Mitsubishi FX Driver Help

Table of Contents

Mitsubishi FX Driver Help

Help version 1.026

CONTENTS

Overview
What is the Mitsubishi FX Driver?

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

Data Types Description
What data types does this driver support?

Address Descriptions
How do I address a data location on a Mitsubishi FX series device?

Error Descriptions
What error messages does the Mitsubishi FX driver produce?

Overview

The Mitsubishi FX Driver provides an easy and reliable way to connect Mitsubishi FX devices to OPC Client applications, including HMI, SCADA, Historian, MES, ERP and countless custom applications. It is intended for use with Mitsubishi FX series devices.

Device Setup

Supported Devices
FX
FX0
FX0N
FX2N
FX3U

Note: FX3U is not supported in Windows CE.

Communication Protocol
Direct Serial

Supported Communication Parameters
Baud Rate: 9600
Parity: Even
Data Bits: 7
Stop Bits: 1

Ethernet Encapsulation
This driver supports Ethernet Encapsulation, which allows the driver to communicate with serial devices attached to an Ethernet network using a terminal server. It may be invoked through the COM ID dialog in Channel Properties. For more information, refer to the OPC server's help file.

When used directly with a serial port, this driver supports only a single connection to a single controller per serial port. When operating in the Ethernet Encapsulation mode, the driver supports up to 100 controllers per channel. In this mode, a single controller can be paired with a terminal server (device server) to form a single node. For more information, refer to **Ethernet Encapsulation Examples**.

Note: Ethernet Encapsulation is not supported by the FX3U model.

Maximum Number of Channels and Devices
The maximum number of channels supported by this driver is 256. The maximum number of devices is 100.

Device IDs
This protocol does not support simultaneous communication with multiple devices.

Flow Control
When using an RS232/RS485 converter, the type of flow control that is required will depend on the needs of the converter. Some converters do not require any flow control whereas others require RTS flow. Consult the converter's documentation in order to determine its flow requirements. An RS485 converter that provides automatic flow control is recommended.

Note: When using the manufacturer's supplied communications cable, it is sometimes necessary to choose a flow control setting of **RTS** or **RTS Always** under the Channel Properties.

Cable Connections
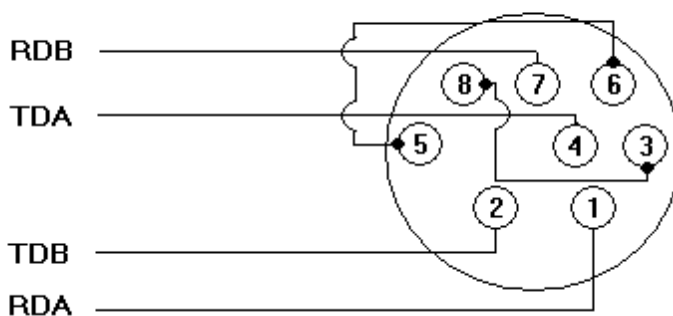
Modem Setup

This driver supports modem functionality. For more information, please refer to the topic "Modem Support" in the OPC Server Help documentation.

Ethernet Encapsulation Examples

Click on a link from the list below in order to jump the specific Ethernet Encapsulation example.

Using COM-ET10-T Ethernet Module
Using the FX2NC-ENET-ADP Ethernet Module
Using the FX3U-ENET Ethernet Module
Writing Network Parameters to the PLC

Note: Ethernet Encapsulation is not supported by the FX3U device model.

Using COM-ET10-T Ethernet Module

Mitsubishi Electric has an Ethernet module that allows peer to peer Ethernet connection to FX1S, FX1N, FX2NC or FX2NC model controllers. The COM-ET10-T module connects to the controller via the FX1N-CNV-BD or the FX2N-CNV-BD communications adapter. For equipment information, refer to the FX hardware manual.

Example Parameters
```
The Module ID is Fixed at ET10
 E = 45
 T = 54
 1 = 31
 0 = 30

IP Address: 192.168.108.10
 192 = C0
 168 = A8
 108 = 6C
 10 = 0A
Subnet Mask: 255.255.255.0
 255 = FF
 255 = FF
 255 = FF
 0 = 00
Gateway IP Address: 192.168.108.01
 192 = C0
 168 = A8
 108 = 6C
 1 = 01
UPD Port: 1024
TCP Port1: 1025
TCP Port2: 1026
The module currently supports only connecting to TCP Port 1025.
```

Mitsubishi FX 2N() Series Ladder

Ladder Logic for initializing the module. The configuration parameters are stored in data registers. Any thousand range with the D register area beginning with D0, D1000, D2000 and etc can be used.

```
|---| M8002 |--------------[ MOV  H81  D8120 ]  'Set communication register
|   |--------[ DMOV H45543130  D1000 ] 'Module ID ET10 (Fixed)
| |-------[ DMOV H0C0A86C0A  D1002 ] 'Module IP Address
|   |-------[ DMOV H0FFFFFF00  D1004 ] 'Subnet Mask
|   |-------[ DMOV H0C0A86C01  D1006 ] 'Gateway IP Address
|   |-------------[ MOV K1024  D1008 ] 'UDP Port Number
|   |-------------[ MOV K1025  D1009 ] 'TCP Port Number 1
|   |-------------[ MOV K1026  D1010 ] 'TCP Port Number 2
|   |---------------[ MOV K0  D1011 ] 'System Area Default Value 0
|   |---------------[ MOV K0  D1012 ] 'System Area Default Value 0
| |---------------[ MOV K0  D1013 ] 'System Area Default Value 0
|   |---------------[ MOV K0  D1014 ] 'System Area Default Value 0
|   |---------------[ MOV K0  D1015 ] 'System Area Default Value 0
```

Note: This information has been taken from the *Mitsubishi Electric Ethernet Communications Module COM ET10-T* manual.

Using the FX2NC-ENET-ADP Ethernet Module

Mitsubishi has an Ethernet module that allows peer to peer Ethernet connection to FX1S, FX1N, FX2NC or FX2NC model controllers. The FX2NC-ENET_ADP module connects to the controller via the FX1N-CNV-BD or the FX2N-CNV-BD communications adapter. For equipment information, refer to the FX hardware manual.

Example Parameters

```
The Module ID is Fixed at ENET
 E = 45
 N = 4E
 E = 45
 T = 54
IP Address: 192.168.108.10
 192 = C0
 168 = A8
 108 = 6C
 10 = 0A
Subnet Mask: 255.255.255.0
 255 = FF
 255 = FF
 255 = FF
 0 = 00
Gateway IP Address: 192.168.108.01
 192 = C0
 168 = A8
 108 = 6C
 1 = 01
TCP Port: 1024
```

Mitsubishi FX1s Series Ladder

Ladder Logic for initializing the module on a FX1s. Set the Ethernet parameters in 9 data registers starting from D128 to D136.

```
|---| M8002 |-----------[ DMOV H454E4554  D128 ]  'Module ID ENET (Fixed)
|  |-------[ DMOV H0C0A86C0A  D130 ]  'Module IP Address
|  |-------[ DMOV H0FFFFFF00  D132 ]  'Subnet Mask
|  |-------[ DMOV H0C0A86C01  D134 ]  'Gateway IP Address
|  |-------------[ MOV K1024  D136 ]  'TCP Port Number
```

Mitsubishi FX1N/2N/2NC Series Ladder

Ladder Logic for initializing the module on a FX1s. Set the Ethernet parameters in 9 data registers starting from D1000 to D1008. If these registers are used for other purposes, set to 9 data registers starting at D2000, D3000, D4000, D5000, D6000 or D7000.

```
|---| M8002 |-----------[ DMOV H454E4554  D1000 ]  'Module ID ENET (Fixed)
|  |-------[ DMOV H0C0A86C0A  D1002 ]  'Module IP Address
|  |-------[ DMOV H0FFFFFF00  D1004 ]  'Subnet Mask
|  |-------[ DMOV H0C0A86C01  D1006 ]  'Gateway IP Address
|  |-------------[ MOV K1024  D1008 ]  'TCP Port Number
```

Note: This information has been taken from the *Mitsubishi FX2NC-ENET-ADP Ethernet Adapter Users* manual.

Using the FX3U-ENET Ethernet Module

To connect to the FX3U PLC via the FX3U-ENET Ethernet module, use Ethernet Encapsulation. In order to do so, the FX3U-ENET module must be configured using the Mitsubishi GX Developer-FX software. For more information, follow the instructions below.

1. To start, create a new GX Developer project for a FX3U model.

2. Then, select **Tools** | **FX special functions utility** | **FX Configurator-EN**.
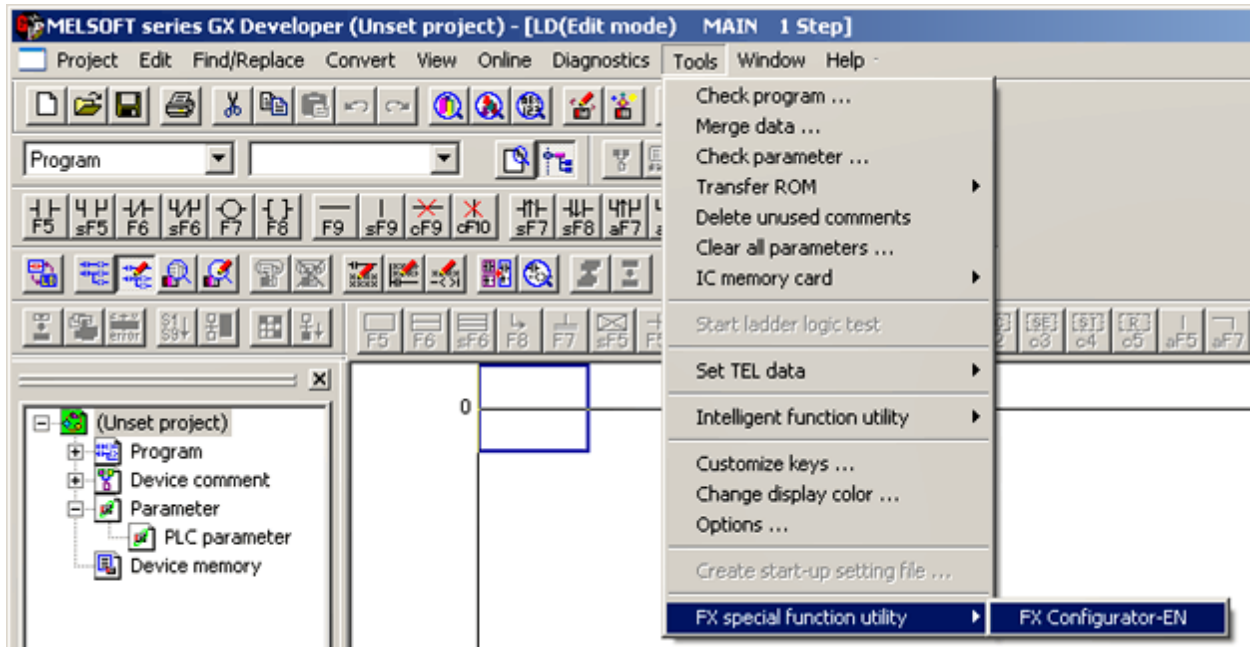
3. The FX Configurator-EN dialog should appear as shown below.
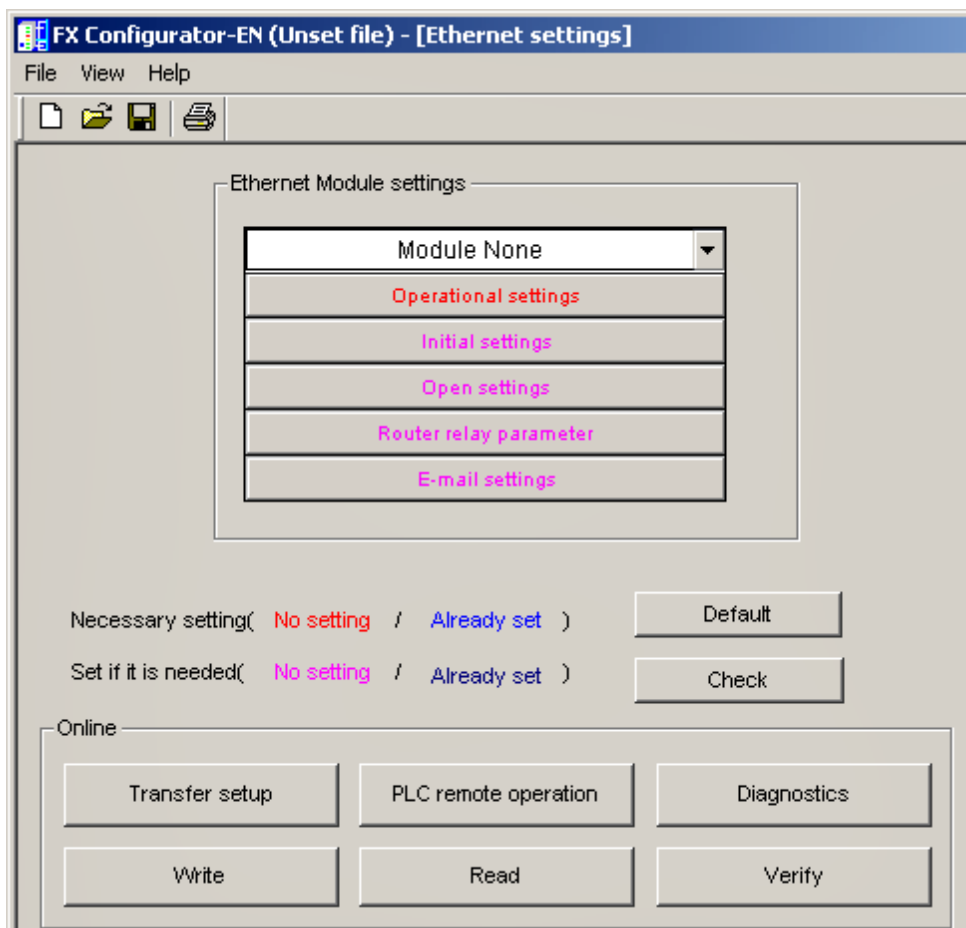
4. Next, fill in the minimum required configuration information for the FX3U-ENET block. Select a module from the first drop-down list, and then click **Operational Settings**. The settings should be similar to those shown in the image below.
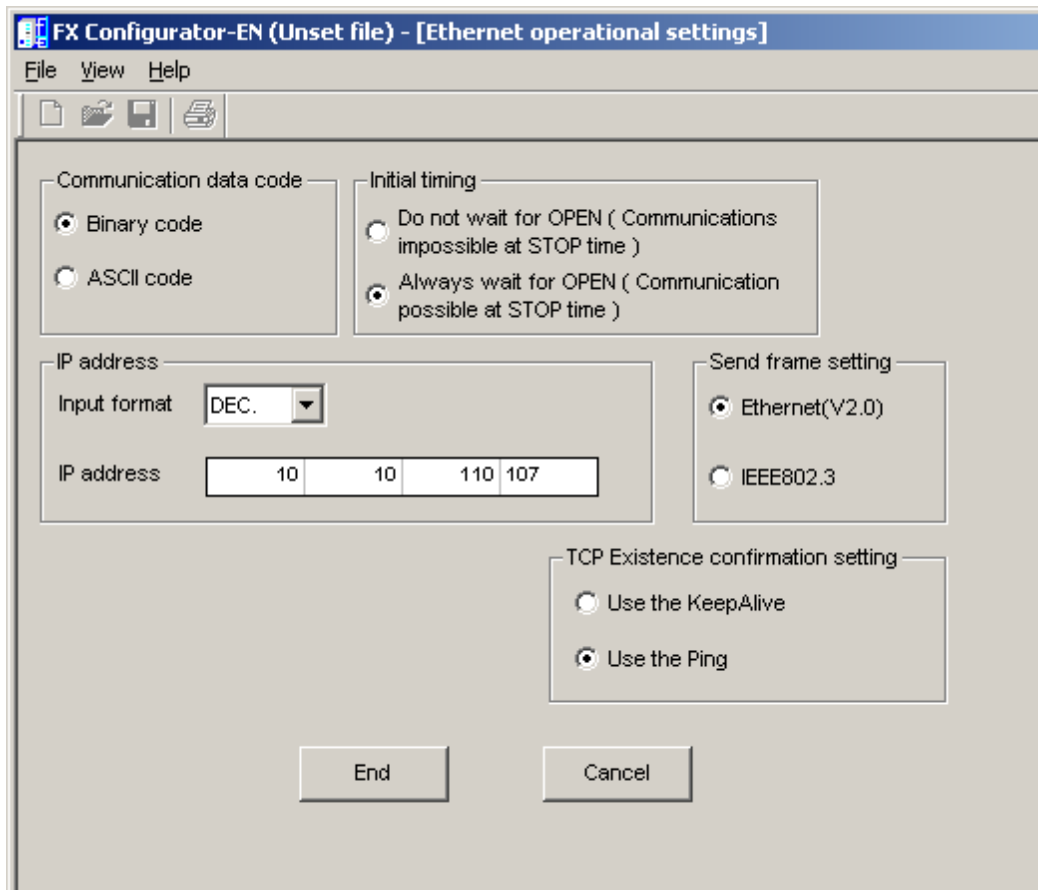
5. Click **End** to continue.

6. In the **FX Configurator-EN** dialog window, click **Open Settings**.
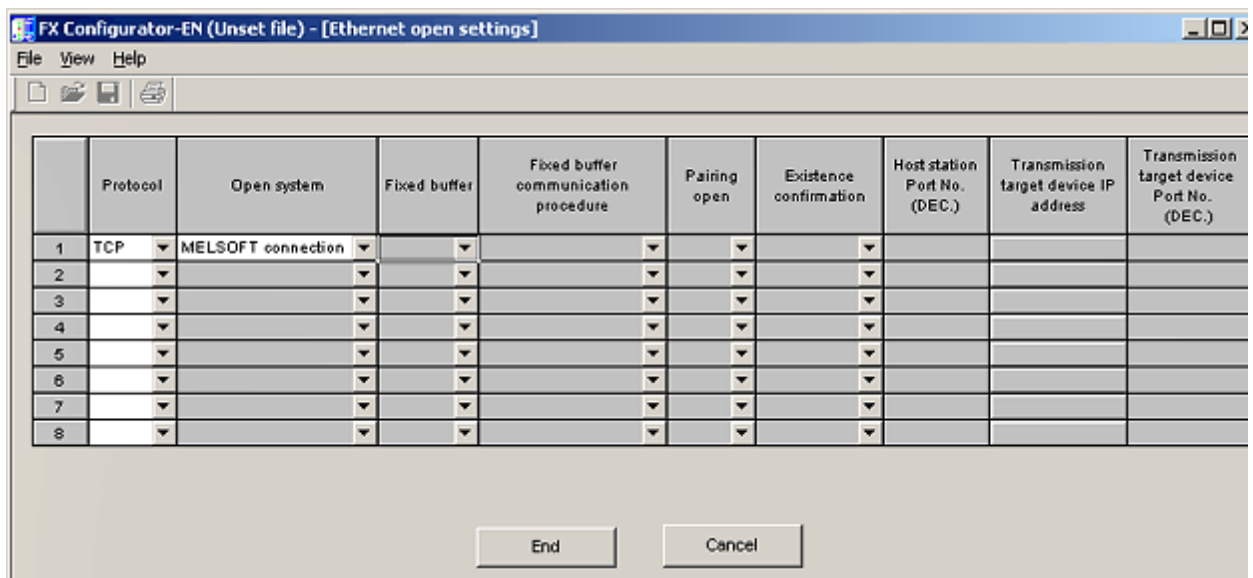
Writing Network Parameters to the PLC

After the network parameters have been specified, they must be written to the PLC. To do so, click **Write** from the **FX-Configurator-EN** window.
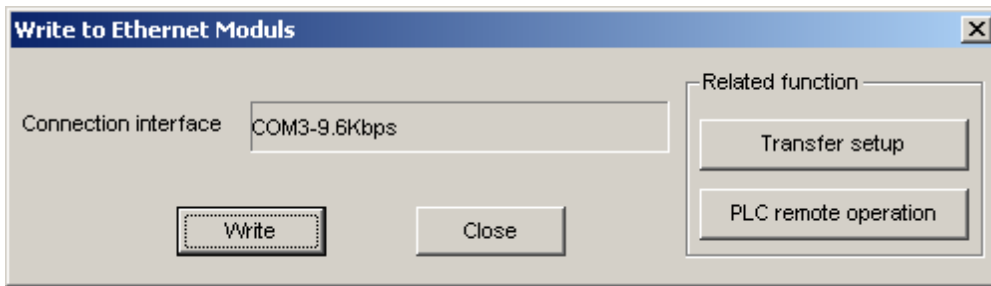
There must be a serial connection to the FX3U PLC because the configuration settings are written to the PLC via this serial link. Make sure that the communication parameters are correct by using the **Transfer setup** button above or by clicking **Online** | **Transfer setup** from the main menu.

After the parameters have been written, cycle the power on the PLC for the network parameter changes to take effect. Now, the FX3U PLC can be communicated with using Ethernet encapsulation via the FX3U-ENET Ethernet module. Use the following parameters:

- **IP Address:** 10.10.110.107 (or the IP address used above)
- **Port Number:** 5551
- **Protocol:** TCP/IP

Data Types Description

The Mitsubishi FX Net device 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 The driver interprets two consecutive registers as a single precision value by making the last register the high word and the first register the low word.
String	Null terminated ASCII string Support includes HiLo LoHi byte order selection and string lengths up to 64 bytes.

Address Descriptions

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

FX Addressing
FX0 Addressing
FX0N Addressing
FX2N Addressing
FX3U Addressing

FX Addressing

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

Device Type	Range	Data Type	Access
Inputs	X000-X377*	**Boolean**	Read Only
Outputs	Y000-Y377*	**Boolean**	Read/Write
Auxiliary Relays	M0000-M1535	**Boolean**	Read/Write
Special Aux. Relays	M8000-M8255	**Boolean**	Read/Write
States	S000-S999	**Boolean**	Read/Write
Timer Contacts	TS000-TS255	**Boolean**	Read Only
Counter Contacts	CS000-CS255	**Boolean**	Read Only
Timer Reset	TR000-TR255	**Boolean**	Read/Write
Counter Reset	CR000-CR255	**Boolean**	Read/Write
Timer Value	T000-T255	Short, **Word**	Read/Write
Counter Value	C000-C199	Short, **Word**	Read/Write
32 Bit Counter Value**	C200-C255	Long, **DWord**	Read/Write
Data Registers**	D000-D999 D000-D998	**Short**, Word Long, DWord, Float	Read/Write
Data Registers String Access HiLo Byte Ordering	DSH0000.02-DSH0967.64 The string length may also be specified using a colon. The string length can range from 2 to 64 bytes, and must be an even number.***	**String**	Read/Write
Data Registers String Access LoHi Byte Ordering	DSL0000.02-DSL0967.64 The string length may also be specified using a colon. The string length can range from 2 to 64 bytes, and must be an even number.***	**String**	Read/Write
Special Data Registers**	D8000-D8255 D8000-D8254	**Short**, Word Long, DWord, Float	Read/Write

*Octal.
**Users can specify a Long data type by appending a space and an "L" to the address. For example, "D000" would be entered as "D000 L". This does not apply to arrays or bit accessed registers.
***For example, a string with HiLo byte ordering at D10 for a length of 12 bytes would be DSH10.12.

FX0 Addressing

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

Device Type	Range	Data Type	Access
Inputs	X000-X017*	**Boolean**	Read Only
Outputs	Y000-Y15*	**Boolean**	Read/Write
Auxiliary Relays	M0000-M0511	**Boolean**	Read/Write
Special Aux. Relays	M8000-M8255	**Boolean**	Read/Write
States	S00-S63	**Boolean**	Read/Write
Timer Contacts	TS00-TS55	**Boolean**	Read Only
Counter Contacts	CS00-CS15 CS235-CS254	**Boolean**	Read Only
Timer Reset	TR00-TR55	**Boolean**	Read/Write
Counter Reset	CR00-CR15 CR235-CR254	**Boolean**	Read/Write
Timer Value	T00-T55	Short, **Word**	Read/Write
Counter Value	C00-C15	Short, **Word**	Read/Write
32 Bit Counter Value**	C235-C254	Long, **DWord**	Read/Write
Data Registers**	D00-D31 D00-D30	**Short**, Word Long, DWord, Float	Read/Write
Data Registers String Access HiLo Byte Ordering	DSH0000.02–DSH30.62 The string length may also be specified using a colon. The string length can range from 2 to 62 bytes, and must be an even number.***	**String**	Read/Write
Data Registers String Access LoHi Byte Ordering	DSL0000.02–DSL30.62 The string length may also be specified using a colon. The string length can range from 2 to 62 bytes, and must be an even number.***	**String**	Read/Write
Special Data Registers**	D8000-D8069 D8000-D8068	**Short**, Word Long, DWord, Float	Read/Write

*Octal.
**Users can specify a Long data type by appending a space and an "L" to the address. For example, "D00" would be entered as "D00 L". This does not apply to arrays or bit accessed registers.
***For example, a string with HiLo byte ordering at D10 for a length of 12 bytes would be DSH10.12.

FX0N Addressing

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

Device Type	Range	Data Type	Access
Inputs	X000-X177*	**Boolean**	Read Only
Outputs	Y000-Y177*	**Boolean**	Read/Write
Auxiliary Relays	M0000-M0511	**Boolean**	Read/Write
Special Aux. Relays	M8000-M8255	**Boolean**	Read/Write
States	S000-S127	**Boolean**	Read/Wri-

			te
Timer Contacts	TS00-TS63	**Boolean**	Read Only
Counter Contacts	CS00-CS31 CS235-CS254	**Boolean**	Read Only
Timer Reset	TR00-TR63	**Boolean**	Read/Wri-te
Counter Reset	CR00-CR31 CR235-CR254	**Boolean**	Read/Wri-te
Timer Value	T00-T63	Short, **Word**	Read/Wri-te
Counter Value	C00-C31	Short, **Word**	Read/Wri-te
32 Bit Counter Value**	C235-C254	Long, **DWord**	Read/Wri-te
Data Registers**	D000-D255 D000-D254	**Short**, Word Long, DWord, Float	Read/Wri-te
Data Registers String Access HiLo Byte Ordering	DSH0000.02-DSH223.64 The string length may also be specified using a colon. The string length can range from 2 to 64 bytes, and must be an even number.***	**String**	Read/Wri-te
Data Registers String Access LoHi Byte Ordering	DSL0000.02-DSL223.64 The string length may also be specified using a colon. The string length can range from 2 to 64 bytes, and must be an even number.***	**String**	Read/Wri-te
Special Data Registers**	D8000-D8255 D8000-D8254	**Short**, Word Long, DWord, Float	Read/Wri-te

*Octal.
**Users can specify a Long data type by appending a space and an "L" to the address. For example, "D000" would be entered as "D000 L". This does not apply to arrays or bit accessed registers.
***For example, a string with HiLo byte ordering at D10 for a length of 12 bytes would be DSH10.12.

FX2N Addressing

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

Device Type	Range	Data Type	Access
Inputs	X000-X377*	**Boolean**	Read Only
Outputs	Y000-Y377*	**Boolean**	Read/Wri-te
Auxiliary Relays	M0000-M3071	**Boolean**	Read/Wri-te
Special Aux. Relays	M8000-M8255	**Boolean**	Read/Wri-te
States	S000-S999	**Boolean**	Read/Wri-te
Timer Contacts	TS000-TS255	**Boolean**	Read Only
Counter Contacts	CS000-CS255	**Boolean**	Read Only
Timer Reset	TR000-TR255	**Boolean**	Read/Wri-te
Counter Reset	CR000-CR255	**Boolean**	Read/Wri-te
Timer Value	T000-T255	Short, **Word**	Read/Wri-te
Counter Value	C000-C199	Short, **Word**	Read/Wri-te
32 Bit Counter Value**	C200-C255	Long, **DWord**	Read/Wri-te

Data Registers**	D000-D7999 D000-D7998	**Short**, Word Long, DWord, Float	Read/Write
Data Registers String Access HiLo Byte Ordering	DSH0000.02-DSH7967.64 The string length may also be specified using a colon. The string length can range from 2 to 64 bytes, and must be an even number.***	**String**	Read/Write
Data Registers String Access LoHi Byte Ordering	DSL0000.02-DSL7967.64 The string length may also be specified using a colon. The string length can range from 2 to 64 bytes, and must be an even number.***	**String**	Read/Write
Special Data Registers**	D8000-D8255 D8000-D8254	**Short**, Word Long, DWord, Float	Read/Write

*Octal.
**Users can specify a Long data type by appending a space and an "L" to the address. For example, "D000" would be entered as "D000 L". This does not apply to arrays or bit accessed registers.
***For example, a string with HiLo byte ordering at D10 for a length of 12 bytes would be DSH10.12.

FX3U Addressing

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

Note: The FX3U model is not supported in Windows CE. It also does not support Ethernet Encapsulation.

Device Type	Range	Data Type	Access
Inputs	X000-X377*	**Boolean**	Read Only
Outputs	Y000-Y377*	**Boolean**	Read/Write
Auxiliary Relays	M0000-M7679	**Boolean**	Read/Write
Special Aux. Relays	M8000-M8511	**Boolean**	Read/Write
States	S000-S4095	**Boolean**	Read/Write
Timer Contacts	TS000-TS511	**Boolean**	Read Only
Counter Contacts	CS000-CS255	**Boolean**	Read Only
Timer Reset	TR000-TR511	**Boolean**	Read/Write
Counter Reset	CR000-CR255	**Boolean**	Read/Write
Timer Value	T000-T511	Short, **Word**	Read/Write
Counter Value	C000-C199	Short, **Word**	Read/Write
32 Bit Counter Value**	C200-C255	Long, **DWord**	Read/Write
Data Registers**	D000-D7999 D000-D7998	**Short**, Word Long, DWord, Float	Read/Write
Data Registers String Access HiLo Byte Ordering	DSH0000.02-DSH7967.64 The string length may also be specified using a colon. The string length can range from 2 to 64 bytes, and must be an even number.***	**String**	Read/Write
Data Registers String Access LoHi Byte Ordering	DSL0000.02-DSL7967.64 The string length may also be specified using a	**String**	Read/Write

	colon. The string length can range from 2 to 64 bytes, and must be an even number.***		
File Registers**	R000-R32767 R000-R32766	**Short**, Word Long, DWord, Float	Read/Write
File Registers String Access HiLo Byte Ordering	RSH0000.02-RSH32735.64 The string length may also be specified using a colon. The string length can range from 2 to 64 bytes, and must be an even number.***	**String**	Read/Write
File Registers String Access LoHi Byte Ordering	RSL0000.02-RSL32735.64 The string length may also be specified using a colon. The string length can range from 2 to 64 bytes, and must be an even number.***	**String**	Read/Write
Special Data Registers**	D8000-D8511 D8000-D8510	**Short**, Word Long, DWord, Float	Read/Write

*Octal.
**Users can specify a Long data type by appending a space and an "L" to the address. For example, "D000" would be entered as "D000 L". This does not apply to arrays or bit accessed registers.
***For example, a string with HiLo byte ordering at D10 for a length of 12 bytes would be DSH10.12

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

Serial Communications
Communications error on '<channel name>' [<error mask>]
COMn does not exist
COMn is in use by another application
Error opening COMn
Unable to set comm parameters on COMn

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

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 statically 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 statically 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 statically 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 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. 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 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.

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.

Serial Communications

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

Serial Communications
Communications error on '<channel name>' [<error mask>]
COMn does not exist
COMn is in use by another application
Error opening COMn
Unable to set comm parameters on COMn

Communications error on '<channel name>' [<error mask>]

Error Type:
Serious

Error Mask Definitions:
B = Hardware break detected.
F = Framing error.
E = I/O error.
O = Character buffer overrun.
R = RX buffer overrun.
P = Received byte parity error.
T = TX buffer full.

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

Solution:
1. Verify the cabling between the PC and the PLC device.
2. Verify the specified communications parameters match those of the device.

COMn does not exist

Error Type:
Fatal

Possible Cause:
The specified COM port is not present on the target computer.

Solution:
Verify that the proper COM port has been selected.

COMn is in use by another application

Error Type:
Fatal

Possible Cause:
The serial port assigned to a device is being used by another application.

Solution:
Verify that the correct port has been assigned to the channel.

Error opening COMn

Error Type:
Fatal

Possible Cause:
The specified COM port could not be opened due an internal hardware or software problem on the target computer.

Solution:
Verify that the COM port is functional and may be accessed by other Windows applications.

Unable to set comm parameters on COMn

Error Type:
Fatal

Possible Cause:
The serial parameters for the specified COM port are not valid.

Solution:
Verify the serial parameters and make any necessary changes.

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 serial 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. 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 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.

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

Error Type:
Serious

Possible Cause:
1. The serial 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.

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.

Index

A

B

C

D

E