select the desired units from this dropdown menu. If the desired units are not available in the dropdown menu, select "Other Units" and enter the appropriate enumerated value (as defined by the BACnet Specification) in the "Unit Value" field.

Unit Value

This field is enabled only when the "Units" selection is set to "Other Units". Enter the appropriate enumerated value (as defined by the BACnet Specification.)

9.4.7.2 Analog Output Object Settings

Object Name

The name of the BACnet object. Enter a string of between 1 and 32 characters in length. All object names must be unique within a node.

Instance

The BACnet object's instance number. Enter a value between 0...4194302 (0x0...0x3FFFFE).

Register

The inverter register number (refer to section 4) that the BACnet object's present value will access.

Units

Select the desired units from this dropdown menu. If the desired units are not available in the dropdown menu, select "Other Units" and enter the appropriate enumerated value (as defined by the BACnet Specification) in the "Unit Value" field.

Unit Value

This field is enabled only when the "Units" selection is set to "Other Units". Enter the appropriate enumerated value (as defined by the BACnet Specification.)

Relinquish Default

Defines the default value to be used for an object's present value property when all entries in the object's priority array are NULL.

9.4.7.3 Analog Value Object Settings

Object Name

The name of the BACnet object. Enter a string of between 1 and 32 characters in length. All object names must be unique within a node.

Instance

The BACnet object's instance number. Enter a value between 0...4194302 (0x0...0x3FFFFE).

Register

The inverter register number (refer to section 4) that the BACnet object's present value will access.

Units

Select the desired units from this dropdown menu. If the desired units are not available in the dropdown menu, select "Other Units" and enter the appropriate enumerated value (as defined by the BACnet Specification) in the "Unit Value" field.

Unit Value

This field is enabled only when the “Units” selection is set to “Other Units”. Enter the appropriate enumerated value (as defined by the BACnet Specification.)

Relinquish Default

Defines the default value to be used for an object’s present value property when all entries in the object’s priority array are NULL.

9.4.7.4 Binary Input Object Settings

Object Name

The name of the BACnet object. Enter a string of between 1 and 32 characters in length. All object names must be unique within a node.

Instance

The BACnet object’s instance number. Enter a value between 0...4194302 (0x0...0x3FFFFE).

Register

The inverter register number (refer to section 4) that the BACnet object’s present value will access.

Bitmask

Specifies which bit(s) in the 16-bit value designated by the “Register” that the binary object will map to. This mechanism allows up to 16 binary objects to be simultaneously assigned to one register (each binary object mapping to a single bit of that 16-bit word). It is possible to map binary objects to multiple bits within the designated register.

The effect of the “Bitmask” field when reading: When the present value property of a binary object is read by a BACnet client, the bitmask is used to determine the active/inactive state of the object by inspecting the value in the designated register at the bit location(s) indicated in the bitmask. If all of the bit locations at the designated register are set, then the object’s state will be returned as “active”. Else, the object’s state will be returned as “inactive”. This resultant state is reversed prior to being placed on the network if the object’s “Polarity” is set to “Reverse”.

Active Text

Specifies the description of the object's "active" state. Enter a string of up to 32 characters in length. This field is optional and may be left blank.

Inactive Text

Specifies the description of the object's "inactive" state. Enter a string of up to 32 characters in length. This field is optional and may be left blank.

Polarity

Indicates the relationship between the physical state of the object (as stored in the register) and the logical state represented by the object's present value property. If the physical state is active high, select "Normal" from this dropdown menu. If the physical state is active low, select "Reverse" from this dropdown menu. For further detail, refer to the "Bitmask" behavioral description.

9.4.7.5 Binary Output Object Settings

Object Name

The name of the BACnet object. Enter a string of between 1 and 32 characters in length. All object names must be unique within a node.

Instance

The BACnet object's instance number. Enter a value between 0...4194302 (0x0...0x3FFFFE).

Register

The inverter register number (refer to section 4) that the BACnet object's present value will access.

Bitmask

Specifies which bit(s) in the 16-bit value designated by the "Register" that the binary object will map to. This mechanism allows up to 16 binary objects to be simultaneously assigned to one register (each binary object mapping to a single bit of that 16-bit word). It is possible to map binary objects to multiple bits within the designated register.

The effect of the “Bitmask” field when writing: When the present value property of a binary object is set to “active” by a BACnet client, then the bit(s) in the designated register indicated by the bitmask are set. Similarly, when the present value property of the object is set to “inactive”, then the bit(s) in the designated register indicated by the bitmask are cleared. This setting/clearing behavior is reversed if the object’s “Polarity” is set to “Reverse”.

The effect of the “Bitmask” field when reading: When the present value property of a binary object is read by a BACnet client, the bitmask is used to determine the active/inactive state of the object by inspecting the value in the designated register at the bit location(s) indicated in the bitmask. If all of the bit locations at the designated register are set, then the object’s state will be returned as “active”. Else, the object’s state will be returned as “inactive”. This resultant state is reversed prior to being placed on the network if the object’s “Polarity” is set to “Reverse”.

Active Text

Specifies the description of the object’s “active” state. Enter a string of up to 32 characters in length. This field is optional and may be left blank.

Inactive Text

Specifies the description of the object’s “inactive” state. Enter a string of up to 32 characters in length. This field is optional and may be left blank.

Polarity

Indicates the relationship between the physical state of the object (as stored in the register) and the logical state represented by the object’s present value property. If the physical state is active high, select “Normal” from this dropdown menu. If the physical state is active low, select “Reverse” from this dropdown menu. For further detail, refer to the “Bitmask” behavioral description.

Relinquish Default

Defines the default value to be used for an object’s present value property when all entries in the object’s priority array are NULL.

9.4.7.6 Binary Value Object Settings

Object Name

The name of the BACnet object. Enter a string of between 1 and 32 characters in length. All object names must be unique within a node.

Instance

The BACnet object's instance number. Enter a value between 0...4194302 (0x0...0x3FFFFE).

Register

The inverter register number (refer to section 4) that the BACnet object's present value will access.

Bitmask

Specifies which bit(s) in the 16-bit value designated by the "Register" that the binary object will map to. This mechanism allows up to 16 binary objects to be simultaneously assigned to one register (each binary object mapping to a single bit of that 16-bit word). It is possible to map binary objects to multiple bits within the designated register.

The effect of the "Bitmask" field when writing: When the present value property of a binary object is set to "active" by a BACnet client, then the bit(s) in the designated register indicated by the bitmask are set. Similarly, when the present value property of the object is set to "inactive", then the bit(s) in the designated register indicated by the bitmask are cleared.

The effect of the "Bitmask" field when reading: When the present value property of a binary object is read by a BACnet client, the bitmask is used to determine the active/inactive state of the object by inspecting the value in the designated register at the bit location(s) indicated in the bitmask. If all of the bit locations at the designated register are set, then the object's state will be returned as "active". Else, the object's state will be returned as "inactive".

Active Text

Specifies the description of the object's "active" state. Enter a string of up to 32 characters in length. This field is optional and may be left blank.

Inactive Text

Specifies the description of the object's "inactive" state. Enter a string of up to 32 characters in length. This field is optional and may be left blank.

Relinquish Default

Defines the default value to be used for an object's present value property when all entries in the object's priority array are NULL.

9.4.7.7 Multi-state Input Object Settings

Object Name

The name of the BACnet object. Enter a string of between 1 and 32 characters in length. All object names must be unique within a node.

Instance

The BACnet object's instance number. Enter a value between 0...4194302 (0x0...0x3FFFFE).

Register

The inverter register number (refer to section 4) that the BACnet object's present value will access.

9.4.7.8 Multi-state Output Object Settings

Object Name

The name of the BACnet object. Enter a string of between 1 and 32 characters in length. All object names must be unique within a node.

Instance

The BACnet object's instance number. Enter a value between 0...4194302 (0x0...0x3FFFFE).

Register

The inverter register number (refer to section 4) that the BACnet object's present value will access.

Relinquish Default

Defines the default value to be used for an object's present value property when all entries in the object's priority array are NULL.

9.4.7.9 Multi-state Value Object Settings

Object Name

The name of the BACnet object. Enter a string of between 1 and 32 characters in length. All object names must be unique within a node.

Instance

The BACnet object's instance number. Enter a value between 0...4194302 (0x0...0x3FFFFE).

Register

The inverter register number (refer to section 4) that the BACnet object's present value will access.

Relinquish Default

Defines the default value to be used for an object's present value property when all entries in the object's priority array are NULL.

9.5 MELSEC SERVER

9.5.1 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 section 4.1). 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 include:

- Supports up to 8 simultaneous connections.
- The TCP port is user-configurable.
- The max MELSEC frame size is 1460 bytes. Refer to Table 35.
- 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.
- The driver can be configured to detect a timeout (communication loss) and perform a timeout action.

9.5.2 Supported Commands

Table 35 defines the commands that are supported by the driver.

Table 35: MELSEC Server Supported Commands

Frame Type	Command Name	Command Code	Subcommand Code	Max Points
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

9.5.3 Server Settings

In the studio's Project panel, navigate to **A8NETH-2P...Ethernet...MELSEC Server**.

TCP Port

Defines the local TCP port (1025...65534) on which the driver will listen for connections from the client. Ensure that this port assignment is unique, and does not conflict with ports utilized by other drivers.

9.5.4 Connection Timeout Options

In the studio's Project panel, navigate to **A8NETH-2P...Ethernet...MELSEC Server**. The following configuration options will determine the actions to be taken if the connection is abnormally terminated or lost. 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. Note that a certain degree of caution must be exercised when using the timeout feature to avoid "nuisance" timeouts from occurring.

Enable Supervisory Timer

This timer provides the ability for the driver to monitor timeout occurrences on the overall receive activity for all connections.

- The timer will start after receiving the first request. Once the timer is started, it cannot be disabled.
- If the driver experiences no receive activity for more than the **Timeout** time setting, then the driver assumes that the client or network has experienced some sort of unexpected problem, and will perform the **Timeout Action**.

Enable Connection Timer

This timer provides the ability for the driver to monitor timeout occurrences and errors within the scope of each client connection.

- If a particular open socket experiences no activity for more than the **Timeout** time setting, then the driver assumes that the client or network has experienced some sort of unexpected problem, and will close that socket and perform the **Timeout Action**.
- If a socket error occurs (regardless of whether the error was due to a communication lapse or abnormal socket error), the driver will perform the **Timeout Action**. Specifically, do not perform inadvisable behavior such as sending a request from the client device, and then closing the socket prior to successfully receiving the server's response. The reason for this is because the server will experience an error when attempting to respond via the now-closed socket. Always be sure to manage socket life cycles "gracefully", and do not abandon outstanding requests.

Timeout

Defines the maximum number of milliseconds for a break in network communications before a timeout event will be triggered.

Timeout Action

Select an action from the drop down menu:

"None" No effect. The inverter will continue to operate with the last available settings.

"Apply Fail-safe Values" Apply the fail-safe values as described in section 5.4.1.

9.6 PROFINET IO

9.6.1 Overview

The PROFINET IO device driver allows a controller to interact with the interface card via cyclic data exchange and acyclic read/write requests. The I/O data is entirely user-configurable, and is utilized when a standard I/O module is chosen during network configuration.

Other notes of interest include:

- Allows simultaneous access to only 1 PROFINET controller.
- Supports conformance class B and real time (RT) communication.
- Supports the highest Netload Class III.
- Supports MRP (Media Redundancy Protocol) client.
- Supports DCP (Discovery Control Protocol).
- Supports alarms.
- Supports I&M.
- The lowest supported I/O Cycle Update Time (via STEP 7 or an equivalent hardware configuration tool) is 1ms.
- The GSDML file can be obtained from the [product web page](#).
- Supports several user configurable I/O modules with up to 32 input words and 32 output words.
- Supports the PROFIdrive profile version 4.1.
- No explicit module selection is required on the interface card: the module will be selected automatically according to the controller's configuration.
- If a timeout (communication loss) occurs on the RT connection, the driver can be configured to perform a timeout action. 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.

9.6.2 Device Settings

In the studio's **Project** panel, navigate to **A8NETH-2P...Ethernet...PROFINET IO**.

Device Name

The device name / station name must be unique across the entire PROFINET network, because it is used by controllers to uniquely identify PROFINET devices. This string must conform to the device name requirements contained in the PROFINET specification.

9.6.3 Connection Timeout Options

In the studio's **Project** panel, navigate to **A8NETH-2P...Ethernet...PROFINET IO**. The following configuration options will determine the actions to be taken by the card if the PROFINET IO connection is abnormally terminated or lost.

Timeout Action

Select an action from the drop down menu:

“None” No effect. The inverter will continue to operate with the last available settings.

“Apply Fail-safe Values” Apply the fail-safe values as described in section 5.4.1.

9.6.4 Cyclic I/O Produced and Consumed Data Access Settings

In the studio's **Project** panel, navigate to **A8NETH-2P...Ethernet...PROFINET IO...Produced Data Word** and/or **Consumed Data Word**.

The Produced Data Word and Consumed Data Word objects are only applicable when using the I/O module “IN: 32 WORDS, OUT: 32 WORDS”, which is typically the case. The Produced Data Word defines the structure of status data sent from the inverter to the controller. The Consumed Data Word objects will define the structure of the command data sent from the controller (for example, a Siemens PLC) to the inverter. These objects allow the creation of custom-built I/O data. Up to 32 “command” register values can be sent to the inverter, and up to 32 “status” register values can be sent back to the controller. Therefore, up to 32 Produced and 32 Consumed Data Word objects can be created. If a consumed word offset is not defined, that data will be ignored by the inverter. If a produce word offset is not defined, the value will default to 0. The size of the actual I/O produced and consumed data is determined by the PROFINET controller. The I/O data format is summarized in Table 36.

Description

This 32-character (max) field is strictly for user reference: it is not used at any time by the driver.

Produced Data Word Offset

The value from the associated inverter register will populate this word offset of the produced data that is to be sent to the controller. It is recommended to start at word offset 0.

Consumed Data Word Offset

The consumed data received from the controller at this word offset will contain the value to be written to the associated inverter register. It is recommended to start at word offset 0.

Register

The inverter register (refer to section 4) associated with the word offset. For the Produced Data Word object, enter a “status register to be monitored. For the Consumed Data Word object, enter a “command” register that can be written.

Data Type

Each data word is fixed to 16-Bit Unsigned. This is equivalent to two bytes. The data word is transferred in little endian format.

Table 36: PROFINET User Configurable Module I/O Data Format

Consumed Data (PLC to Inverter)		Produced Data (Inverter to PLC)	
Word Offset	Inverter Register	Word Offset	Inverter Register
0	Any	0	Any
1	Any	1	Any
:	Any	:	Any
30	Any	30	Any
31	Any	31	Any

The default I/O configuration is described in Table 37. Always use the studio to confirm the configuration before commissioning the device.

Table 37: PROFINET Default User Configurable Module I/O Data Format

Consumed Data (PLC to Inverter)		Produced Data (Inverter to PLC)	
Word Offset	Inverter Register	Word Offset	Inverter Register
0	1	0	1227
1	1229	1	1231
:	None	:	None

9.6.5 PROFIdrive Profile

For optimal interoperability, the interface card supports the PROFIdrive profile version 4.1. Use of the PROFIdrive profile is optional and is not recommended unless specifically required in the PROFINET system specification. No explicit configuration of the interface card is necessary in the studio when using the PROFIdrive profile. The controller **must** support the PROFIdrive profile and **must** be configured to use the “Standard Telegram 1” module on the interface card. If the controller does not support the PROFIdrive profile, use the configurable I/O “IN: 32 WORDS, OUT: 32 WORDS” module. The PROFIdrive profile is only partially described in this manual due to its complexity. The complete PROFIdrive 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

9.6.5.1 PROFIdrive Standard Telegram 1

The standard telegram 1 mapping is described in Table 38.

Table 38: 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

9.6.5.2 PROFIdrive 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 39. The status word, ZSW1, returns status information from the inverter to the controller. The bitmapping for the status word is described in Table 40.

Table 39: 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
	0	Freeze Ramp Generator	Freeze the RFG with the current output frequency

Bit	Value	Significance	Description
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 40: 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
	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 1227 bit 3
	0	No Fault	No trip present as indicated by register 1227 bit 3
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	---	---

Bit	Value	Significance	Description
8	1	Speed Within Tolerance	Actual value equals the reference value and is within the tolerance as indicated by register 1227 bit 10.
	0	Speed Out Of Tolerance	Actual value differs from the reference value or is outside of the tolerance as indicated by register 1227 bit 10.
9	1	Control Requested	Control by PLC is possible as indicated by register 1227 bit 8 and 9.
	0	No Control Requested	Control is not possible by the controller as indicated by register 1227 bit 8 and 9.
10	1	Frequency Reached Or Exceeded	The actual value \geq max reference value. Refer to register 1236 (RatedSpeed).
	0	Frequency Not Reached	The actual value $<$ max reference value. Refer to register 1236 (RatedSpeed).
11 - 15	Not used	---	---

9.6.5.3 PROFdrive 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.

The “Max RPM” equates to the value of register 1236 (RatedSpeed).

$$\text{RPM} = \frac{\text{NSOLL_A} \times \text{Max RPM}}{0x4000} \quad \text{Equation 4}$$

$$\text{NIST_A} = \frac{\text{RPM} \times 0x4000}{\text{Max RPM}} \quad \text{Equation 5}$$

NSOLL_A: The reference speed setpoint is a normalized value. The interface card applies the Normalize-to-RPM conversion indicated by Equation 4. The resulting value is written to inverter register 1229 (SetpointSpeed).

NIST_A: The inverter operating actual speed is a normalized value that is calculated from inverter register 1231 (ActualSpeed). The interface card applies the RPM-to-Normalize conversion indicated by Equation 5.

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 value of register 1236 is read by the interface card only at boot-up. If this value is changed, then the interface card must be rebooted in order for it to read and apply the new value when performing the conversions.

9.6.5.4 PROFdrive-Specific Parameters

The PROFdrive-specific parameters are shown in Table 41. The parameters are read-only.

Table 41: 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 (Fault codes 284..291)
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

9.6.5.5 PROFdrive State Diagram

The state diagram is displayed in Figure 78.

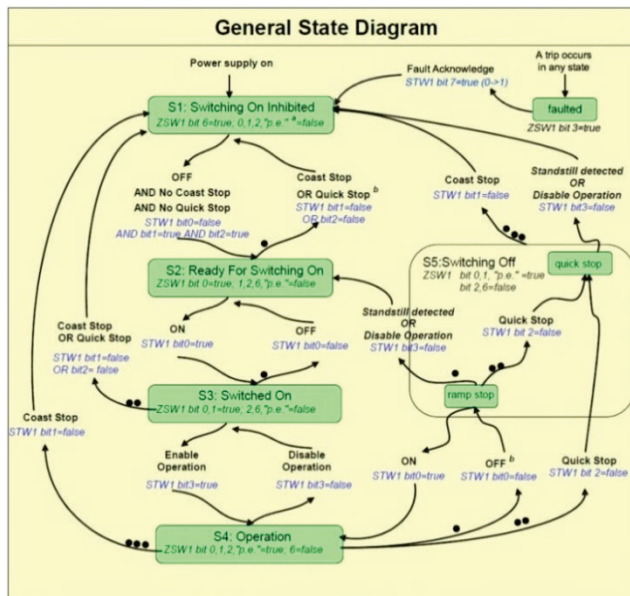


Figure 78: PROFdrive State Diagram

9.6.6 Acyclic Data Access

Any inverter register can be accessed via PROFINET acyclic services. To accomplish this, set the API to 0, Slot to 1, and SubSlot to 1. The record number/index value is equivalent to the desired register number described in section 4.1. The length is specified according to the number of bytes to access. Because each register corresponds to 2 bytes of data, the length must be an even number.

9.6.7 TIA Portal (STEP 7) Hardware Configuration Example

The following example will use TIA Portal V13 (STEP 7) to demonstrate the basic hardware configuration procedure to configure a PROFINET device. The procedure, in general, will apply to similar configuration software. The example will not cover all features of TIA Portal. Any questions regarding TIA Portal (or similar configuration software) must be directed to the vendor of the software.

This example assumes that there is already an existing TIA Portal project with the desired PLC.

9.6.7.1 Register the GSDML File

1. Open the TIA Portal project. Navigate to **Options...Manage general station description files (GSD)** as shown in Figure 79.
2. Locate and select the GSDML file and click the **Install** button. Confirm that the installation was completed successfully as shown in Figure 80 and click the **Close** button. It is recommended to use the latest GSDML file, which is available via the [product web page](#) on the internet.
3. This will update the **Hardware catalog**. Locate the device in the **Hardware catalog**. In the **Project tree**, double-click on **Device & networks**. Select the **Network view** tab and locate the device in the **Hardware catalog** as shown in Figure 81.

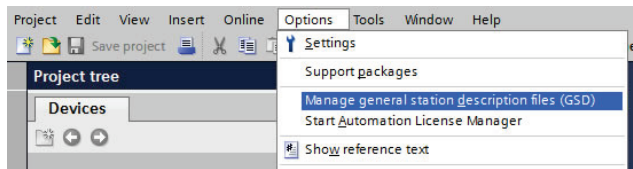


Figure 79: Install GSD File Menu Option

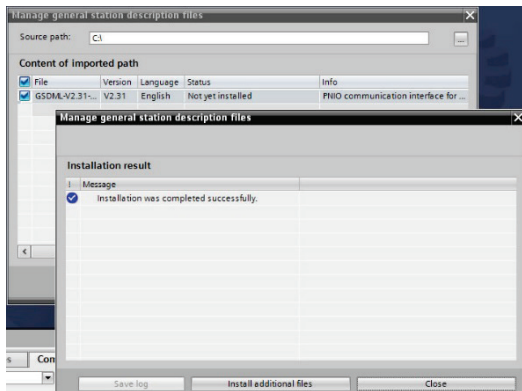


Figure 80: Successfully Installed GSDML File

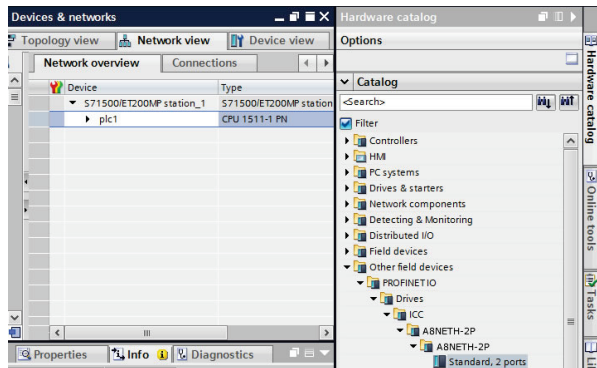


Figure 81: Updated Hardware Catalog

9.6.7.2 Add the Device to the Configuration

Select the device in the device tree and drag the device onto the PROFINET IO system in the configuration as shown in Figure 82.

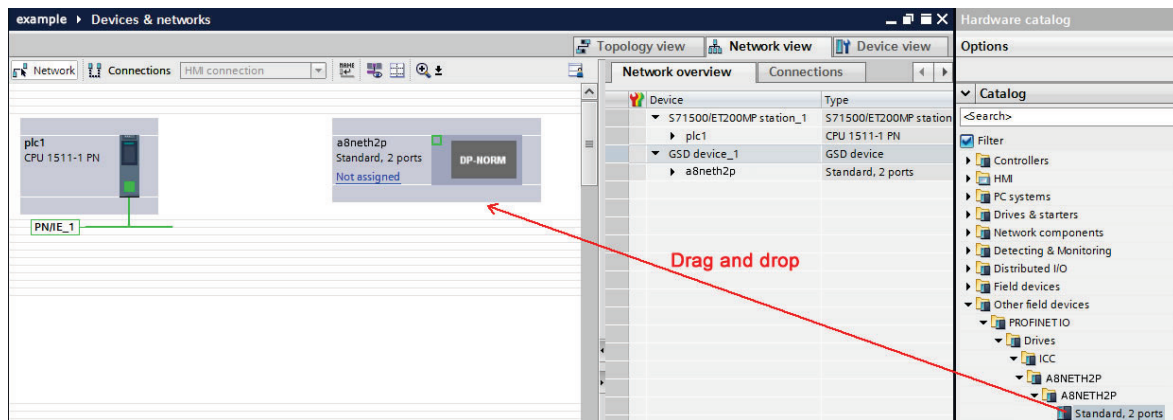


Figure 82: Add Device to Configuration

9.6.7.3 Select the IO Controller

On the device, click “Not assigned” and select the appropriate PLC PROFINET interface as shown in Figure 83. This will assign the device to the PROFINET IO system as shown in Figure 84.



Figure 83: Select IO Controller

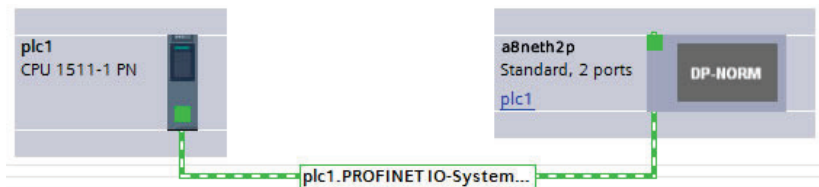


Figure 84: PROFINET IO System

9.6.7.4 Assign the I/O Module

Click on the device and then click on the **Device view** tab. In the **Hardware catalog**, expand **Module** and add a module “IN: XX WORDS, OUT: YY WORDS” into **Slot 1**. The module will determine the input and output sizes. In this example, the module “IN: 02 WORDS, OUT: 02 WORDS” is selected. Select a module with the appropriate input and output sizes for your specific application. Refer to Figure 85.

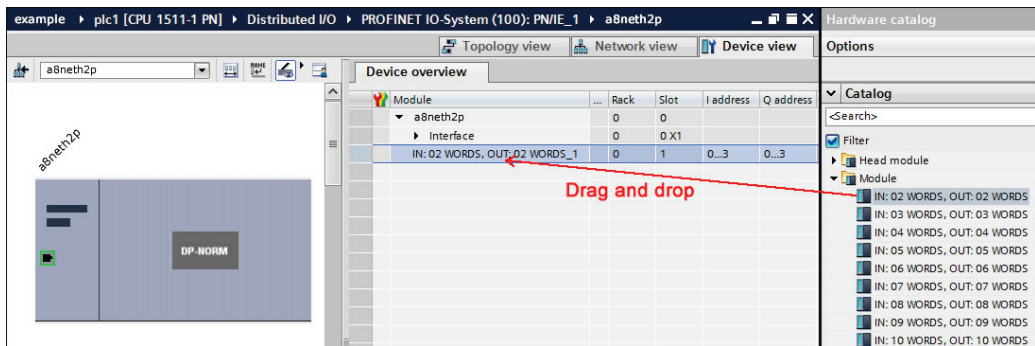


Figure 85: Add IO Module

9.6.7.5 Configure the Device Properties

1. Select the device and navigate to the **Properties** tab. Select the **PROFINET interface [X1]** node. Assign a unique and compatible **IP address** for this device as shown in Figure 86.
2. Assign a unique **PROFINET device name** as shown in Figure 87.
3. Set the I/O cycle **Update time** and **Watchdog time** as shown in Figure 88.

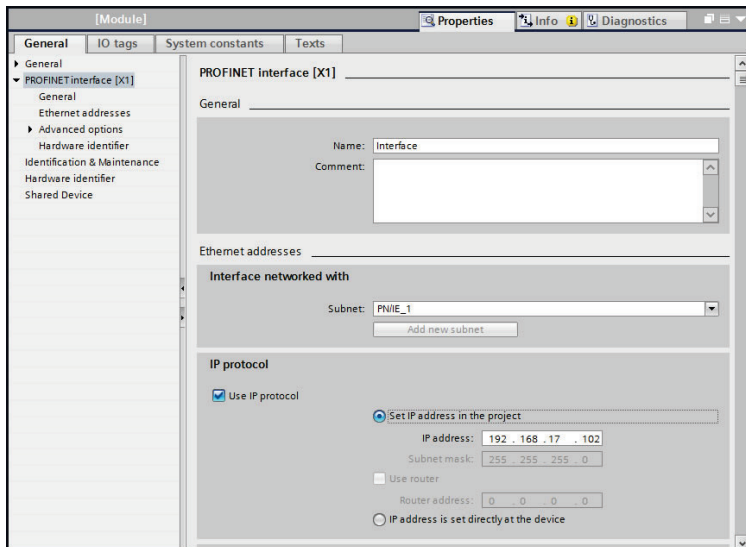


Figure 86: Assign Unique Compatible IP Address

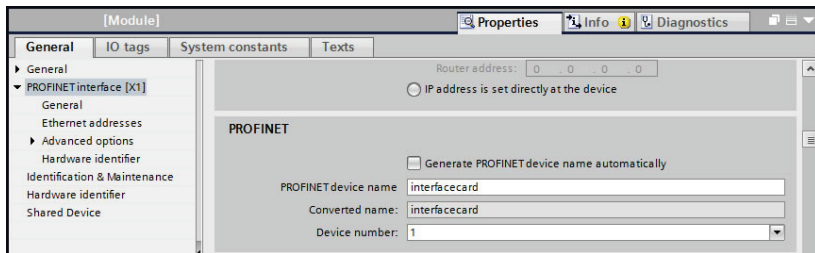


Figure 87: Assign Unique Device Name

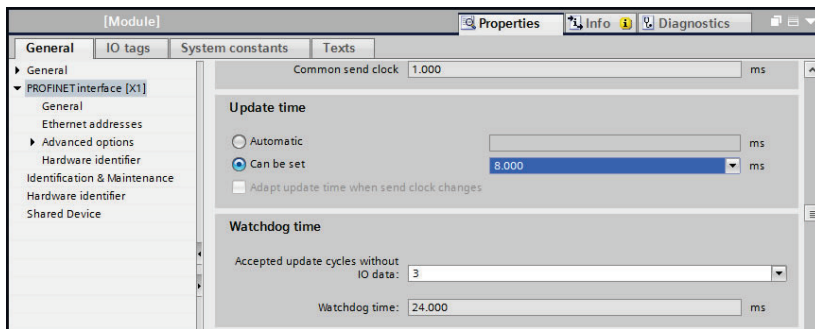


Figure 88: Set I/O Cycle Update Time

9.6.7.6 Online Device Discovery and Configuration

In the **Project tree**, expand **plc1...Distributed I/O...PROFINET IO-System (100):PN/IE_1**. Expand the device and double-click **Online & diagnostics**. In the next panel, expand **Functions** and select the **Assign IP address** node. Click the **Accessible devices** button. Select the appropriate **PG/PC interface** and click the **Start search** button to discover and display the PROFINET devices on the network as shown in Figure 89. Select the device and click the **Apply** button.

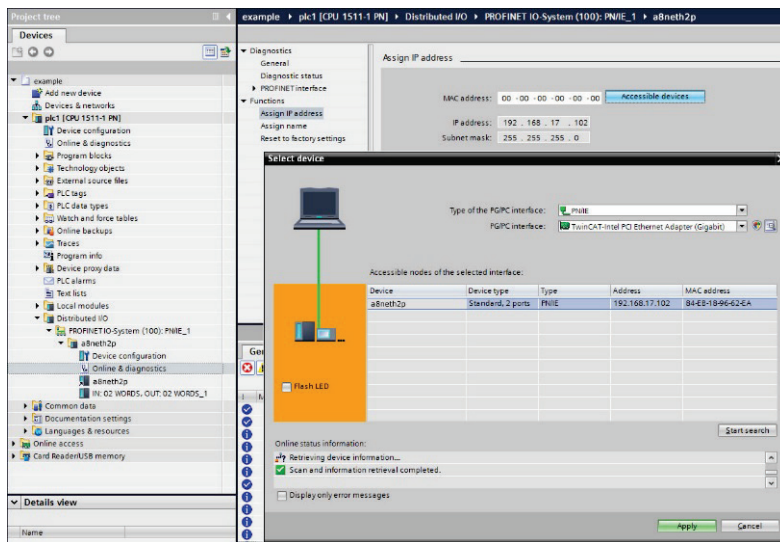


Figure 89: Discover PROFINET Devices on the Network

If the **IP address** does not match the values set in the configuration, click the **Assign IP address** button as shown in Figure 90.

Navigate to **Functions...Assign name**. If the **PROFINET device name** does not match, select the device and click the **Assign name** button as shown in Figure 91.

9.6.7.7 Save the Configuration

The hardware configuration is now complete. Save and perform any necessary compilation of the configuration. Download the application and configuration to the PLC. The PLC application program can then be started. Please consult with the vendor of your PROFINET PLC software for additional programming and configuration details.

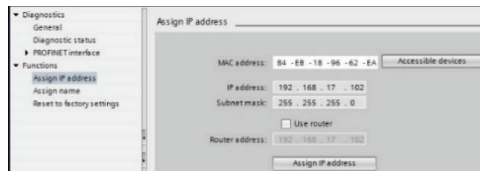


Figure 90: Assign IP Address

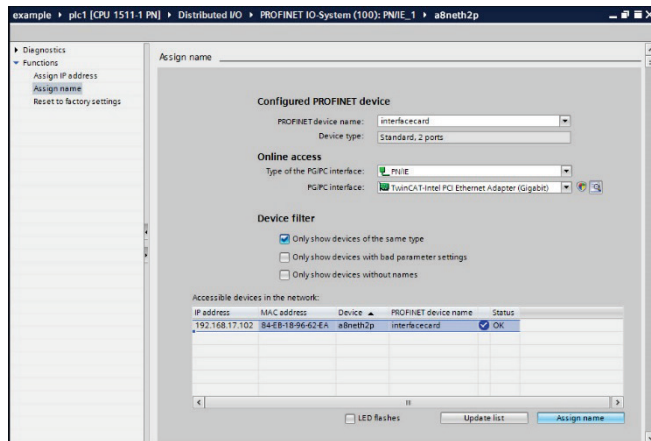


Figure 91: Assign Name

9.6.8 GE Proficy Configuration Example

The following example will use GE Proficy Machine Edition SIM11 to demonstrate the basic procedure for configuring a PROFINET device. The example will not cover all features of Proficy. Any questions regarding Proficy (or similar configuration software) must be directed at the vendor of the software.

This example assumes that there is already an existing Proficy project with the desired PLC.

9.6.8.1 Register the GSDML File

Open the Proficy project. In the **Navigator** panel, right-click **Profinet Controller** and select **Add IO-Device...** as shown in Figure 92.

Click the **Have GSDML...** button as shown in Figure 93.

Locate and select the GSDML file. Click the **Open** button to register the GSDML as shown in Figure 94. It is recommended to use the latest GSDML, which is available via the [product web page](#) on the internet.

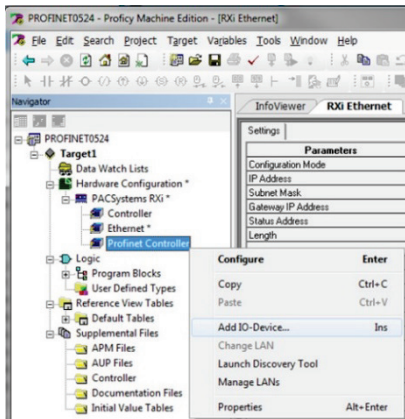


Figure 92: Add IO-Device

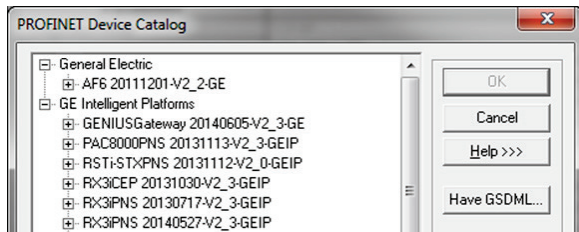


Figure 93: Have GSDML



Figure 94: Updated PROFINET Device Catalog

9.6.8.2 Add the Device to the Configuration

In the **Navigator** panel, right-click on the device and select **Change Module List...** as shown in Figure 95.

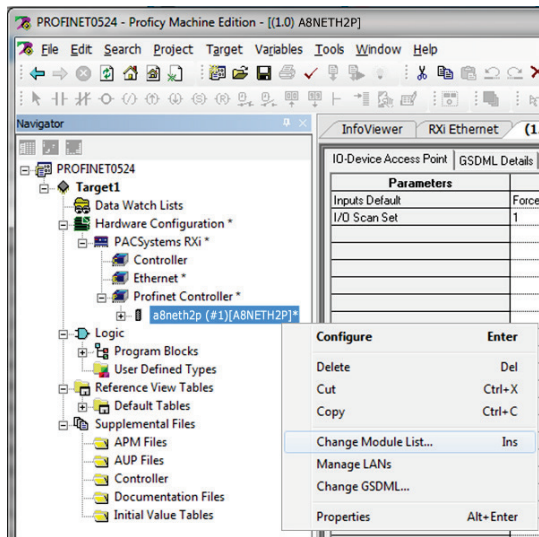


Figure 95: Change Module List

Select a module and drag the module into the available slot. The available slots and modules will vary depending on the specific device. Select a module appropriate for your application. Click the **OK** button as shown in Figure 96.

The module will be reflected in the **Navigator** panel, under the device as shown in Figure 97.

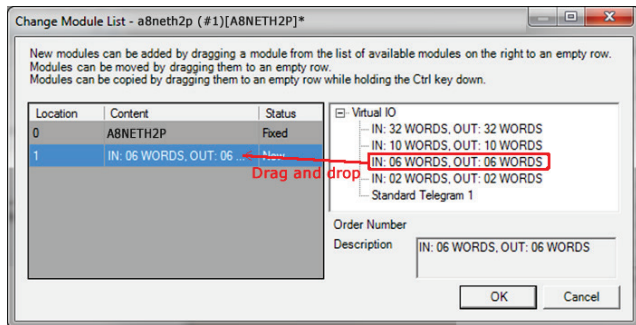


Figure 96: Add IO Module

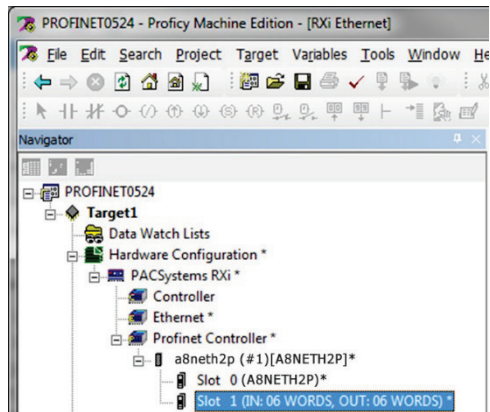


Figure 97: Added IO Module

9.6.8.3 Configure the Device Properties

In the **Navigator** panel, right-click on the device and select **Properties** as shown in Figure 98.

Set the properties to match the configuration on the device. The properties must be appropriate for the application and the PROFINET network.

Set the **Update Rate (ms)**. For this example, the **Update Rate (ms)** is set to “8” ms.

Assign a unique **Device Name**. For this example, the **Device Name** is set to “interfacecard”.

Assign a unique and compatible **IP Address**. For this example the **IP Address** is set to “192.168.17.102”.

The resulting properties are shown in Figure 99.

9.6.8.4 Save the Configuration

The device configuration is now complete. Save and perform any necessary compilation of the configuration. Download the application and configuration to the PLC. The PLC application program can then be started. Please consult with the vendor of your PROFINET PLC software for additional programming and configuration details.

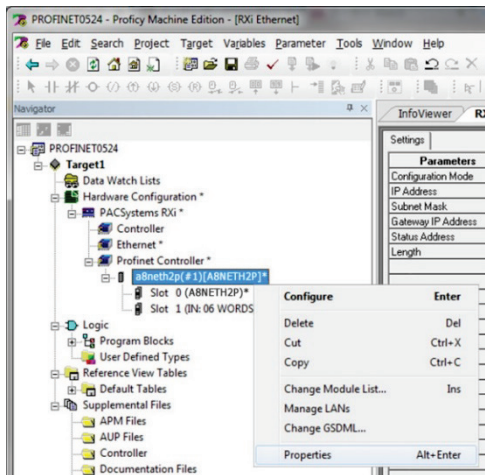


Figure 98: Select Device Properties

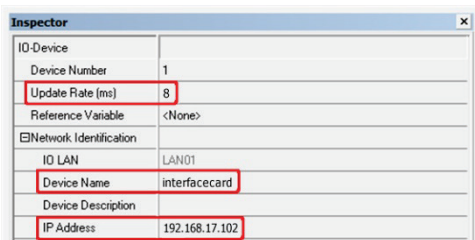


Figure 99: Set Device Properties

9.7 EtherCAT

9.7.1 Overview

The EtherCAT slave driver allows an EtherCAT master (e.g. TwinCAT) to interact with the interface card via cyclic exchange of I/O process data and acyclic requests. The cyclic I/O process data is entirely user-configurable.

Other notes of interest include:

- EtherCAT is a dedicated real-time protocol and cannot coexist with other Ethernet (SUE) protocols.
- A USB connection is required for use with the studio.
- The lowest supported FreeRun Cycle Time is 1ms.
- Supports up to 32 transmit data words and 32 receive data words.
- Slave device profile 5001.
- The ESI file can be obtained from the included CD-ROM or downloaded from the [product web page](#).

9.7.2 Device Settings

In the studio's **Project** panel, navigate to **A8NETH-2P...Ethernet...EtherCAT**.

Currently, there are no configurable device settings.

9.7.3 Transmit and Receive Process Data Word Settings

In the studio's **Project** panel, navigate to **A8NETH-2P...Ethernet...EtherCAT...Transmit Process Data...Transmit Data Word** and/or **Receive Process Data... Receive Data Word**.

The Produced Data Word objects define the structure of status data sent from the inverter to the master. The Consumed Data Word objects will define the structure of the command data sent from the master (for example, a TwinCAT PC) to the inverter. These objects allow the creation of custom-built I/O process data. Up to 32 "command" register values can be sent to the inverter, and up to 32 "status" register values can be sent back to the controller. Therefore, up to 32 Transmit and 32 Receive Data Word objects can be created. The I/O process data format is summarized in Table 42.

Table 42: EtherCAT User Configurable I/O Process Data Format

Consumed Data (Master to Inverter)		Produced Data (Inverter to Master)	
Word Offset	Inverter Register	Word Offset	Inverter Register
0	Any	0	Any
1	Any	1	Any
:	Any	:	Any
30	Any	30	Any
31	Any	31	Any

The default I/O configuration is described in Table 43. Always use the studio to confirm the configuration before commissioning the device.

Table 43: EtherCAT Default User Configurable I/O Process Data Format

Consumed Data (Master to Inverter)		Produced Data (Inverter to Master)	
Word Offset	Inverter Register	Word Offset	Inverter Register
0	1	0	1227
1	1229	1	1231
:	None	:	None

Description

This 32-character (max) field is strictly for user reference: it is not used at any time by the driver.

Register

The inverter register as described in section 4. For the Transmit Data Word object, enter a “status” register to be monitored. For the Receive Data Word object, enter a “command” register that can be written.

Data Type

Each data word is fixed to 16-Bit Unsigned. This is equivalent to two bytes. The data word is transferred in big endian format.

10 TROUBLESHOOTING

Although by no means exhaustive, the following table provides possible causes behind some of the most common errors experienced when using this option card.

Problem	Symptom	Solution
Inverter displays E.1, E.2, or E.3 alarm, or the card is unresponsive	Inverter cannot recognize the option	<ul style="list-style-type: none">• Confirm that the interface card connector is properly seated. Refer to section 2.• Rebooting the interface card via the studio disrupts the communication with the inverter. Reset the fault.• If the card is connected in a ring topology, all devices in the ring must be configured with the same ring redundancy protocol (i.e. MRP, DLR). The appropriate ring redundancy protocol must also be enabled on the card. Otherwise a ring topology will create an Ethernet loop and produce undefined/erratic behavior.

Problem	Symptom	Solution
<p>No communications between the network and the card</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"> • Confirm that the card is running normally (Module Status LED is not blinking red) and connected to the local Ethernet network. • Ensure that the card's is programmed with compatible network settings. Consult with your network administrator to determine the compatible settings. • Confirm that the destination IP address programmed into the controller equipment or computer matches that of the interface card, as displayed by the studio. • Confirm that intermediate firewalls or routers have been configured to allow access to the interface via the applicable TCP/UDP ports. • Try a known working Ethernet cable and switch. • 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.
<p>No PROFINET communication</p>	<p>PROFINET I/O communication cannot be established. The "Network Status" LED is not solid green.</p>	<ul style="list-style-type: none"> • Confirm that the card's PROFINET device name matches the name assigned in the controller's configuration. • Confirm that the card's network settings match the settings assigned in the controller's configuration. • Confirm that the I/O cycle update time is set to 1ms or larger. • Ensure that the card is connected to a 100Mbps full duplex capable switch. • Ensure that the card can be discovered using the controller's discovery tool.

Problem	Symptom	Solution
Unable to control the inverter via network communications	Cannot write to command parameters via network communications, or writing to these parameters has no apparent effect	<ul style="list-style-type: none"> • 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. • If using the inverter's terminal contacts, refer to the inverter's instruction manual to determine the appropriate behavior and priority • Clear all parameter settings to default and reconfigure the parameters.
XML socket connection failed	Message on web server tab information window	TCP port 843 is blocked by a firewall, router or some other intermediate network equipment.
New web server content not loading after web server update	Old web server content is displayed	The internet browser has cached the old web server content. Clear the internet browser's cache before attempting to load the new web server content.
Web page does not display properly	Corrupt web server or outdated flash player plug	<ul style="list-style-type: none"> • Ensure that USB and FTP are disconnected. • Download and install the latest flash player plug from Adobe. • Delete the "WEB" folder from the card's file system and copy a valid default "WEB" folder to the card's file system.
Studio cannot discover the card	The studio does not display the card under "Online Devices"	<ul style="list-style-type: none"> • Confirm that the card is running normally and connected via USB or to the local Ethernet network. It is preferable to connect via USB as there are scenarios in which the Ethernet discovery is not available or disabled. • Confirm that the module and network status LEDs blink the green/red startup sequence when power is first applied. • Add the studio as an exception to the computer's firewall. • Add UDP port 4334 as an exception to the firewall. • Temporarily disable the computer's firewall.

Problem	Symptom	Solution
Studio cannot access file system	The studio displays an error when uploading and downloading the configuration.	If the studio continually displays an error regarding access to the file system, the card's file system may be corrupt. Please format the card's file system and then restore the configuration (refer to section 5.8). If the card cannot be formatted, please contact technical support for instructions to manually format the card's file system.
Firmware-generated error	"MODULE STATUS" LED is flashing red. The number of times the LED flashes indicates an error code.	Record the error code blinking pattern and contact technical support for further assistance.

REVISIONS

Date	Details
June 2016	Initial release
September 2016	Minor wording updates and specifications corrections Update Configuration Studio section
January 2017	Add EtherNet/IP section for explicit messaging using get/set attribute single services