Changes for the Better

MITSUBISHI ELECTRIC
AUTOMATION, INC.

Using ICC ETH-1000 EtherNet/IP Interface with Mitsubishi FX PLC

Contents

FURTHER READING REFERENCE LIST

Mitsubishi

Q Corresponding MELSEC Communication Protocol Reference Manual SH(NA)-080008-K

QnUCPU User's Manual Communication via Built-in Ethernet Port SH(NA)-080811ENG-B

Q Corresponding Ethernet Interface Module User's Manual (Basic) SH (NA)-080009-N

ICC

Instruction Manual: ETH-1000 Multiprotocol Ethernet / RS-485 Gateway

Millennium Gateway Series Protocol Driver Manuals: EtherNet/IP Server Driver Manual

Millennium Gateway Series Protocol Driver Manuals: Mitsubishi MELSEC Client Driver Manual

Chapter 1 **Introduction**

This document provides instructions and examples on how to configure a system consisting of a Rockwell ControlLogix PLC, an ICC ETH-1000 Gateway, and Mitsubishi FX PLC system. An example of the system configuration is shown in Figure 1 below.

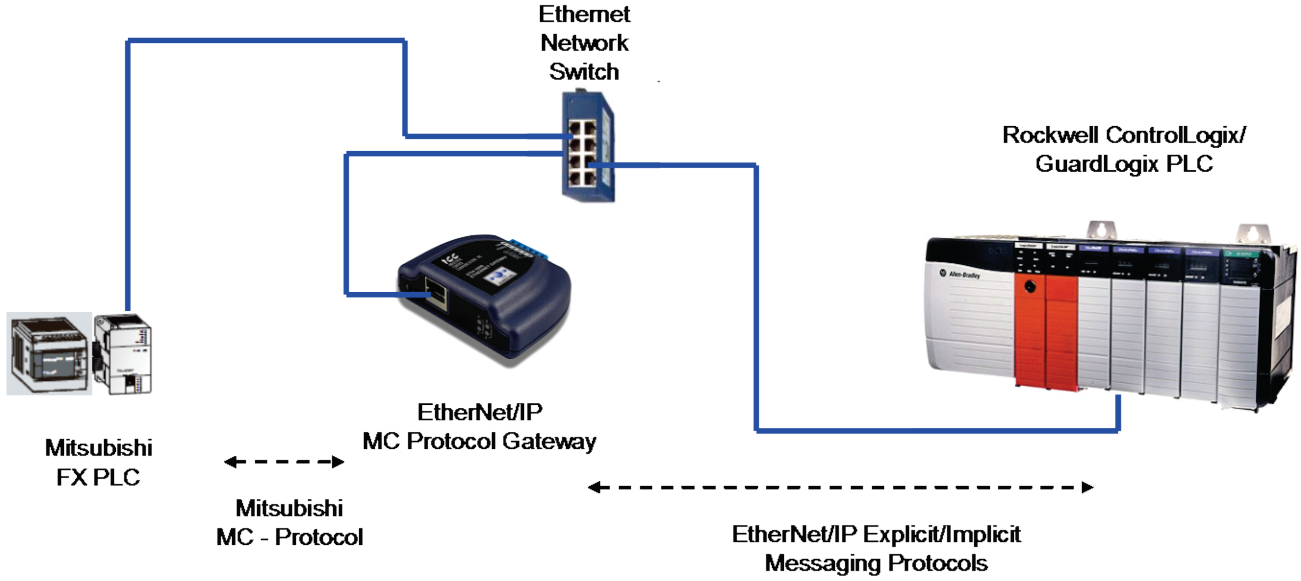

Ethernet Network Switch
Rockwell ControlLogix/ GuardLogix PLC
Mitsubishi FX PLC
EtherNet/IP MC Protocol Gateway
Mitsubishi MC - Protocol
EtherNet/IP Explicit/Implicit Messaging Protocols

Figure 1: EtherNet/IP Connectivity – Rockwell PLC to Mitsubishi FX PLC

The system configurations enable Rockwell PLCs to read and write both bit and register data of Mitsubishi FX PLCs using either EtherNet/IP Implicit or Explicit Messaging. An external FX3U-ENET Ethernet module is used for this communication.

The ICC ETH-1000 gateway module is used to convert the EtherNet/IP protocol to Mitsubishi MELSEC Communication (MC) Protocol that is supported by Mitsubishi FX PLCs.

It is assumed that the user of this guide is familiar with the Rockwell RSLogix5000 environment, the operation of Mitsubishi FX PLCs, and has sufficient knowledge of the ICC ETH-1000 Gateway. It is critical for the users to refer to the appropriate manuals when setting up the system parameters for Ethernet applications.

1 Introduction
2 System Overview
3 Connecting Devices to the Network
4 ControlLogix PLC Project Configuration
5 ETH-1000 Configuration
6 Using EtherNet/IP Explicit Messaging
7 Terminology

Chapter 2 **System Overview**

A Verification System is used as a test bed for verifying the steps documented in this Quick Start guide. The Verification System is shown in Figure 2 below with the IP address assignments of all the devices.

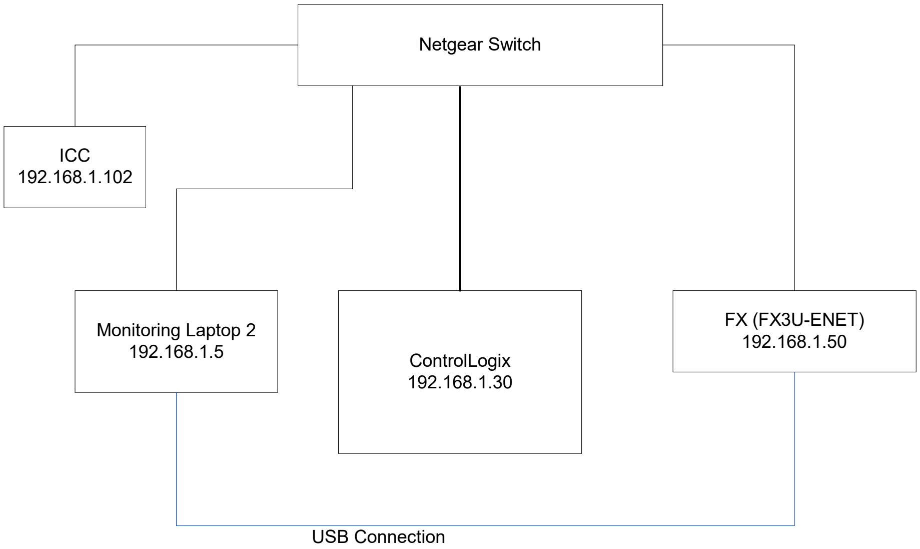


Figure 2: Architecture of an Example Verification System

The Verification System consists of one monitoring PC (e.g. Monitoring Laptop 2), a ControlLogix system, an ICC ETH-1000 module, and one Mitsubishi FX3G PLC with FX3U-ENET Ethernet module.

The following list contains high-level steps to establish proper EtherNet/IP communication of this Verification System. Each of these steps will be further detailed in subsequent chapters:

1. Connect the programming/monitoring PC, Rockwell PLC, ICC ETH-1000, and Mitsubishi PLC system to the Ethernet network
   a. Configure all devices to have proper IP addresses and subnet masks as shown above.
   b. Ensure that RSLogix5000, the ICC Configuration Studio, and Mitsubishi GX Works2 or GX Developer software packages are installed on the programming PC.
2. Create a project using the RSLogix5000 software to interface with the Mitsubishi PLC.
   a. Add the ETH-1000 as a Generic Ethernet Module in the RSLogix5000 project.
3. Configure the ETH-1000
   a. Configure the EtherNet/IP Parameters
   b. Map the ControlLogix Tags to the ETH-1000 internal database locations
   c. Map the ETH-1000 internal database locations to corresponding PLC register and/or bit locations
4. Configure FX system
   a. Configure the Ethernet port on the external FX3U-ENET module

Chapter 3 **Connecting Devices to the Network**

The steps of configuring the IP addresses of the ControlLogix PLC system, the ICC ETH-1000 module, and the PLC are documented in this chapter.

3.1 Changing the IP Address of the ControlLogix System

The minimum configuration of a ControlLogix system consists of a ControlLogix chassis (e.g. 1756-A7), a power supply (e.g. 1756-PA72), a ControlLogix Controller (e.g. 1756-L61), and an EtherNet/IP module (e.g. 1756-ENBT).

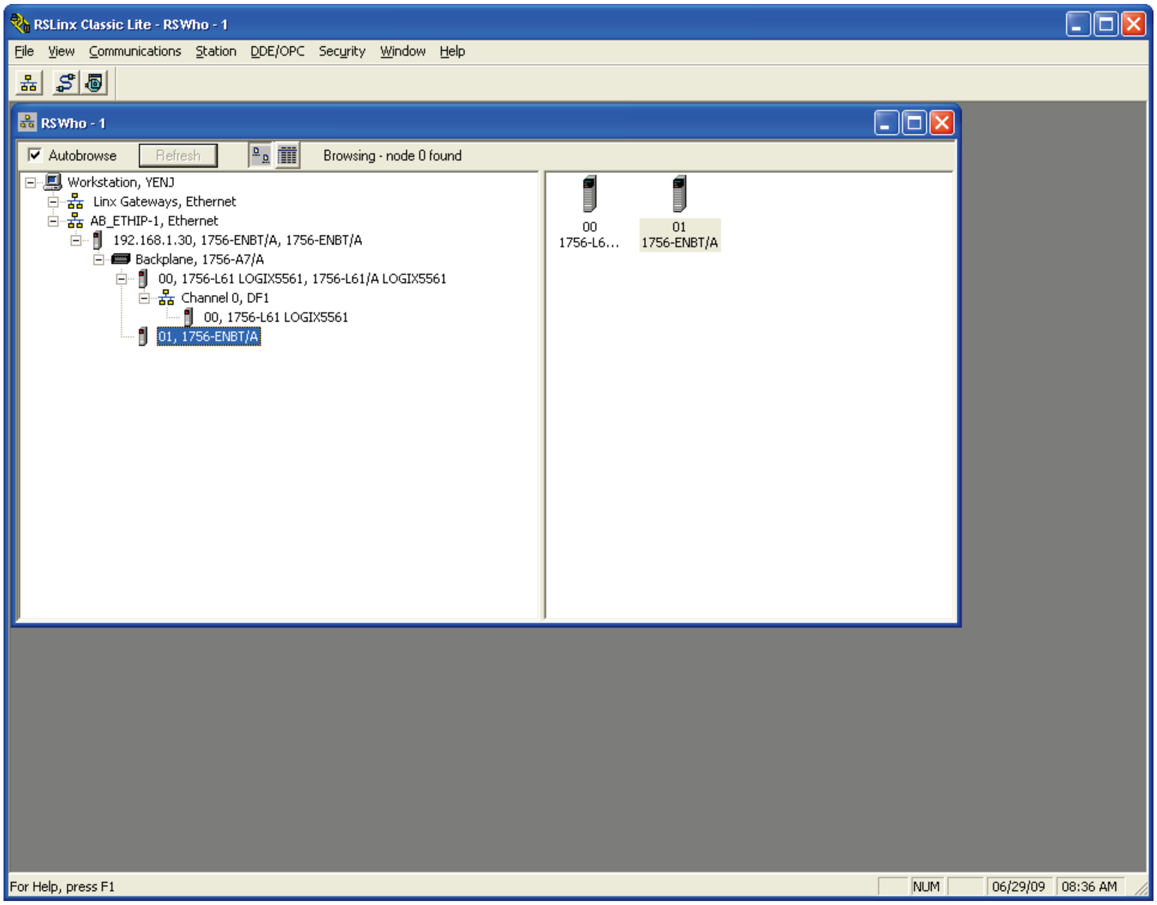
The steps here document the procedure to modify the IP address of a 1756-ENBT module to add the ControlLogix system on the same network with the Mitsubishi PLC and the ICC ETH-1000 module. It is assumed that there is already an IP address assigned to the ENBT module. If configuring a brand new ENBT module is necessary, please consult the user manual of the Rockwell ENBT module.

1. Power on the Rockwell system and monitor the display scrolling across the front of the ENBT module. If the module is working properly, the message "*OK Rev x.x.x IP1.IP2.IP3.IP4*" should scroll across the display. Rev x.x.x is the revision number of the module firmware, and IP1.IP2.IP3.IP4 forms the current IP address of the module. For example, the message is "*OK Rev 2.3.1 192.168.1.30*"

2. Connect a configuration PC on the network. Change the IP address of the PC to be in the same subnet as the 1756-ENBT module.

   - *Note: it is assumed that changing the IP address on a PC is known by the users of this manual. If required, please consult Windows OS Help File.*

3. Open the RSLinx Classic on the configuration PC and select Communications → RSWho and expand the tree to see the following screen showing the system configuration:

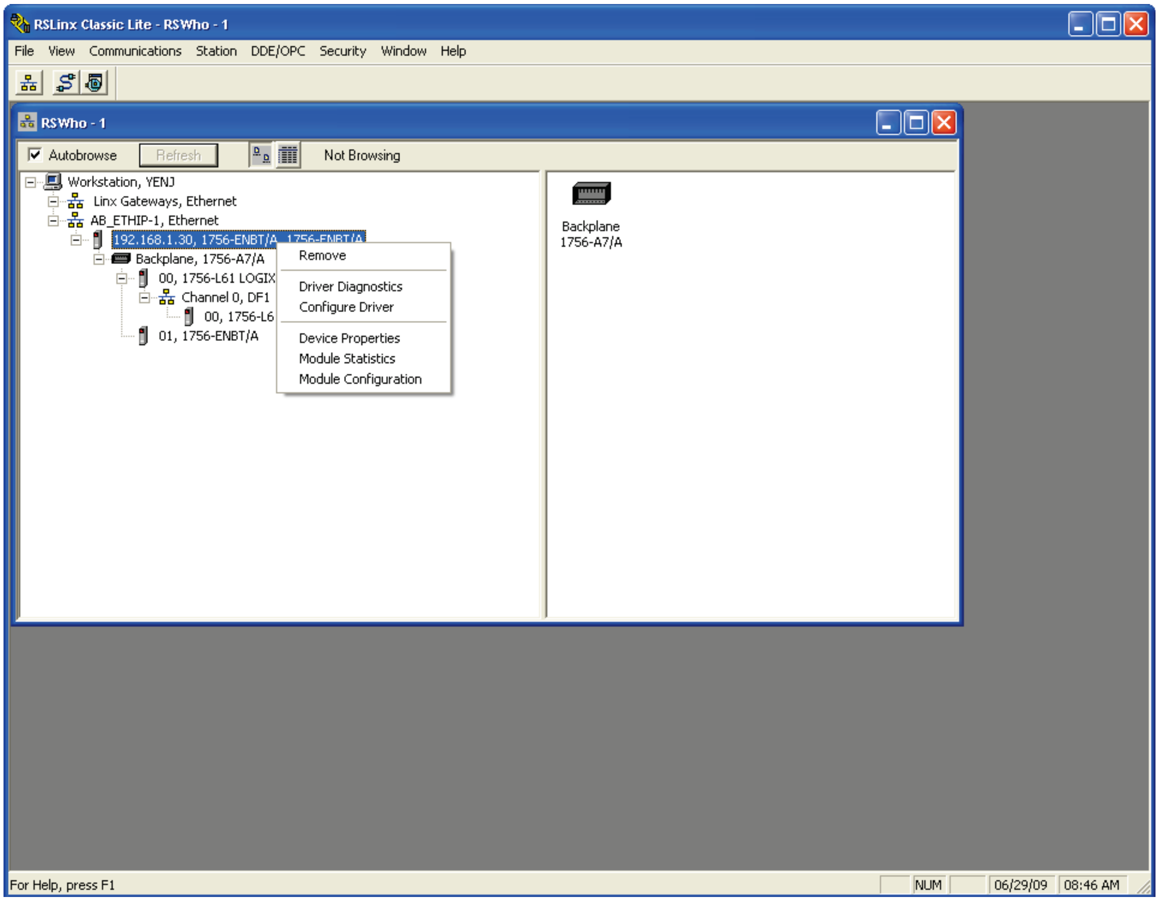

RSLinx Classic Lite - RSWho - 1
File View Communications Station DDE/OPC Security Window Help
RSWho - 1
☑ Autobrowse
Refresh
00
1756-L6...
01
1756-ENBT/A

Browsing - node 0 found

```
Workstation, YENJ
  Linx Gateways, Ethernet
  AB_ETHIP-1, Ethernet
    192.168.1.30, 1756-ENBT/A, 1756-ENBT/A
      Backplane, 1756-A7/A
        00, 1756-L61 LOGIX5561, 1756-L61/A LOGIX5561
          Channel 0, DF1
            00, 1756-L61 LOGIX5561
        01, 1756-ENBT/A
```

For Help, press F1 NUM 06/29/09 08:36 AM

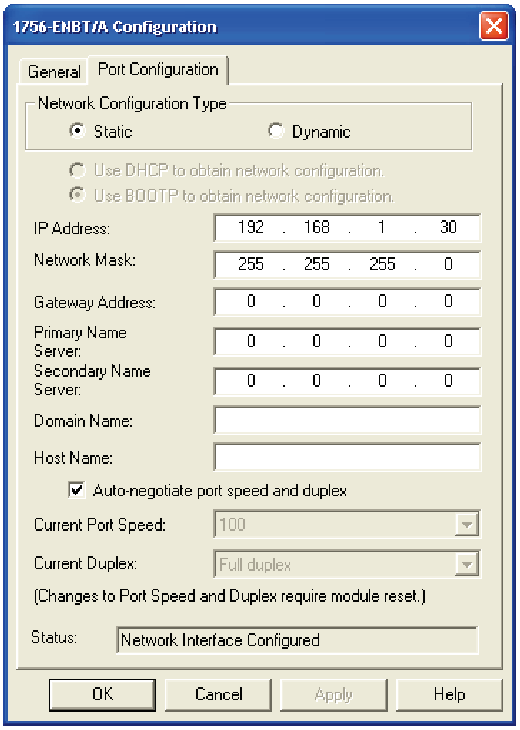
4. Right-Click on the ENBT Module and select “Module Configuration” from the drop down list:



Workstation, YENJ
Linx Gateways, Ethernet
AB_ETHIP-1, Ethernet
192.168.1.30, 1756-ENBT/A, 1756-ENBT/A
Backplane, 1756-A7/A
00, 1756-L61 LOGIX
Channel 0, DF1
00, 1756-L6
01, 1756-ENBT/A

5. Under the “Port Configuration” tab, select the Network Configuration Type to be “Static”, and one can modify the IP address and the Network Mask to the desired values. Click “OK” and save the new IP address configuration. The new IP address should scroll across the front of the ENBT module.

6. Once the new IP address is set, it is very likely (depending on what IP address and Network mask were assigned to the ENBT module) that the Configuration PC will no longer be able to communicate with the ENBT module. The IP address of the Configuration PC will need to be changed to be in the same subnet of the ENBT module before the communication can be re-established.



Network Configuration Type
● Static ○ Dynamic
○ Use DHCP to obtain network configuration.
● Use BOOTP to obtain network configuration.

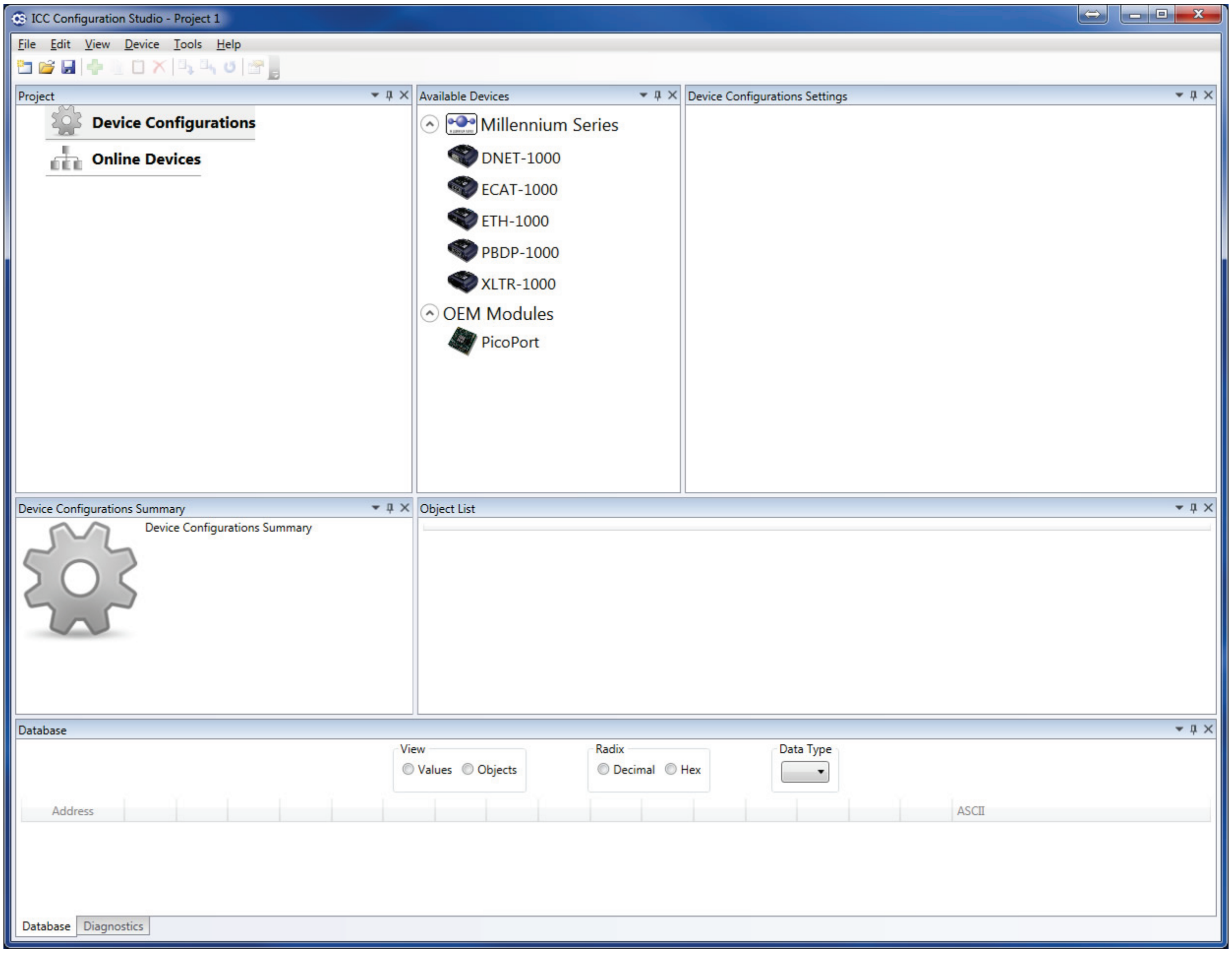
IP Address: 192 . 168 . 1 . 30
Network Mask: 255 . 255 . 255 . 0
Gateway Address: 0 . 0 . 0 . 0
Primary Name Server: 0 . 0 . 0 . 0
Secondary Name Server: 0 . 0 . 0 . 0
Domain Name:
Host Name:
☑ Auto-negotiate port speed and duplex
Current Port Speed: 100
Current Duplex: Full duplex
(Changes to Port Speed and Duplex require module reset.)
Status: Network Interface Configured

3.2 Changing the IP Address of the ETH-1000

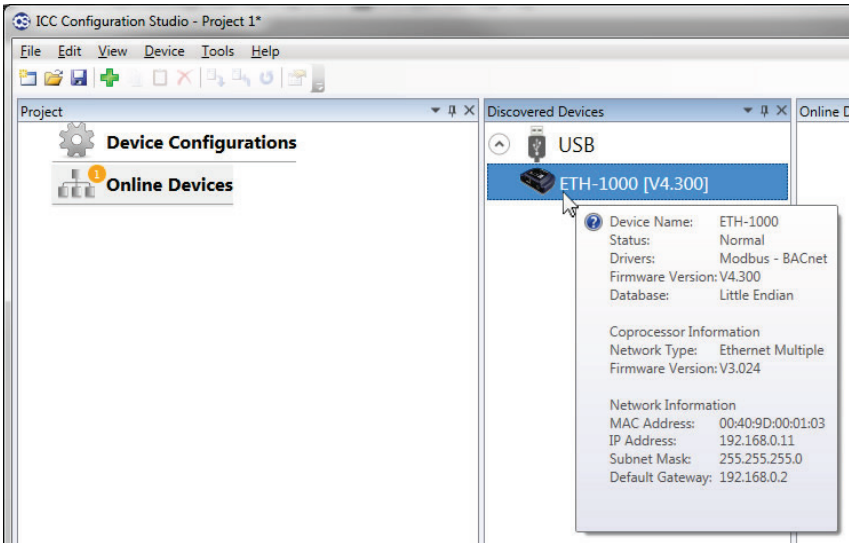
The ICC Configuration Studio software should be installed onto the PC that will be used to configure the ETH-1000. The ETH-1000 can be powered using an USB connection, a Power over Ethernet (PoE) connection or an external 7- 24V DC power supply.

A USB cable must be connected between the Configuration PC and the ETH-1000 to configure the ETH-1000.

1. Launch the ICC Configuration Studio on the PC and connect a USB cable to the ETH-1000.



2. Click on the “Online Devices” heading to display the “Discovered Devices” panel. Move the mouse cursor over the ETH-1000 to display device information such as the firmware versions, database endianness, and network info.

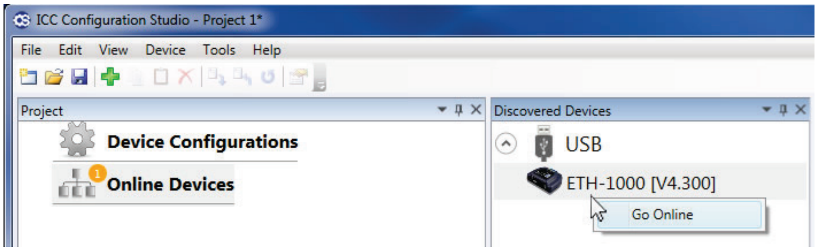


Device Name: ETH-1000
Status: Normal
Drivers: Modbus - BACnet
Firmware Version: V4.300
Database: Little Endian

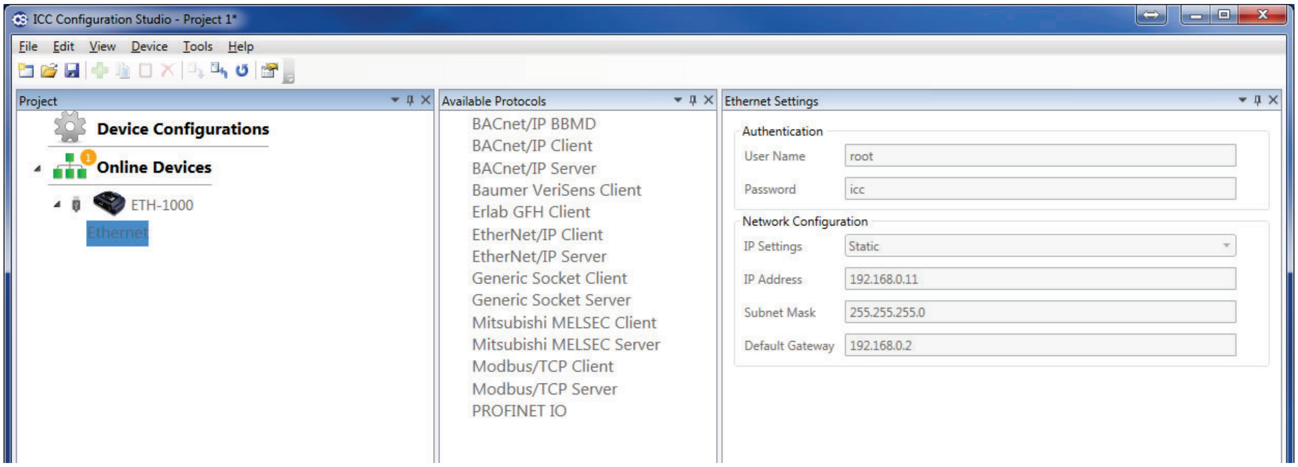
Coprocessor Information
Network Type: Ethernet Multiple
Firmware Version: V3.024

Network Information
MAC Address: 00:40:9D:00:01:03
IP Address: 192.168.0.11
Subnet Mask: 255.255.255.0
Default Gateway: 192.168.0.2

3. In the “Discovered Devices” panel, right-click on the ETH-1000 and select “Go Online”.



4. The ETH-1000 module will be connected, and the screen will be populated with the current configuration:



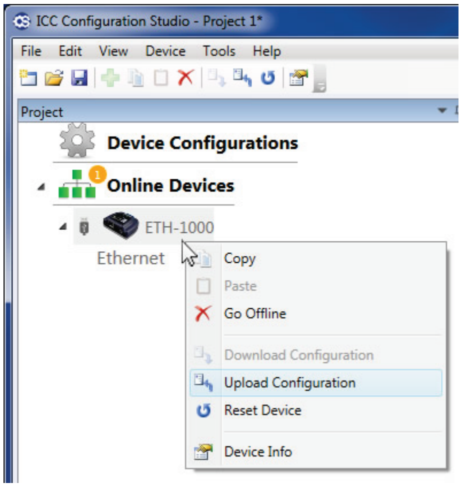
Available Protocols
- BACnet/IP BBMD
- BACnet/IP Client
- BACnet/IP Server
- Baumer VeriSens Client
- Erlab GFH Client
- EtherNet/IP Client
- EtherNet/IP Server
- Generic Socket Client
- Generic Socket Server
- Mitsubishi MELSEC Client
- Mitsubishi MELSEC Server
- Modbus/TCP Client
- Modbus/TCP Server
- PROFINET IO

Ethernet Settings

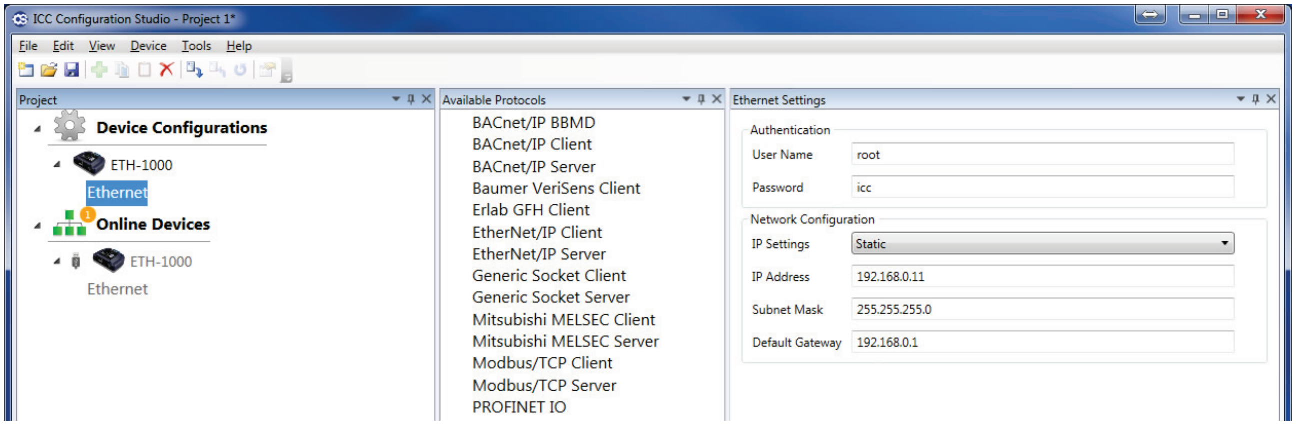
Authentication
User Name: root
Password: icc

Network Configuration
IP Settings: Static
IP Address: 192.168.0.11
Subnet Mask: 255.255.255.0
Default Gateway: 192.168.0.2

5. Note that it is NOT possible to edit the “Online Devices” configuration. The configuration must be uploaded into the “Device Configurations” heading before the configuration can be modified. Right-click on the online ETH-1000 and select “Upload Configuration”.



6. The user can now modify the IP Address and Subnet Mask.



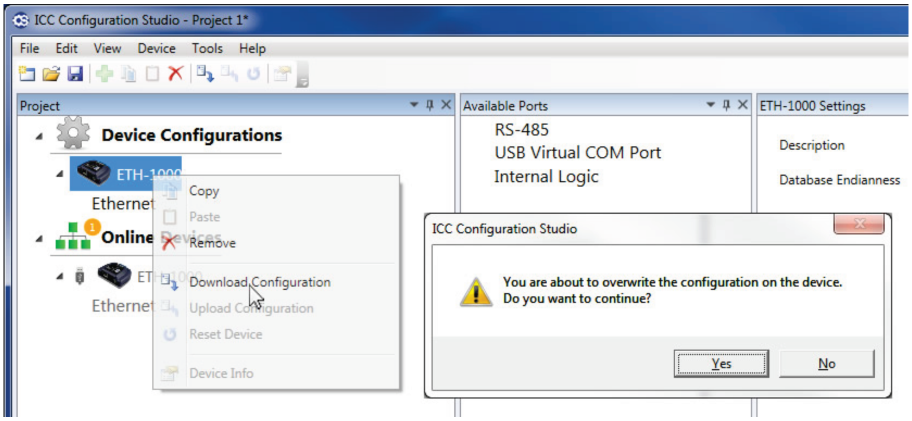
Authentication

User Name	root
Password	icc

Network Configuration

IP Settings	Static
IP Address	192.168.0.11
Subnet Mask	255.255.255.0
Default Gateway	192.168.0.1

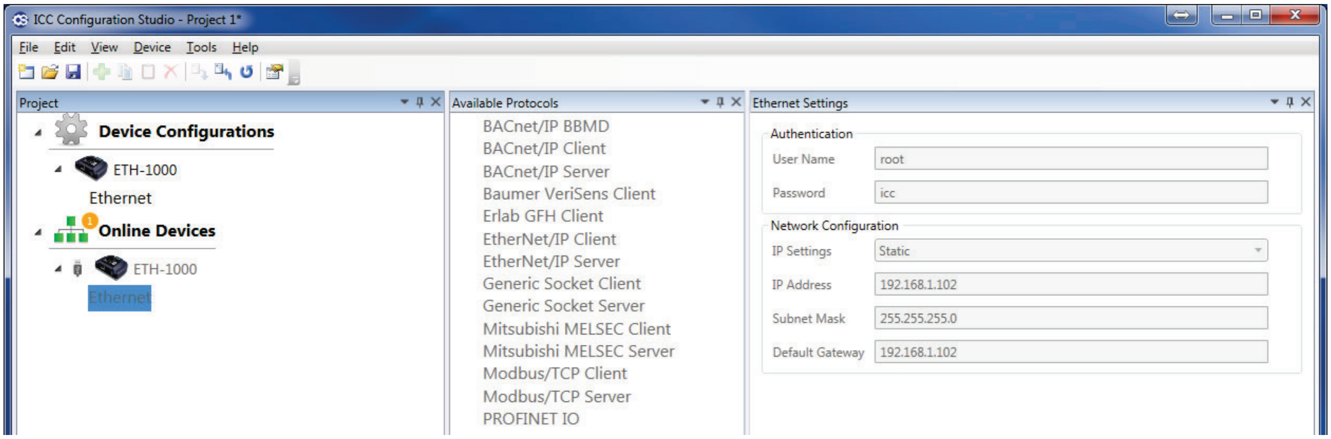
7. After making the changes, right-click on the ETH-1000 and select “Download Configuration” to load the new configuration into the online ETH-1000. A warning message will pop-up. Click the “Yes” button to continue downloading the new IP address. The ETH-1000 will automatically reset for changes to take effect.



ICC Configuration Studio

You are about to overwrite the configuration on the device.
Do you want to continue?

8. After the download and reset is complete, the online ETH-1000 configuration will show the new IP Address. In this example, the IP Address was changed to 192.168.1.102.


ICC Configuration Studio - Project 1*
File Edit View Device Tools Help
Project
Available Protocols
Ethernet Settings

Project

- Device Configurations
  - ETH-1000
    - Ethernet
- Online Devices
  - ETH-1000
    - Ethernet

Available Protocols

- BACnet/IP BBMD
- BACnet/IP Client
- BACnet/IP Server
- Baumer VeriSens Client
- Erlab GFH Client
- EtherNet/IP Client
- EtherNet/IP Server
- Generic Socket Client
- Generic Socket Server
- Mitsubishi MELSEC Client
- Mitsubishi MELSEC Server
- Modbus/TCP Client
- Modbus/TCP Server
- PROFINET IO

Ethernet Settings

Authentication

User Name: root

Password: icc

Network Configuration

IP Settings: Static

IP Address: 192.168.1.102

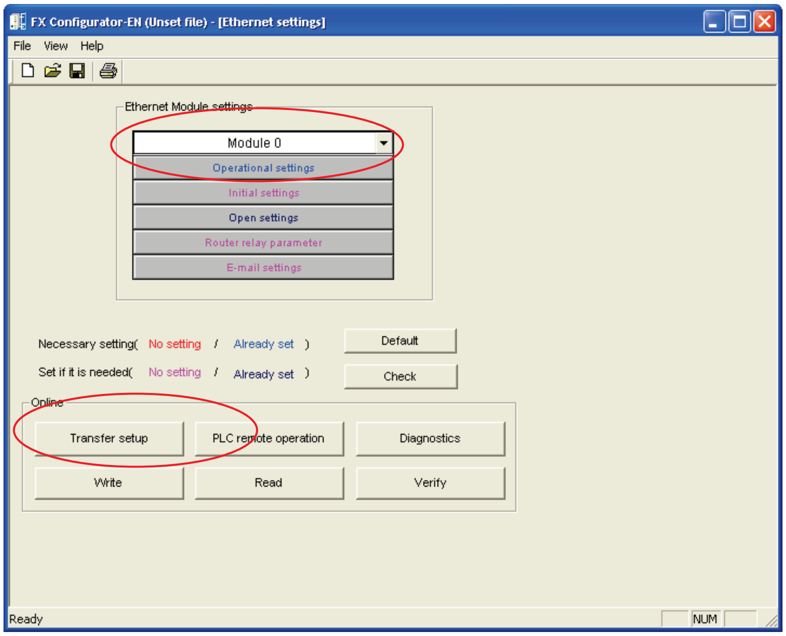
Subnet Mask: 255.255.255.0

Default Gateway: 192.168.1.102

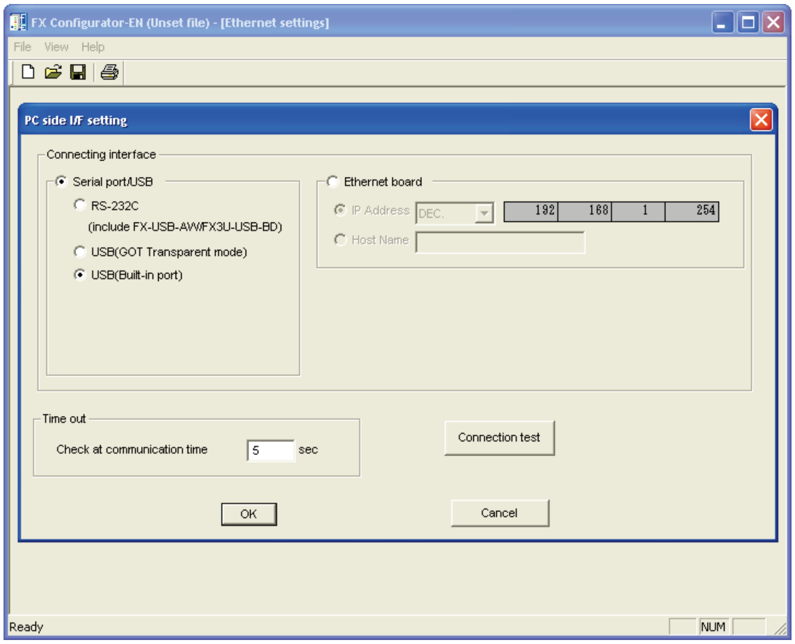
3.3 Changing the IP Address of the FX3U-ENET Ethernet Module

The following are steps to change the IP address of the FX3U-ENET Ethernet module:

1. The FX Configurator-EN utility software must be installed on the Configuration PC to configure the module. Contact Mitsubishi technical support to obtain the latest version of the FX Configurator-EN utility.
2. Run the configuration software and select “Module 0”.



3. Connect a USB cable from the Configuration PC and click the “Transfer Setup” button. On the window that appears, select “USB (Built-in port)” for connection to the FX CPU. Click the “Connection test” button to ensure the connection to the FX CPU has been properly established. Then, click “OK” to return to the main menu.

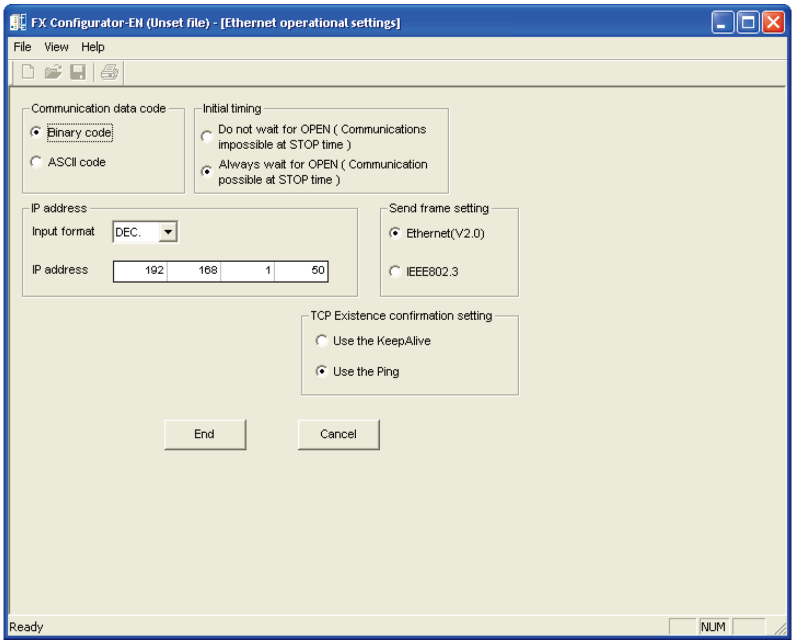


● Serial port/USB
○ RS-232C (include FX-USB-AW/FX3U-USB-BD)
○ USB(GOT Transparent mode)
● USB(Built-in port)

○ Ethernet board
IP Address DEC. 192 168 1 254
○ Host Name

Time out
Check at communication time 5 sec

4. At the Main Menu, select “Operational Settings” to assign the IP address and perform other configuration. In the example Verification System, the IP address is set to 192.168.1.50. “Initial Timing” must be set to “Always wait for Open”. When configuration is complete, click the “End” button to save the configuration and return to the Main Menu.



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

Communication data code
- ● Binary code
- ○ ASCII code

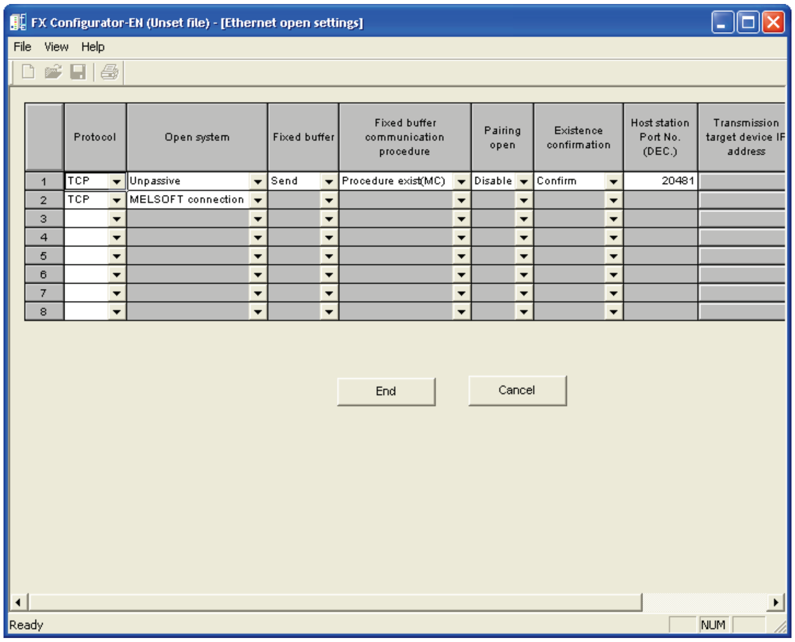
Initial timing
- ○ Do not wait for OPEN (Communications impossible at STOP time)
- ● Always wait for OPEN (Communication possible at STOP time)

IP address
- Input format: DEC.
- IP address: 192 . 168 . 1 . 50

Send frame setting
- ● Ethernet(V2.0)
- ○ IEEE802.3

TCP Existence confirmation setting
- ○ Use the KeepAlive
- ● Use the Ping

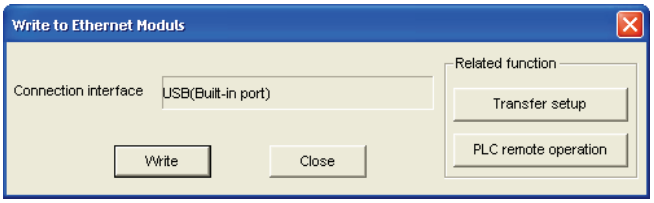
5. At the Main Menu, select “Initial Settings” and enter the parameters as shown below. The Channel 1 configuration shown allows MC Protocol to be used with the FX3U-ENET module. It is important to ensure that the “Host Station Port Number” setting matches the ETH-1000’s Connection Object port number (to be configured later in this document). The Channel 2 configuration shown also allows GX Works2 or GX Developer to interface with the FX PLC via Ethernet. Click the “End” button to accept the configuration and return to the Main Menu.



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 IF address
1	TCP	Unpassive	Send	Procedure exist(MC)	Disable	Confirm	20481	
2	TCP	MELSOFT connection						
3								
4								
5								
6								
7								
8								

6. At the Main Menu, click on the “Write” button. The following pop-up window will then appear. Click on the “Write” button in the pop-up window to transfer the configuration to the FX3U-ENET module.



Write to Ethernet Moduls

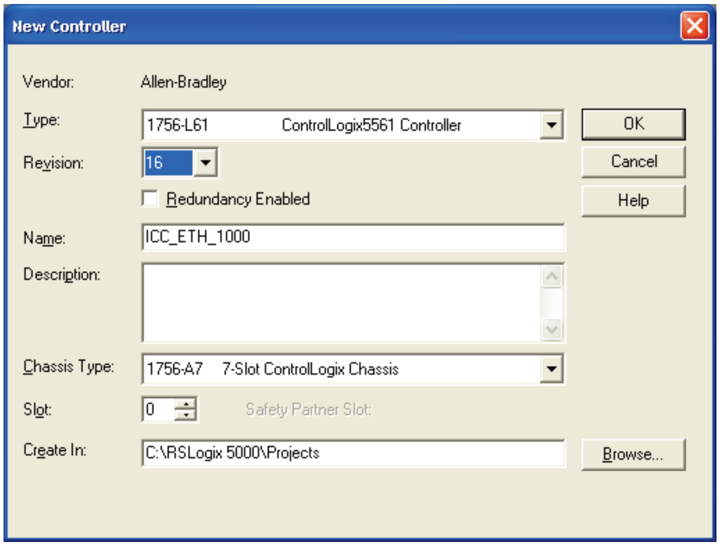
Connection interface: USB(Built-in port)

Chapter 4 ControlLogix PLC Project Configuration

The configuration steps of a ControlLogix project are described in this chapter. These steps are used to communicate with an ICC ETH-1000 module. It is assumed that the user has basic knowledge in using RSLogix5000 software to perform the basic configuration steps.

4.1 Adding the 1756 ENBT Module

1. Create a new RSLogix5000 project using the proper revision level of the ControlLogix controller. In this example, the revision level is 16.



Vendor: Allen-Bradley

Type: 1756-L61 ControlLogix5561 Controller

Revision: 16

☐ Redundancy Enabled

Name: ICC_ETH_1000

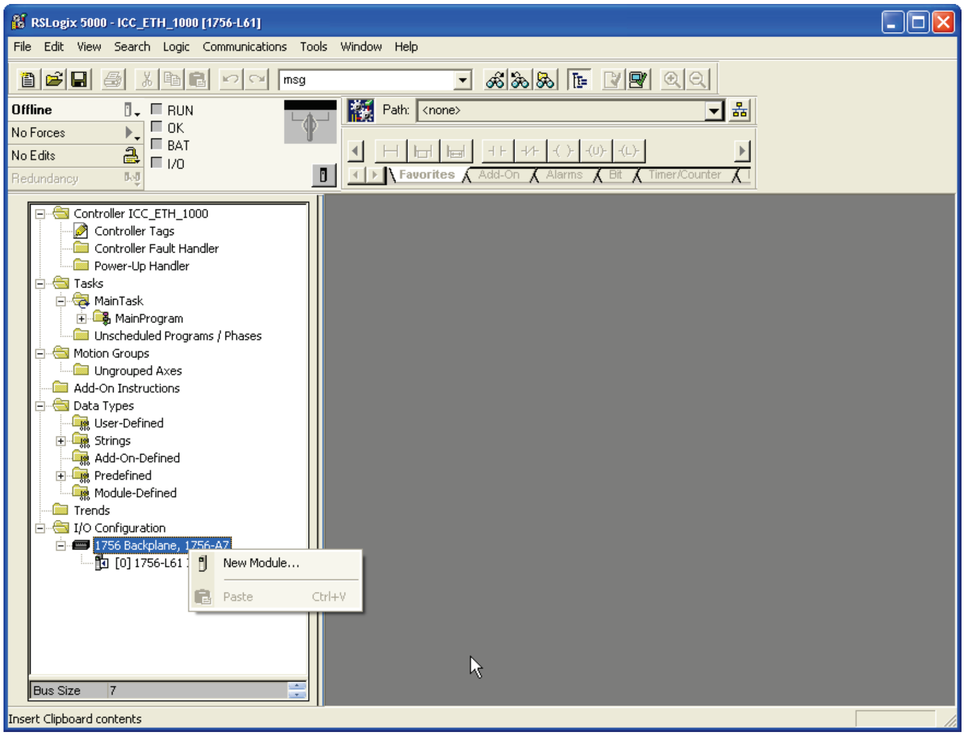
Description:

Chassis Type: 1756-A7 7-Slot ControlLogix Chassis

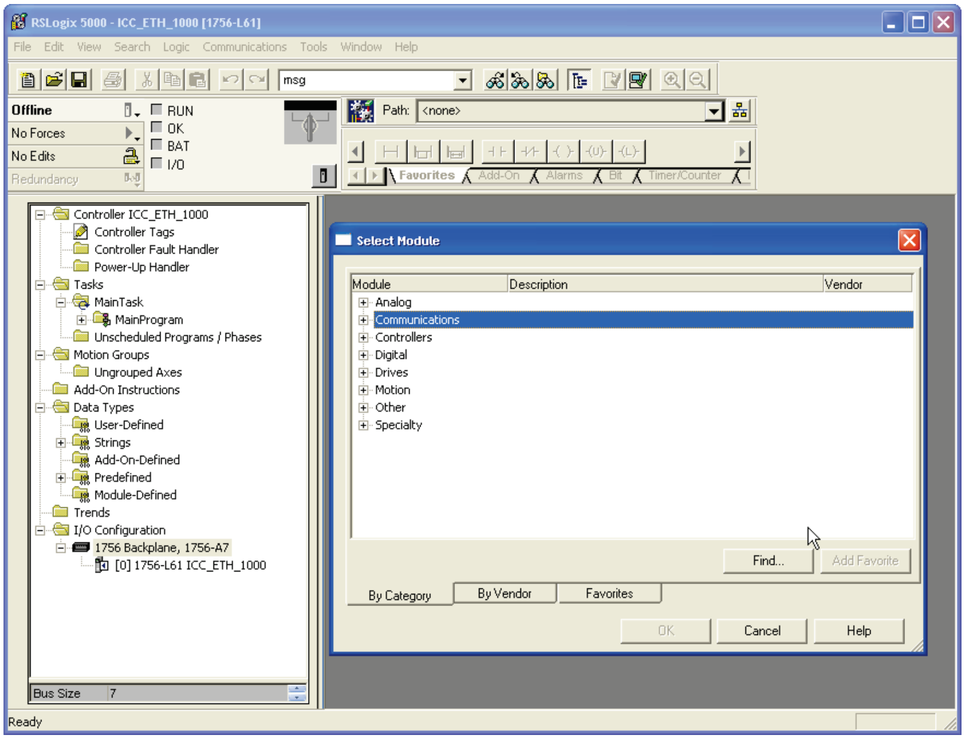
Slot: 0 Safety Partner Slot:

Create In: C:\RSLogix 5000\Projects

2. Right Click on the 1756 Backplane Selection and choose “New Module…”

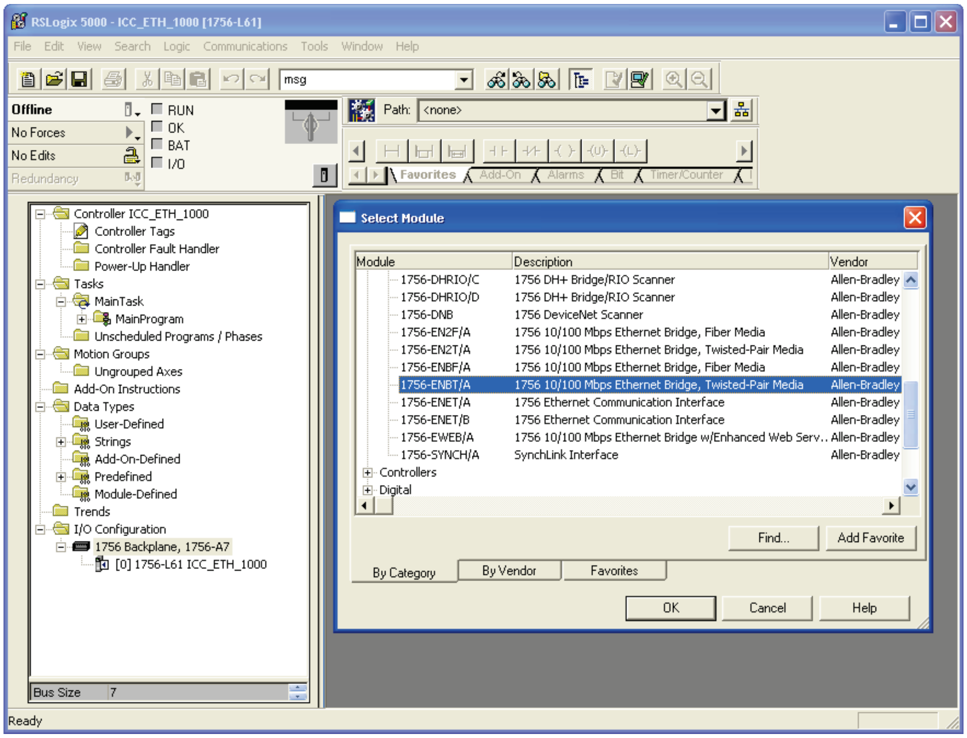


3. In the “Select Module” pop-up window, choose “Communications”.



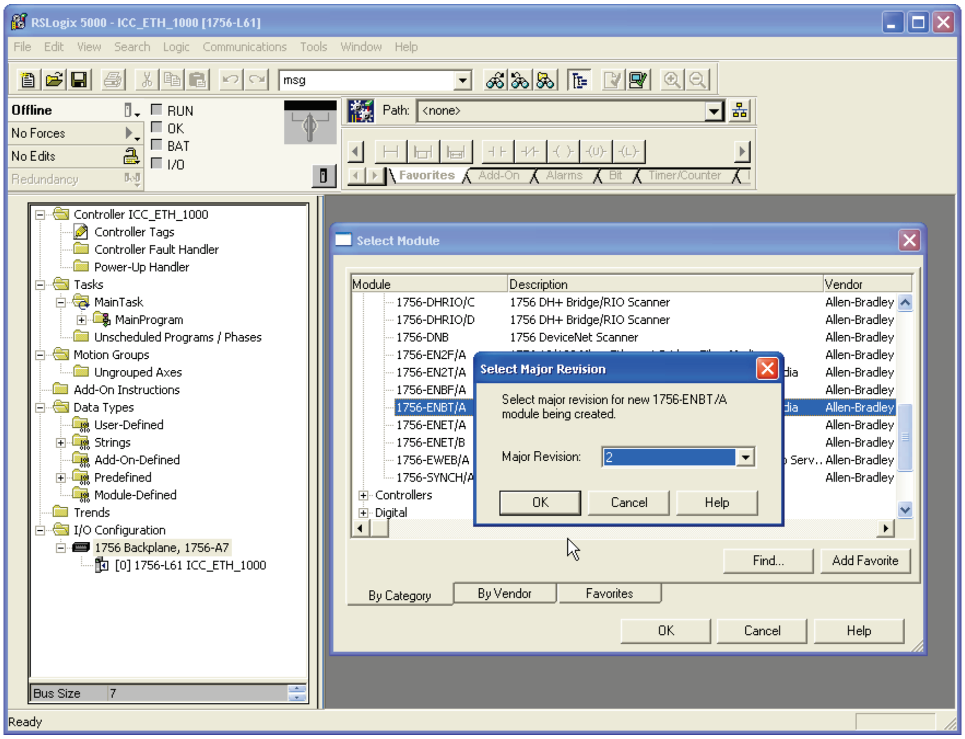
Module	Description	Vendor
Analog		
Communications		
Controllers		
Digital		
Drives		
Motion		
Other		
Specialty		

4. Select the 1756-ENBT module.



Module	Description	Vendor
1756-DHRIO/C	1756 DH+ Bridge/RIO Scanner	Allen-Bradley
1756-DHRIO/D	1756 DH+ Bridge/RIO Scanner	Allen-Bradley
1756-DNB	1756 DeviceNet Scanner	Allen-Bradley
1756-EN2F/A	1756 10/100 Mbps Ethernet Bridge, Fiber Media	Allen-Bradley
1756-EN2T/A	1756 10/100 Mbps Ethernet Bridge, Twisted-Pair Media	Allen-Bradley
1756-ENBF/A	1756 10/100 Mbps Ethernet Bridge, Fiber Media	Allen-Bradley
1756-ENBT/A	1756 10/100 Mbps Ethernet Bridge, Twisted-Pair Media	Allen-Bradley
1756-ENET/A	1756 Ethernet Communication Interface	Allen-Bradley
1756-ENET/B	1756 Ethernet Communication Interface	Allen-Bradley
1756-EWEB/A	1756 10/100 Mbps Ethernet Bridge w/Enhanced Web Serv..	Allen-Bradley
1756-SYNCH/A	SynchLink Interface	Allen-Bradley
Controllers		
Digital		

5. Select the Major Revision level of the ENBT firmware. In the Verification System, the major revision level of the ENBT module is 2.



RSLogix 5000 - ICC_ETH_1000 [1756-L61]

Select Module

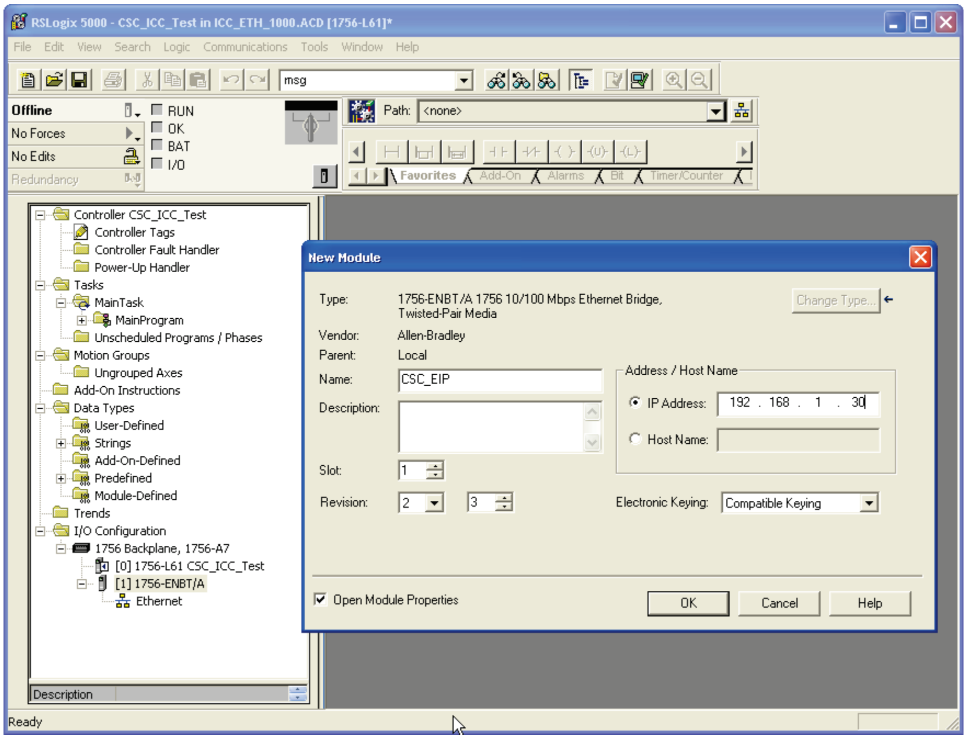
Module	Description	Vendor
1756-DHRIO/C	1756 DH+ Bridge/RIO Scanner	Allen-Bradley
1756-DHRIO/D	1756 DH+ Bridge/RIO Scanner	Allen-Bradley
1756-DNB	1756 DeviceNet Scanner	Allen-Bradley
1756-EN2F/A		Allen-Bradley
1756-EN2T/A	dia	Allen-Bradley
1756-ENBF/A		Allen-Bradley
1756-ENBT/A	dia	Allen-Bradley
1756-ENET/A		Allen-Bradley
1756-ENET/B		Allen-Bradley
1756-EWEB/A	Serv...	Allen-Bradley
1756-SYNCH/A		Allen-Bradley
Controllers		
Digital		

Select Major Revision

Select major revision for new 1756-ENBT/A module being created.

Major Revision: 2

6. Enter the Name, Slot Location, Revision Level and IP Address of the ENBT module. In the Verification System, the module name is “CSC_EIP”, the firmware Revision is 2.3, the module is in Slot 1 of the ControlLogix chassis, and the IP address is set to 192.168.1.30, matching the previous configuration.



RSLogix 5000 - CSC_ICC_Test in ICC_ETH_1000.ACD [1756-L61]*

New Module

Type: 1756-ENBT/A 1756 10/100 Mbps Ethernet Bridge, Twisted-Pair Media

Vendor: Allen-Bradley

Parent: Local

Name: CSC_EIP

Description:

Slot: 1

Revision: 2 3

Address / Host Name

● IP Address: 192 . 168 . 1 . 30

○ Host Name:

Electronic Keying: Compatible Keying

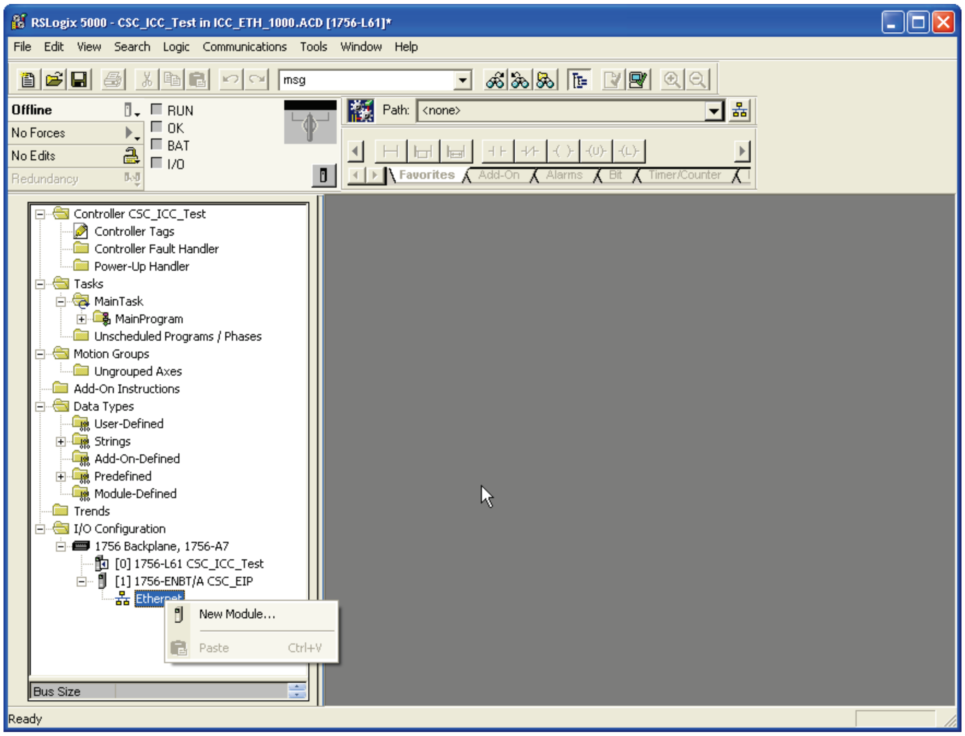
☑ Open Module Properties

7. Click “OK” to accept the configuration and make no additional configuration changes to the “Connection” tab. Click “OK” again to accept the configuration.

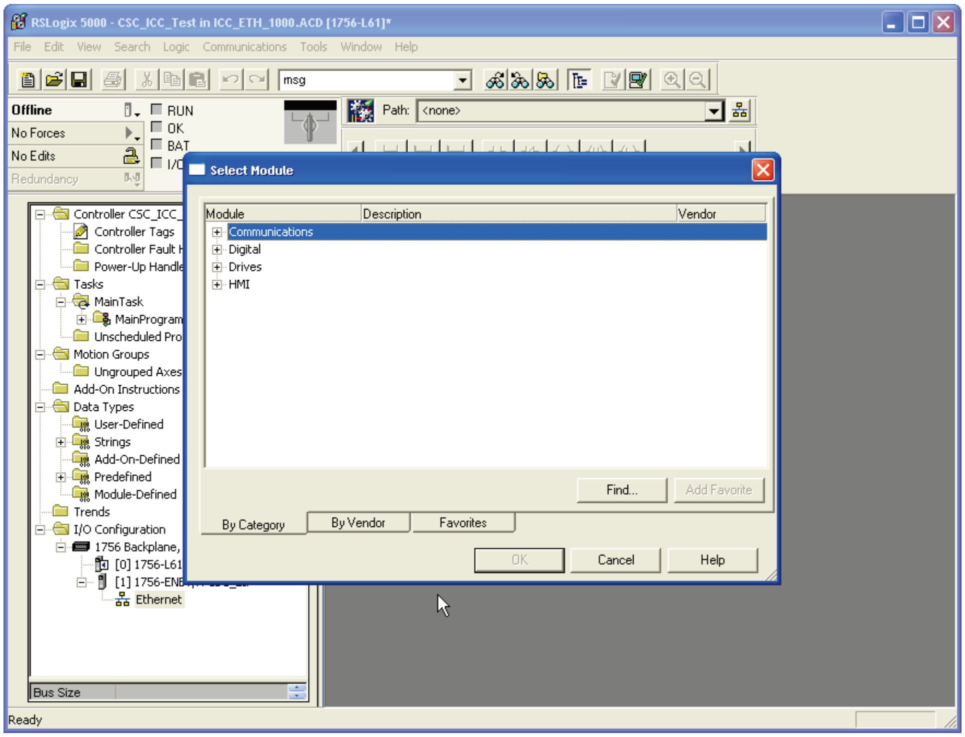
4.2 Adding the ETH-1000 Module

The following steps are used to add the ETH-1000 module for communication using I/O (implicit) messaging.

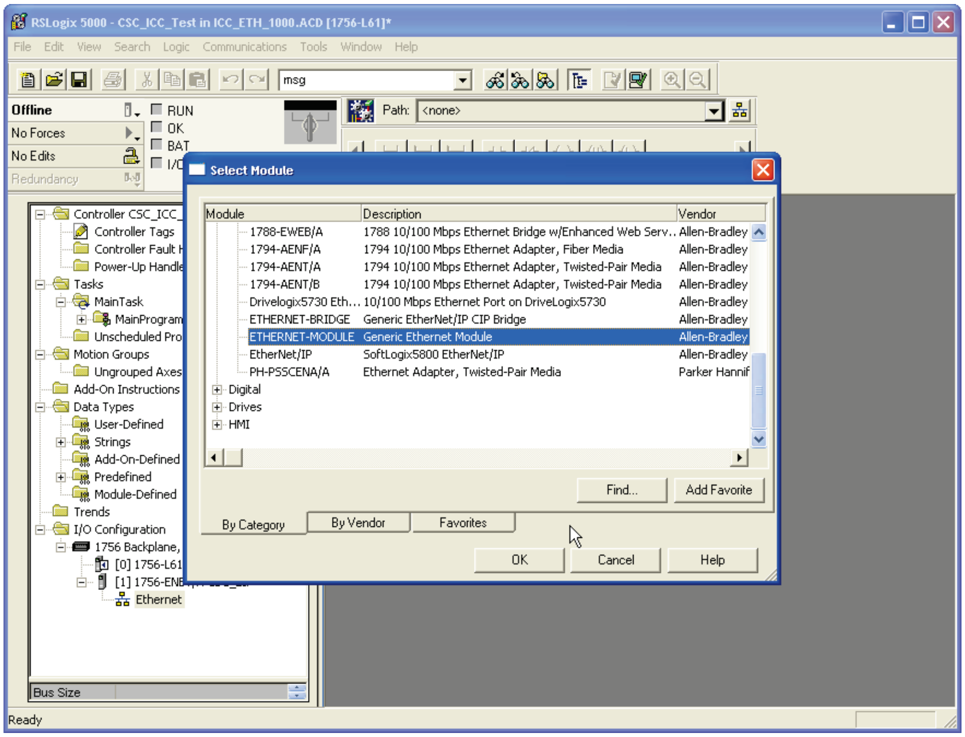
1. Right click on the “Ethernet” icon under the ENBT module and select “New Module…”



2. Select “Communications” and expand the tree for additional selections.

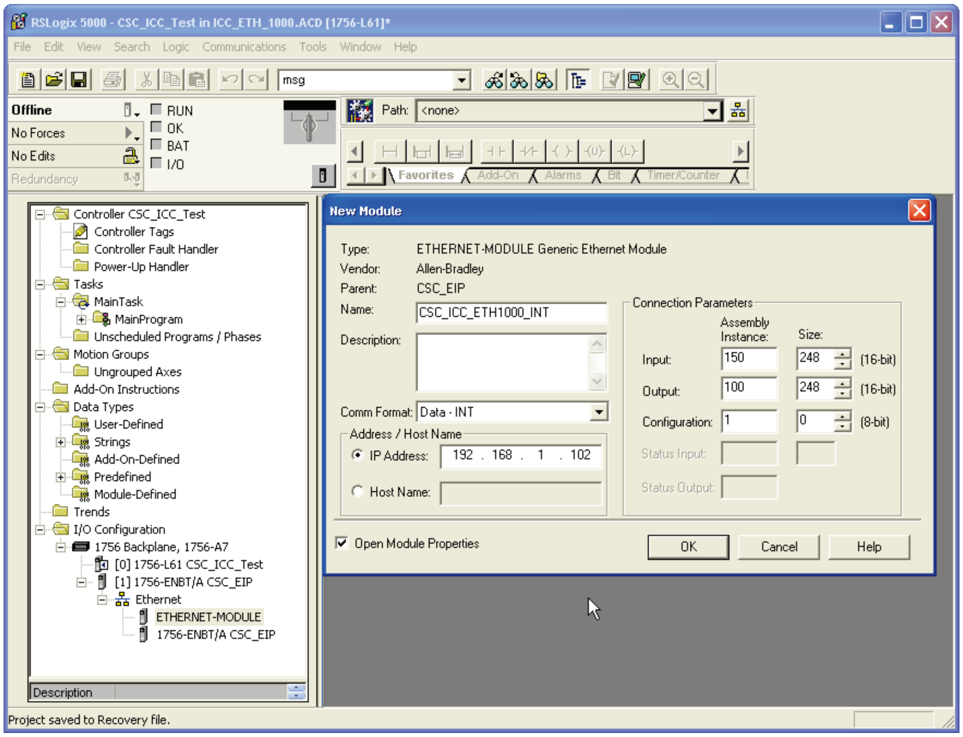


3. Choose “Ethernet-Module / Generic Ethernet Module”.



Module	Description	Vendor
1788-EWEB/A	1788 10/100 Mbps Ethernet Bridge w/Enhanced Web Serv...	Allen-Bradley
1794-AENF/A	1794 10/100 Mbps Ethernet Adapter, Fiber Media	Allen-Bradley
1794-AENT/A	1794 10/100 Mbps Ethernet Adapter, Twisted-Pair Media	Allen-Bradley
1794-AENT/B	1794 10/100 Mbps Ethernet Adapter, Twisted-Pair Media	Allen-Bradley
Drivelogix5730 Eth...	10/100 Mbps Ethernet Port on DriveLogix5730	Allen-Bradley
ETHERNET-BRIDGE	Generic EtherNet/IP CIP Bridge	Allen-Bradley
ETHERNET-MODULE	Generic Ethernet Module	Allen-Bradley
EtherNet/IP	SoftLogix5800 EtherNet/IP	Allen-Bradley
PH-PSSCENA/A	Ethernet Adapter, Twisted-Pair Media	Parker Hannif
Digital		
Drives		
HMI		

4. Double-click on the selection and configure the module accordingly. This is a critical configuration step to ensure the ETH-1000 will work properly in the system as the application requires. Please also consult the ICC ETH-1000 User's Manual regarding configuration of these items.



New Module

Type: ETHERNET-MODULE Generic Ethernet Module
Vendor: Allen-Bradley
Parent: CSC_EIP
Name: CSC_ICC_ETH1000_INT
Description:
Comm Format: Data - INT

Address / Host Name
● IP Address: 192 . 168 . 1 . 102
○ Host Name:

Connection Parameters

	Assembly Instance:	Size:	
Input:	150	248	(16-bit)
Output:	100	248	(16-bit)
Configuration:	1	0	(8-bit)
Status Input:			
Status Output:			

☑ Open Module Properties

a. Configure the "Comm Format" as "Data-INT" for the overall system to work best with the ETH-1000 and the PLC registers. This will allow the transfers to be done in 16 bit integers.

- *Note: The Comm Format data type should be appropriately configured to match the requirements of each particular application.*

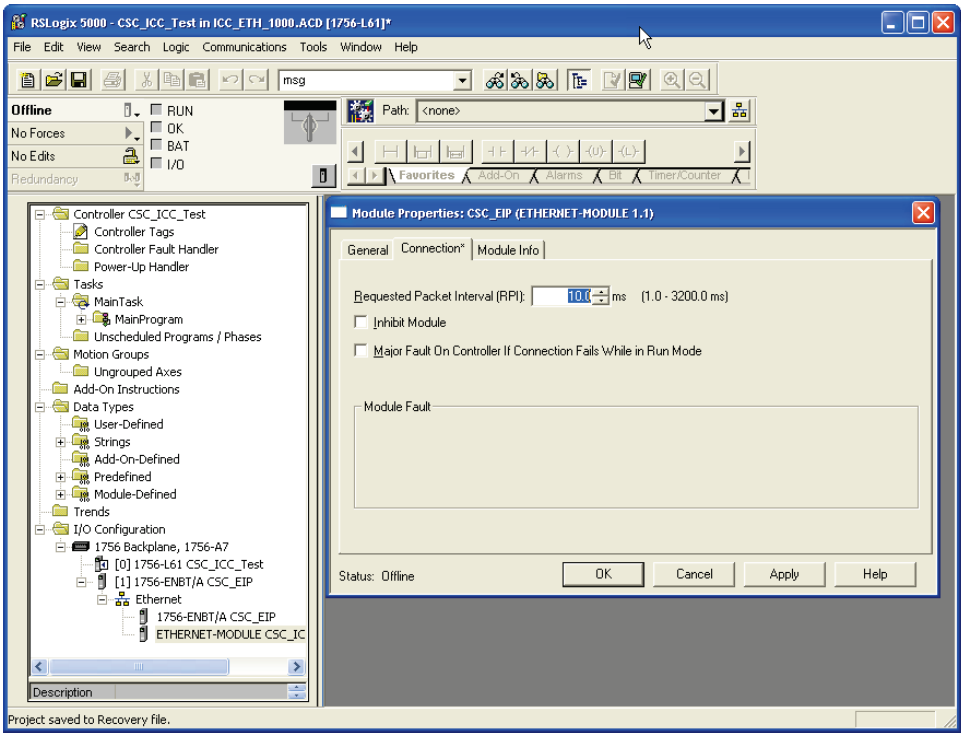
b. Set the IP Address of the generic Ethernet module to the IP address previously assigned to the ETH-1000. For example, the IP address in the Verification System is 192.168.1.102.

c. Configure the "Connection Parameters" as follows:

- The "Input" Assembly Instance should be set to "150." The Input Assembly buffer size should be set to the size appropriate for the application. In the verification system, the input buffer size is set to 248 16-bit words.
- The "Output" Assembly Instance should be set to "100." The Output Assembly buffer size should be set to the size appropriate for the application. In the verification system, the output buffer size is set to 248 16-bit words.
- The "Configuration" Assembly Instance is not used. The Assembly Instance number should therefore be set to "1", and the buffer size set to 0.

d. Check the "Open Module Properties" box and click "OK" to accept the configuration.

5. Configure the RPI to 10.0ms.



RSLogix 5000 - CSC_ICC_Test in ICC_ETH_1000.ACD [1756-L61]*

Module Properties: CSC_EIP (ETHERNET-MODULE 1.1)

General | Connection* | Module Info

Requested Packet Interval (RPI): 10.0 ms (1.0 - 3200.0 ms)

□ Inhibit Module

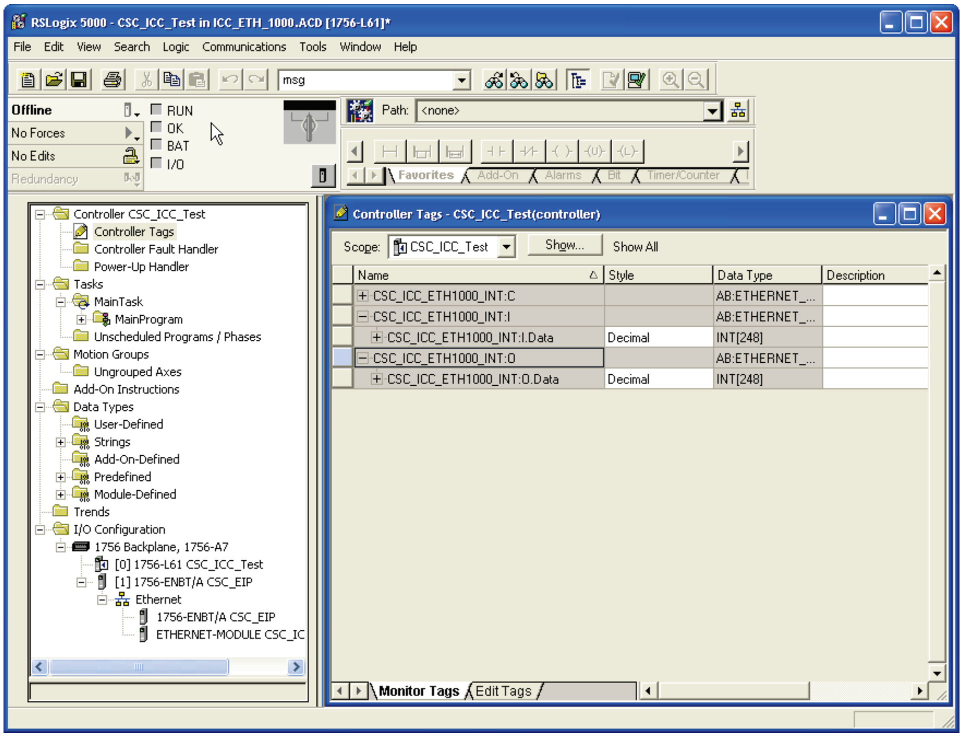
□ Major Fault On Controller If Connection Fails While in Run Mode

Module Fault

Status: Offline

6. Select “OK” to accept and complete the ETH-1000 module configuration.

7. Double-click the “Controller Tags” selection to view the automatically-created module tags:



RSLogix 5000 - CSC_ICC_Test in ICC_ETH_1000.ACD [1756-L61]*

Controller Tags - CSC_ICC_Test(controller)

Scope: CSC_ICC_Test Show... Show All

Name	Style	Data Type	Description
+ CSC_ICC_ETH1000_INT:C		AB:ETHERNET_...	
− CSC_ICC_ETH1000_INT:I		AB:ETHERNET_...	
+ CSC_ICC_ETH1000_INT:I.Data	Decimal	INT[248]	
− CSC_ICC_ETH1000_INT:O		AB:ETHERNET_...	
+ CSC_ICC_ETH1000_INT:O.Data	Decimal	INT[248]	

a. 248 integer tags were created for CSC_ICC_ETH1000_INT:I. These are the tag locations where the ETH-1000 will send data to the ControlLogix PLC via implicit messaging at the RPI rate.

b. 248 integer tags were created for CSC_ICC_ETH1000_INT:O. These are the tag locations where data will be sent to the ETH-1000 via implicit messaging at the RPI rate.

c. The ETH-1000 database locations where data will be written to and read from will be configured using the steps described in the following Chapter.

Chapter 5 **ETH-1000 Configuration**

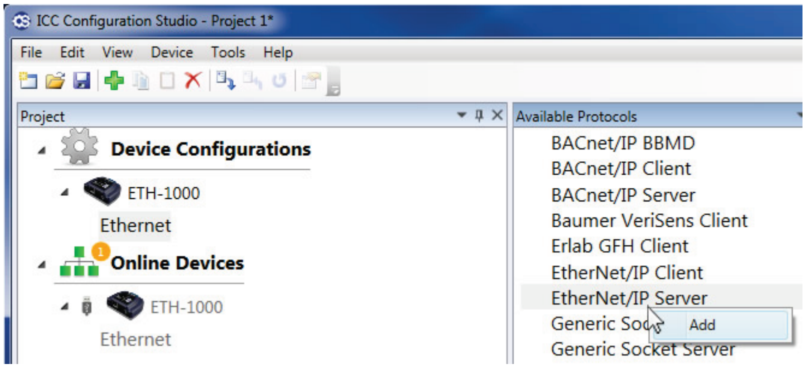
This chapter documents the steps to configure the ETH-1000 to allow communication between a ControlLogix PLC (using EtherNet/IP class 1 I/O messaging) and a Mitsubishi PLC (using MC protocol). This configuration is performed with the ICC Configuration Studio software.

Configuring the Validation System as shown in Figure 2 is used as the example. Some parameters will need different values to properly reflect the actual system a user is configuring. However, the example configuration can be used to simplify the overall configuration effort.

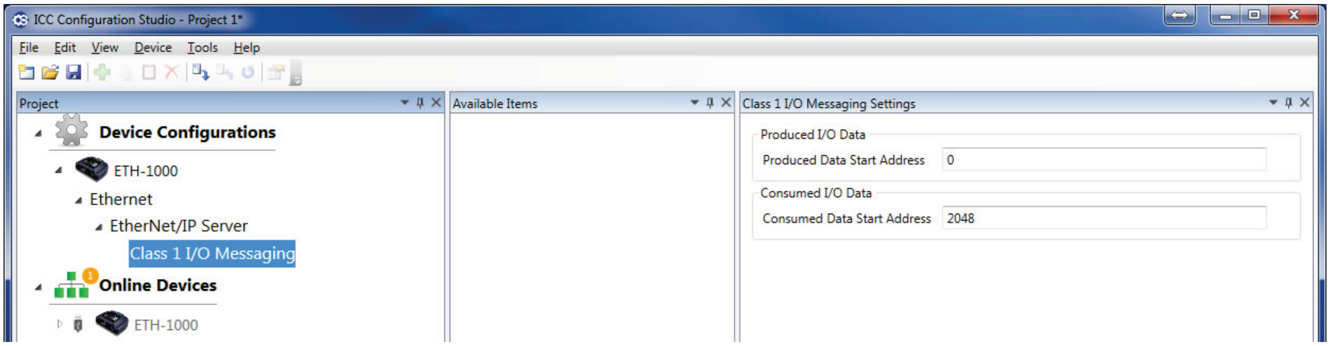
Prior to performing the following procedure, the configuration steps as shown in Section 3.2 of this document should first be completed.

5.1 Configuring EtherNet/IP Implicit Messaging

1. In the "Project" panel "Device Configurations" heading, expand the ETH-1000 configuration and select the "Ethernet" node. In the "Available Protocols" panel, right-click on "EtherNet/IP Server" and select "Add" to activate the EtherNet/IP server driver. "EtherNet/IP Server" will then appear beneath the "Ethernet" node in the project tree.



2. Expand the "EtherNet/IP Server" driver and select "Class 1 I/O Messaging" to define the ETH-1000 database locations. These database locations define the start addresses of the buffers where data will be exchanged between the PLC and the ETH-1000 through the class 1 I/O messaging connection.



Produced I/O Data
Produced Data Start Address: 0

Consumed I/O Data
Consumed Data Start Address: 2048

In this example, the "Produced Data Start Address" is set to 0 and the "Consumed Data Start Address" is set to 2048. These addresses define the ETH-1000 database locations that will be used to transfer data between the ETH-1000 and the ControlLogix tags. The "Produced" and "Consumed" terms are defined from the perspective of the ETH-1000. Figure 3 is helpful in explaining how the data items are assigned and transferred.

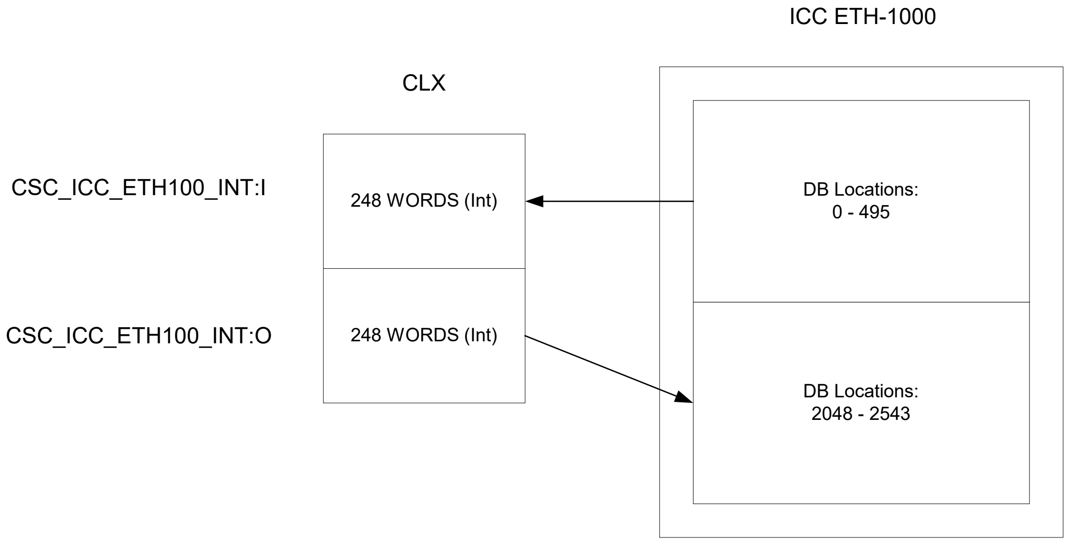
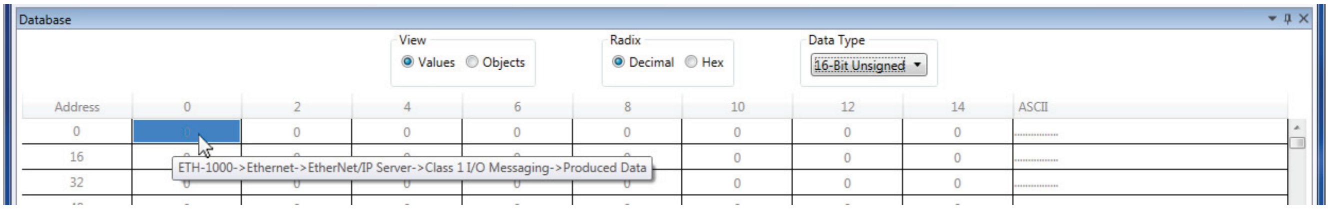


Figure 3: Mapping ControlLogix Tags to ETH-1000 Database Locations

The ETH-1000's internal database is byte-addressable. Therefore, 248 tag words consume 496 bytes in the database.

3. In the "Database" panel, locate the starting addresses that are configured for use by both the Produced (Figure 4) and Consumed (Figure 5) data.

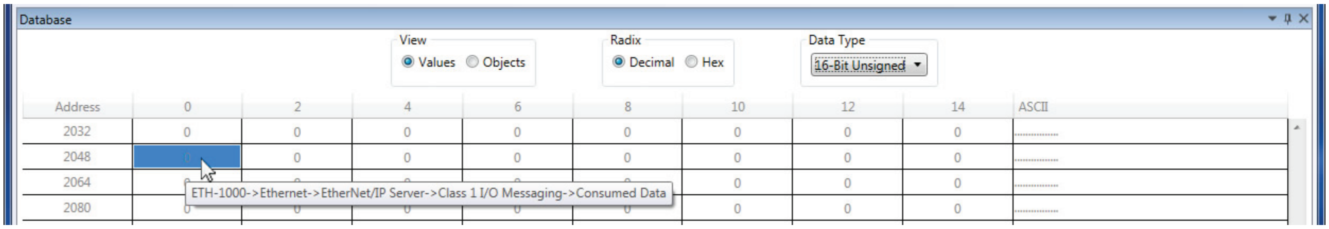


View: ● Values ○ Objects
Radix: ● Decimal ○ Hex
Data Type: 16-Bit Unsigned

Address	0	2	4	6	8	10	12	14	ASCII
0	0	0	0	0	0	0	0	0	....
16						0	0	0	....
32	0	0	0	0	0	0	0	0	....

ETH-1000->Ethernet->EtherNet/IP Server->Class 1 I/O Messaging->Produced Data

Figure 4: Produced Data Database Buffer



View: ● Values ○ Objects
Radix: ● Decimal ○ Hex
Data Type: 16-Bit Unsigned

Address	0	2	4	6	8	10	12	14	ASCII
2032	0	0	0	0	0	0	0	0	....
2048	0	0	0	0	0	0	0	0	....
2064						0	0	0	....
2080	0	0	0	0	0	0	0	0	....

ETH-1000->Ethernet->EtherNet/IP Server->Class 1 I/O Messaging->Consumed Data

Figure 5: Consumed Data Database Buffer

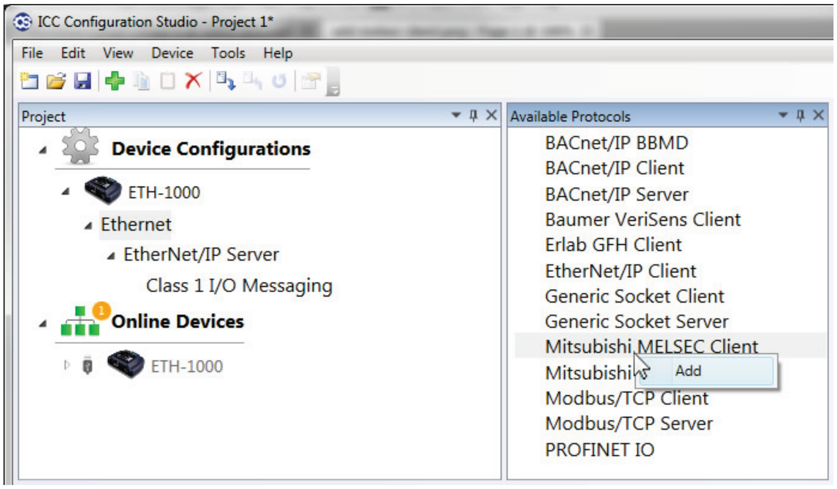
5.2 Configuring the MELSEC Protocol Driver

This section documents the steps to properly configure the MELSEC protocol driver. Note that this configuration is relevant only for the example Verification System architecture as shown in Figure 2: *the specific Connection Objects and Service Objects should be appropriately configured to match the requirements of each particular application*. Users should consult the ETH-1000 user's manual and understand the information and timing requirements necessary for transferring data to and from Mitsubishi PLCs.

The critical steps are the configuration of a MELSEC Connection Object and the services that need to be accomplished using this Connection Object. A Connection Object can be configured to represent a physical connection between an ETH-1000 and a Mitsubishi PLC. Depending on the application requirements, however, a single physical connection can support multiple “logical” connections via multiple Connection Objects.

Once a Connection Object has been established, multiple Service Objects can then be configured for this Connection Object. Service Objects define the tasks that need to be accomplished across the connection. For example, a Service Object could be configured to read 20 words of a PLC’s Data Registers starting from D12287, or to write 10 words to the PLC Internal Relay area.

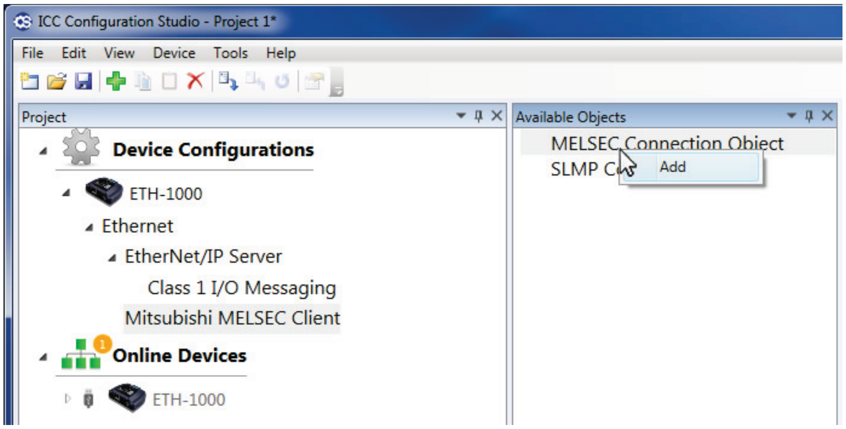
1. In the “Project” panel “Device Configurations” heading, expand the ETH-1000 configuration and select the “Ethernet” node. In the “Available Protocols” panel, right-click on “Mitsubishi MELSEC Client” and select “Add” to activate the MELSEC client driver. “Mitsubishi MELSEC Client” will then appear beneath the “Ethernet” node in the project tree.

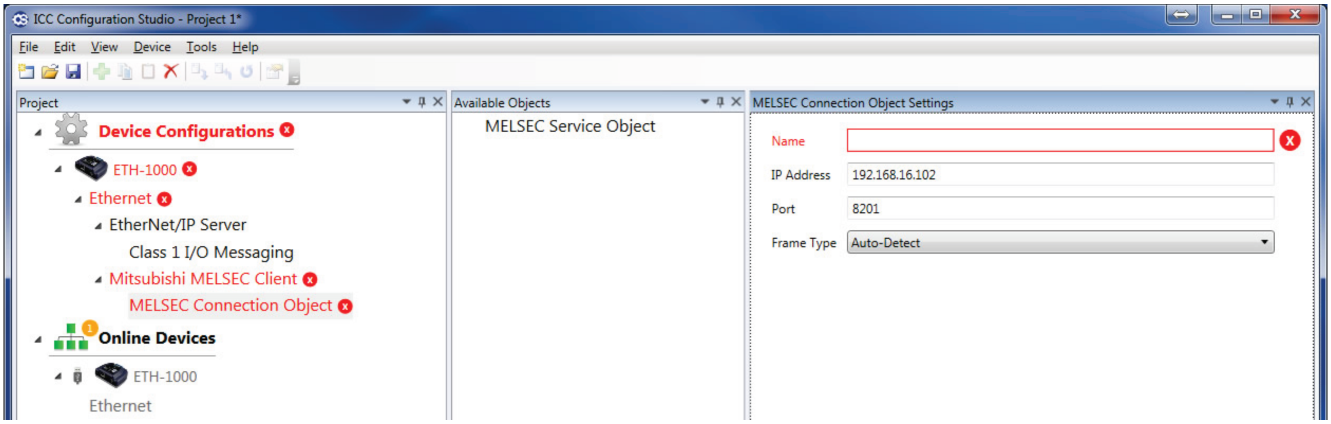


5.3 Configuring Connection Objects

The following steps will create a Connection Object to communicate with an FX PLC residing at IP address 192.168.1.50 using UDP port 0x5001 (20481 decimal).

1. In the “Project” panel “Device Configurations” heading, select the “Mitsubishi MELSEC Client” driver added in section 5.2.
2. In the “Available Objects” panel, right-click on “MELSEC Connection Object” and select “Add”. “MELSEC Connection Object” will then appear beneath “Mitsubishi MELSEC Client” in the project tree.





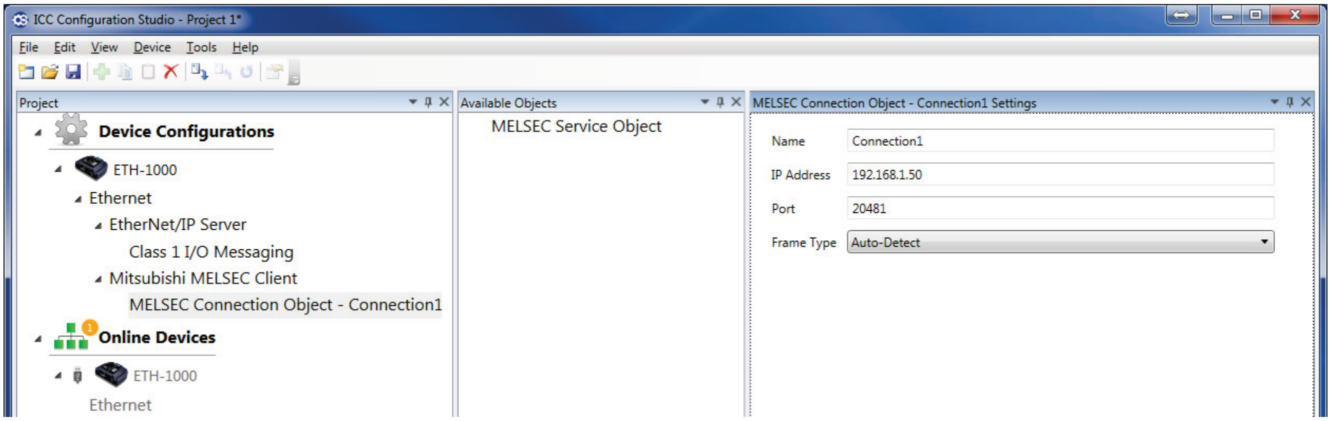
Name

IP Address 192.168.16.102

Port 8201

Frame Type Auto-Detect

3. Provide a name for this Connection Object. The name “Connection1” has been used for this example.
4. Enter the IP Address of the targeted PLC. The IP Address is 192.168.1.50 for this example.
5. Enter the Port to use on the targeted PLC. The Port is 0x5001 (20481 decimal) for this example.
6. Set the Frame Type to “Auto-Detect” or “3E Frame”.



Name Connection1

IP Address 192.168.1.50

Port 20481

Frame Type Auto-Detect

5.4 Configuring Service Objects

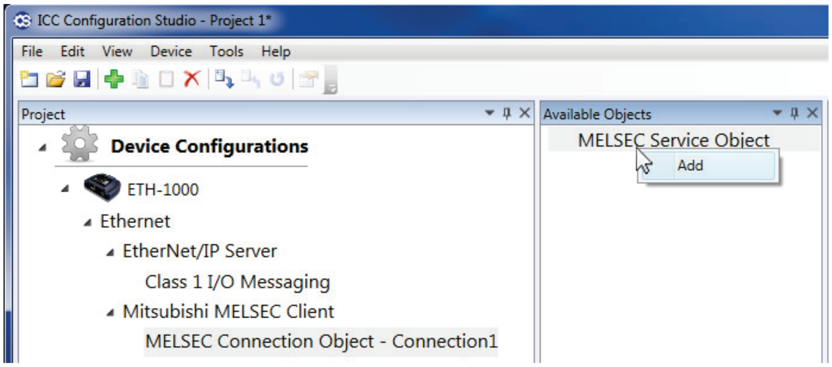
For the Verification System architecture example, four Service Objects will be created for the Connection Object:

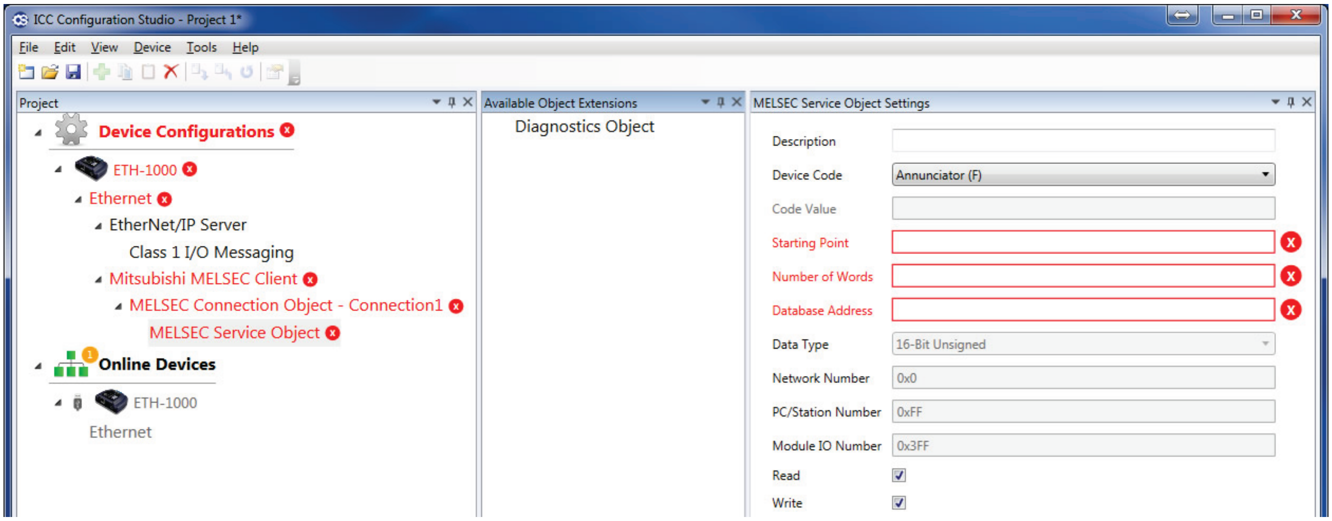
- Write 20 words from ETH-1000 to Data Register 1000…1019
- Read 20 words from Data Register 2000…2019 to ETH-1000
- Write 10 words (160 bits) from ETH-1000 to Internal Relay M0…M159
- Read 10 words (160 bits) from Internal Relay M400…M559 to ETH-1000

Perform the following steps to create and configure the Service Object that will write 20 words from the ETH-1000 to Data Registers 1000…1019:

1. In the “Project” panel, select the Connection Object titled “MELSEC Connection Object – Connection1”, that was created in section 5.3.

2. In the “Available Objects” panel, right-click on “MELSEC Service Object” and select “Add”. “MELSEC Service Object” will then appear beneath “MELSEC Connection Object – Connection1” in the project tree.





Description	
Device Code	Annunciator (F)
Code Value	
Starting Point	
Number of Words	
Database Address	
Data Type	16-Bit Unsigned
Network Number	0x0
PC/Station Number	0xFF
Module IO Number	0x3FF
Read	☑
Write	☑

3. Enter an optional “Description” string to document this Service Object’s purpose.

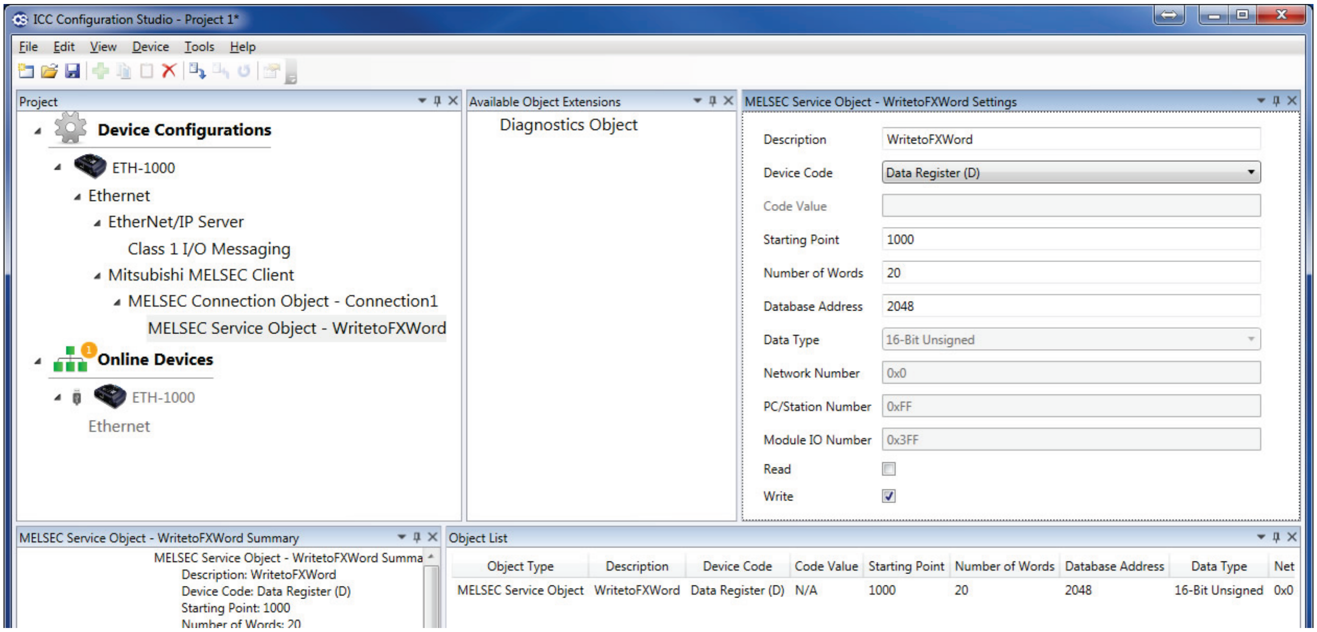
4. Select the targeted PLC “Device Code” to which data will be written. For this example, select “Data Register (D).”

5. Enter the “Starting Point” of the Data Register area where the data will be written to. In this example, the Starting Point is 1000.

6. Enter the “Number of Words” to be transferred. In this example, the Number of Words is 20.

7. Enter the “Database Address” which designates the starting location where stored data will be written to the Data Registers. In this example, the starting Database Address is 2048.

8. Enable the appropriate “Read” or “Write” checkbox(s), depending on the purpose of the specific Service Object. It is important to note that “Read” and “Write” terms are defined from the perspective of the ETH-1000. In other words, selecting the “Write” checkbox enables the Service Object to transfer data stored in the database to the designated Data Register locations on the PLC. For this example, only the “Write” checkbox is selected.


ICC Configuration Studio - Project 1*
File Edit View Device Tools Help
Project
Available Object Extensions
Diagnostics Object
MELSEC Service Object - WritetoFXWord Settings
MELSEC Service Object - WritetoFXWord Summary
Object List

- Device Configurations
  - ETH-1000
    - Ethernet
      - EtherNet/IP Server
        - Class 1 I/O Messaging
      - Mitsubishi MELSEC Client
        - MELSEC Connection Object - Connection1
          - MELSEC Service Object - WritetoFXWord
- Online Devices
  - ETH-1000
    - Ethernet

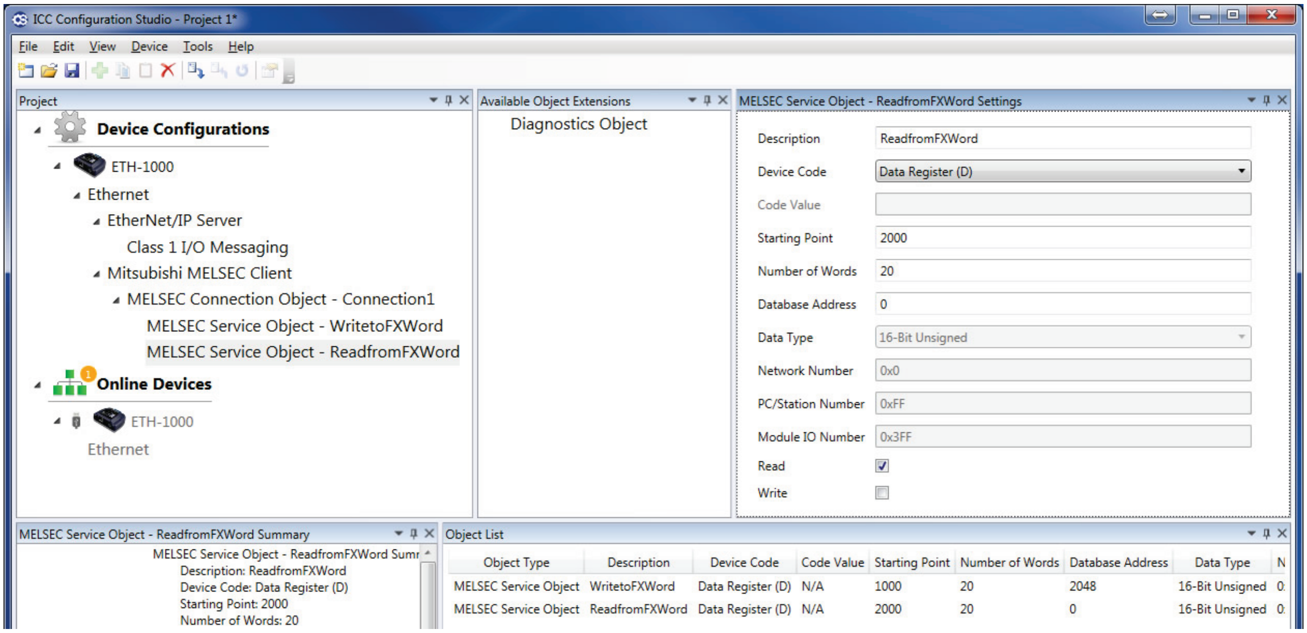
Description	WritetoFXWord
Device Code	Data Register (D)
Code Value	
Starting Point	1000
Number of Words	20
Database Address	2048
Data Type	16-Bit Unsigned
Network Number	0x0
PC/Station Number	0xFF
Module IO Number	0x3FF
Read	□
Write	☑

MELSEC Service Object - WritetoFXWord Summa
Description: WritetoFXWord
Device Code: Data Register (D)
Starting Point: 1000
Number of Words: 20

Object Type	Description	Device Code	Code Value	Starting Point	Number of Words	Database Address	Data Type	Net
MELSEC Service Object	WritetoFXWord	Data Register (D)	N/A	1000	20	2048	16-Bit Unsigned	0x0

9. Repeat steps 1-8 above to create and configure the other three Service Objects for this Connection Object.

   a. Define the Service Object that will read Data Registers 2000…2019 to the ETH-1000:


ICC Configuration Studio - Project 1*
File Edit View Device Tools Help
Project
Available Object Extensions
Diagnostics Object
MELSEC Service Object - ReadfromFXWord Settings
MELSEC Service Object - ReadfromFXWord Summary
Object List

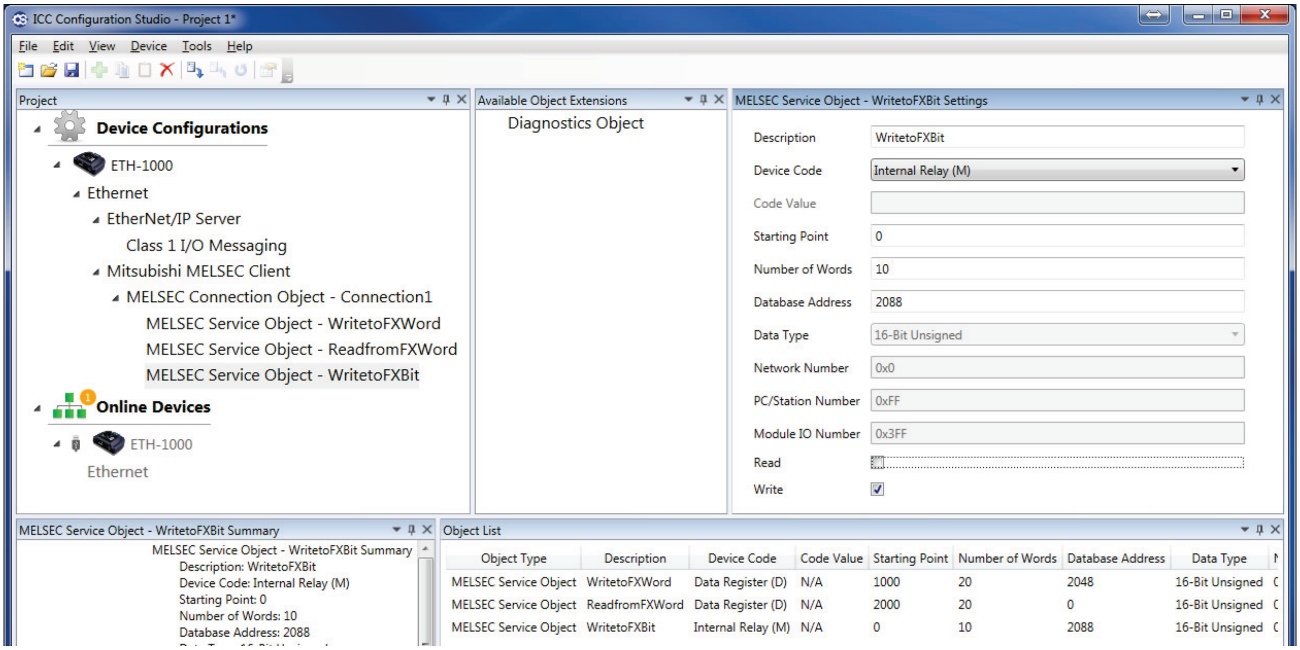
- Device Configurations
  - ETH-1000
    - Ethernet
      - EtherNet/IP Server
        - Class 1 I/O Messaging
      - Mitsubishi MELSEC Client
        - MELSEC Connection Object - Connection1
          - MELSEC Service Object - WritetoFXWord
          - MELSEC Service Object - ReadfromFXWord
- Online Devices
  - ETH-1000
    - Ethernet

Description	ReadfromFXWord
Device Code	Data Register (D)
Code Value	
Starting Point	2000
Number of Words	20
Database Address	0
Data Type	16-Bit Unsigned
Network Number	0x0
PC/Station Number	0xFF
Module IO Number	0x3FF
Read	☑
Write	□

MELSEC Service Object - ReadfromFXWord Sumr
Description: ReadfromFXWord
Device Code: Data Register (D)
Starting Point: 2000
Number of Words: 20

Object Type	Description	Device Code	Code Value	Starting Point	Number of Words	Database Address	Data Type	N
MELSEC Service Object	WritetoFXWord	Data Register (D)	N/A	1000	20	2048	16-Bit Unsigned	0:
MELSEC Service Object	ReadfromFXWord	Data Register (D)	N/A	2000	20	0	16-Bit Unsigned	0:

b. Define the Service Object that will write 10 words (160 bits) from the ETH-1000 to Internal Relay M0…M159. It is critical to note that the FX PLC internal relays can only be written as words on "word boundaries" because all bits are packaged and transferred as words. For example, the "Starting Point" of a service object that writes to Internal Relays cannot be 200, because 200 is not divisible by 16. However, a "Starting Point" of 208 (i.e. M208) would be valid.


ICC Configuration Studio - Project 1*
File Edit View Device Tools Help
Project
Device Configurations
ETH-1000
Ethernet
EtherNet/IP Server
Class 1 I/O Messaging
Mitsubishi MELSEC Client
MELSEC Connection Object - Connection1
MELSEC Service Object - WritetoFXWord
MELSEC Service Object - ReadfromFXWord
MELSEC Service Object - WritetoFXBit
Online Devices
ETH-1000
Ethernet
Available Object Extensions
Diagnostics Object

MELSEC Service Object - WritetoFXBit Settings

Description	WritetoFXBit
Device Code	Internal Relay (M)
Code Value	
Starting Point	0
Number of Words	10
Database Address	2088
Data Type	16-Bit Unsigned
Network Number	0x0
PC/Station Number	0xFF
Module IO Number	0x3FF
Read	☐
Write	☑

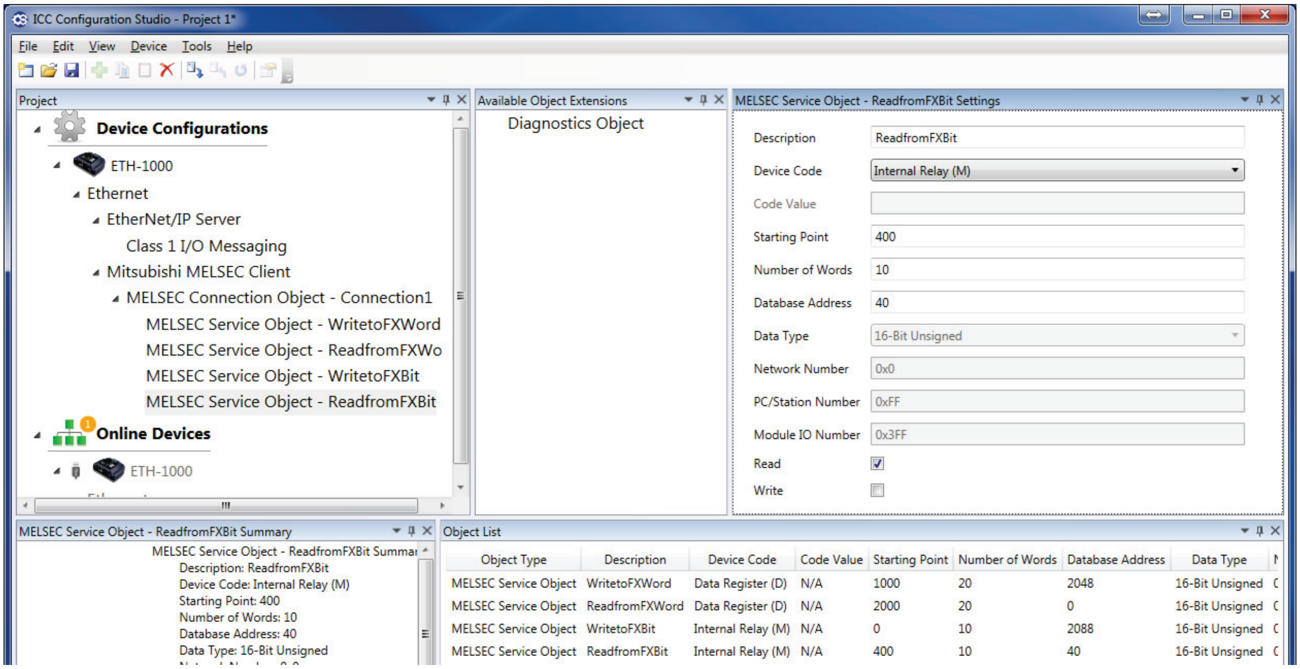
MELSEC Service Object - WritetoFXBit Summary

MELSEC Service Object - WritetoFXBit Summary
Description: WritetoFXBit
Device Code: Internal Relay (M)
Starting Point: 0
Number of Words: 10
Database Address: 2088

Object List

Object Type	Description	Device Code	Code Value	Starting Point	Number of Words	Database Address	Data Type
MELSEC Service Object	WritetoFXWord	Data Register (D)	N/A	1000	20	2048	16-Bit Unsigned
MELSEC Service Object	ReadfromFXWord	Data Register (D)	N/A	2000	20	0	16-Bit Unsigned
MELSEC Service Object	WritetoFXBit	Internal Relay (M)	N/A	0	10	2088	16-Bit Unsigned

c. Define the Service Object that will read 10 words (160 bits) from Internal Relay M400…M559 to the ETH-1000:


ICC Configuration Studio - Project 1*
File Edit View Device Tools Help
Project
Device Configurations
ETH-1000
Ethernet
EtherNet/IP Server
Class 1 I/O Messaging
Mitsubishi MELSEC Client
MELSEC Connection Object - Connection1
MELSEC Service Object - WritetoFXWord
MELSEC Service Object - ReadfromFXWo
MELSEC Service Object - WritetoFXBit
MELSEC Service Object - ReadfromFXBit
Online Devices
ETH-1000
Available Object Extensions
Diagnostics Object

MELSEC Service Object - ReadfromFXBit Settings

Description	ReadfromFXBit
Device Code	Internal Relay (M)
Code Value	
Starting Point	400
Number of Words	10
Database Address	40
Data Type	16-Bit Unsigned
Network Number	0x0
PC/Station Number	0xFF
Module IO Number	0x3FF
Read	☑
Write	☐

MELSEC Service Object - ReadfromFXBit Summary

MELSEC Service Object - ReadfromFXBit Summar
Description: ReadfromFXBit
Device Code: Internal Relay (M)
Starting Point: 400
Number of Words: 10
Database Address: 40
Data Type: 16-Bit Unsigned

Object List

Object Type	Description	Device Code	Code Value	Starting Point	Number of Words	Database Address	Data Type
MELSEC Service Object	WritetoFXWord	Data Register (D)	N/A	1000	20	2048	16-Bit Unsigned
MELSEC Service Object	ReadfromFXWord	Data Register (D)	N/A	2000	20	0	16-Bit Unsigned
MELSEC Service Object	WritetoFXBit	Internal Relay (M)	N/A	0	10	2088	16-Bit Unsigned
MELSEC Service Object	ReadfromFXBit	Internal Relay (M)	N/A	400	10	40	16-Bit Unsigned

5.5 Calculating the ETH-1000 Database Addresses

One of the most important steps in configuring a Service Object is determining the required database address that must be entered. Because the ETH-1000 implements a “shared database” accessible equally by all protocol drivers, one must be careful in defining where data items are written to and read from.

Figure 3 in section 5.1 shows the mapping of data from the ControlLogix to the ETH-1000’s database. This concept can now be expanded upon to reveal the complete end-to-end mapping of data from the ControlLogix to the Mitsubishi PLC as defined in the system. Figure 6 illustrates the complete mapping for the Validation System shown in Figure 2.

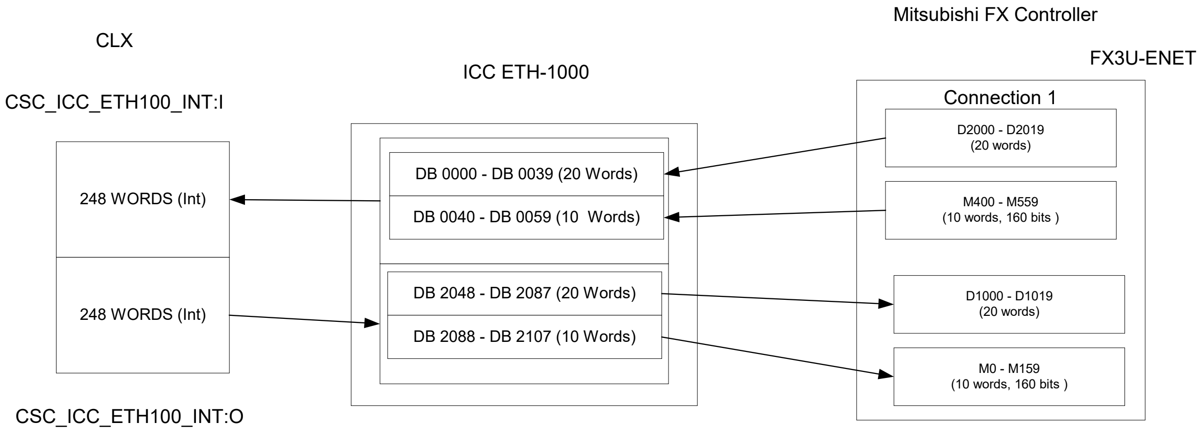


Figure 6: Complete Mapping of ControlLogix Tags to Mitsubishi PLC Elements

Note that the database addresses defined in the MELSEC Service Objects are the starting addresses of each block of data shown above. The ETH-1000 database addresses are “Byte” addresses, so that starting address locations need to be adjusted according to the relevant data types.

A Microsoft Excel-based database address calculation tool is available from Mitsubishi Electric Automation, Inc. upon request. The tool can be used to calculate the ETH-1000 database locations of MELSEC Service Objects to assist in configuration efforts. Please contact your MEAU representatives to obtain a copy of this tool. Additionally, the Database panel of the ICC Configuration Studio can be used to view the database usage for each mapped protocol object, thereby providing a convenient visual reference for current database usage.

Chapter 6 Using EtherNet/IP Explicit Messaging

The ControlLogix PLC can communicate with the Mitsubishi PLC asynchronously using class 3 EtherNet/IP explicit messaging through the ETH-1000 gateway. This communication is accomplished through the use of MSG instructions in RSLogix5000. Refer to the Mitsubishi MELSEC Bypass -related sections in the "Millennium Gateway Series Protocol Driver Manuals: EtherNet/IP Server Driver Manual" and the "Millennium Gateway Series Protocol Driver Manuals: Mitsubishi MELSEC Client Driver Manual" for details pertaining to the configuration and operation of these MSG instructions.

Chapter 7 **Terminology**

Implicit Messaging	Also known as I/O messaging or class 1 messaging. Connections are established to move application-specific I/O data at regular intervals. These connections may be configured in a one-to-many relationship in order to take full advantage of the producer-consumer multicast model. Implicit messaging utilizes the UDP/IP transport mechanism.
Explicit Messaging	Also known as class 3 messaging. Explicit messaging is a point-to-point relationship that is established to facilitate request-response transactions between two nodes. These connections are general purpose in nature and can be used to reach any network-accessible items within a device. Explicit messaging utilizes the TCP/IP transport mechanism.
EtherNet/IP	EtherNet/IP is the name given to the Common Industrial Protocol (CIP), as implemented over standard Ethernet (IEEE 802.3 and the TCP/IP protocol suite).
Common Industrial Protocol (CIP)	The Common Industrial Protocol (CIP) is a media independent, connection-based, object-oriented protocol designed for automation applications. It encompasses a comprehensive set of communication services for automation applications: control, safety, synchronization, motion, configuration and information.
Connection Object	The MELSEC protocol driver uses connection objects to target each server (PLC). A connection object defines a connection to a unique endpoint (IP address and port number) and specifies the frame type used for all underlying service objects. A connection object can be thought of as a communication channel or “pipe” which is created between the driver and the server device, independent of the service objects that make use of that communication channel to transfer data requests.
Service Object	The MELSEC protocol driver uses service objects to define what data transfer functions (read and/or write) are to be performed. Service objects define attributes such as the targeted PLC internal element type, starting element, number of elements, and associated storage location in the ETH-1000’s internal database.

Revisions

July 2009 – Document created and Released, Version 1.0

January 2016 – Updated to reflect ICC Configuration Studio, Version 1.1