Mitsubishi CNC Ethernet Driver Help

Table of Contents

Mitsubishi CNC Ethernet Driver Help

Help version 1.026

CONTENTS

Overview
What is the Mitsubishi CNC Ethernet Driver?

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

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

Data Types Description
What data types does this driver support?

Address Descriptions
How do I address a data location on a Mitsubishi CNC Ethernet device?

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

Overview

The Mitsubishi CNC Ethernet Driver provides an easy and reliable way to connect Mitsubishi CNC Ethernet controllers to OPC Client applications, including HMI, SCADA, Historian, MES, ERP and countless custom applications.

Device Setup

Supported Devices
C64 CNC Controller

Communication Protocol
Ethernet with Winsock V1.1 or higher.
TCP/IP.

Supported Communication Parameters
Binary Format only.

Model
Mitsubishi C64 with an AJ71QE71 compatible Ethernet module.

Device ID
Device IDs are specified as *YYY.YYY.YYY.YYY*, where *YYY* designates the device IP address. Each *YYY* byte should be in the range of 0 to 255.

32-Bit Data First Word Low
Two consecutive registers addresses in a Mitsubishi device are used for 32 bit data types. This parameter specifies whether the driver should assume the first word is the low or the high word of the 32-bit value. The default setting is first word low.

Connection Timeout
This parameter specifies the amount of time that the driver will wait for a connection to be made with a device. Depending on network load, the connection time may vary with each attempt. The valid range is 1 to 60 seconds. The default setting is 3 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 valid range is 50 to 30000 milliseconds. The default setting is 1000 milliseconds.

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

Cable Connections and Diagrams

Patch Cable (Straight Through)

TD +	1	OR/WHT	OR/WHT	1	TD +
TD -	2	OR	OR	2	TD -
RD +	3	GRN/WHT	GRN/WHT	3	RD +
	4	BLU	BLU	4	
	5	BLU/WHT	BLU/WHT	5	
RD -	6	GRN	GRN	6	RD -
	7	BRN/WHT	BRN/WHT	7	
	8	BRN	BRN	8	

RJ45   RJ45

Crossover Cable

TD +	1	OR/WHT	GRN/WHT	1	TD +
TD -	2	OR	GRN	2	TD -
RD +	3	GRN/WHT	OR/WHT	3	RD +
	4	BLU	BLU	4	
	5	BLU/WHT	BLU/WHT	5	
RD -	6	GRN	OR	6	RD -
	7	BRN/WHT	BRN/WHT	7	
	8	BRN	BRN	8	

RJ45   RJ45

10 BaseT

1 2 3 4 5 6 7 8

8-pin RJ45

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

Network Parameters
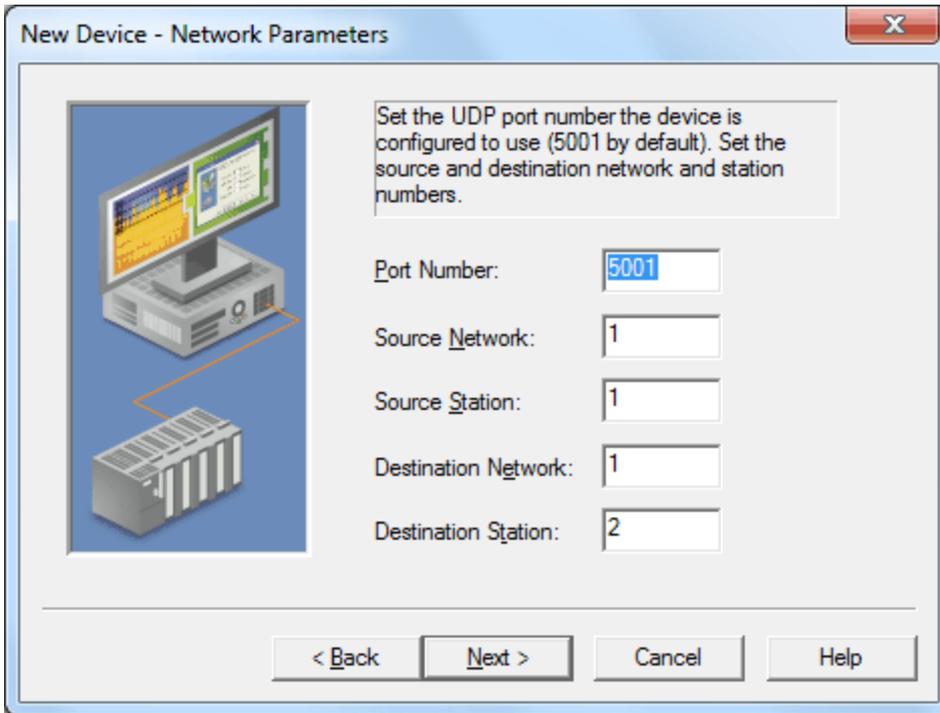
Descriptions of the parameters are as follows:

- **Port Number:** This parameter specifies the UDP port for the destination CNC (or the gateway device if a multi-layered network is configured to receive requests). The default setting is 5001.

   Note: It is recommended that the default setting be used because it is always available for programming and monitoring tools.
- **Source Network:** This parameter specifies the source network number on which the PC resides. The valid range is 1 to 239. The default setting is 1. This setting is irrelevant if the driver communicates directly to CNC (no gateway device).
- **Source Station:** This parameter specifies the station number assigned to the PC. The valid range is 1 to 239. The default setting is 1.

   Note: All devices on the source network should have unique station numbers. This setting is irrelevant if the driver communicates directly to CNC (no gateway device).
- **Destination Network:** This parameter specifies the network number on which the CNC resides. The valid range is 0 to 239. The default setting is 1. This setting is irrelevant if the driver communicates directly to CNC (no gateway device).
- **Destination Station:** This parameter specifies the station number assigned to CNC. The valid range is 0 to 239. The default setting is 2.

   Note: All devices on the destination network should have unique station numbers. This setting is irrelevant if the driver communicates directly to CNC (no gateway device).

Multi-level Networks

This driver can be used to communicate with devices on remote networks. In the example shown below, CNC 1 and CNC 2 are on the local Ethernet network. CNC 3 and CNC 4 are on Network 2 - NET/10. PLC 1 serves as a relay device connecting the two networks.

See Also: Device Setup
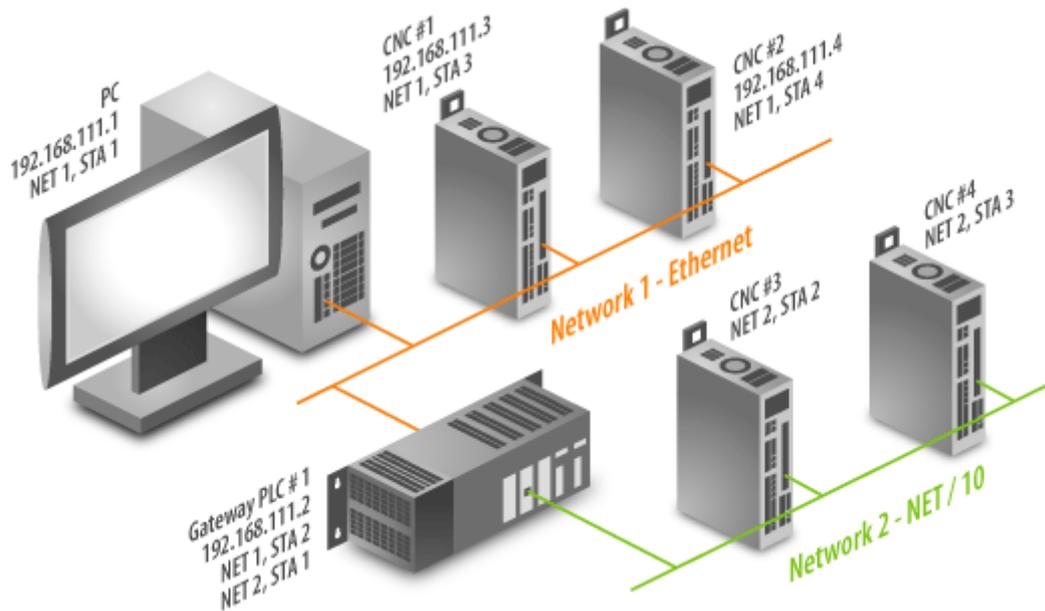
The gateway has an AJ71QE71 Ethernet module and NET/10 module. CNC 1 and CNC 2 have an AJ71QE71 Ethernet module and NET/10 module. CNC 3 and CNC 2 have a NET/10 module. In this example, four devices are created in the server project using the Device IDs listed in the table below.

CNC	Device ID	SRC NET	SRC STA	DST NET	DST STA	Comment
1	192.168.111.2	1	1	1	3	Direct
2	192.168.111.2	1	1	1	4	Direct
3	192.168.111.2	1	1	2	2	Via PLC 1
4	192.168.111.2	1	1	2	3	Via PLC 1

Users can configure the Ethernet card in gateway (Open Method UDP and IP 192.168.111.2). The destination IP (255.255.255.255) and destination port (0xFFFF) can be used to accommodate any IP and Port that may be utilized by the PC.

Note: 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 be set for remote devices accordingly.

## Optimizing Your Mitsubishi CNC Ethernet Communications

The Mitsubishi CNC 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 CNC 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 CNC Ethernet Driver 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 can then be defined under that channel. Each of these devices represents a single 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 CNC Ethernet Driver or the network. An example of how the application may appear when configured using a single channel is shown below.
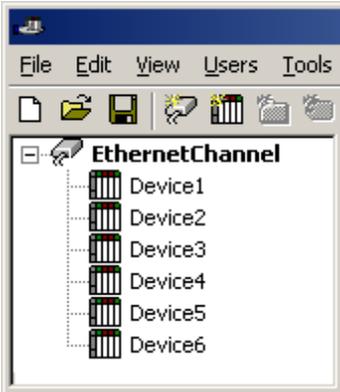
Each device appears under a single Mitsubishi CNC Ethernet device channel. In this configuration, the driver must move from one device to the next as quickly as possible in order to gather information at an effective rate. As more devices are added or more information is requested from a single device, the overall update rate begins to suffer.

If the Mitsubishi CNC Ethernet Driver could only define one single channel, then the example shown above would be the only option available; however, the Mitsubishi CNC Ethernet 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.
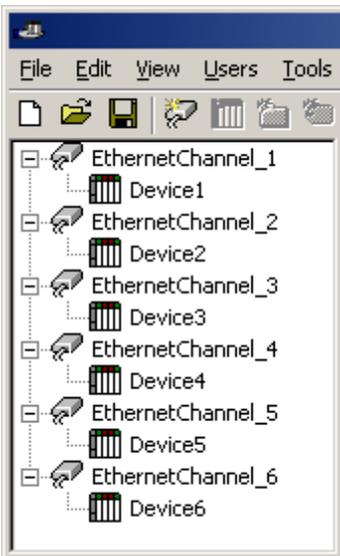
Each device has now been defined under its own channel. In this new configuration, a single path of execution is dedicated to the task of gathering data from each device. If the application has 256 or fewer devices, it can be optimized exactly how it is shown here.

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 256 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.

Protocol Choice

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

Data Types Description

The Mitsubishi CNC Ethernet Driver supports the following data types.

Data Type	Description
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
Long Example	If register 40001 is specified as a long, bit 0 of register 40001 would be bit 0 of the 32 bit data type and bit 15 of register 40002 would be bit 31 of the 32 bit data type. The reverse is true when this is not selected.
Float*	32 bit floating point value
Float Example	If register 40001 is specified as a float, bit 0 of register 40001 would be bit 0 of the 32 bit data type and bit 15 of register 40002 would be bit 31 of the 32 bit data type. The reverse is true when this is not selected.

*The driver interprets two consecutive registers as a single precision value by making the first register the low word and the second register the high word. The reverse is true when this is not selected.

Mitsubishi CNC Ethernet Address Descriptions

Address specifications vary depending on the model in use. The default data types for dynamically defined tags are shown in **bold**.

Device Type	Range*	Data Type	Access
Inputs	X0000-X0AFF (Hex)	**Boolean**, Short, Word, Long, DWord	Read/Write
Outputs	Y0000-Y0E7F (Hex)	**Boolean**, Short, Word, Long, DWord	Read/Write
Link Relays	B0000-B1FFF (Hex)	**Boolean**, Short, Word, Long, DWord	Read/Write
Special Link Relays	SB0000-SB01FF (Hex)	**Boolean**, Short, Word, Long, DWord	Read/Write
Internal Relays	M0000-M8191	**Boolean**, Short, Word, Long, DWord	Read/Write
Special Internal Relays	SM0000-SM0127	**Boolean**, Short, Word, Long, DWord	Read/Write
Latch Relays	L0000-L0255	**Boolean**, Short, Word, Long, DWord	Read/Write
Annunciator Relays	F0000-F0127	**Boolean**, Short, Word, Long, DWord	Read/Write
Timer Contacts	TS0000-TS0255	**Boolean**, Short, Word, Long, DWord	Read/Write
Timer Coils	TC0000-TC0255	**Boolean**, Short, Word, Long, DWord	Read/Write
Counter Contacts	CS0000-CS0127	**Boolean**, Short, Word, Long, DWord	Read/Write
Counter Coils	CC0000-CC0127	**Boolean**, Short, Word, Long, DWord	Read/Write

*This device will respond to block reads that extend past its memory range if the starting address is within the valid memory range. When this happens, the device will return zeros for all values outside this memory range.

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

Device Type	Range	Data Type	Access
Timer Value	TN0000-TN0255	**Short**, Word	Read/Write
Counter Value	CN0000-CN0127	Short, **Word**	Read/Write
Data Register	D00000-D08191 D00000-D08190	**Short**, Word Long, DWord, Float	Read/Write
Data Register Bit Access	D00000.00-D08191.15* D00000.00-D08190.31*	**Short**, Word, Boolean** Long, DWord,	Read/Write
Data Register String Access HiLo Byte Ordering	DSH00000.002-DSH08190.002 DSH00000.128-DSH08127.128 The .Bit number is used to specify the string length 2-128 bytes. Length must be even.	**String**	Read/Write
Data Register String Access LoHi Byte Ordering	DSL00000.002-DSL08190.002 DSL00000.128-DSL08127.128 The .Bit number is used to specify the string length 2-128 bytes. Length must be even.	**String**	Read/Write
Special Data Register	SD0000-SD0127 SD0000-SD0126	**Short**, Word Long, DWord, Float	Read/Write
Special Data Register Bit Access	SD0000.00-SD0127.15* SD0000.00-SD0126.31*	**Short**, Word, Boolean** Long, DWord	Read/Write
Link Register	W0000-W1FFF (Hex) W0000-W1FFE (Hex)	**Short**, Word Long, DWord, Float	Read/Write
Link Register Bit Access	W0000.00-W1FFF.15* W0000.00-W1FFE.31*	**Short**, Word, Boolean** Long, DWord	Read/Write
Link Register String Access HiLo Byte Ordering	WSH0000.002-WSH1FFE.002 WSH0000.128-WSH1FBF.128 The .Bit number is used to specify the string length 2-128 bytes. Length must be even.	**String**	Read/Write

Link Register String Access LoHi Byte Ordering	WSL0000.002-WSL1FFE.002 WSL0000.128-WSL1FBF.128 The .Bit number is used to specify the string length 2-128 bytes. Length must be even.	**String**	Read/Write
Special Link Register	SW0000-SW01FF (Hex) SW0000-SW01FE (Hex)	**Short**, Word Long, DWord, Float	Read/Write
Special Link Register Bit Access	SW0000.00-SW01FF.15* SW0000.00-SW01FE.31*	**Short**, Word, Boolean** Long, DWord	Read/Write
File Register	R00000-R08191 R00000-R08190	**Short**, Word Long, DWord, Float	Read/Write
File Register Bit Access	R00000.00-R08191.15* R00000.00-R08190.31*	**Short**, Word, Boolean** Long, DWord	Read/Write
File Register String Access HiLo Byte Ordering	RSH0000.002-RSH08190.002 RSH0000.128-RSH08127.128 The .bit number is used to specify the string length 2-128 bytes. Length must be even.	**String**	Read/Write
File Register String Access LoHi Byte Ordering	RSL0000.002-RSL08190.002 RSL0000.128-RSL08127.128 The .bit number is used to specify the string length 2-128 bytes. Length must be even.	**String**	Read/Write
Index Register	Z00-Z13 Z00-Z12	**Short**, Word Long, DWord, Float	Read/Write
Index Register Bit Access	Z00.00-Z13.15* Z00.00-Z12.31*	**Short**, Word, Boolean** Long, DWord	Read/Write

*For register memory, the data types Short, Word, DWord, Long, and Boolean may use an optional .bb (dot bit) that can be appended to the address to reference a bit in a particular value. The valid ranges for the optional bit are 0 to 15 for Short, Word, Boolean, and 0 to 31 for Long and DWord. 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 also 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.

Array Access
All device types can be accessed in arrays of Short, Word, Long, DWord, or Float format. The size of the array depends on the data type and device type. All Register device types can access up to 254 elements for Short and Word and 127 elements for Long, DWord, and Floats. All Bit memory types can be accessed with up to 125 elements for Short and Word and 62 elements for Long, DWord, and Float. Arrays can either 1 dimension or 2. Regardless of the dimensions, the array size must not exceed the limits already stated. Appending array notation onto a normal device reference enters arrays.

Note: The default for array tags (all device types) is Word.

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

Error Descriptions

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

Address Validation
Address '<address>' is out of range for the specified device or register
Data Type '<type>' is not valid for device address '<address>'
Device address '<address>' contains a syntax error
Device address '<address>' is not supported by model '<model name>'
Device address '<address>' is Read Only
Missing address

Device Status Messages
Device '<device name>' is not responding
Unable to write to '<address>' on device '<device name>'

Device Specific Messages
Block read failed for <number> points starting at addresses <address name> on device <device name>. Connection error
Block read failed for <number> points starting at addresses <address name> on device <device name>. Device responded with error code: <error code>
Block read failed for <number> points starting at addresses <address name> on device <device name>. Framing error
Failed to read <address> for device <device name>
Winsock initialization failed (OS Error = n)
Winsock V1.1 or higher must be installed to use the Mitsubishi A Series Ethernet device driver
Write failed for addresses <address name> on device <device name>. Connection error
Write failed for addresses <address name> on device <device name>. Device responded with error code: <error code>
Write failed for addresses <address name> on device <device name>. Framing error

Address Validation

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

Address Validation
Address '<address>' is out of range for the specified device or register
Data Type '<type>' is not valid for device address '<address>'
Device address '<address>' contains a syntax error
Device address '<address>' is not supported by model '<model name>'
Device address '<address>' is Read Only
Missing address

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

Error Type:
Warning

Possible Cause:
A tag address that has been specified dynamically 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.

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

Error Type:
Warning

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

Solution:

Modify the requested data type in the client application.

Device address '<address>' contains a syntax error

Error Type:
Warning

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

Solution:
Re-enter the address 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 dynamically 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. Also verify that the selected model name for the device is correct.

Device address '<address>' is Read Only

Error Type:
Warning

Possible Cause:
A tag address that has been specified dynamically 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.

Missing address

Error Type:
Warning

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

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

Device Status Messages

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

Device Status Messages
Device '<device name>' is not responding
Unable to write to '<address>' on device '<device name>'

Device '<device name>' is not responding

Error Type:
Serious

Possible Cause:
1. The connection between the device and the host PC is broken.
2. The communications parameters for the serial connection are incorrect.

3. The named device may have been assigned an incorrect Network ID.
4. If using the driver in UDP wireless mode, it is possible that the wireless router sends duplicate packets. When the driver receives duplicate packets, it ignores them and waits for the expected packet. However, if the expected packet (read confirmation) is not received in the given retry/timeout interval, the above error may be posted. The error is not posted due to an actual read failure but due to a faulty router sending duplicate packets (in response to an earlier request).
5. The response from the device took longer to receive than the amount of time specified in the "Request Timeout" device setting.

Solution:
1. Verify the cabling between the PC and the PLC device.
2. Verify that the specified communications parameters are correct.
3. Verify that the Network ID given to the named device matches that of the actual device.
4. Increase the retry and/or timeout value so that the driver can wait longer to receive the correct packet.
5. Increase the Request Timeout setting so that the entire response can be handled.

See Also:
Device Setup

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

Error Type:
Serious

Possible Cause:
1. The connection between the device and the host PC is broken.
2. The communications parameters are incorrect.
3. The named device may have been assigned an incorrect Network ID.
4. If the driver is in UDP wireless mode, it is possible that the wireless router sends duplicate packets. When the driver receives duplicate packets, it ignores them and waits for the expected packet. However, if the expected packet (write confirmation) is not received in the given retry/timeout interval, the above error may be posted (it is very possible that the write request has been already completed by the device). The error is not posted due to an actual write failure but due to a faulty router sending duplicate packets (in response to an earlier request).

Solution:
1. Verify the cabling between the PC and the PLC device.
2. Verify that the specified communications parameters are correct.
3. Verify that the Network ID given to the named device matches that of the actual device.
4. Increase the retry and/or timeout value so that the driver can wait longer to receive the correct packet.

See Also:
Device Setup

Device Specific Messages

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

Device Specific Messages
Block read failed for <number> points starting at addresses <address name> on device <device name>. Connection error
Block read failed for <number> points starting at addresses <address name> on device <device name>. Device responded with error code: <error code>
Block read failed for <number> points starting at addresses <address name> on device <device name>. Framing error
Failed to read <address> for device <device name>
Winsock initialization failed (OS Error = n)
Winsock V1.1 or higher must be installed to use the Mitsubishi A Series Ethernet device driver
Write failed for addresses <address name> on device <device name>. Connection error
Write failed for addresses <address name> on device <device name>. Device responded with error code: <error code>
Write failed for addresses <address name> on device <device name>. Framing error

Block read failed for <number> points starting at addresses <address name> on device <device name>. Connection error

Error Type:

Serious

Possible Cause:
This error is due to a Winsock error (such as socket creation failure and so forth).

Solution:
N/A

Block read failed for <number> points starting at addresses <address name> on device <device name>. Device responded with error code: <error code>

Error Type:
Serious

Possible Cause:
The error code should point to the reason for this error message.

Solution:
Depends on the error code.

Block read failed for <number> points starting at addresses <address name> on device <device name>. Framing error

Error Type:
Serious

Possible Cause:
A packet has been received with incorrect values for fields such as source station number and so forth.

Solution:
N/A

Failed to read <address> for device <device name>

Error Type:
Serious

Possible Cause:
The specified address is out of range of device.

Solution:
Verify the address range supported by device and modify tag configuration accordingly.

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 device 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.

Write failed for addresses <address name> on device <device name>. Connection error

Error Type:
Serious

Possible Cause:
This error is due to a Winsock error (such as socket creation failure and so forth).

Solution:
N/A

Write failed for addresses <address name> on device <device name>. Device responded with error code: <error code>

Error Type:
Serious

Possible Cause:
The error code should point to the reason for this error message.

Solution:
Depends on the error code.

Write failed for addresses <address name> on device <device name>. Framing error

Error Type:
Serious

Possible Cause:
A packet has been received with incorrect values for fields such as source station number and so forth.

Solution:
N/A

Index