Mitsubishi Ethernet Driver

Table of Contents

Table of Contents 2
- Mitsubishi Ethernet Driver 4
- Overview 4

Device Setup 5
- First Word Low 5
- Communications Parameters 6
- Time and Date Synchronization 7
- Multi-level Networks 7

PLC Setup 9
- A Series PLC Setup 9
- QnA Series PLC Setup 10
- Q Series PLC Setup 12
- Q Series Built-in Ethernet Port PLC Setup 17
- FX3U Series PLC Setup 20

Optimizing Mitsubishi Ethernet Communications 24

Data Types Description 25

Address Descriptions 26
- Mitsubishi A Series Address Descriptions 26
- Mitsubishi Q Series Address Descriptions 28
- Mitsubishi FX3U Series Address Descriptions 32

Error Descriptions 34
- Missing address. 35
- Device address <address> contains a syntax error. 35
- Address <address> is out of range for the specified device or register. 35
- Device address <address> is not supported by model <model name>. 35
- Data Type <type> is not valid for device address <address>. 35
- Device address <address> is read only. 36
- Device <device name> is not responding. 36
- Unable to write to <address> on device <device name>. 36
- Failed to sync time and date for device <device>. Will retry in <time> minutes. 37
- Unable to bind to adapter: <adapter>. Connect failed. 37
- Unable to read from address <start address> to <end address> on device <device name>. 37
- Unable to read from address <start address> to <end address> on device <device name>. Device returned error code <hexadecimal error code>. 37
- Unable to read from address <start address> to <end address> on device <device name>. The device reported an invalid address or an error. 38
- Unable to read tag <tag address> on device <device name>. 38
- Unable to read tag <tag address> on device <device name>. Device returned error code <hexadecimal error code>. 38
- Unable to read tag <tag address> on device <device name>. The device reported an invalid address or an error. 38
- Unable to read to tag(s) on device <device name>. The device returned a PC number error. 39

Unable to write to tag <tag address> on device <device name>. Device must be configured to allow writes while in RUN mode. 39

Unable to write to tag <tag address> on device <device name>. Device returned error code <hexadecimal error code>. 39

Unable to write to tag <tag address> on device <device name>. The device reported an invalid address or an error. 39

Unable to write to tag <tag address> on device <device name>. The device returned a PC number error. 40

Winsock initialization failed (OS Error = n). 40

Winsock V1.1 or higher must be installed to use the Mitsubishi Ethernet Driver 40

Index 41

Mitsubishi Ethernet Driver

Help version 1.068

CONTENTS

Overview
What is the Mitsubishi Ethernet Driver?

Device Setup
How do I configure a device for use with this driver?

PLC Setup
What steps do I need to take in my PLC to use this driver?

Optimizing Mitsubishi Ethernet Communications
How do I get the best performance from the Mitsubishi Ethernet Driver?

Data Types Description
What data types does this driver support?

Address Descriptions
How do I address a data location on a Mitsubishi A Series and Q Series Ethernet devices?

Error Descriptions
What error messages does the Mitsubishi Ethernet Driver produce?

Overview

The Mitsubishi Ethernet Driver provides a reliable way to connect Mitsubishi Ethernet devices to OPC Client applications, including HMI, SCADA, Historian, MES, ERP, and countless custom applications. It is intended for use with Mitsubishi A Series and Mitsubishi Q Series devices communicating via the AJ71E71, A1SJ71E71, AJ71QE71, A1SJ71QE71 or QJ71E71 Ethernet communications cards. A built-in Ethernet Port is supported for Q Series devices. This driver also supports the FX3U series PLC via the FX3U-ENET Ethernet module.

Note: Communications Card model numbers listed are the base model number only. All suffixes are supported.

Device Setup

Supported Devices

A Series PLCs
QnA Series PLCs
Q (Q mode) Series PLCs
FX3U Series PLCs

Communication Protocol

Ethernet: using Winsock V1.1 or higher.
TCP/IP, UDP

Supported Communication Parameters

Binary Format only

Model

A Series for all A Series PLCs
Q Series for all QnA and Q Series PLCs
FX3U for all FX3U Series PLCs

Maximum Number of Channels and Devices

The maximum number of channels that are supported is 256. The maximum amount of devices supported is 255.

Device ID (PLC Network Address)

The Device ID is used to specify the device IP address along with a PC Number and Net Number if the device is a Q series PLC. For more information, refer to the selected model.

- **A Series:** Device IDs are specified as YYY.YYY.YYY.YYY:XXX. The YYY designates the device IP address (each YYY byte should be in the range of 0 to 255). The XXX designates the PC Number of the target device and can be in the range of 0 to 64 or 255 for the local PC.
- **QnA and Q Series:** Device IDs are specified as YYY.YYY.YYY.YYY:Nzzz:XXX or YYY.YYY.YYY.YYY:nzzz:XXX. The YYY designates the device IP address (each YYY byte should be in the range of 0 to 255). The zzz designates the Network Number of the target device and can be in the range of 0 to 255.

  Note: For a local connection, which is network 0, the network number can be omitted, resulting in the format YYY.YYY.YYY.YYY:XXX. The XXX designates the PC Number of the target device and can be in the range of 0 to 64 or 255 for the Local PC. For more information, refer to **Multi-level Networks**.
- **FX3U:** Device IDs are specified as YYY.YYY.YYY.YYY:XXX. The YYY designates the device IP address (each YYY byte should be in the range of 0 to 255). The XXX designates the PC Number of the target device and can be in the range of 0 to 15 or 255 for the local PC.

Connection Timeout

This parameter specifies the time that the driver will wait for a connection to be made with a device. Depending on network load the connect time may vary with each connection attempt. The default setting is 3 seconds. The valid range is 1 to 60 seconds.

Request Timeout

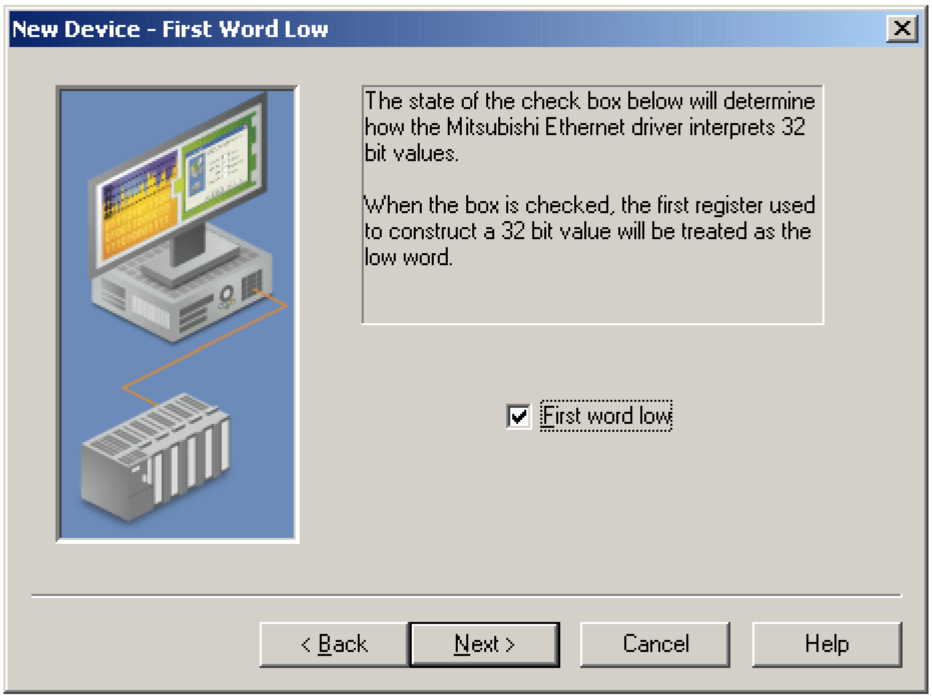
This parameter specifies the time that the driver will wait on a response from the device before giving up and going on to the next request. Longer timeouts only affect performance if a device is not responding. The default setting is 250 milliseconds. The valid range is 50 to 9999 milliseconds.

Retry Attempts

This parameter specifies the number of times the driver will retry a message before giving up and going on to the next message. The default setting is 3 retries. The valid range is 1 to 10.

Note: The AJ71E71, A1SJ71E71, AJ71QE71, A1SJ71QE71 and QJ71E71 families of communications cards occupy ranges of X and Y memory. Writing to this memory with the Mitsubishi Ethernet Driver may disable the card causing a loss of communications. For more information, refer to the communications card manual.

First Word Low



The state of the check box below will determine how the Mitsubishi Ethernet driver interprets 32 bit values.

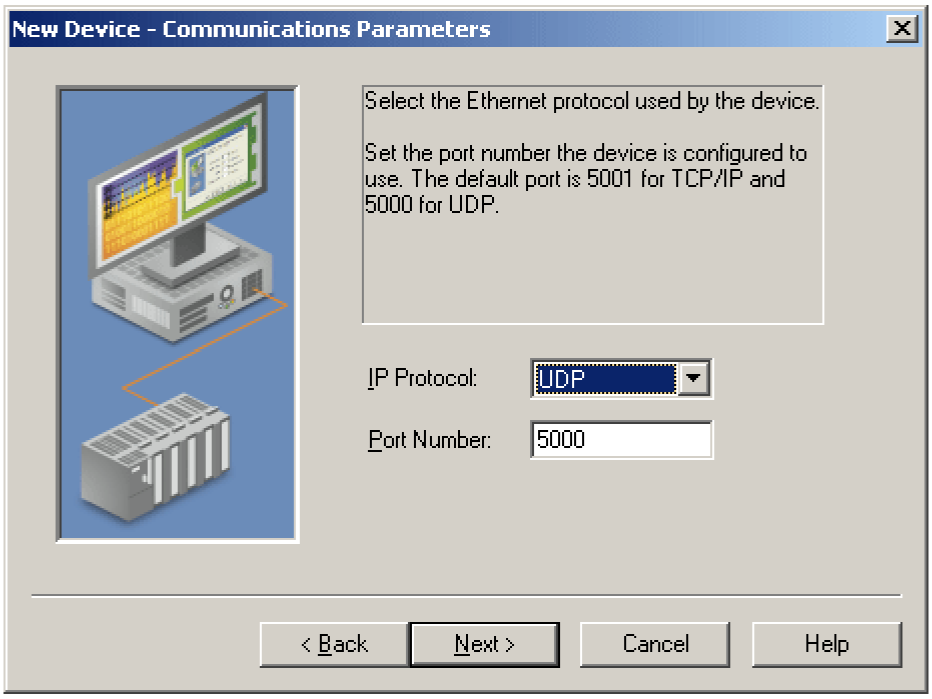
When the box is checked, the first register used to construct a 32 bit value will be treated as the low word.

First word low [✓]

Description of the option is as follows:

- **First Word Low:** In a Mitsubishi Ethernet device, the addresses of two consecutive registers are used for 32-bit data types. When this option is checked, the driver will assume that the first word is low for the 32-bit value. When this option is unchecked, the driver will assume that the first word is high for the 32-bit value. The default setting is checked.

Communications Parameters



Select the Ethernet protocol used by the device.

Set the port number the device is configured to use. The default port is 5001 for TCP/IP and 5000 for UDP.

IP Protocol: UDP

Port Number: 5000

Descriptions of the parameters are as follows:

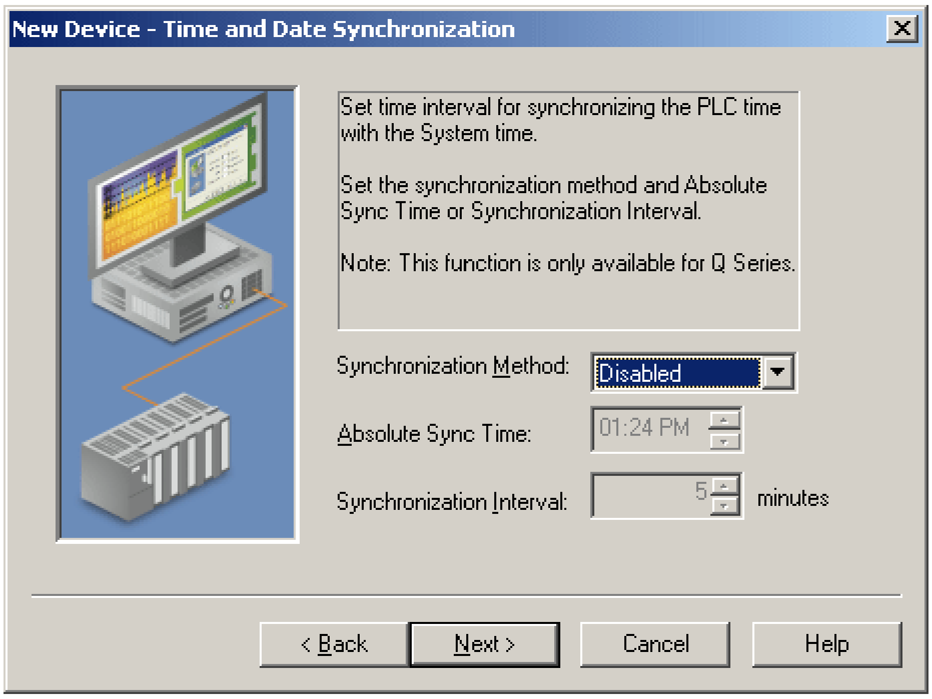
- **IP Protocol:** This parameter specifies the IP protocol. Options include TCP/IP and UDP. TCP/IP is less efficient than UDP and requires a special ladder for network error recovery in the A and QnA series PLCs. Furthermore, Q series users planning to communicate with devices on a remote network must configure multiple ports in the relay device when using TCP/IP. As such, UDP is recommended wherever possible. For more information, refer to **Multi-level Networks**.
- **Port Number:** This parameter specifies the port number. The default setting for UDP is 5000. The default setting for TCP is 5001.

  Note: The default settings are based on GX Developer version 8.25B.

See Also: **PLC Setup**

Time and Date Synchronization

The Time and Date Synchronization options are only available to the Q Series PLCs.


New Device - Time and Date Synchronization

Set time interval for synchronizing the PLC time with the System time.

Set the synchronization method and Absolute Sync Time or Synchronization Interval.

Note: This function is only available for Q Series.

Synchronization Method: Disabled

Absolute Sync Time: 01:24 PM

Synchronization Interval: 5 minutes

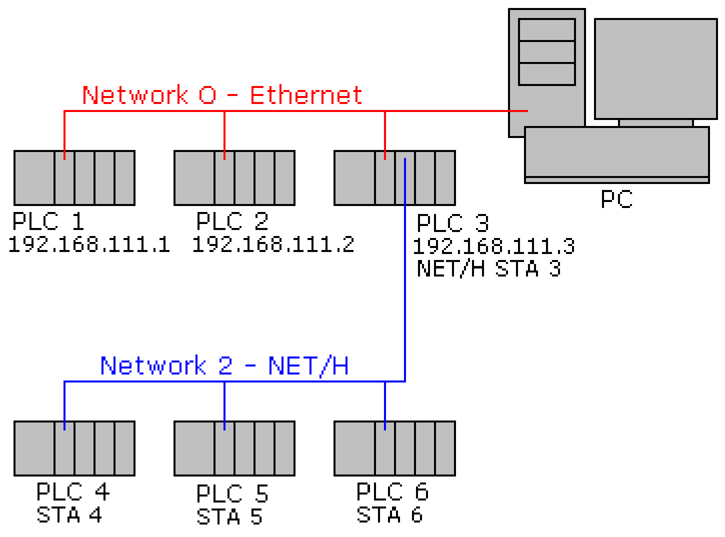
Descriptions of the parameters are as follows:

- **Synchronization Method:** This parameter specifies the synchronization method. Options include Disabled, Interval, and Absolute. The default setting is Disabled.
- **Absolute Sync Time:** This parameter specifies an absolute time that the synchronization will occur at each day. It is only available when the synchronization method is Absolute.
- **Synchronization Interval:** This parameter specifies the interval of time between synchronizations. The driver can periodically synchronize a Q Series PLC's time and date with the system time and date of the host computer. The valid range is 5 to 1440 minutes (24 hours). The default setting is 5 minutes. It is only available when the synchronization method is Interval.

  Note: For example, if 240 minutes is entered, the driver will set the PLC date and time every 4 hours.

Multi-level Networks

The Q Series model is used to communicate with devices on remote networks. In the example shown below, PLC 1, PLC 2 and PLC 3 are on the local Ethernet network (Network 0). PLC 4, PLC 5 and PLC 6 are on a remote NET/H network. PLC 3 serves as a relay device connecting the two networks.



If PLC 1, PLC 2 and PLC 3 have QJ71E71-100 Ethernet modules configured with IPs 192.168.111.1, 192.168.111.2 and 192.168.111.3 respectively. In addition to the Ethernet module, PLC 3 also has a QJ71BR11 NET/H module configured as station 3. Assume that PLC 4, PLC 5 and PLC 6 have NET/H modules configured as stations 4, 5 and 6 respectively.

To communicate with all six PLCs, six devices would need to be created in the server project. The Device IDs would be as follows:

PLC	Device ID	Comment
1	192.168.111.1:N0:255*	Local network, local PC
2	192.168.111.2:N0:255*	Local network, local PC
3	192.168.111.3:N0:255*	Local network, local PC
4	192.168.111.3:N2:4	Network 2, PC 4, via PLC 3
5	192.168.111.3:N2:5	Network 2, PC 5, via PLC 3
6	192.168.111.3:N2:6	Network 2, PC 6, via PLC 3

*This example shows :N0 as the network number for the local network. It is also possible to omit the network number when it is Network 0 (local network), thus, the Device ID 192.168.111.1:255 would also be valid in this case.

Note 1: For performance and reliability, the driver is designed to use a separate socket for each device. Thus, if TCP/IP is used, the relay device in this example would need to have at least 4 ports configured - one to connect to each of the driver's sockets for PLC 3, PLC 4, PLC 5 and PLC 6. However, only a single port needs to be configured in the relay device if UDP and the "unspecified" destination IP (255.255.255.255) and port number (0xFFFF) are being used. Therefore, UDP is generally recommended for this type of application. For more information, refer to **PLC Setup**.

Note 2: A relay device may take 5 or more seconds to report a failed read and write to a remote device. It is recommended that the request timeout for remote devices be set accordingly. For more information, refer to **Device Setup**.

PLC Setup

The hardware must be configured before Ethernet communications is possible. For information on a specific hardware series, select a link from the list below.

A Series PLC Setup
QnA Series PLC Setup
Q Series PLC Setup
Q Series Built-in Ethernet Port PLC Setup
FX3U Series PLC Setup

A Series PLC Setup

Hardware Settings

The DIP switches on the AJ71E71 Ethernet interface card must be set as follows.

- DIP switches 1-6 must be set to OFF.
- DIP switch 7 must be set to ON.
- DIP switch 8 must be set to OFF.

Ladder Program

The Mitsubishi A Series PLC requires that a ladder program be used to initialize the AJ71E71 or A1SJ71E71 Ethernet interface card and define the desired open system. TCP/IP and UDP open systems may be used with this driver. In the case of TCP/IP, error handling code should also be implemented.

Note: TCP/IP is less efficient than UDP and requires special ladder to handle network error recovery. Also, if planning to communicate with devices on a remote network, TCP/IP requires that multiple ports be configured in the relay device. Thus, UDP is recommended wherever possible. For more information, refer to **Multi-level Networks**.

Initialization Ladder

The following initialization code sets the IP address of the device and triggers execution of the open code. For this example, an IP of 192.168.111.123 (C0.A8.6F.7B Hex) is assumed.

```
| M9038                                 H             |
+-| |--+--------------------------[DMOV C0A86F7B D100]|
|      |                               H    K       K |
+      +--------------------------[TO 0000 0  D100 2 ]|
|      |                                              |
+      +------------------------------------[SET M40 ]|
| M40                                                 |
+-| |------------------------------------------<Y0019>|
| X0019  Y0019                                        |
+-| |----| |--------------------------------[PLS M41 ]|
| M41                                                 |
+-| |---------------------------------------[SET M42 ]|
|                                                     |
```

Open and Error Handling Ladder for TCP/IP

The following open and error handling code assumes TCP/IP communications, unpassive mode, on port 5001 (1389 Hex).

This code is for the first communications buffer of the AJ71E71 card. Similar code must be implemented for each additional buffer needed. Simply ensure that the proper interface bits are used as well as separate error handling bits and timers for each buffer.

Note: It is strongly recommended that users follow the code fragment as closely as possible. Without proper error handling and recovery on the PLC side of the connection, communications with the PLC may not be reestablished after a physical error, such as a cable break, occurs. Without the error handling represented here, PLC might have to be reset in order to reestablish communications.

```
| M42   X0010  Y0008                  H    K  H    K  |
+-| |---|/|----|/|-----+----------[TO 0000 16 8002 1 ]|
|                      |              H    K  H    K  |
+                      +----------[TO 0000 24 1389 1 ]|
|                      |                              |
+                      +-------------------[SET Y0008]|
| X0010                                               |
+-| |---------------------------------------[PLF M50 ]|
| M50                                                 |
+-| |--+-----------------------------------[RST Y0008]|
|      |                                              |
+      +------------------------------------[RST M42 ]|
|      |                                              |
+      +------------------------------------[SET M51 ]|
| M51                                             K20 |
+-| |--------------------------------------------<TO >|
| TO                                                  |
+-| |--+------------------------------------[RST M51 ]|
|      |                                              |
+      +------------------------------------[SET M42 ]|
```

Given the ladder fragment shown here for TCP/IP port operation, the AJ71E71 will be forced to close and re-enable the port for a connection if the current connection is lost. This will occur 2 seconds after the error is detected as controlled by T0. Reloading the port mode and port number and the set of Y008 resets the port.

Open Ladder for UDP

The following open code assumes UDP communications on port 5000 (1388 Hex). The UDP open system requires that the destination address be specified. This would be the IP and port that the driver will use to communicate with the PLC. To prevent issues with conflicting port usage, the Mitsubishi Ethernet Driver allows Windows to assign any unused UDP port to each device configured in the driver on startup. Thus, the port that the driver will use is not predictable. Therefore, the destination port must be configured in the PLC as "unspecified". This is done by entering FFFF (Hex) as shown below. The exact IP address that the driver will use may be specified. This example assumes 192.168.111.24 (C0.A8.6F.18 Hex). However, the destination may also be left as "unspecified" with 255.255.255.255 (FF.FF.FF.FF Hex).

Note: If a specific IP address is put into the ladder code, only the machine with that IP address will be able to communicate with the PLC via UDP. If the IP address is left as "unspecified," then any IP address can communicate with the PLC.

```
| M42   X0010  Y0008                  H    K  H    K  |
+-| |---|/|----|/|-----+----------[TO 0000 16 110  1 ]|
|                      |              H    K  H    K  |
+                      +----------[TO 0000 24 1388 1 ]|
|                      |              H    K  H    K  |
+                      +----------[TO 0000 25 6F18 1 ]|
|                      |              H    K  H    K  |
+                      +----------[TO 0000 26 C0A8 1 ]|
|                      |              H    K  H    K  |
+                      +----------[TO 0000 27 FFFF 1 ]|
|                      |                              |
+                      +-------------------[SET Y0008]|
```

QnA Series PLC Setup

Hardware Settings

The DIP switches on the A1SJ71QE71 Ethernet interface card must be set as follows:

- DIP switches 1-2 must be set to OFF.
- DIP switch 3 must be set to ON.
- DIP switches 4-6 must be set to OFF.

- DIP switch 7 must be set to ON.
- DIP switch 8 must be set to OFF.

Ladder Program

The Mitsubishi QnA Series PLC requires that a ladder program be used to initialize the AJ71QE71 or A1SJ71QE71 Ethernet interface card and define the desired open system. TCP/IP and UDP open systems may be used with this driver. In the case of TCP/IP, error handling code should also be implemented. Note that TCP/IP is less efficient than UDP and requires a special ladder to handle network error recovery. Also, if planning to communicate with devices on a remote network, TCP/IP requires that multiple ports be configured in the relay device. Thus, UDP is recommended wherever possible. For more information, refer to **Multi-level Networks**.

Note: Power must be cycled to the PLC in order for any network configuration to take effect.

Initialization Ladder

The following initialization code sets the IP address of the device and triggers execution of the open code. For this example, an IP of 192.168.111.123 (C0.A8.6F.7B Hex) is assumed.

```
| SM1038                                 H             |
+-| |--+--------------------------[DMOV C0A86F7B D100]|
|       |                             H    K        K  |
+       +--------------------------[TO 0000 0  D100 2 ]|
|       |                                              |
+       +------------------------------------[SET M40 ]|
| M40                                                  |
+-| |------------------------------------------<Y0019>|
| X0019  Y0019                                         |
+-| |----| |---------------------------------[PLS M41 ]|
| M41                                                  |
+-| |----------------------------------------[SET M42 ]|
|                                                      |
```

Open and Error Handling Ladder for TCP/IP

The following open and error handling code assumes TCP/IP communications, unpassive mode, on port 5001 (1389 Hex).

This code is for the first communications buffer of the A1SJ71QE71 card. Similar code must be implemented for each addition buffer needed. Simply ensure that the proper interface bits are used as well as separate error handling bits and timers for each buffer.

Note: It is strongly recommended that users follow the code fragment as closely as possible. Without proper error handling and recovery on the PLC side of the connection, communications may not able to be reestablished with the PLC after a physical error, such as a cable break, occurs. Without the error handling represented here, the PLC might need to be reset in order to reestablish communications.

```
| M42    X0010   Y0008                   H    K  H     K  |
+-| |---|/|----|/|-----+----------[TO 0000 32 8000 1 ]|
|                      |                 H    K  H     K  |
+                      +----------[TO 0000 40 1389 1 ]|
|                      |                                  |
+                      +------------------[SET Y0008]|
| X0010                                                   |
+-| |-------------------------------------[PLF M50 ]|
| M50                                                     |
+-| |--+----------------------------------[RST Y0008]|
|      |                                                  |
+      +----------------------------------[RST M42 ]|
|      |                                                  |
+      +----------------------------------[SET M51 ]|
| M51                                              K20 |
+-| |---------------------------------------------<TO >|
| T0                                                      |
+-| |--+----------------------------------[RST M51 ]|
|      |                                                  |
+      +----------------------------------[SET M42 ]|
```

Given the ladder fragment shown here for TCP/IP port operation, the A1SJ71QE71 will be forced to close and re-enable the port for a connection if the current connection is lost. This will occur 2 seconds after the error is detected as controlled by T0. Reloading the port mode and port number and the set of Y008 resets the port.

Open Ladder for UDP

The following open code assumes UDP communications on port 5000 (1388 Hex). The UDP open system requires that the destination address be specified. This would be the IP and port that the driver will use to communicate with the PLC. To prevent issues with conflicting port usage, the Mitsubishi Ethernet Driver allows Windows to assign any unused UDP port to each device configured in the driver on startup. Thus, the port that the driver will use is not predictable. Users must configure the destination port in the PLC as "unspecified". This is done by entering FFFF (Hex) as shown below. The exact IP address the driver will use may be specified. This example assumes 192.168.111.24 (C0.A8.6F.18 Hex). However, The destination may also be left as "unspecified" with 255.255.255.255 (FF.FF.FF.FF Hex).

Note: If a specific IP address is put into the ladder code, only the machine with that IP address will be able to communicate with the PLC via UDP. If the IP address is left as "unspecified," then any IP address can communicate with the PLC.

```
| M42    X0010   Y0008                   H    K  H     K  |
+-| |---|/|----|/|-----+----------[TO 0000 32 110  1 ]|
|                      |                 H    K  H     K  |
+                      +----------[TO 0000 40 1388 1 ]|
|                      |                 H    K  H     K  |
+                      +----------[TO 0000 41 6F18 1 ]|
|                      |                 H    K  H     K  |
+                      +----------[TO 0000 42 C0A8 1 ]|
|                      |                 H    K  H     K  |
+                      +----------[TO 0000 43 FFFF 1 ]|
|                      |                                  |
+                      +------------------[SET Y0008]|
```

Q Series PLC Setup

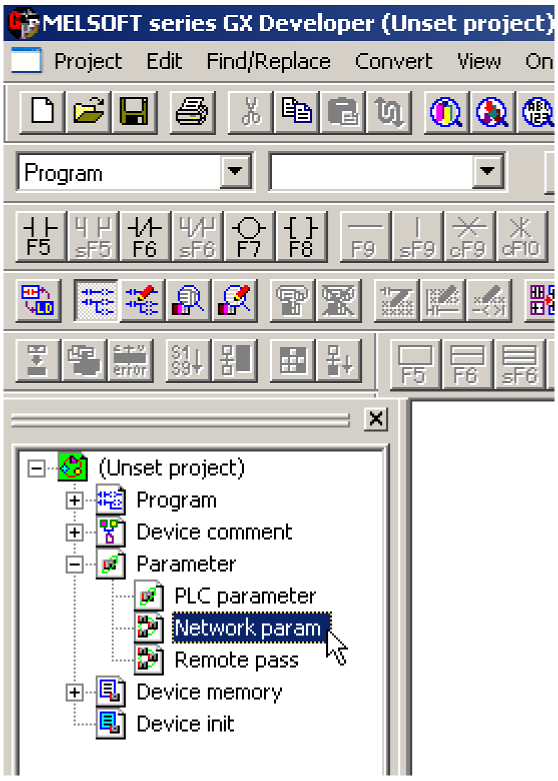
Unlike the A and QnA series, the newest Q Series Ethernet modules (QJ71E71-100) do not have DIP switches that need to be set. Furthermore, special ladder logic to enable Ethernet communications is not required. Users must set network related parameters in the controller, however, using the Mitsubishi GX Developer software. Ports may be configured to use TCP/IP or UDP.

Note:

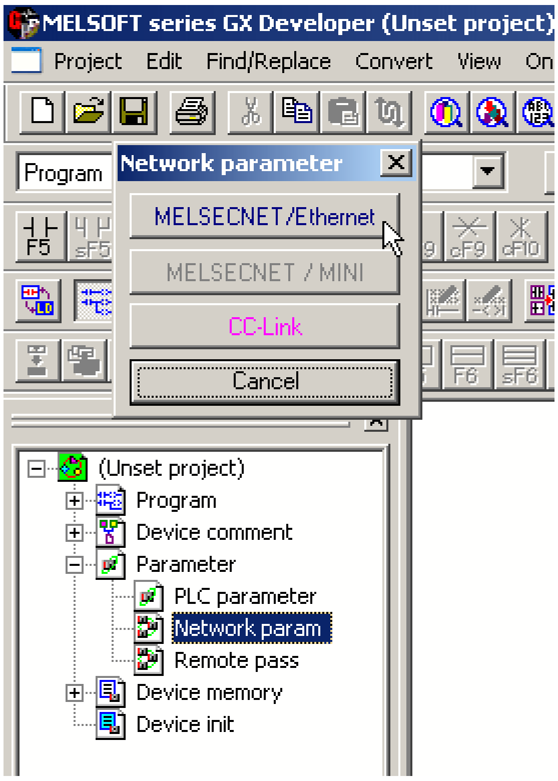
TCP/IP is less efficient than UDP. Users planning to communicate with devices on a remote network should note that TCP/IP requires multiple ports be configured in the relay device. Thus, UDP is recommended wherever possible. For more information, refer to **Multi-level Networks**.

Device Configuration

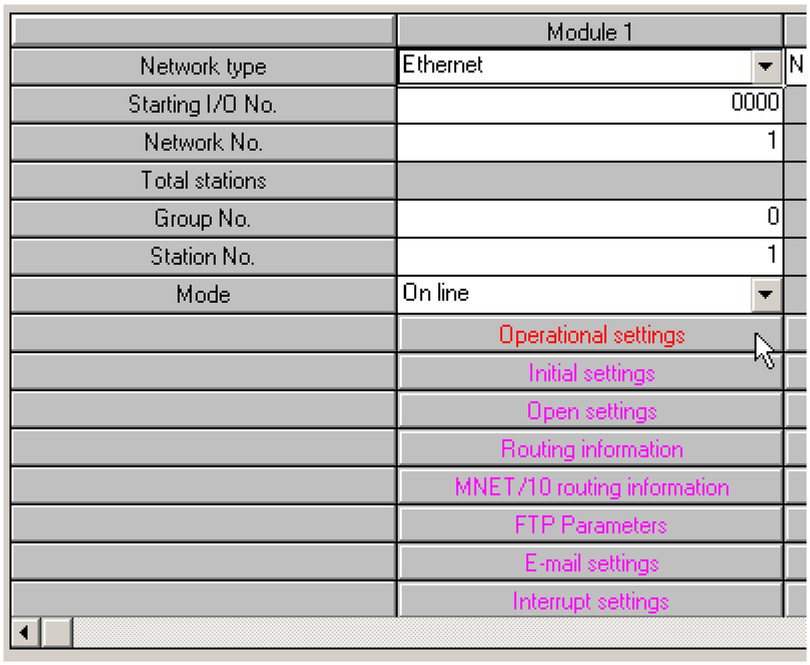
1. To start, create a new GX Developer project for a Q Series (Q mode) PLC. Alternatively, open and edit an existing project.
2. Next, select **Network Param**.



3. In **Network Parameter**, click **MELSECNET/Ethernet**.



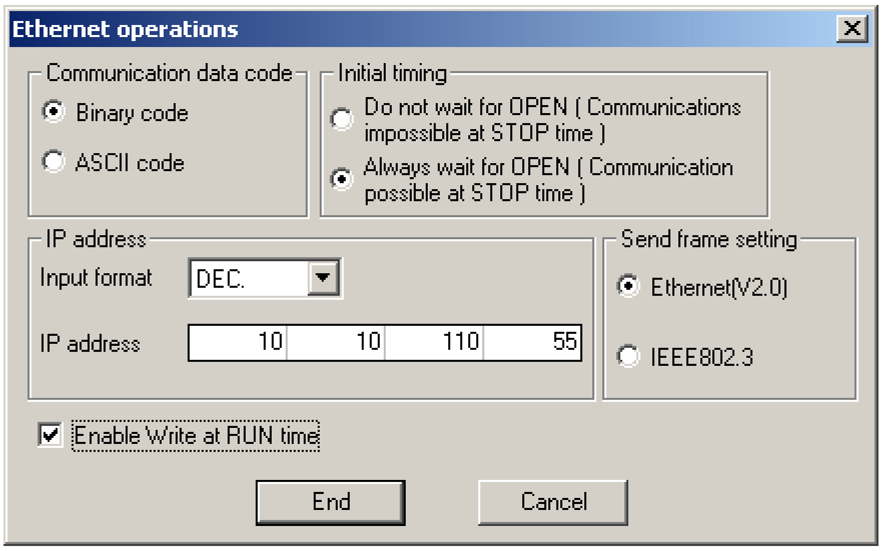
4. Fill in the required information for the Ethernet module. Although the network type must be Ethernet, other settings will depend on the particular application. The example below is for station 1 on network 1. The starting I/O No. is 0 in this case because the QJ71E71 Ethernet module is installed in the slot adjacent to the CPU. If there are other modules between the CPU and Ethernet unit, determine the total I/O mapped to those and set the starting I/O of the Ethernet unit accordingly. Once these basic network settings are specified, click on **Operational Settings**.



	Module 1
Network type	Ethernet
Starting I/O No.	0000
Network No.	1
Total stations	
Group No.	0
Station No.	1
Mode	On line
	Operational settings
	Initial settings
	Open settings
	Routing information
	MNET/10 routing information
	FTP Parameters
	E-mail settings
	Interrupt settings

5. The **Ethernet Operations** dialog is used to define the device's IP address. Except for the IP address, the settings should be as shown below.

   Note: Unless security or safety concerns require otherwise, make sure "Enable Write at RUN time" is checked. If this is left unchecked, all writes will fail when the PLC is in Run mode.



Ethernet operations

Communication data code: [● Binary code] ○ ASCII code

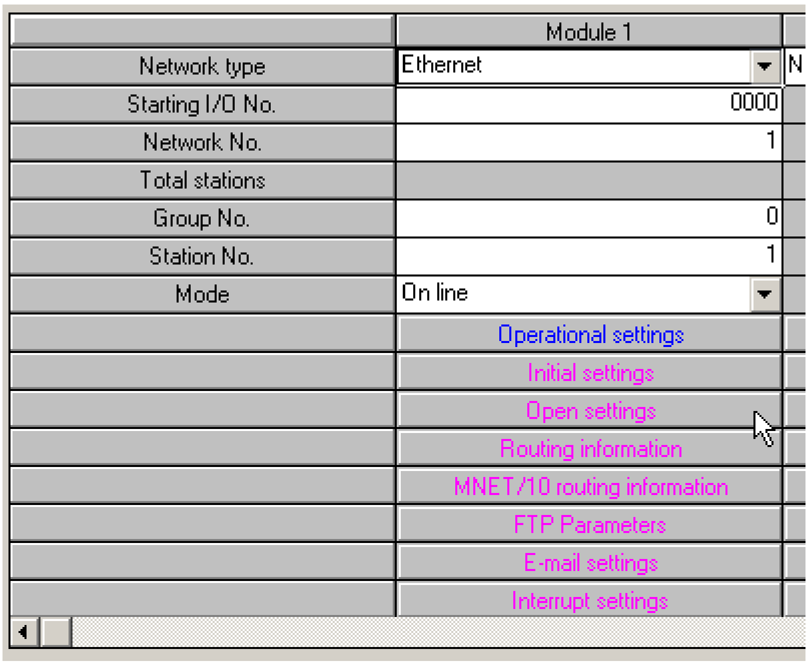
Initial timing: ○ Do not wait for OPEN (Communications impossible at STOP time) [● Always wait for OPEN (Communication possible at STOP time)]

IP address — Input format: DEC. — IP address: 10 10 110 55

Send frame setting: [● Ethernet(V2.0)] ○ IEEE802.3

[✓] Enable Write at RUN time

6. Click **End**.
7. Upon returning to the basic network parameters dialog, click **Open settings**.

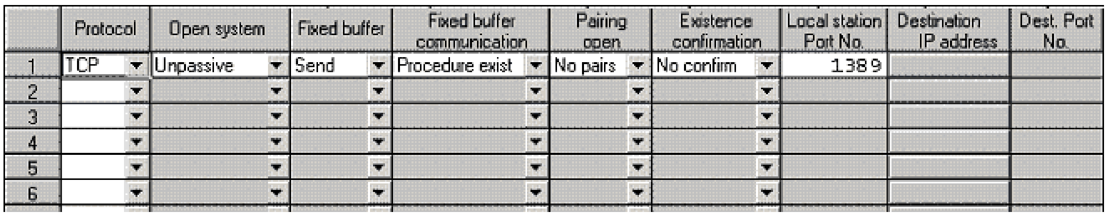


	Module 1
Network type	Ethernet
Starting I/O No.	0000
Network No.	1
Total stations	
Group No.	0
Station No.	1
Mode	On line
	Operational settings
	Initial settings
	Open settings
	Routing information
	MNET/10 routing information
	FTP Parameters
	E-mail settings
	Interrupt settings

8. Specify the desired open settings. These depend on the chosen IP protocol, which may be TCP or UDP.

Open Settings for TCP

Enter **TCP** for the protocol. For simplicity, the **Unpassive** open system is recommended. By using the unpassive open system, users will not have to configure the IP and port that the driver will use. In the example below, the local port number 5001 (1389 Hex) is specified.

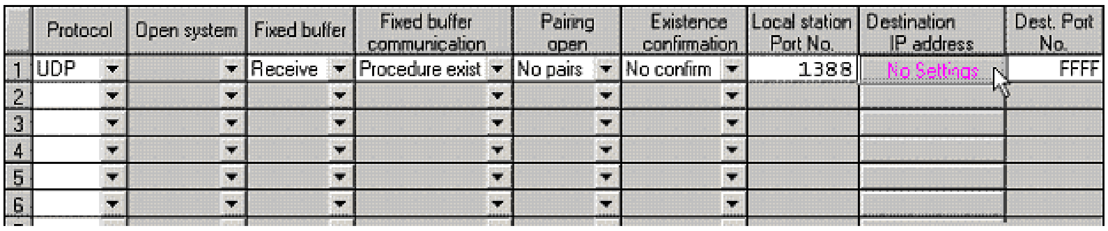


	Protocol	Open system	Fixed buffer	Fixed buffer communication	Pairing open	Existence confirmation	Local station Port No.	Destination IP address	Dest. Port No.
1	TCP	Unpassive	Send	Procedure exist	No pairs	No confirm	1389		
2									
3									
4									
5									
6									

Tip:
Consult the Knowledge Base and the Mitsubishi Technical Bulletin "Existence Confirmation Configuration using Fixed Buffer Communications with a QJ71E71-100 Ethernet Module" for detailed information about device configuration for TCP communications.

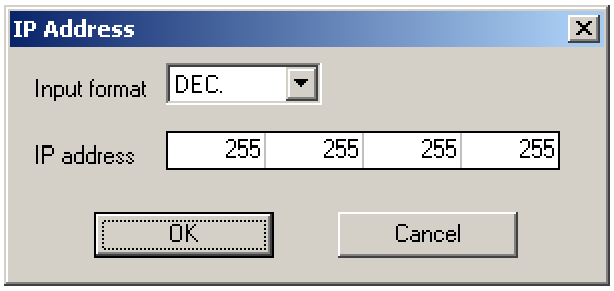
Open Settings for UDP

1. Enter **UDP** for the protocol. There are no open system options for UDP. In the example below, the local port number 5000 (1388 Hex) is specified.
2. Next, specify the destination IP and port. This would be the IP and port that the driver will use to communicate with the PLC. To prevent issues with conflicting port usage, the Mitsubishi Ethernet Driver allows Windows to assign any unused UDP port to each device configured in the driver on startup. Thus, the port that the driver will use is not predictable. Users must configure the destination port in the PLC as "unspecified". This is done by entering FFFF (Hex) as shown below.
3. Finally, click on the Destination IP address button.



	Protocol	Open system	Fixed buffer	Fixed buffer communication	Pairing open	Existence confirmation	Local station Port No.	Destination IP address	Dest. Port No.
1	UDP		Receive	Procedure exist	No pairs	No confirm	1388	No Settings	FFFF
2									
3									
4									
5									
6									

4. Either specify the IP address that the driver will be using or leave it at the "unspecified" address of 255.255.255.255 as shown below.



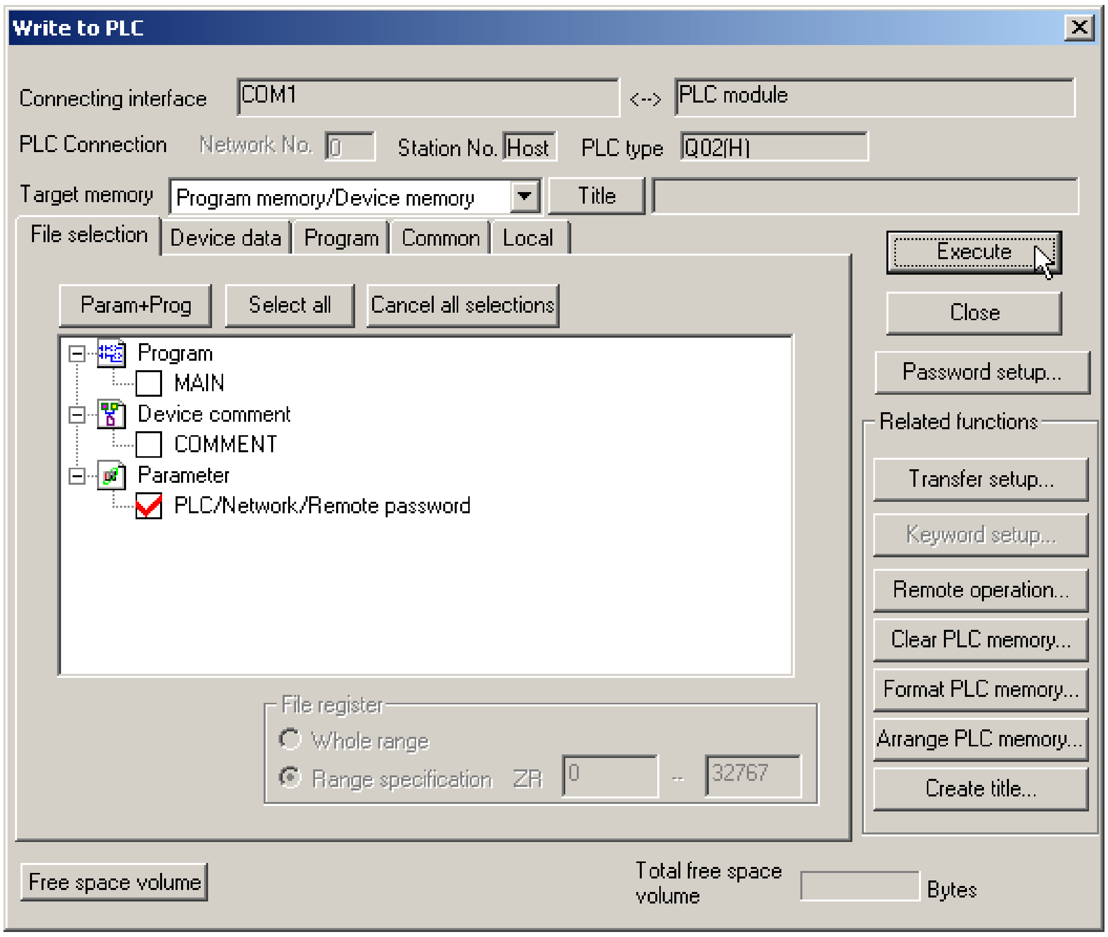
IP Address

Input format: DEC.

IP address: 255 255 255 255

Write Network Parameters to PLC

After all of the network parameters have been specified, they must be written to the PLC. This can be done by selecting the **Online** | **Write To PLC...** menu option. Check the network parameters file selection and then click **Execute**.



Note:
Users must cycle the power on the PLC in order for the network parameter changes to take effect.

Q Series Built-in Ethernet Port PLC Setup

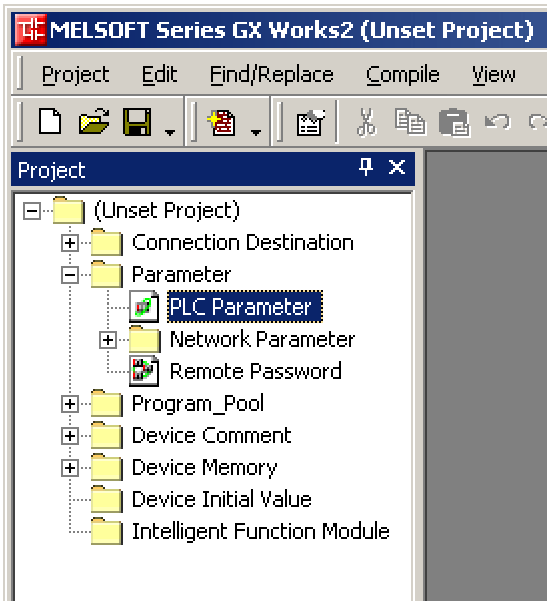
In order for the Mitsubishi Ethernet Driver to communicate with the Mitsubishi Q Series CPU's built-in Ethernet port, some network parameters must be configured in the PLC.

Device Configuration

The following instructions were created using Mitsubishi GX Works2 software.

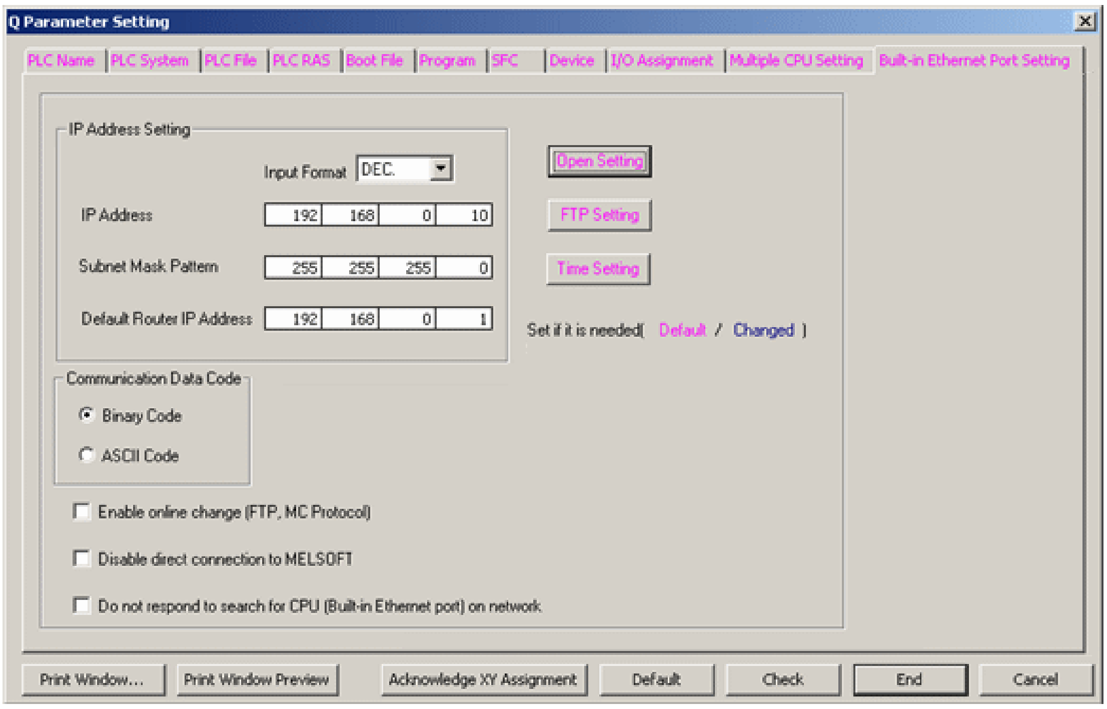
1. To start, create a new project for a Q Series (Q mode) PLC. Alternatively, open and edit an existing project.

2. Next, select **PLC Parameter**.



3. Open the **Built-in Ethernet Port Setting** tab, and then make the following changes:

- Beneath **IP Address Setting**, fill in all required information.
- Beneath **Communication Data Code**, select **Binary Code**.


Q Parameter Setting

PLC Name | PLC System | PLC File | PLC RAS | Boot File | Program | SFC | Device | I/O Assignment | Multiple CPU Setting | Built-in Ethernet Port Setting

IP Address Setting

Input Format DEC.

IP Address 192 168 0 10

Subnet Mask Pattern 255 255 255 0

Default Router IP Address 192 168 0 1

Open Setting

FTP Setting

Time Setting

Set if it is needed(Default / Changed)

Communication Data Code

(●) Binary Code

( ) ASCII Code

[ ] Enable online change (FTP, MC Protocol)

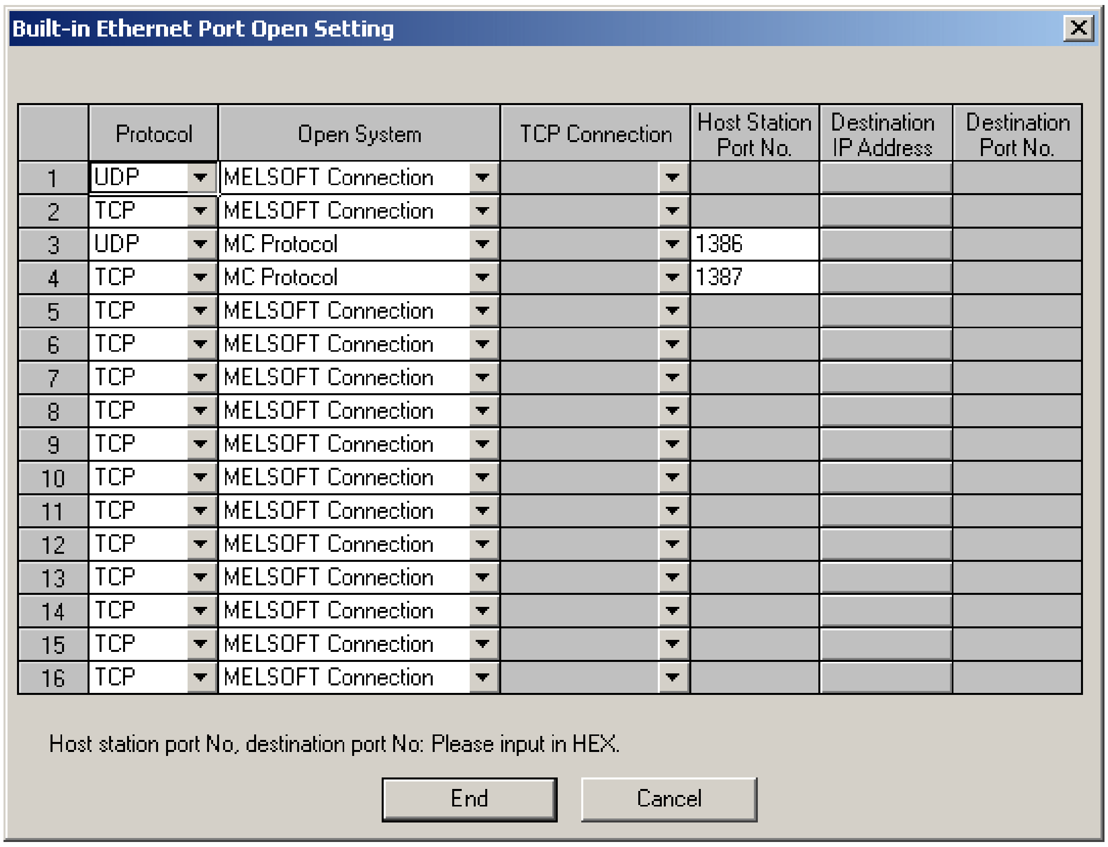
[ ] Disable direct connection to MELSOFT

[ ] Do not respond to search for CPU (Built-in Ethernet port) on network

Print Window... | Print Window Preview | Acknowledge XY Assignment | Default | Check | End | Cancel

4. Next, click **Open Setting**, and then make the following changes:

- Specify the **Protocol**. Options include **UDP** or **TCP**.
- Specify the **Open System** as **MC Protocol**.
- Specify the **Host Station Port No**.


Built-in Ethernet Port Open Setting

	Protocol	Open System	TCP Connection	Host Station Port No.	Destination IP Address	Destination Port No.
1	UDP	MELSOFT Connection				
2	TCP	MELSOFT Connection				
3	UDP	MC Protocol		1386		
4	TCP	MC Protocol		1387		
5	TCP	MELSOFT Connection				
6	TCP	MELSOFT Connection				
7	TCP	MELSOFT Connection				
8	TCP	MELSOFT Connection				
9	TCP	MELSOFT Connection				
10	TCP	MELSOFT Connection				
11	TCP	MELSOFT Connection				
12	TCP	MELSOFT Connection				
13	TCP	MELSOFT Connection				
14	TCP	MELSOFT Connection				
15	TCP	MELSOFT Connection				
16	TCP	MELSOFT Connection				

Host station port No, destination port No: Please input in HEX.

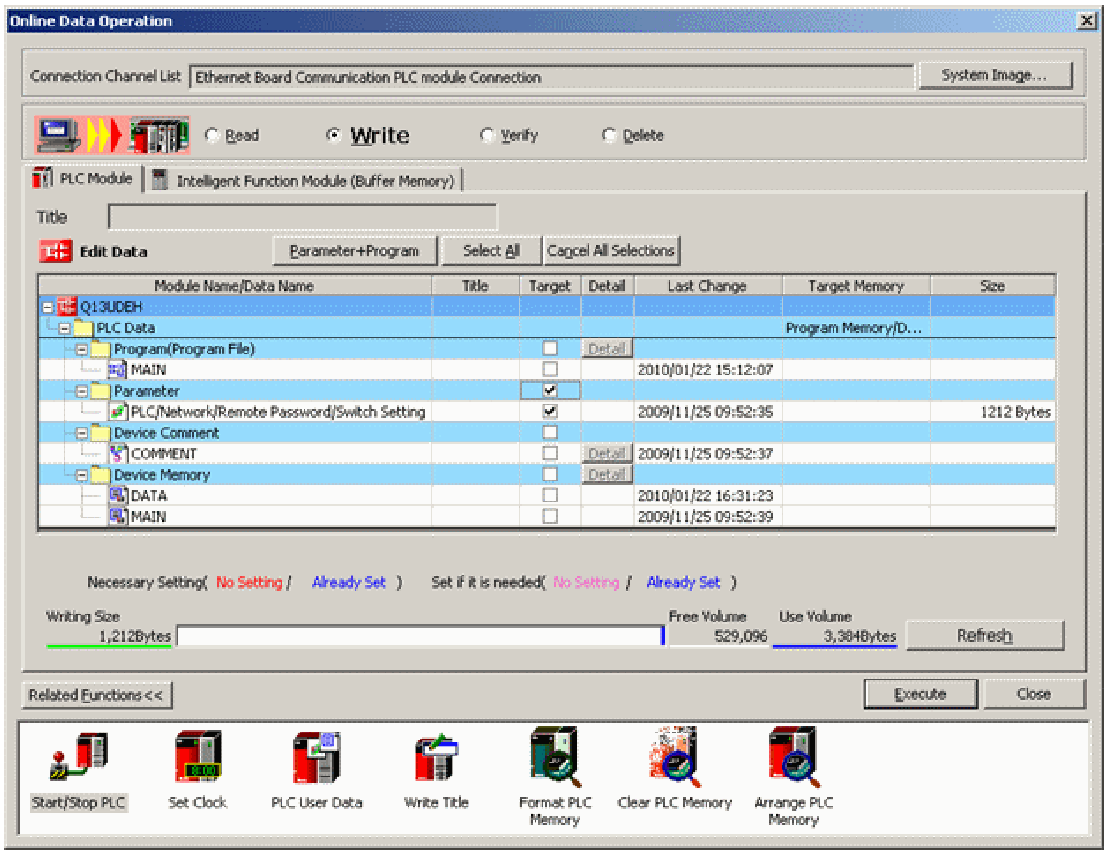
Note: In the example above, the local port numbers 4998 (1386H) and 4999 (1387H) are used.

Important: The driver's default port settings of 5000 UDP and 5001 TCP are not valid port numbers for the built-in Ethernet port. The driver uses decimal numbers for the port number; GX Works2 uses hexadecimal number for the port numbers. Valid port number setting ranges are 0401H (1025) to 1387H (4999), and 1392H (5010) to FFFEH (65534).

5. Click **End**.

Writing the Network Parameters to the PLC

After all network parameters have been specified, they must be written to the PLC. To do so, click **Online** | **Write To PLC...**. Then, check **Parameter** (located beneath **Target**) and then click **Execute**.


Online Data Operation

Connection Channel List: Ethernet Board Communication PLC module Connection

System Image...

Read / [✓ Write] / Verify / Delete

PLC Module | Intelligent Function Module (Buffer Memory)

Title:

Edit Data — Parameter+Program | Select All | Cancel All Selections

Module Name/Data Name	Title	Target	Detail	Last Change	Target Memory	Size
Q13UDEH						
PLC Data					Program Memory/D...	
Program(Program File)		☐	Detail			
MAIN		☐		2010/01/22 15:12:07		
Parameter		✓				
PLC/Network/Remote Password/Switch Setting		✓		2009/11/25 09:52:35		1212 Bytes
Device Comment		☐				
COMMENT		☐	Detail	2009/11/25 09:52:37		
Device Memory		☐	Detail			
DATA		☐		2010/01/22 16:31:23		
MAIN		☐		2009/11/25 09:52:39		

Necessary Setting(No Setting / Already Set) Set if it is needed(No Setting / Already Set)

Writing Size 1,212Bytes

Free Volume 529,096 Use Volume 3,384Bytes

Refresh

Related Functions<< Execute Close

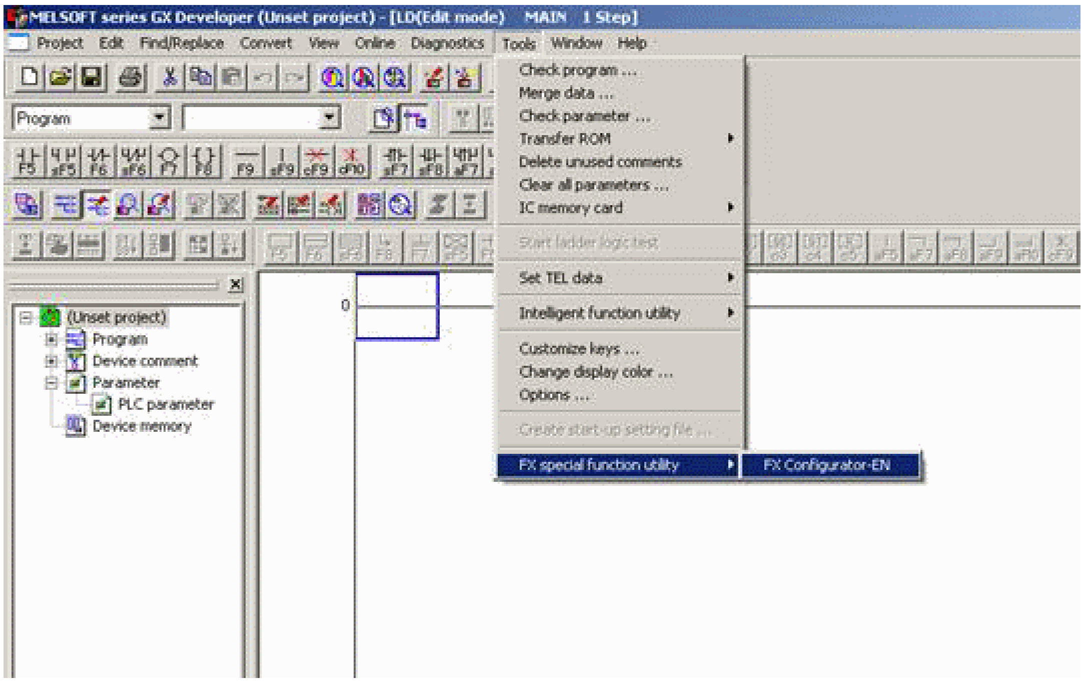
Note: Users must cycle the power on the PLC in order for the network parameter changes to take effect.

FX3U Series PLC Setup

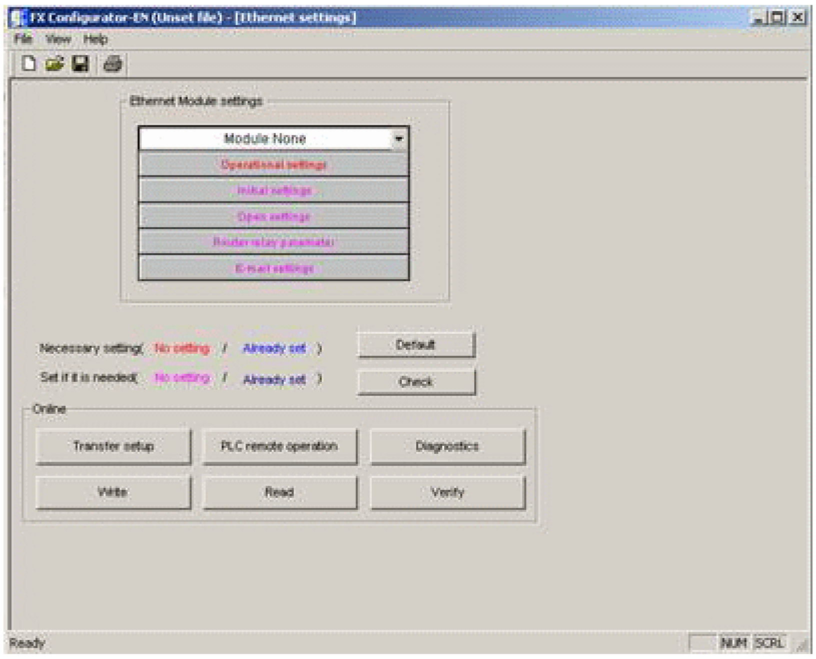
In order for the Mitsubishi Ethernet Driver to communicate with the FX3U PLC via the FX3U-ENET block, some network parameters have to be configured in the FX3U PLC. The Mitsubishi GXDeveloper-FX software is necessary for the following process.

Device Configuration

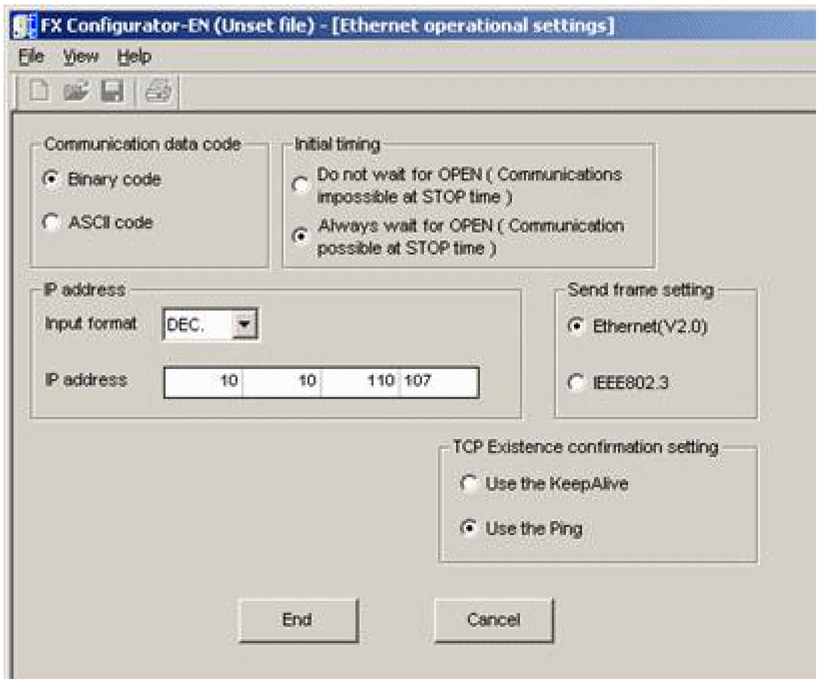
1. To start, create a new GXDeveloper project for a FX3U model. Then click **Tools** | **FX Special Function Utility**.
2. Next, select **FX Configurator-EN**.



Note: The **FX Configurator-EN** dialog should appear as shown below.



3. Next, specify the FX3U-ENET block's minimum required configuration information. Select a module from the first drop-down list and then click **Operational Settings**.



FX Configurator-EN (Unset file) - [Ethernet operational settings]

Communication data code: [✓ Binary code] ASCII code

Initial timing: Do not wait for OPEN (Communications impossible at STOP time) [✓ Always wait for OPEN (Communication possible at STOP time)]

IP address — Input format: DEC. — IP address: 10 10 110 107

Send frame setting: [✓ Ethernet(V2.0)] IEEE802.3

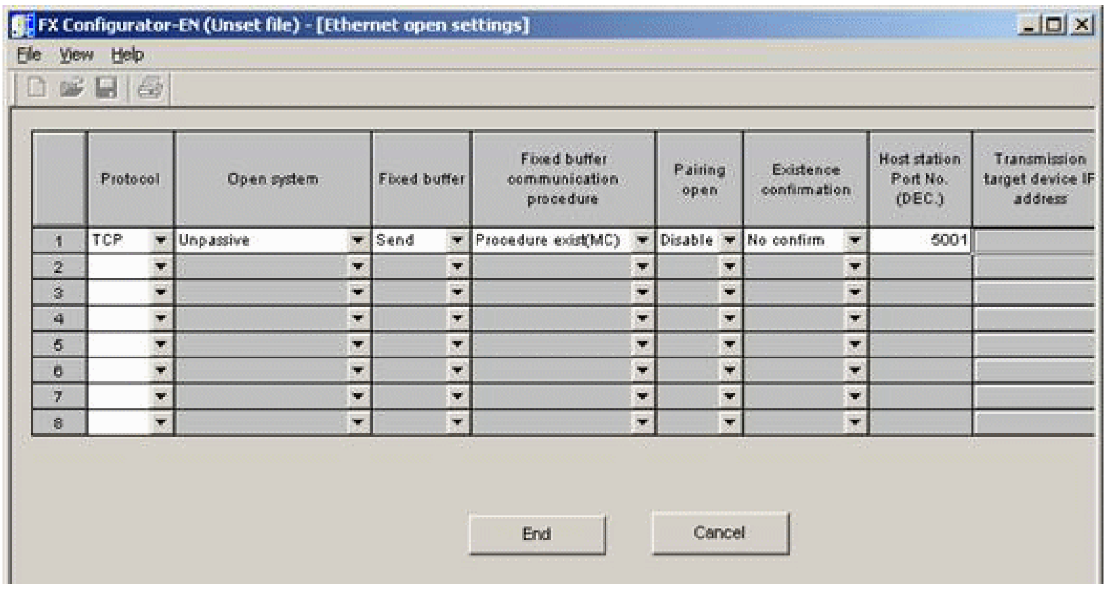
TCP Existence confirmation setting: Use the KeepAlive [✓ Use the Ping]

4. Specify the settings so that they appear similar to the ones shown above.
5. Click **End**.
6. In **FX Configurator-EN**, click **Open Settings**.
7. The open settings depend on the chosen IP protocol: TCP or UDP.

Open Settings for TCP

Enter **TCP** in the Protocol field. For simplicity, the **Unpassive** open system is recommended. By using the unpassive open system, the IP and port that the driver will use do not need to be configured. The **Procedure exist(MC)** communications procedure sets the correct protocol in the FX3U-ENET block to communicate with this driver. In the example below, **5001** (1389 Hex) is specified in the Host station Port No. field.

Note: The example shown below includes only one connection. In order to make multiple connections to the device from the OPC server, add another entry on this screen and configure another open port (such as, Port 5002). Check the device's manual to verify the device's available ports.

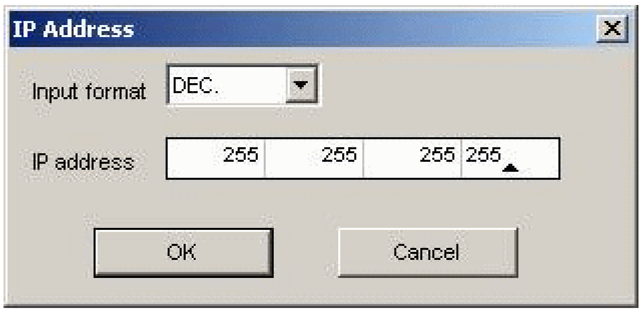


FX Configurator-EN (Unset file) - [Ethernet open settings]

	Protocol	Open system	Fixed buffer	Fixed buffer communication procedure	Pairing open	Existence confirmation	Host station Port No. (DEC.)	Transmission target device IP address
1	TCP	Unpassive	Send	Procedure exist(MC)	Disable	No confirm	5001	
2								
3								
4								
5								
6								
7								
8								

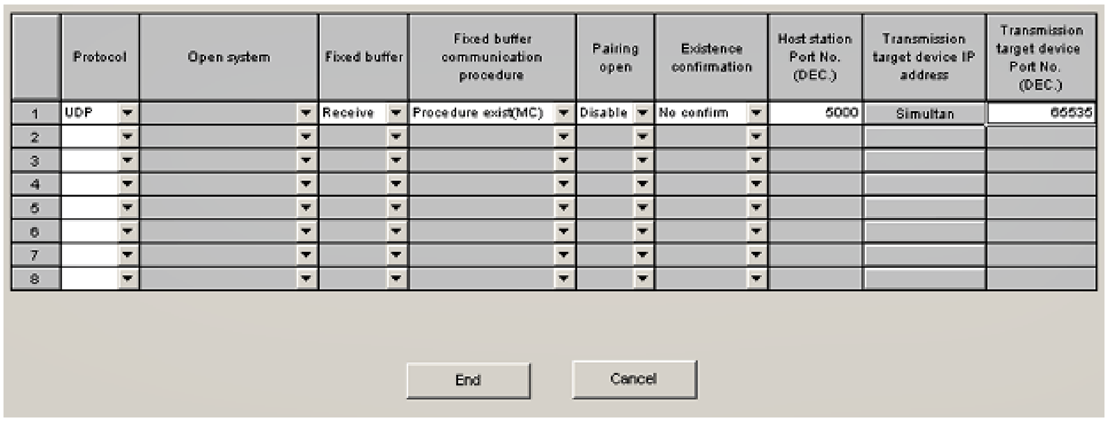
Open Settings for UDP

1. Enter **UDP** in the Protocol field. There are no open system options for UDP. The **Procedure exist(MC)** communications procedure sets the correct protocol in the FX3U-ENET block to communicate with this driver. In the example below, 5000 (1388 Hex) is specified in the Host station Port No. field.
2. In order to allow this driver to choose any port for communications, configure the target port as "unspecified" by entering 65535 (FFFFHex) in the Transmission target device Port No. field. The IP address that the driver uses can be specified or not. To enter the "unspecified" address of 255.255.255.255, do as shown below.



Input format: DEC.

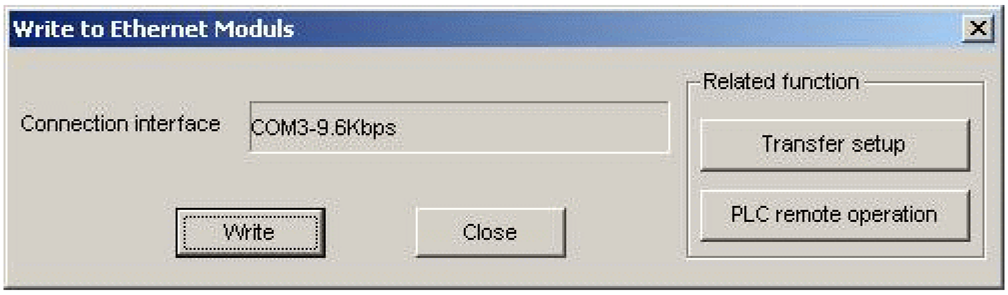
IP address: 255 255 255 255



	Protocol	Open system	Fixed buffer	Fixed buffer communication procedure	Pairing open	Existence confirmation	Host station Port No. (DEC.)	Transmission target device IP address	Transmission target device Port No. (DEC.)
1	UDP		Receive	Procedure exist(MC)	Disable	No confirm	5000	Simultan	65535
2									
3									
4									
5									
6									
7									
8									

Write Network Parameters to PLC

After all of the network parameters have been specified, they must be written to the PLC. To do so, click **Write** from the main FX-Configurator-ENwindow.



Connection interface: COM3-9.6Kbps

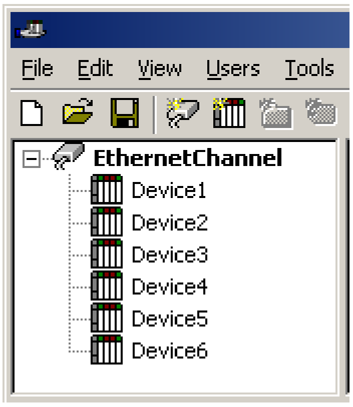
Note: There must be a serial connection to the FX3U PLC. The configuration settings are written to the PLC via this serial link. Also make sure that the communication parameters are correct. Settings can be checked by clicking **Transfer Setup** or be selecting **Online** | **Transfer Setup** from the main menu.

Users must cycle the power on the PLC in order for the network parameter changes to take effect.

Optimizing Mitsubishi Ethernet Communications

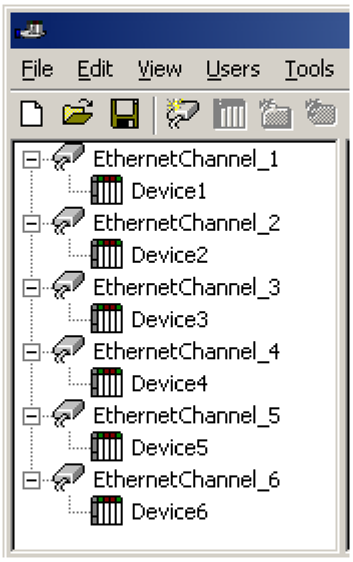
The Mitsubishi Ethernet Driver has been designed to provide the best performance with the least amount of impact on the system's overall performance. While the Mitsubishi Ethernet Driver is fast, there are a couple of guidelines that can be used in order to control and optimize the application and gain maximum performance.

This server refers to communications protocols like Mitsubishi Ethernet Device as a channel. Each channel defined in the application represents a separate path of execution in the server. Once a channel has been defined, a series of devices must then be defined under that channel. Each of these devices represents a single Mitsubishi Ethernet device from which data will be collected. While this approach to defining the application will provide a high level of performance, it won't take full advantage of the Mitsubishi Ethernet Driver or the network. An example of how the application may appear when configured using a single channel is shown below.



Each device appears under a single Mitsubishi Ethernet Device channel. In this configuration, the driver must move from one device to the next as quickly as possible in order to gather information at an effective rate. As more devices are added or more information is requested from a single device, the overall update rate begins to suffer.

If the Mitsubishi Ethernet Driver could only define one single channel, then the example shown above would be the only option available; however, the driver can define up to 256 channels. Using multiple channels distributes the data collection workload by simultaneously issuing multiple requests to the network. An example of how the same application may appear when configured using multiple channels to improve performance is shown below.



Each device has now been defined under its own channel. In this new configuration, a single path of execution is dedicated to the task of gathering data from each device. If the application has 256 or fewer devices, it can be optimized exactly how it is shown here.

The performance will improve even if the application has more than 256 devices. While 256 or fewer devices may be ideal, the application will still benefit from additional channels. Although by spreading the device load across all channels will cause the server to move from device to device again, it can now do so with far less devices to process on a single channel.

Note: An additional performance gain can be achieved by using UDP instead of TCP/IP. For more information, refer to **Device Setup** and **PLC Setup**.

Data Types Description

The Mitsubishi Ethernet Driver supports the following data types.

Data Type	Description
Boolean	Single bit
Word	Unsigned 16-bit value bit 0 is the low bit bit 15 is the high bit
Short	Signed 16-bit value bit 0 is the low bit bit 14 is the high bit bit 15 is the sign bit
DWord	Unsigned 32-bit value bit 0 is the low bit bit 31 is the high bit
Long	Signed 32-bit value bit 0 is the low bit bit 30 is the high bit bit 31 is the sign bit
Float	32-bit floating point value
String	Null terminated ASCII string Support, includes HiLo and LoHi byte order selection and string lengths up to 128 bytes.
BCD	Two byte packed BCD Value range is 0-9999. Behavior is undefined for values beyond this range.
LBCD	Four byte packed BCD Value range is 0-99999999. Behavior is undefined for values beyond this range.
Date	32-bit value
Date Example	Date format:YYYY-MM-DDTHH:MM:SS.000 2000-01-01T12:30:45.000
Double*	64-bit floating point value The driver interprets four consecutive registers as a Double precision value by making the first two registers the low DWord and the last two registers the high DWord.
Double Example*	If register D0000000 is specified as a Double, bit 0 of register D0000000 would be bit 0 of the 64-bit data type. Bit 15 of register D0000003 would be bit 63 of the 64-bit data type.

*The descriptions above assume the default first word low data handling of 32-bit data types.

Address Descriptions

Address specifications vary depending on the model in use. Select a link from the following list to obtain specific address information for the model of interest.

A Series
Q Series
FX3U Series

Mitsubishi A Series Address Descriptions

The default data types for dynamically defined tags are shown in **bold**.

Device Type	Range	Data Type	Access
Inputs*	X000-X1FFF (Hex) X000-X1FF0 (Hex) X000-X1FE0 (Hex)	**Boolean** Short, Word, BCD Long, DWord, LBCD	Read/Write
Outputs*	Y000-Y1FFF (Hex) Y000-Y1FF0 (Hex) Y000-Y1FE0 (Hex)	**Boolean** Short, Word, BCD Long, DWord, LBCD	Read/Write
Link Relays*	B000-B1FFF (Hex) B000-B1FF0 (Hex) B000-B1FE0 (Hex)	**Boolean** Short, Word, BCD Long, DWord, LBCD	Read/Write
Internal Relays*	M0000-M8191 M0000-M8176 M0000-M8160	**Boolean** Short, Word, BCD Long, DWord, LBCD	Read/Write
Special Int. Relays*	M9000-M9255 M9000-M9240 M9000-M9224	**Boolean** Short, Word, BCD Long, DWord, LBCD	Read Only
Latch Relays*	L0000-L8191 L0000-L8176 L0000-L8160	**Boolean** Short, Word, BCD Long, DWord, LBCD	Read/Write
Annunciator Relays*	F0000-F2047 F0000-F2032 F0000-F2016	**Boolean** Short, Word, BCD Long, DWord, LBCD	Read/Write
Timer Contacts*	TS0000-TS2047 TS0000-TS2032 TS0000-TS2016	**Boolean** Short, Word, BCD Long, DWord, LBCD	Read/Write
Timer Coils*	TC0000-TC2047 TC0000-TC2032 TC0000-TC2016	**Boolean** Short, Word, BCD Long, DWord, LBCD	Read/Write
Counter Contacts*	CS0000-CS1023 CS0000-CS1008 CS0000-CS0992	**Boolean** Short, Word, BCD Long, DWord, LBCD	Read/Write
Counter Coils*	CC0000-CC1023 CC0000-CC1008 CC0000-CC0992	**Boolean** Short, Word, BCD Long, DWord, LBCD	Read/Write

*Users can specify a Long data type by appending a space and an "L" to the address. For example, "CS0000" would be entered as "CS0000 L". This does not apply to arrays or bit accessed registers.

Note: All Boolean device types can be accessed as Short, Word, BCD, Long, DWord and LBCD; however, the device must be addressed on a 16-bit boundary.

Device Type	Range	Data Type	Access
Timer Value	TN0000-TN2047	**Short**, Word, BCD	Read/Write
Counter Value	CN0000-CN1023	Short, **Word**, BCD	Read/Write
Data Registers***	D0000-D8191 D0000-D8190 D0000-D8188	**Short**, Word, BCD Long, DWord, LBCD, Float, Date Double	Read/Write

Device Type	**Range**	**Data Type**	**Access**
Data Register Bit Access	D0000.00-D8191.15* D0000.00-D8190.31*	**Short**, Word, BCD, Boolean Long, DWord, LBCD	Read/Write
Data Registers String Access HiLo Byte Ordering	DSH00000.002-DSH08190.002 DSH00000.128-DSH08127.128 The string length may also be specified using a colon. The string length must be between 2-128 bytes and even.	**String**	Read/Write
Data Registers String Access LoHi Byte Ordering	DSL00000.002–DSL08190.002 DSL00000.128-DSL08127.128 The string length may also be specified using a colon. The string length must be between 2-128 bytes and even.	**String**	Read/Write
Special Data Registers***	D9000-D9255 D9000-D9254 D9000-D9252	**Short**, Word, BCD Long, DWord, LBCD, Float, Date Double	Read Only
Data Register Bit Access	D9000.00-D9255.15* D9000.00-D9254.31*	**Short**, Word, BCD, Boolean** Long, DWord, LBCD	Read Only
Link Registers***	W0000-W1FFF (Hex) W0000-W1FFE (Hex) W0000-W1FFC (Hex)	**Short**, Word, BCD Long, DWord, LBCD, Float, Date Double	Read/Write
Link Register Bit Access	W0000.00-W1FFF.15* W0000.00-W1FFE.31*	**Short**, Word, BCD, Boolean** Long, DWord, LBCD	Read/Write
Link Registers String Access HiLo Byte Ordering	WSH0000.002-WSH1FFE.002 WSH0000.128-WSH1FBF.128 The string length may also be specified using a colon. The string length must be between 2-128 bytes and even.	**String**	Read/Write
Link Registers String Access LoHi Byte Ordering	WSL0000.002-WSL1FFE.002 WSL0000.128-WSL1FBF.128 The string length may also be specified using a colon. The string length must be between 2-128 bytes and even.	**String**	Read/Write
File Register***	R0000-R8191 R0000-R8190 R0000-R8188	**Short**, Word, BCD Long, DWord, LBCD, Float, Date Double	Read/Write
File Register Bit Access	R0000.00-R8191.15* R0000.00-R8190.31*	**Short**, Word, BCD, Boolean** Long, DWord, LBCD	Read/Write
File Registers String Access HiLo Byte Ordering	RSH00000.002-RSH08190.002 RSH00000.128-RSH08127.128 The string length may also be specified using a colon. The string length must be between 2-128 bytes and even.	**String**	Read/Write
File Registers String Access LoHi Byte Ordering	RSL00000.002-RSL08190.002 RSL00000.128-RSL08127.128 The string length may also be specified using a colon.	**String**	Read/Write

Device Type	Range	Data Type	Access
	The string length must be between 2-128 bytes and even.		

*For register memory, the data types Short, Word, BCD, DWord, Long, LBCD and Boolean may append an optional ".bb" (dot bit) or ":bb" (colon bit) to the address in order to reference a bit in a particular value. The valid ranges for the optional bit are 0-15 for Short, Word, BCD, Boolean; and 0-31 for Long, DWord and LBCD. Strings use the bit number to specify length. The valid length of a string in D memory is 2 to 128 bytes. The string length must be an even number. Float types do not support bit operations. The bit number is always in decimal notation.
**When accessing register memory as Boolean, a bit number is required.
***Users can specify a Long data type by appending a space and an "L" to the address. For example, "CS0000" would be entered as "CS0000 L". This does not apply to arrays or bit accessed registers.

Array Access
All device types can be accessed as arrays. The default array tag for all device types is Word. The size of the array depends on both the data type and the device type. All Register device types can access up to the following: 254 elements for Short, Word and BCD; 127 elements for Long, DWord, LBCD and Float; and 63 elements for Double. All Bit memory types can access up to the following: 127 elements for Short, Word and BCD; and 63 elements for Long, DWord and LBCD. Arrays may be 1 or 2 dimensions, but the array size may not exceed the limits stated above.

Note: An array is created when array notation is appended onto a normal device reference.

Examples

1. D100 [4] Single dimension includes the following register addresses: D100, D101, D102, D103.
2. M016 [3][4] Two Dimensions includes the following device addresses as words: M016, M032, M048, M064, M080, M096, M112, M128, M144, M160, M176, M192 3 rows x 4 columns = 12 words 12 x 16 (word) =192 total bits.

Additional Device Examples

1. Access X device memory as Word: X??? where the ??? is a hex number on 16-bit boundaries such as 010, 020, 030, and so forth.
2. Access M device memory as Long: M???? where the ???? is a decimal number on 16-bit boundaries such as 0, 16, 32, 48, and so forth.

Mitsubishi Q Series Address Descriptions
The default data types for dynamically defined tags are shown in **bold**.

Device Type	Range	Data Type	Access
Inputs*	X0000-X3FFF (Hex) X0000-X3FF0 (Hex) X0000-X3FE0 (Hex)	**Boolean** Short, Word, BCD Long, DWord, LBCD	Read/Write
Direct Inputs*	DX0000-DX3FFF (Hex) DX0000-DX3FF0 (Hex) DX0000-DX3FE0 (Hex)	**Boolean** Short, Word, BCD Long, DWord, LBCD	Read/Write
Outputs*	Y0000-Y3FFF (Hex) Y0000-Y3FF0 (Hex) Y0000-Y3FE0 (Hex)	**Boolean** Short, Word, BCD Long, DWord, LBCD	Read/Write
Direct Outputs*	DY0000-DY3FFF (Hex) DY0000-DY3FF0 (Hex) DY0000-DY3FE0 (Hex)	**Boolean** Short, Word, BCD Long, DWord, LBCD	Read/Write
Link Relays*	B0000-BEA60 (Hex) B0000-BEA50 (Hex) B0000-BEA40 (Hex)	**Boolean** Short, Word, BCD Long, DWord, LBCD	Read/Write

Special Link Relays*	SB0000-SB7D00(Hex)
	SB0000-SB7CF0(Hex)
	SB0000-SB7CE0(Hex)
		Boolean
		Short, Word, BCD
		Long, DWord, LBCD
			Read/Write

Internal Relays*	M0000-M60000	**Boolean**	Read/Write

Device Type	Range	Data Type	Access
	M0000-M59984 M0000-M59968	Short, Word, BCD Long, DWord, LBCD	
Special Int. Relays*	SM0000-SM2047 SM0000-SM2032 SM0000-SM2016	**Boolean** Short, Word, BCD Long, DWord, LBCD	Read/Write
Latch Relays*	L0000-L32000 L0000-L31984 L0000-L31968	**Boolean** Short, Word, BCD Long, DWord, LBCD	Read/Write
Annunciator Relays*	F0000-F32000 F0000-F31984 F0000-F31968	**Boolean** Short, Word, BCD Long, DWord, LBCD	Read/Write
Edge Relays*	V0000-V32000 V0000-V31984 V0000-V31968	**Boolean** Short, Word, BCD Long, DWord, LBCD	Read/Write
Step Relays*	S0000-S16383 S0000-S16368 S0000-S16352	**Boolean** Short, Word, BCD Long, DWord, LBCD	Read/Write
Timer Contacts*	TS0000-TS32000 TS0000-TS31984 TS0000-TS31968	**Boolean** Short, Word, BCD Long, DWord, LBCD	Read/Write
Timer Coils*	TC0000-TC32000 TC0000-TC31984 TC0000-TC31968	**Boolean** Short, Word, BCD Long, DWord, LBCD	Read/Write
Integrating Timer Contacts*	SS0000-SS2047 SS0000-SS2032 SS0000-SS2016	**Boolean** Short, Word, BCD Long, DWord, LBCD	Read/Write
Integrating Timer Coils*	SC0000-SC2047 SC0000-SC2032 SC0000-SC2016	**Boolean** Short, Word, BCD Long, DWord, LBCD	Read/Write
Counter Contacts*	CS0000-CS32000 CS0000-CS31984 CS0000-CS31968	**Boolean** Short, Word, BCD Long, DWord, LBCD	Read/Write
Counter Coils*	CC0000-CC32000 CC0000-CC31984 CC0000-CC31968	**Boolean** Short, Word, BCD Long, DWord, LBCD	Read/Write

*Users can specify a Long data type by appending a space and an "L" to the address. For example, "CS0000" would be entered as "CS0000 L". This does not apply to arrays or bit accessed registers.

Note: All Boolean device types can be accessed as Short, Word, BCD, Long, DWord and LBCD; however, the device must be addressed on a 16-bit boundary.

Device Type	Range	Data Type	Access
Timer Value	TN0000-TN32000	**Short**, Word, BCD	Read/Write
Integrating Timer Value	SN0000-SN2047	**Short**, Word, BCD	Read/Write
Counter Value	CN0000-CN32000	Short, **Word, BCD**	Read/Write
Data Registers***	D0000000-D4184063 D0000000-D4184062 D0000000-D4184060 **See Also: Extended Registers**	**Short**, Word, BCD Long, DWord, LBCD, Float, Date Double	Read/Write
Data Register Bit Access	D0000000.00–D4184063.15* D0000000.00–D4184062.31* **See Also: Extended Registers**	**Short**, Word, BCD, Boolean** Long, DWord, LBCD	Read/Write
Data Registers String Access HiLo Byte Ordering	DSH00000.002-DSH4184062.002 DSH00000.128-DSH4183999.128 The string length may also be specified using a colon. The string length must be between	**String**	Read/Write

Device Type	Range	Data Type	Access
	2-128 bytes and even.		
Data Registers String Access LoHi Byte Ordering	DSL00000.002-DSL4184062.002 DSL00000.128-DSL4183999.128 The string length may also be specified using a colon. The string length must be between 2-128 bytes and even.	**String**	Read/Write
Special Data Registers***	SD0000-SD2047 SD0000-SD2046 SD0000-SD2044	**Short**, Word, BCD Long, DWord, LBCD, Float, Date Double	Read/Write
Data Register Bit Access	SD0000.00-SD2047.15* SD0000.00-SD2046.31*	**Short**, Word, BCD, Boolean** Long, DWord, LBCD	Read/Write
Link Registers***	W0000-W3FD7FF (Hex) W0000-W3FD7FE (Hex) W0000-W3FD7FC (Hex) **See Also: Extended Registers**	**Short**, Word, BCD Long, DWord, LBCD, Float, Date Double	Read/Write
Link Register Bit Access	W0000.00-W3FD7FF.15* W0000.00-W3FD7FE.31* **See Also: Extended Registers**	**Short**, Word, BCD, Boolean** Long, DWord, LBCD	Read/Write
Link Registers String Access HiLo Byte Ordering	WSH0000.002-WSH3FD7FE.002 WSH0000.128-WSH3FD7BF.128 The string length may also be specified using a colon. The string length must be between 2-128 bytes and even.	**String**	Read/Write
Link Registers String Access LoHi Byte Ordering	WSL0000.002-WSL3FD7FE.002 WSL0000.128-WSL3FD7BF.128 The string length may also be specified using a colon. The string length must be between 2-128 bytes and even.	**String**	Read/Write

Special Link Registers***	SW0000-SW7D00(Hex)
	SW0000-SW7CFF(Hex)
	SW0000-SW7CFD (Hex)
		Short, Word, BCD
		Long, DWord, LBCD, Float, Date
		Double
			Read/Write

Link Register Bit Access	SW0000.00-SW7D00.15* SW0000.00-SW7CFF.31*	**Short**, Word, BCD, Boolean** Long, DWord, LBCD	Read/Write
File Register***	R00000-R32767 R00000-R32766 R00000-R32764 ZR0000-ZR3FD7FF (Hex) ZR0000-ZR3FD7FE (Hex) ZR0000-ZR3FD7FC (Hex)	**Short**, Word, BCD Long, DWord, LBCD, Float, Date Double **Short**, Word, BCD Long, DWord, LBCD, Float, Date Double	Read/Write
File Register Bit Access	R00000.00-R32767.15* R00000.00-R32766.31* ZR0000.00-ZR3FD7FF.15* ZR0000.00-ZR3FD7FE.31*	**Short**, Word, BCD, Boolean** Long, DWord, LBCD **Short**, Word, BCD, Boolean** Long, DWord, LBCD	Read/Write

Device Type	Range	Data Type	Access
File Registers String Access HiLo Byte Ordering	RSH00000.002-RSH32766.002 RSH00000.128-RSH32703.128 ZRSH0000.002-ZRSH3FD7FE.002 ZRSH0000.128-ZRSH3FD7BF.128 The string length may also be specified using a colon. The string length must be between 2-128 bytes and even.	**String** **String**	Read/Write
File Registers String Access LoHi Byte Ordering	RSL00000.002-RSL32766.002 RSL00000.128-RSL32703.128 ZRSL0000.002-ZRSL3FD7FE.002 ZRSL0000.128-ZRSL3FD7BF.128 The string length may also be specified using a colon. The string length must be between 2-128 bytes and even.	**String** **String**	Read/Write
Index Registers***	Z00-Z20 Z00-Z19 Z00-Z17	**Short**, Word, BCD Long, DWord, LBCD, Float, Date Double	Read/Write
Index Register Bit Access	Z00.00-Z20.15* Z00.00-Z19.31*	**Short**, Word, BCD, Boolean** Long, DWord, LBCD	Read/Write

*For register memory, the data types Short, Word, BCD, DWord, Long, LBCD and Boolean may append an optional ".bb" (dot bit) or ":bb" (colon bit) to the address in order to reference a bit in a particular value. The valid ranges for the optional bit are 0-15 for Short, Word, BCD and Boolean; and 0-31 for Long, DWord and LBCD. Strings use the bit number to specify length. The valid length of a string in D memory is 2 to 128 bytes. The string length must be an even number. Float types do not support bit operations. The bit number is always in decimal notation.
**When accessing register memory as Boolean, a bit number is required.
***Users can specify a Long data type by appending a space and an "L" to the address. For example, "CS0000" would be entered as "CS0000 L". This does not apply to arrays or bit accessed registers.

Extended Registers
The extended range for Data Registers is D12288 to D4184063. The extended range for Link Registers is W3FFF (Hex) to W3FD7FF (Hex). These must be configured on the device.

Array Access
All device types can be accessed as arrays. The default array tag for all device types is Word. The size of the array depends on both the data type and the device type. All Register device types can access up to the following: 254 elements for Short, Word and BCD; 127 elements for Long, DWord, LBCD and Float; and 63 elements for Double. All Bit memory types can access up to the following: 127 elements for Short, Word and BCD; and 63 elements for Long, DWord and LBCD. Arrays may be 1 or 2 dimensions, but the array size may not exceed the limits stated above.

Note: An array is created when array notation is appended onto a normal device reference.

Examples
1. D100 [4] Single dimension includes the following register addresses: D100, D101, D102, D103.

2. M016 [3][4] Two Dimensions includes the following device addresses as words: M016, M032, M048, M064, M080, M096, M112, M128, M144, M160, M176, M192 3 rows x 4 columns =12 words 12 x 16 (word) =192 total bits.

Additional Device Examples
1. Access X device memory as Word: X??? where the ??? is a hex number on 16-bit boundaries such as 010, 020, 030, and so forth.

2. Access M device memory as Long: M???? where the ???? is a decimal number on 16-bit boundaries such as 0, 16, 32, 48, and so forth.

Mitsubishi FX3U Series Address Descriptions

The default data types for dynamically defined tags are shown in **bold**.

Device Type	Range	Data Type	Access
Inputs*	X000-X377 (Oct) X000-X360 (Oct) X000-X340 (Oct)	**Boolean** Short, Word, BCD Long, DWord, LBCD	Read/Write
Outputs*	Y000-Y377 (Oct) Y000-Y360 (Oct) Y000-Y340 (Oct)	**Boolean** Short, Word, BCD Long, DWord, LBCD	Read/Write
Internal Relays*	M0000-M7679 M0000-M7664 M0000-M7648	**Boolean** Short, Word, BCD Long, DWord, LBCD	Read/Write
Special Int. Relays*	M8000-M8511 M8000-M8496 M8000-M8480	**Boolean** Short, Word, BCD Long, DWord, LBCD	Read/Write
Step Relays*	S0000-S4095 S0000-S4080 S0000-S4064	**Boolean** Short, Word, BCD Long, DWord, LBCD	Read/Write
Timer Contacts*	TS000-TS511 TS000-TS496 TS000-TS480	**Boolean** Short, Word, BCD Long, DWord, LBCD	Read/Write
Counter Contacts*	CS000-CS255 CS000-CS240 CS000-CS224	**Boolean** Short, Word, BCD Long, DWord, LBCD	Read/Write

*Users can specify a Long data type by appending a space and an "L" to the address. For example, "CS0000" would be entered as "CS0000 L". This does not apply to arrays or bit accessed registers.

Note: All Boolean device types can be accessed as Short, Word, BCD, Long, DWord and LBCD; however, the device must be addressed on a 16-bit boundary.

Device Type	Range	Data Type	Access
Timer Value	TN000-TN511	**Short**, Word, BCD	Read/Write
Counter Value***	CN000-CN199 CN200-CN255	Short, **Word**, BCD Long, **DWord**, LBCD	Read/Write
Data Registers***	D0000-D7999 D0000-D7998 D0000-D7996	**Short**, Word, BCD Long, DWord, LBCD, Float, Date Double	Read/Write
Data Register Bit Access	D0000.00-D7999.15* D0000.00-D7998.31*	**Short**, Word, BCD, Boolean Long, DWord, LBCD	Read/Write
Data Registers String Access HiLo Byte Ordering	DSH0000.002-DSH7998.002 DSH0000.128-DSH7935.128 The string length may also be specified using a colon. The string length must be between 2-128 bytes and even.	**String**	Read/Write
Data Registers String Access LoHi Byte Ordering	DSL0000.002-DSL7998.002 DSL0000.128-DSL7935.128 The string length may also be specified using a colon. The string length must be between 2-128 bytes and even.	**String**	Read/Write
Special Data Registers***	D8000-D8511 D8000-D8510 D8000-D8508	**Short**, Word, BCD Long, DWord, LBCD, Float, Date Double	Read/Write
Special Data	D8000.00-D8511.15*	**Short**, Word, BCD,	Read/Write

Device Type	Range	Data Type	Access
Register Bit Access	D8000.00-D8510.31*	Boolean** Long, DWord, LBCD	
File Register***	R00000-R32767 R00000-R32766 R00000-R32764	**Short**, Word, BCD Long, DWord, LBCD, Float, Date Double	Read/Write
File Register Bit Access	R00000.00-R32767.15* R00000.00-R32766.31*	**Short**, Word, BCD, Boolean** Long, DWord, LBCD	Read/Write
File Registers String Access HiLo Byte Ordering	RSH00000.002-RSH32766.002 RSH00000.128-RSH32703.128 The string length may also be specified using a colon. The string length must be between 2-128 bytes and even.	**String**	Read/Write
File Registers String Access LoHi Byte Ordering	RSL00000.002-RSL32766.002 RSL00000.128-RSL32703.128 The string length may also be specified using a colon. The string length must be between 2-128 bytes and even.	**String**	Read/Write

*For register memory, the data types Short, Word, BCD, DWord, Long, LBCD and Boolean append an optional ".bb" (dot bit) or ":bb" (colon bit) to the address in order to reference a bit in a particular value. The valid ranges for the optional bit are 0-15 for Short, Word, BCD and Boolean; and 0-31 for Long, DWord and LBCD. Strings use the bit number to specify length. The valid length of a string is 2 to 128 bytes. The string length must be an even number. Float types do not support bit operations. The bit number is always in decimal notation.
**When accessing register memory as Boolean, a bit number is required.
***Users can specify a Long data type by appending a space and an "L" to the address. For example, "CS0000" would be entered as "CS0000 L". This does not apply to arrays or bit accessed registers.

Array Access

All device types can be accessed as arrays. The default array tag for all device types is Word, excepting CN200-255 (which is DWord). The size of the array depends on both the data type and the device type. All Register device types can access up to the following: 64 elements for Short, Word and BCD; 32 elements for Long, DWord, LBCD and Float; and 16 elements for Double. All Bit memory types can access up to the following: 32 elements for Short, Word and BCD; and 16 elements for Long, DWord and LBCD. Arrays may be 1 or 2 dimensions, but the array size may not exceed the limits stated above.

Note 1: An array is created when array notation is appended onto a normal device reference.

Note 2: Due to a limit of the protocol, the largest bit memory array that can be written to is 10 Words/Shorts/BCDs (or 5 DWords/Longs/LBCDs). Although this limit differs from the largest bit memory array that can be read (32 words), the maximum Read/Write array size for register memory type is the same (64 words).

Examples
1. D100 [4] Single dimension includes the following register addresses: D100, D101, D102, D103.

2. M016 [3][4] Two Dimensions includes the following device addresses as words: M016, M032, M048, M064, M080, M096, M112, M128, M144, M160, M176, M192 3 rowsx4 columns=12 words 12 x 16 (word) = 192 total bits.

Additional Device Examples

1. Access M device memory as Long: M???? where the ???? is a decimal number on 16-bit boundaries such as 0, 16, 32, 48, and so forth.

2. Access Y device memory as Short: Y??? where the ??? is an Octal number on 16-bit boundaries such as 020, 040, 060, and so forth.

Error Descriptions

The following messages may be generated. Click on the link for a description of the message.

Address Validation

Missing address.
Device address <address> contains a syntax error.
Address <address> is out of range for the specified device or register.
Device address <address> is not supported by model <model name>.
Data type <type> is not valid for device address <address>.
Device address <address> is read only.

Device Status Messages

Device <device name> is not responding.
Unable to write to <address> on device <device name>.

Device Specific Messages

Failed to sync time and date for device <device>. Will retry in <time> minutes.
Unable to bind to adapter: <adapter>. Connect failed.
Unable to read from address <start address> to <end address> on device <device name>.
Unable to read from address <start address> to <end address> on device <device name>. Device returned error code <hexadecimal error code>.
Unable to read from address <start address> to <end address> on device <device name>. The device reported an invalid address or an error.
Unable to read tag <tag address> on device <device name>.
Unable to read tag <tag address> on device <device name>. Device returned error code <hexadecimal error code>.
Unable to read tag <tag address> on device <device name>. The device reported an invalid address or an error.
Unable to read to tag(s) on device <device name>. The device returned a PC Number error.
Unable to write to tag <tag address> on device <device name>. Device must be configured to allow writes while in RUN mode.
Unable to write to tag <tag address> on device <device name>. Device returned error code <hexadecimal error code>.
Unable to write to tag <tag address> on device <device name>. The device reported an invalid address or an error.
Unable to write to tag <tag address> on device <device name>. The device returned a PC Number error
Winsock initialization failed (OS Error = n).
Winsock V1.1 or higher must be installed to use the Mitsubishi Ethernet Driver.

Missing address.

Error Type:
Warning

Possible Cause:
A tag address that has been specified statically has no length.

Solution:
Re-enter the address in the client application.

Device address <address> contains a syntax error.

Error Type:
Warning

Possible Cause:
A tag address that has been specified statically contains one or more invalid characters.

Solution:
Re-enter the address in the client application.

Address <address> is out of range for the specified device or register.

Error Type:
Warning

Possible Cause:
A tag address that has been specified statically via DDE references a location that is beyond the range of supported locations for the device.

Solution:
Verify that the address is correct; if it is not, re-enter it in the client application.

Device address <address> is not supported by model <model name>.

Error Type:
Warning

Possible Cause:
A tag address that has been specified statically references a location that is valid for the communications protocol but not supported by the target device.

Solution:
Verify that the address is correct; if it is not, re-enter it in the client application. Verify also that the selected model name for the device is correct.

Data Type <type> is not valid for device address <address>.

Error Type:
Warning

Possible Cause:
A tag address that has been specified statically has been assigned an invalid data type.

Solution:
Modify the requested data type in the client application.

Device address <address> is read only.

Error Type:
Warning

Possible Cause:
A tag address that has been specified statically has a requested access mode that is not compatible with what the device supports for that address.

Solution:
Change the access mode in the client application.

Device <device name> is not responding.

Error Type:
Serious

Possible Cause:

1. The Ethernet connection between the device and the host PC is broken.
2. The communications parameters for the Ethernet connection are incorrect.
3. The named device may have been assigned an incorrect Network ID.
4. The response from the device took longer to receive than the amount of time specified in the "Request Timeout" device setting.
5. A server project running on any operating system other than Windows Server 2012 can connect and receive data from a Mitsubishi Ethernet device. When using the same project on a Windows Server 2012 machine, it fails to connect to the device. With the introduction of Windows Server 2012, Microsoft enabled, by default, the Explicit Congestion Notification (ECN) on all network adapters. ECN adds information to the packet header to allow properly equipped network gear to reduce network traffic speed as needed. This feature, while available on older server class operating systems, was not enabled by default until the release of Server 2012. With ECN enabled, some older devices drop or reject the packets due to the altered header.

Solution:

1. Verify the cabling between the PC and the PLC device.
2. Verify that the specified communications parameters match those of the device.
3. Verify that the Network ID given to the named device matches that of the actual device.
4. Increase the Request Timeout setting so that the entire response can be handled.
5. Disable ECN on the device.
    1. While logged in as an administrator, open a DOS command prompt.
    2. Type “netsh int tcp set global ecncapability=disabled” and press Enter.
    3. Restart the computer.
    4. Attempt to connect to the device again.

Unable to write to <address> on device <device name>.

Error Type:
Serious

Possible Cause:
1. The Ethernet connection between the device and the host PC is broken.
2. The communications parameters for the Ethernet connection are incorrect.
3. The named device may have been assigned an incorrect Network ID.

Solution:

1. Verify the cabling between the PC and the PLC device.
2. Verify that the specified communications parameters match those of the device.
3. Verify that the Network ID given to the named device matches that of the actual device.

Failed to sync time and date for device <device>. Will retry in <time> minutes.

Error Type:
Warning

Possible Cause:
The driver failed to write time and date data to the PLC.

Solution:
1. Verify the cabling between the PC and the PLC device.
2. Verify that the specified communications parameters match those of the device.
3. Verify that the Network ID given to the named device matches that of the actual device.

Note:
The driver will automatically retry after the indicated time interval.

Unable to bind to adapter: <adapter>. Connect failed.

Error Type:
Fatal

Possible Cause:
1. The operating system could not find an unused port to use for communication with this device.
2. Network system failure (such as Winsock or network adapter).
3. Other applications have claimed all available ports. This is possible but not likely.

Solution:
1. Reboot the computer and check the network adapter.
2. Check for applications that could be causing conflicts and then shut them down.

Unable to read from address <start address> to <end address> on device <device name>.

Error Type:
Serious

Solutions:
1. A Series and QnA Series PLCs: Configure the AJ71E71 card to allow reads to occur during RUN by setting DIP switch 7 to the ON position.
2. Q Series PLCs: Use GX Developer to enable the setting "Enable Write at RUN time" in Ethernet Operations.

See Also:
A Series PLC Setup
QnA Series PLC Setup
Q Series PLC Setup

Unable to read from address <start address> to <end address> on device <device name>. Device returned error code <hexadecimal error code>.

Error Type:
Warning

Possible Cause:
Unknown

Solutions:

For the meaning of the error code, refer to the manufacturer's documentation.

Unable to read from address <start address> to <end address> on device <device name>. The device reported an invalid address or an error.

Error Type:
Serious

Possible Cause:
1. An attempt has been made to read from a nonexistent location in the specified device.
2. An attempt has been made to read from an address in a device that is not located on the specified network node.

Solution:
1. Verify the tags assigned to the addresses in the specified range on the device, and eliminate those that reference invalid locations.
2. Verify that the Node ID referenced in the device address is correct.

Unable to read tag <tag address> on device <device name>.

Error Type:
Serious

Solutions:
1. A Series and QnA Series PLCs: Configure the AJ71E71 card to allow reads to occur during RUN by setting DIP switch 7 to the ON position.
2. Q Series PLCs: Use GX Developer to enable the setting "Enable Write at RUN time" in Ethernet Operations.

See Also:
A Series PLC Setup
QnA Series PLC Setup
Q Series PLC Setup

Unable to read tag <tag address> on device <device name>. Device returned error code <hexadecimal error code>.

Error Type:
Warning

Possible Cause:
Unknown

Solutions:
For the meaning of the error code, refer to the manufacturer's documentation.

Unable to read tag <tag address> on device <device name>. The device reported an invalid address or an error.

Error Type:
Serious

Possible Cause:
An attempt has been made to read a nonexistent location in the specified device.

Solution:
Verify the tags assigned to addresses in the specified range on the device and eliminate those that reference invalid locations.

Unable to read to tag(s) on device <device name>. The device returned a PC number error.

Error Type:
Serious

Possible Cause:
The PC number that has been entered for the Device ID is invalid. This may occur if the desired MelsecNet station is not available.

Solution:
1. If attempting to communicate with a PC located on MelsecNet, verify the PC number of the desired target PC.
2. If intending to talk directly to the local PC that contains the Ethernet connection, specify a PC number of 255.

Unable to write to tag <tag address> on device <device name>. Device must be configured to allow writes while in RUN mode.

Error Type:
Serious

Solutions:
1. A Series and QnA Series PLCs: Configure the AJ71E71 card to allow writes to occur during RUN by setting DIP switch 7 to the ON position.
2. Q Series PLCs: Use GX Developer to enable the setting "Enable Write at RUN time" in Ethernet Operations.

See Also:
A Series PLC Setup
QnA Series PLC Setup
Q Series PLC Setup

Unable to write to tag <tag address> on device <device name>. Device returned error code <hexadecimal error code>.

Error Type:
Warning

Possible Cause:
Unknown

Solutions:
For the meaning of the error code, refer to the manufacturer's documentation.

Unable to write to tag <tag address> on device <device name>. The device reported an invalid address or an error.

Error Type:
Serious

Possible Cause:
1. An attempt has been made to write to a nonexistent location in the specified device.
2. An attempt has been made to write to an address in a device that is not located on the specified network node.

Solution:
1. Verify the tags assigned to the addresses in the specified range on the device, and eliminate those that reference invalid locations.
2. Verify that the Node ID referenced in the device address is correct.

Unable to write to tag <tag address> on device <device name>. The device returned a PC number error.

Error Type:
Serious

Possible Cause:
The PC number that has been entered for the Device ID is invalid. This may occur if the desired MelsecNet station is not available.

Solution:
1. If attempting to communicate with a PC located on MelsecNet, verify the PC number of the desired target PC.
2. If intending to talk directly to the local PC that contains the Ethernet connection, specify a PC number of 255.

Winsock initialization failed (OS Error = n).

Error Type:
Fatal

OS Error:	Indication	Possible Solution
10091	Indicates that the underlying network subsystem is not ready for network communication.	Wait a few seconds and restart the driver.
10067	Limit on the number of tasks supported by the Windows Sockets implementation has been reached.	Close one or more applications that may be using Winsock and restart the driver.

Winsock V1.1 or higher must be installed to use the Mitsubishi Ethernet Driver

Error Type:
Fatal

Possible Cause:
The version number of the Winsock DLL found on the system is less than 1.1.

Solution:
Upgrade Winsock to version 1.1 or higher.

Index

A

A Series PLC Setup 9

Address <address> is out of range for the specified device or register. 35

Address Descriptions 26

B

Boolean 25

C

Communications Parameters 6

D

Data Type <type> is not valid for device address <address>. 35

Data Types Description 25

Device <device name> is not responding. 36

Device address <address> contains a syntax error. 35

Device address <address> is not supported by model <model name>. 35

Device address <address> is read only. 36

Device ID 5

Device Setyp 5

DWord 25

E

Error Descriptions 34

F

Failed to sync time and date for device <device>. Will retry in <time> minutes. 37

First Word Low 5

FX3U Series PLC Setup 20

H

Help Contents 4

L

Long 25

M

Missing address. 35
Mitsubishi A Series Address Descriptions 26
Mitsubishi FX3U Series Address Descriptions 32
Mitsubishi Q Series Address Descriptions 28
Multi-Level Networks 7

N

Network 5

O

Optimizing Your Mitsubishi Ethernet Communications 24
Overview 4

P

PLC Setup 9

Q

Q Series PLC Setup 12
QnA Series PLC Setup 10

S

Short 25

T

Time and Date Synchronization 7

U

Unable to bind to adapter: <adapter>. Connect failed. 37

Unable to read from address <start address> to <end address> on device <device name>. 37

Unable to read from address <start address> to <end address> on device <device name>. Device returned error code <hexadecimal error code>. 37

Unable to read from address <start address> to <end address> on device <device name>. The device reported an invalid address or an error. 38

Unable to read tag <tag address> on device <device name>. 38

Unable to read tag <tag address> on device <device name>. Device returned error code <hexadecimal error code>. 38

Unable to read tag <tag address> on device <device name>. The device reported an invalid address or an error. 38

Unable to read to tag(s) on device <device name>. The device returned a PC number error. 39

Unable to write tag <address> on device <device name>. 36

Unable to write to tag <tag address> on device <device name>. Device must be configured to allow writes while in RUN mode. 39

Unable to write to tag <tag address> on device <device name>. Device returned error code <hexadecimal error code>. 39

Unable to write to tag <tag address> on device <device name>. The device reported an invalid address or an error. 39

Unable to write to tag <tag address> on device <device name>. The device returned a PC number error. 40

W

Winsock initialization failed (OS Error = n). 40

Winsock V1.1 or higher must be installed to use this driver. 40

Word 25