

**Using ICC ETH-1000
EtherNet/IP Interface with
Mitsubishi Motion Controllers**

Quick Start Guide

START



Contents

Contents.....	i
FURTHER READING REFERENCE LIST	ii
Chapter 1 Introduction.....	1-1
Chapter 2 System Overview.....	2-1
Chapter 3 Connecting Devices to the Network.....	3-1
3.1 Changing the IP Address of the ControlLogix System	3-1
3.2 Changing the IP Address of the ETH-1000.....	3-3
3.3 Changing the IP Addresses of Motion Controllers	3-6
Chapter 4 ControlLogix PLC Project Configuration	4-1
4.1 Adding the 1756 ENBT Module	4-1
4.2 Adding the ETH-1000 Module.....	4-4
Chapter 5 ETH-1000 Configuration.....	5-1
5.1 Configuring EtherNet/IP Implicit Messaging.....	5-1
5.2 Configuring the MELSEC Protocol Driver	5-2
5.3 Configuring Connection Objects	5-3
5.4 Configuring Service Objects	5-5
5.5 Calculating the ETH-1000 Database Addresses.....	5-9
Chapter 6 Using EtherNet/IP Explicit Messaging	6-1
Chapter 7 Terminology.....	7-1
Revisions	A

FURTHER READING REFERENCE LIST

Mitsubishi

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

Q170MCPU Quick Start Guide

Motion Controller MR-MQ100 User's Manual (Details) IB(NA)-0300150-A

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 motion controllers. An Example of the system configuration is shown in Figure 1 below.

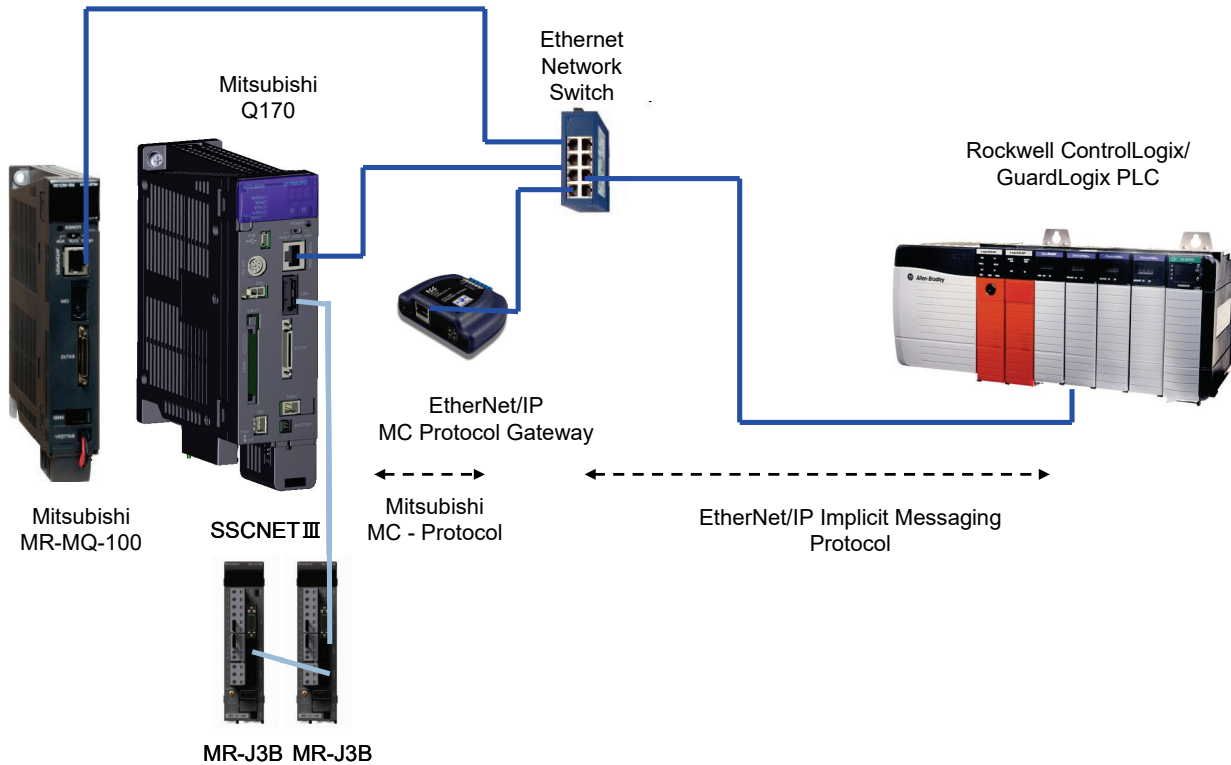


Figure 1: EtherNet/IP Connectivity – Rockwell PLC to Mitsubishi Motion Controllers

The system configurations enable Rockwell PLCs to read and write both bit and register data of Mitsubishi motion controllers using either EtherNet/IP Implicit or Explicit Messaging.

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 stand-alone motion controllers such as Q170's or MR-MQ100's.

The initial applications are limited to those whose motion profiles are being controlled directly by the Mitsubishi motion controllers. Rockwell PLCs are used in a supervisory role to (1) command (i.e. start or stop) motion programs already stored in the motion controllers, (2) configure motion controller parameters, and (3) monitor motion controller status information.

It is assumed that the user of this guide is familiar with the Rockwell RSLogix5000 environment, the operation of Mitsubishi MR-MQ100 and Q170 motion controllers, and has sufficient knowledge of the ICC ETH-1000 Gateway. It is critical for the users to refer to the motion controller manuals when setting up the system parameters for Ethernet applications.

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.

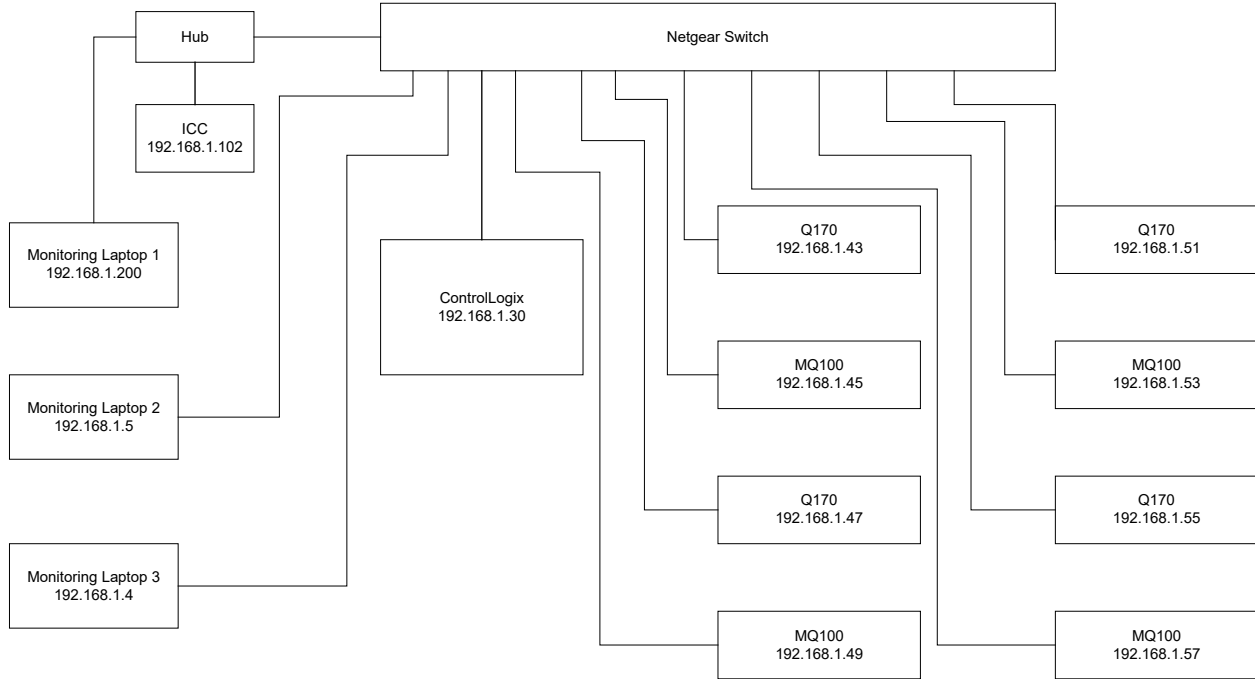


Figure 2: Architecture of an Example Verification System

The Verification System consists of three monitoring PCs (e.g. Monitoring Laptop 1, 2 and 3), a ControlLogix system, an ICC ETH-1000 module, and four Mitsubishi MR-MQ-100 1.5Axis Motion Controllers and four Mitsubishi Q170 Stand-Alone Motion Controllers.

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(s), Rockwell PLC, ICC ETH-1000, and Mitsubishi Motion Controller(s) 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 MT Developer software packages are installed on the programming PC(s).
2. Create a project using the RSLogix5000 software to control the Mitsubishi motion controllers.
 - 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 motion controller register and/or bit locations for each motion controller

Chapter 3 Connecting Devices to the Network

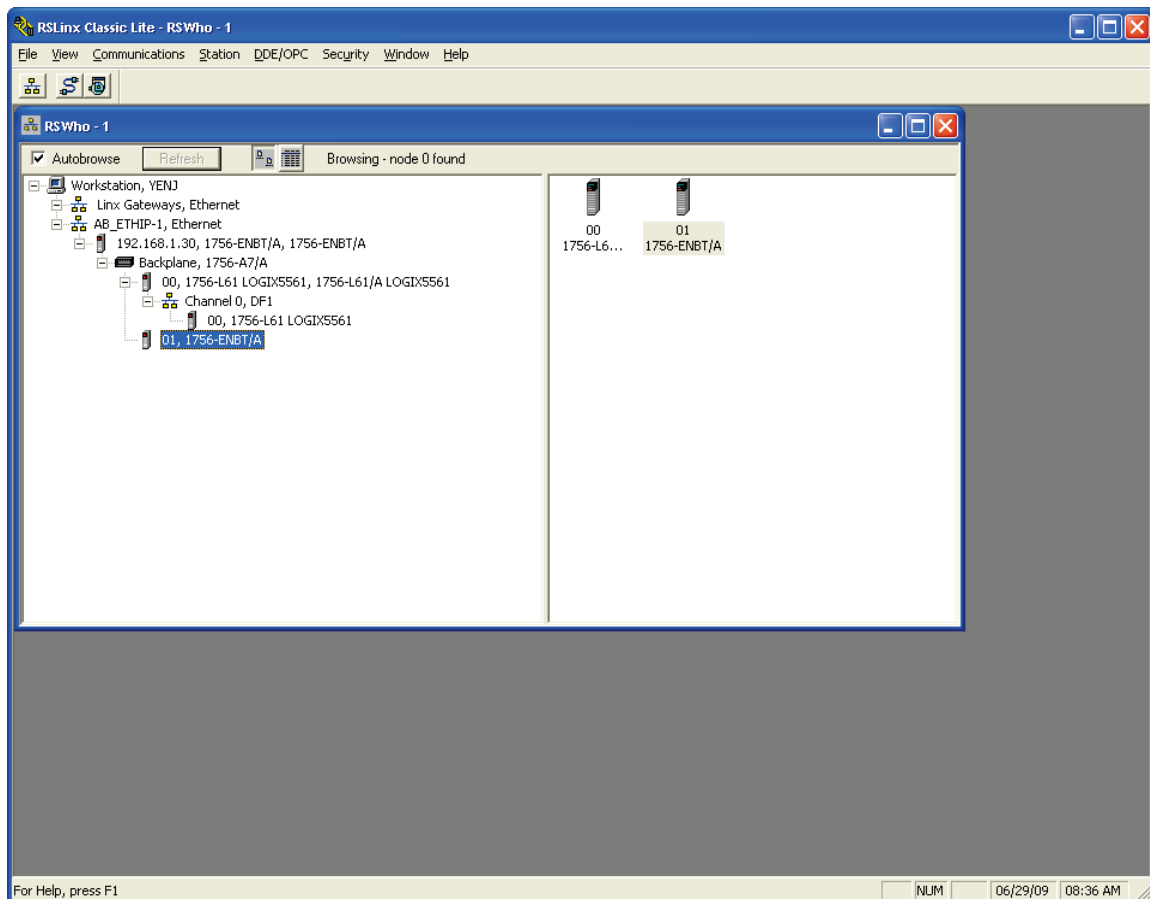
The steps of configuring the IP addresses of the ControlLogix PLC system, the ICC ETH-1000 module, and the motion controller(s) are documented in this chapter.

3.1 Changing the IP Address of the ControlLogix System

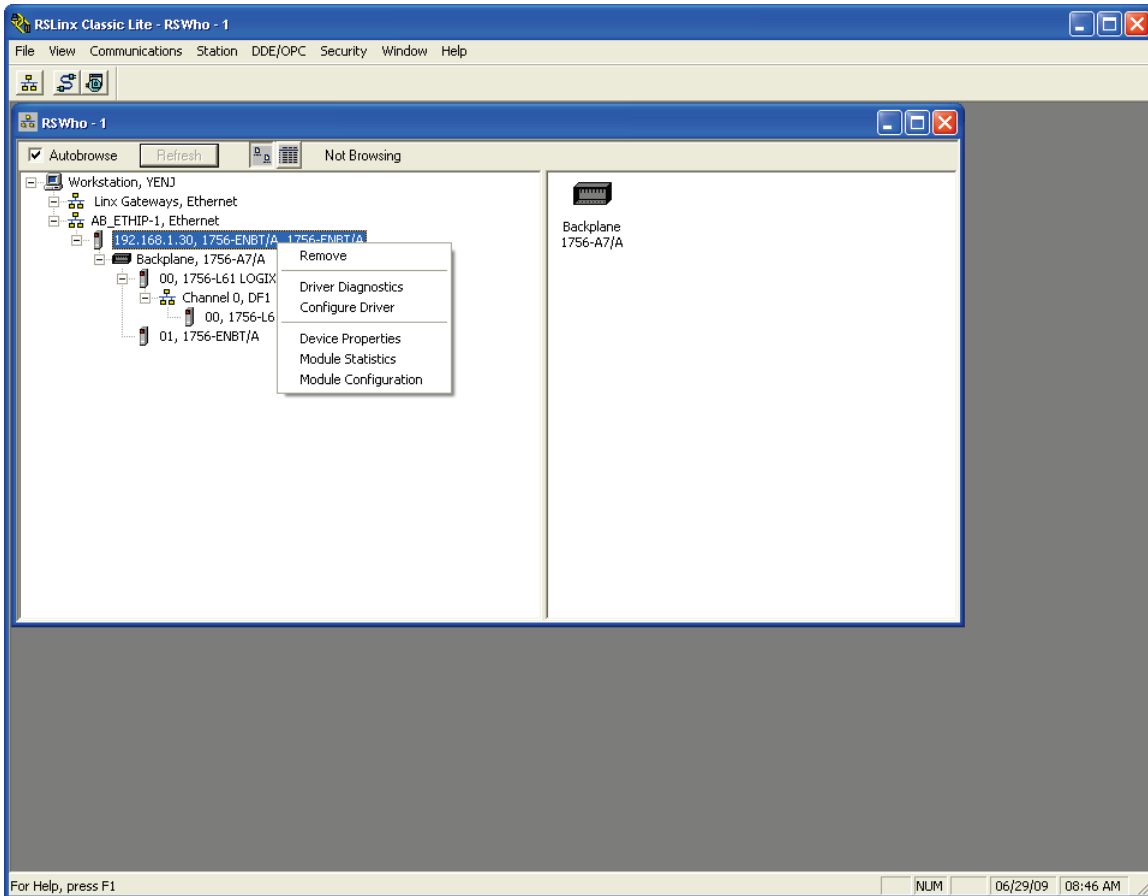
The minimum configuration of a ControlLogix system consists of a ControlLogix chassis (e.g. 1756-A7), a power supply (e.g. 1756-PA72), a ControlLogix Controller (e.g. 1756-L61), and an EtherNet/IP module (e.g. 1756-ENBT).

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 motion controllers 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.

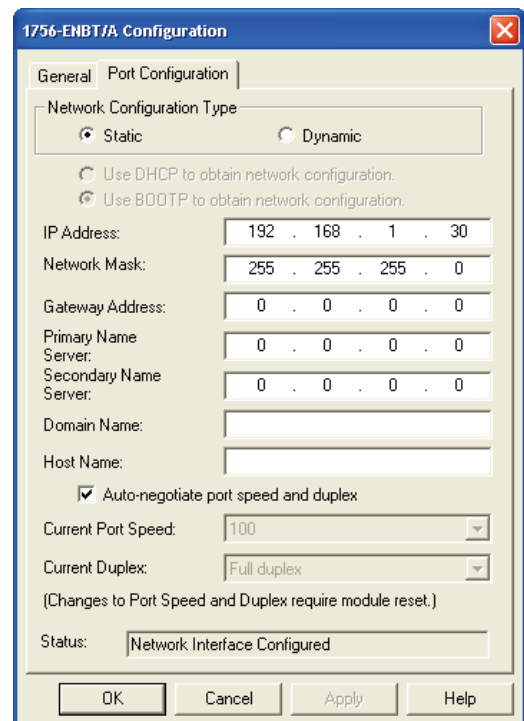
1. Power on the Rockwell system and monitor the display scrolling across the front of the ENBT module. If the module is working properly, the message “OK Rev x.x.x IP1.IP2.IP3.IP4” should scroll across the display. Rev x.x.x is the revision number of the module firmware, and IP1.IP2.IP3.IP4 forms the 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:



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



- 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.
- 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.

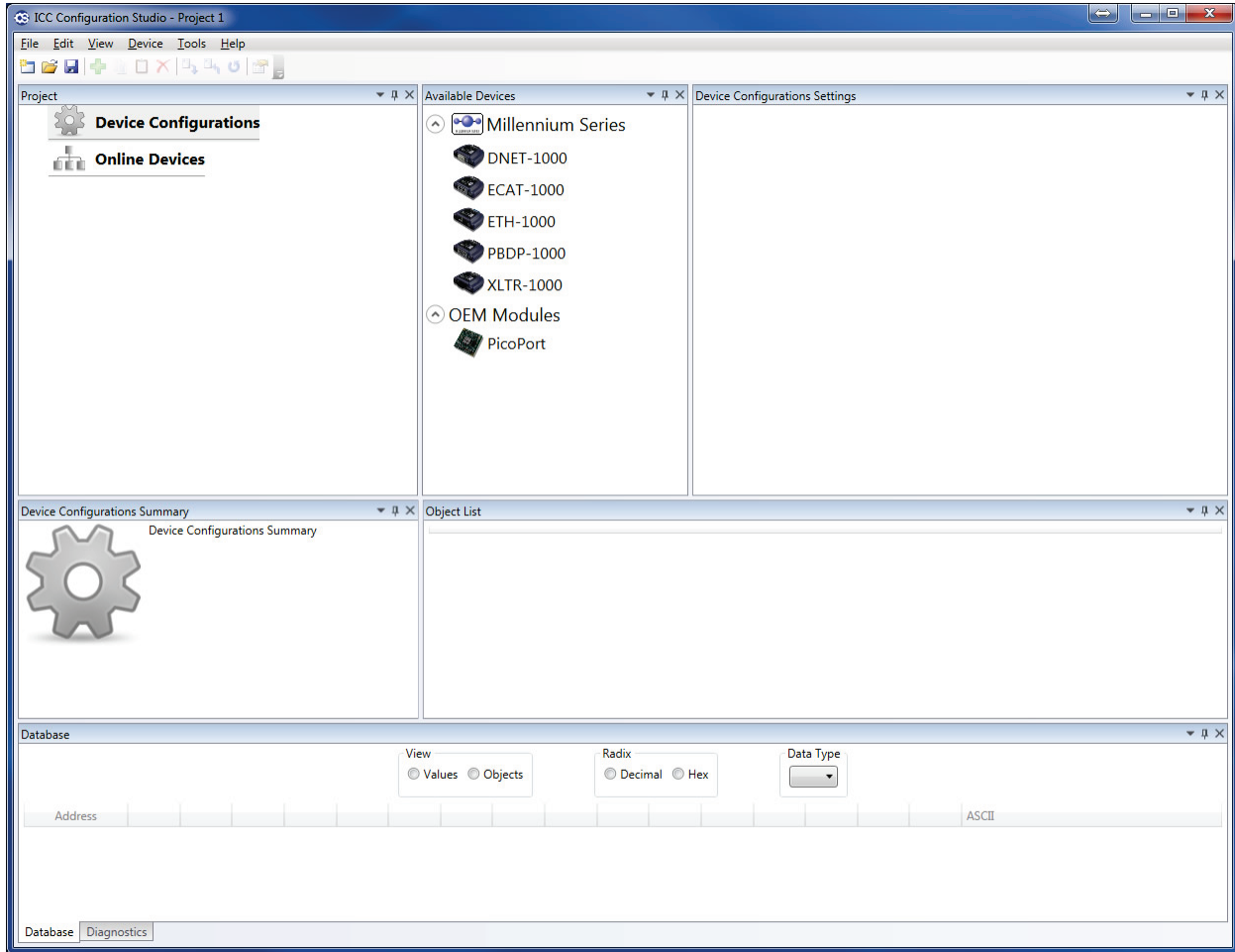


3.2 Changing the IP Address of the ETH-1000

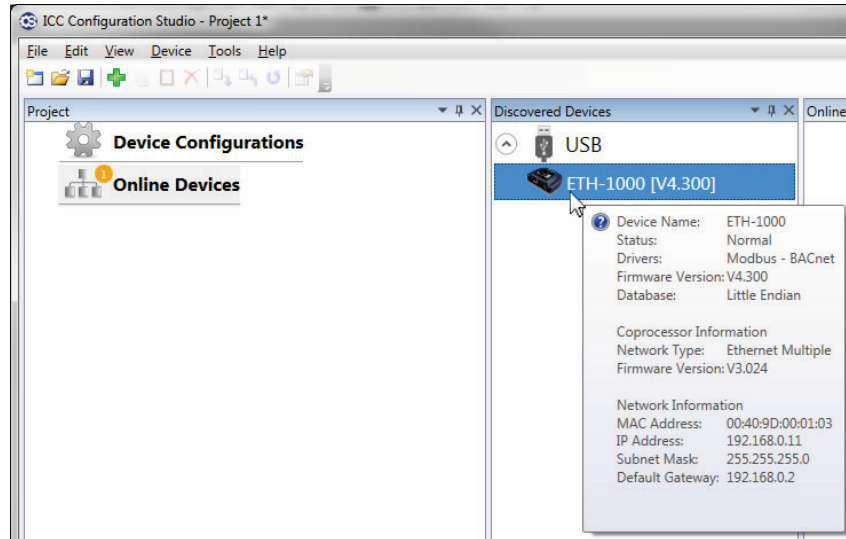
The ICC Configuration Studio software should be installed onto the PC that will be used to configure the ETH-1000. The ETH-1000 can be powered using an USB connection, a Power over Ethernet (PoE) connection or an external 7- 24V DC power supply.

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

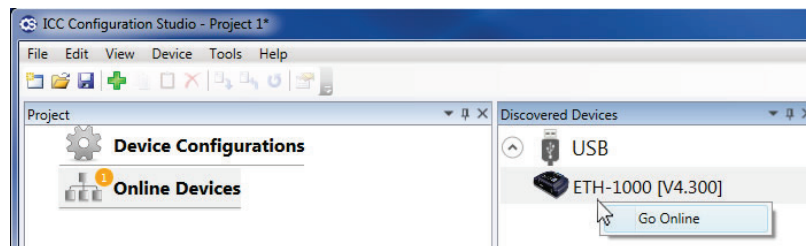
1. Launch the ICC Configuration Studio on the PC and connect a USB cable to the ETH-1000.



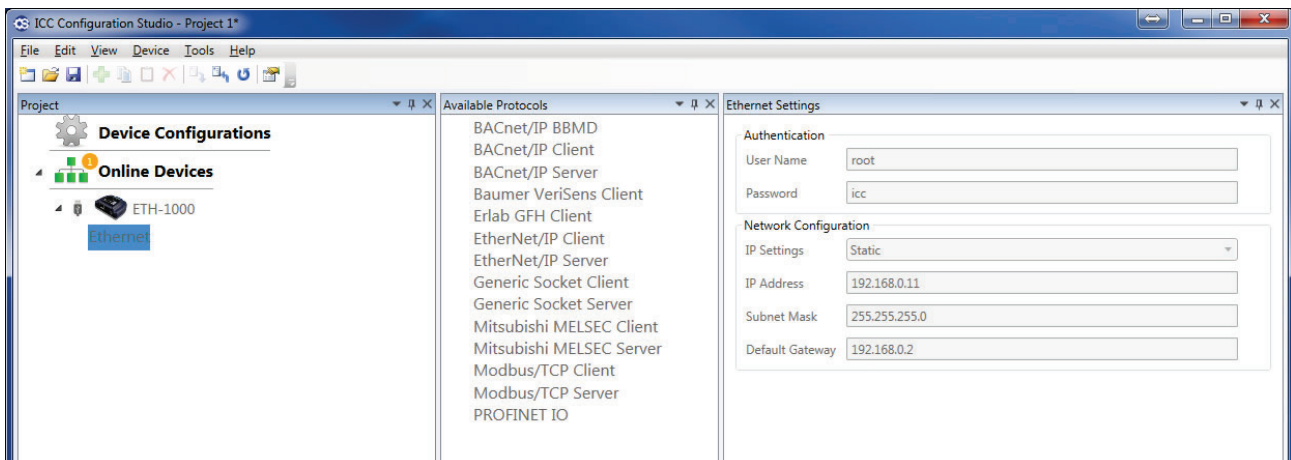
- Click on the “Online Devices” heading to display the “Discovered Devices” panel. Move the mouse cursor over the ETH-1000 to display device information such as the firmware versions, database endianness, and network info.



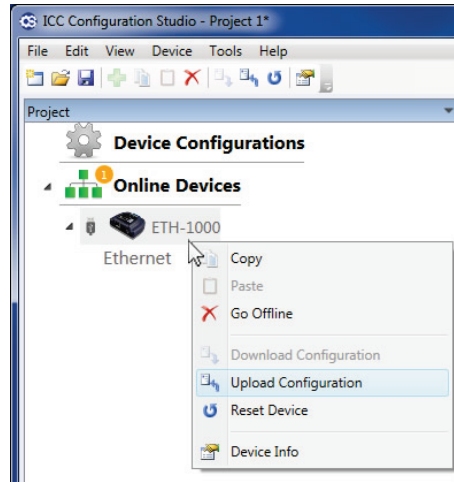
- In the “Discovered Devices” panel, right-click on the ETH-1000 and select “Go Online”.



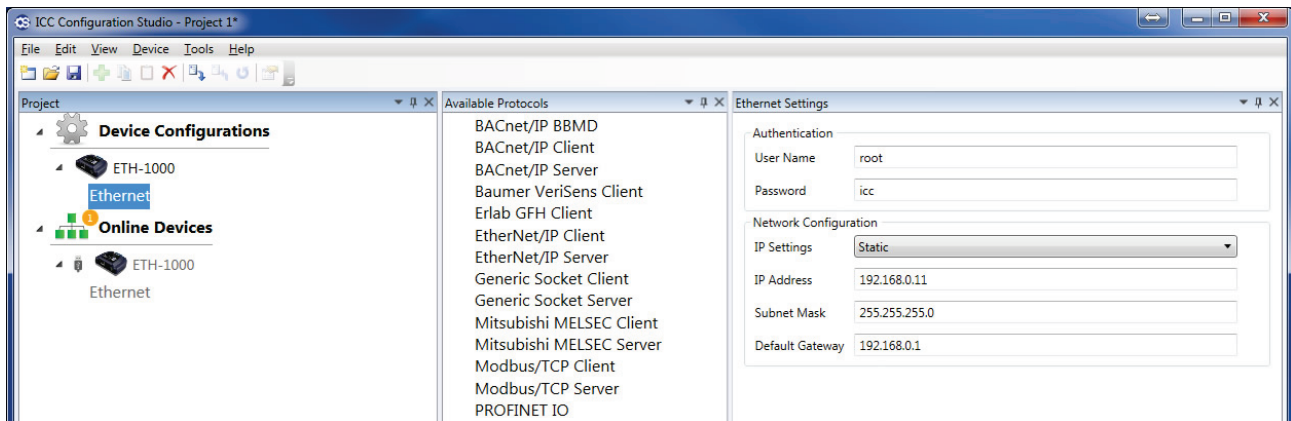
- The ETH-1000 module will be connected, and the screen will be populated with the current configuration:



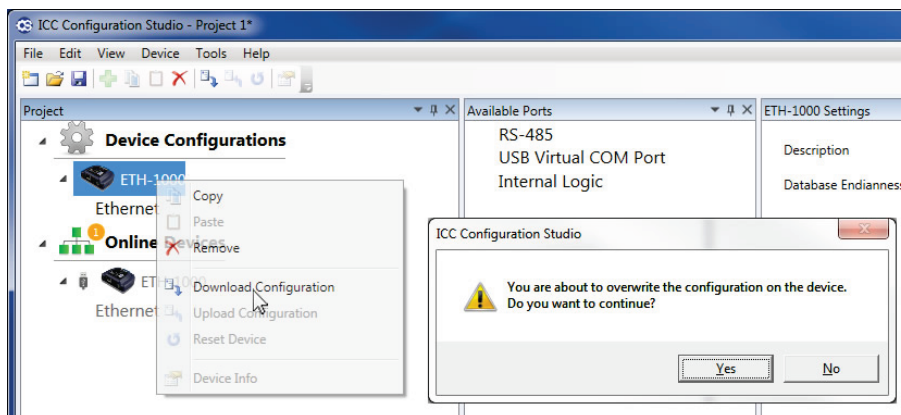
- Note that it is NOT possible to edit the “Online Devices” configuration. The configuration must be uploaded into the “Device Configurations” heading before the configuration can be modified. Right-click on the online ETH-1000 and select “Upload Configuration”.



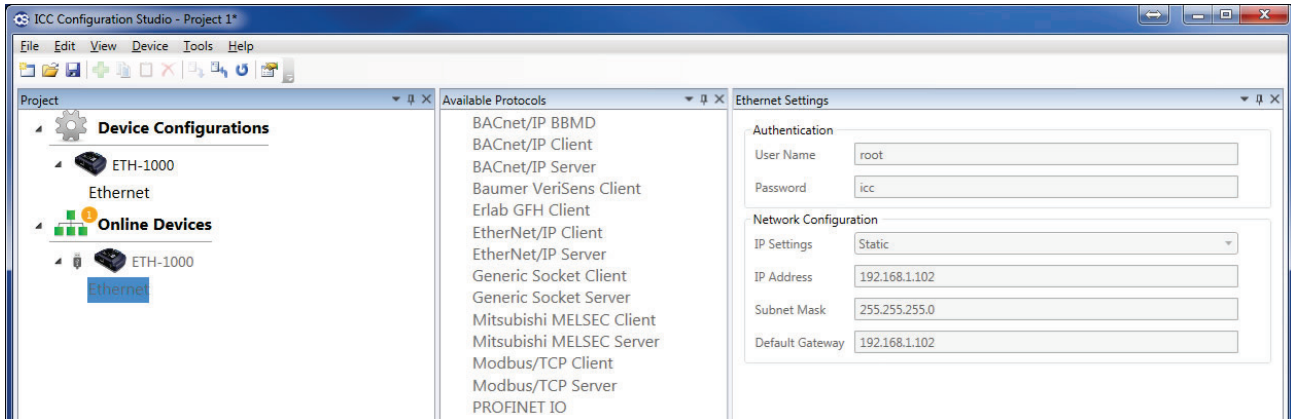
- The user can now modify the IP Address and Subnet Mask.



- 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.



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.



3.3 Changing the IP Addresses of Motion Controllers

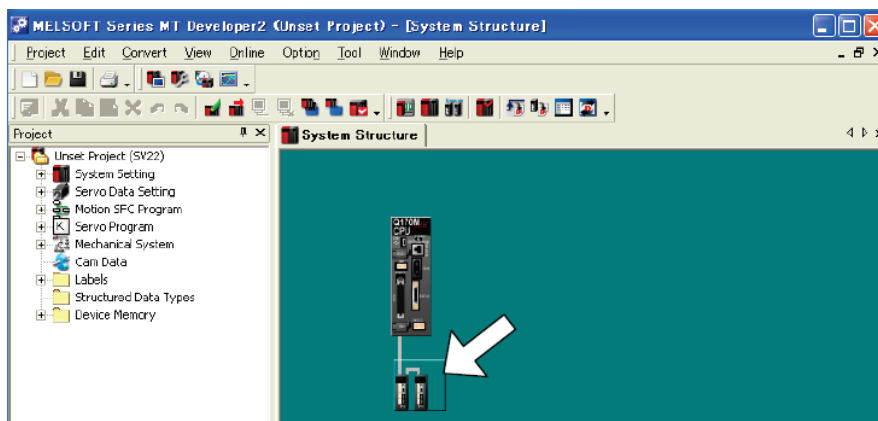
If a brand new motion controller is used, the motion controller needs to have the firmware installed first. The firmware version needs to support MELSEC Communication Protocol (MC-Protocol). The firmware update procedure is the same for both MR-MQ100 and Q170.

The following are brief summary steps of the firmware update procedure. Please consult the Q170M Quick Start Guide for complete firmware update procedure.

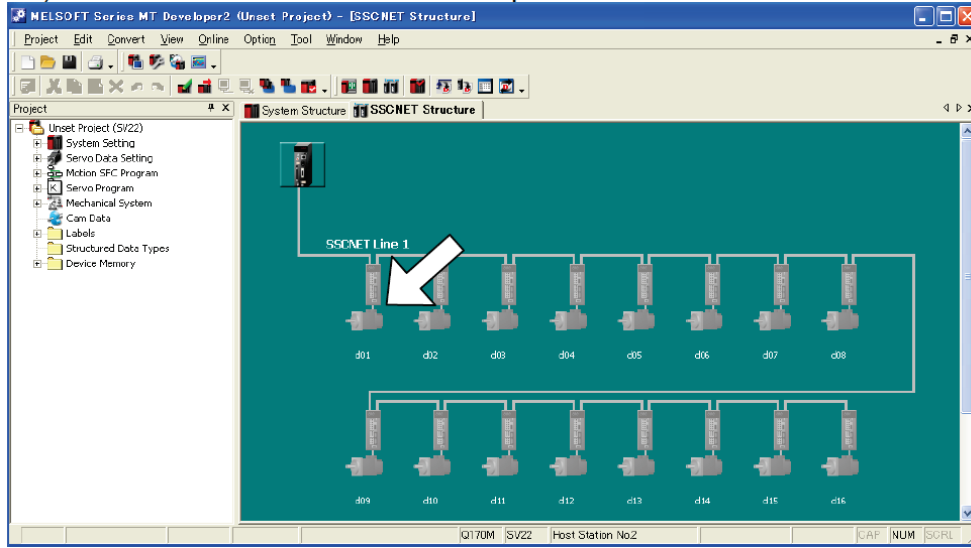
1. Turn the rotary switches on a motion controller to INSTALL mode with the following setting: SW1: A, SW2: 0, and then reset the power of the motion controller.
2. Use the Configuration PC to install the motion controller operating system firmware version SV22. Open the MT Developer 2 Version 1.05F on the Configuration PC and use the 'Install Software' function and follow the on screen steps to install the firmware on the Motion Controller.
3. Turn rotary switches on the Motion Controller back to the NORMAL mode with the following settings: SW1: 0, SW2: 0; then reset the power of the motion controller.

The following are steps to change the IP address of the Motion Controller:

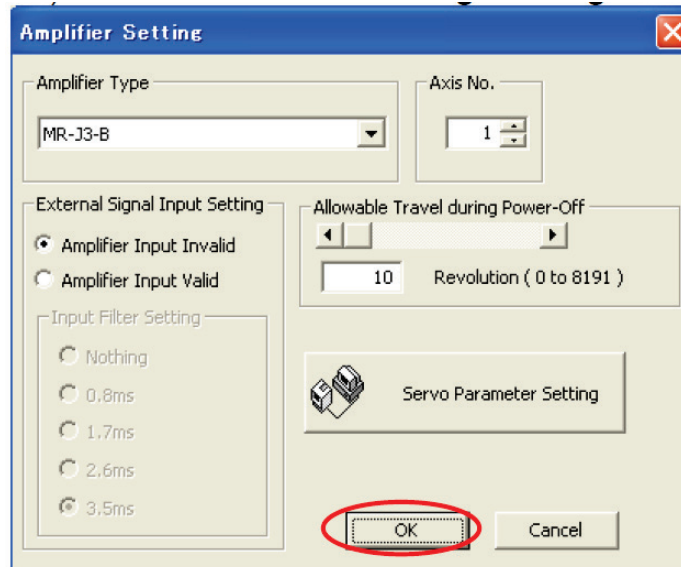
1. Open the MT Developer 2 on the Configuration PC and create a new project for the appropriate controller.
2. Activate the MR-J3 amp/motor set from the 'SSCNET Structure' in MT Developer 2:
 - a. Double-Click the Servo Amplifier icon.



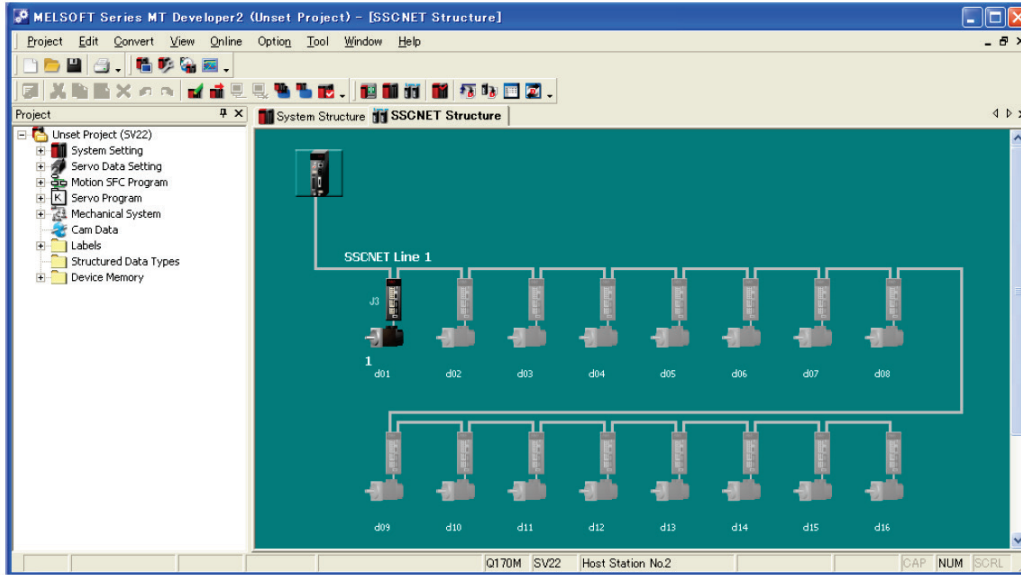
b. Double-Click the first d01 amplifier/servo icon.



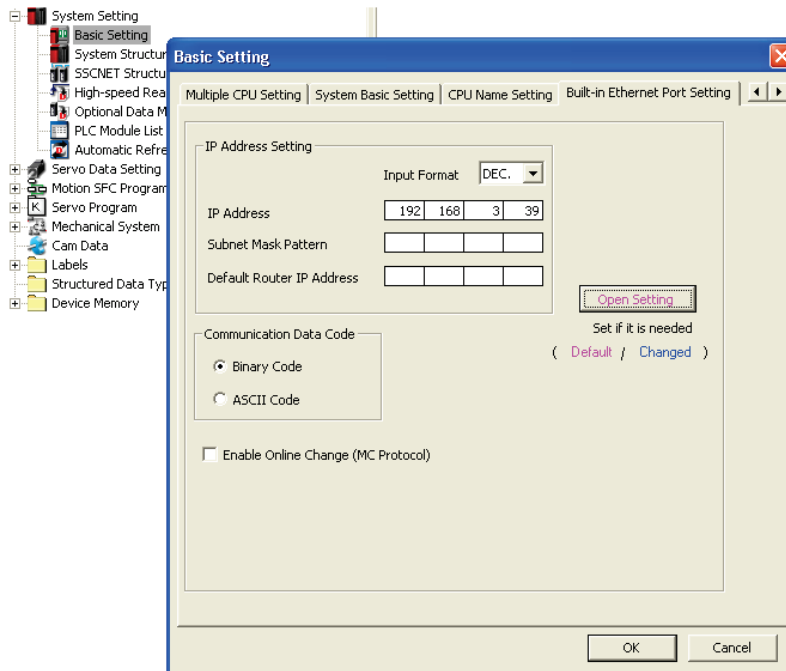
c. Accept the setting in the “Amplifier Setting” window without making changes. Click “OK” to accept.



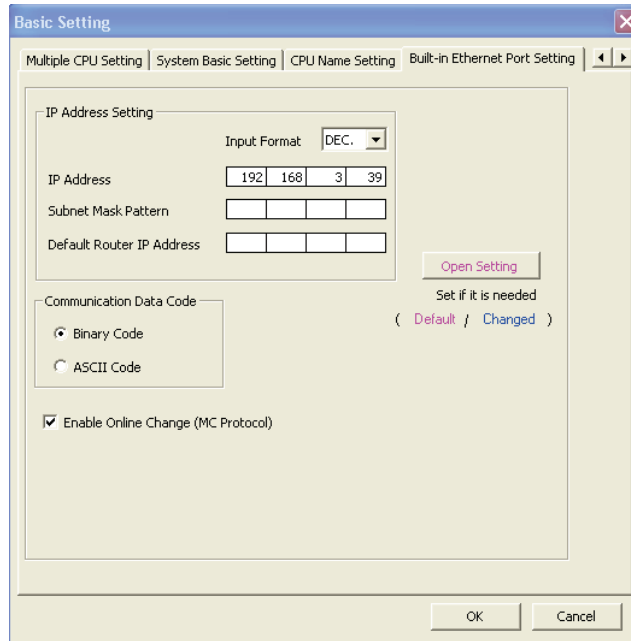
d. Observe the first d01 amplifier/servo icon is enabled and defined as Axis 1.



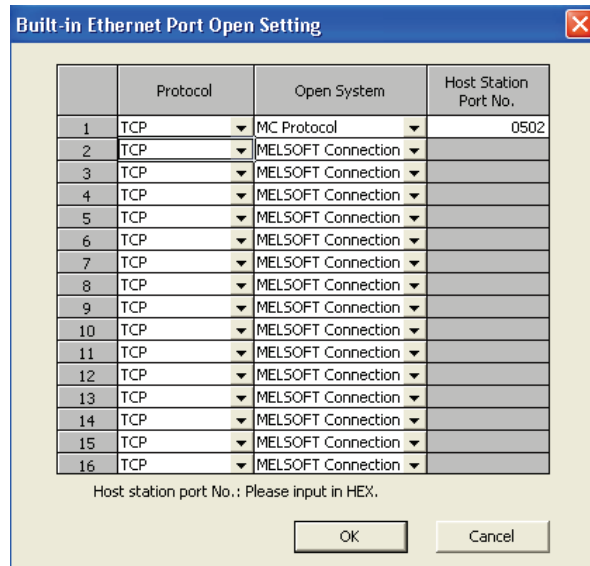
3. Using the 'Built-in Ethernet Port Setting' tab within the 'Basic Settings', configure the IP address.



- a. Change the IP address to the desired address and then select the Enable Online Change (MC Protocol) Check box:



- b. Click the “Open Setting” button and enter the first channel as MC Protocol with a proper Host Station Port number (In this example, the Port Number is set to Hex 0502):



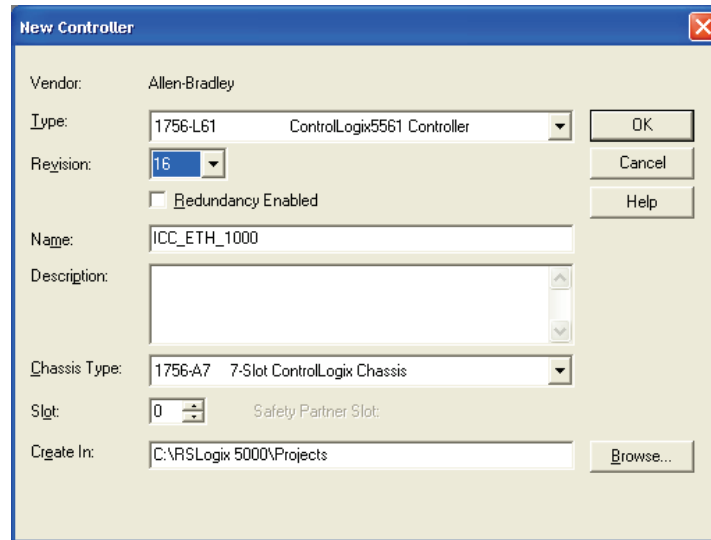
4. Download the parameter settings to the motion controller, then reset the controller by holding the “STOP” switch to the left for two seconds and then returning it to the RUN position.
5. Repeat the steps for all motion controllers on the network.

Chapter 4 ControlLogix PLC Project Configuration

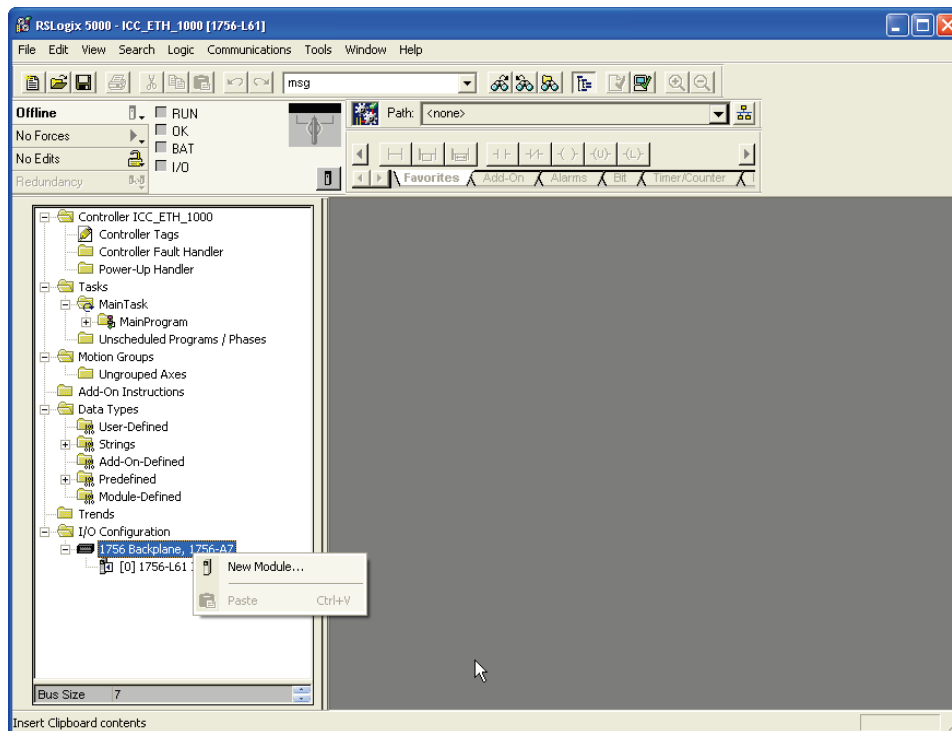
The configuration steps of a ControlLogix project are described in this chapter. These steps are used to communicate with an ICC ETH-1000 module. It is assumed that the user has basic knowledge in using RSLogix5000 software to perform the basic configuration steps.

4.1 Adding the 1756 ENBT Module

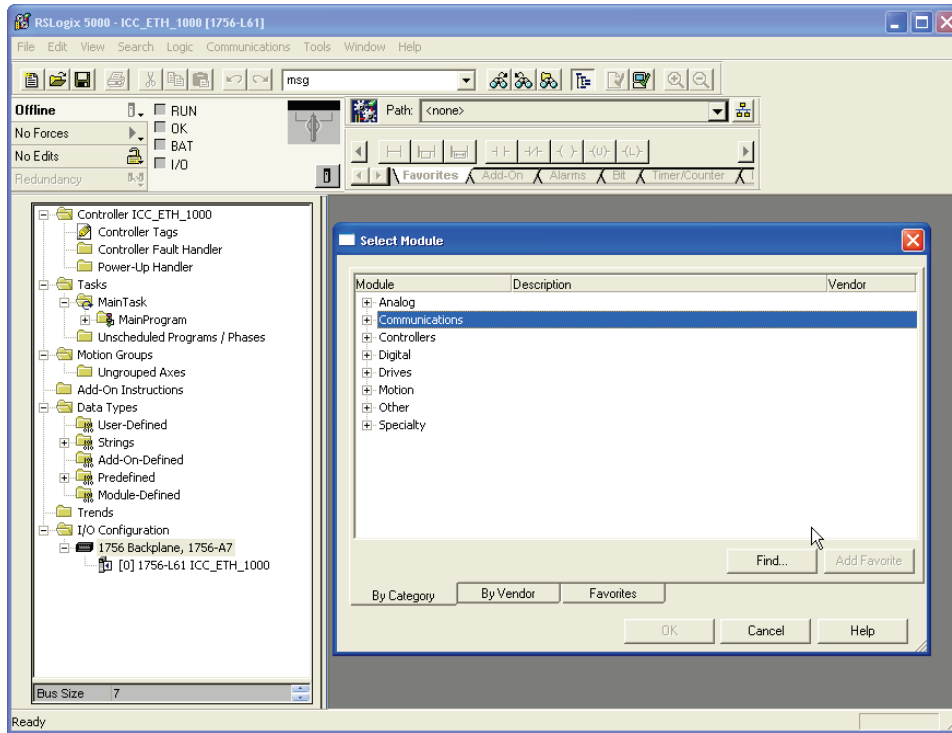
1. Create a new RSLogix5000 project using the proper revision level of the ControlLogix controller. In this example, the revision level is 16.



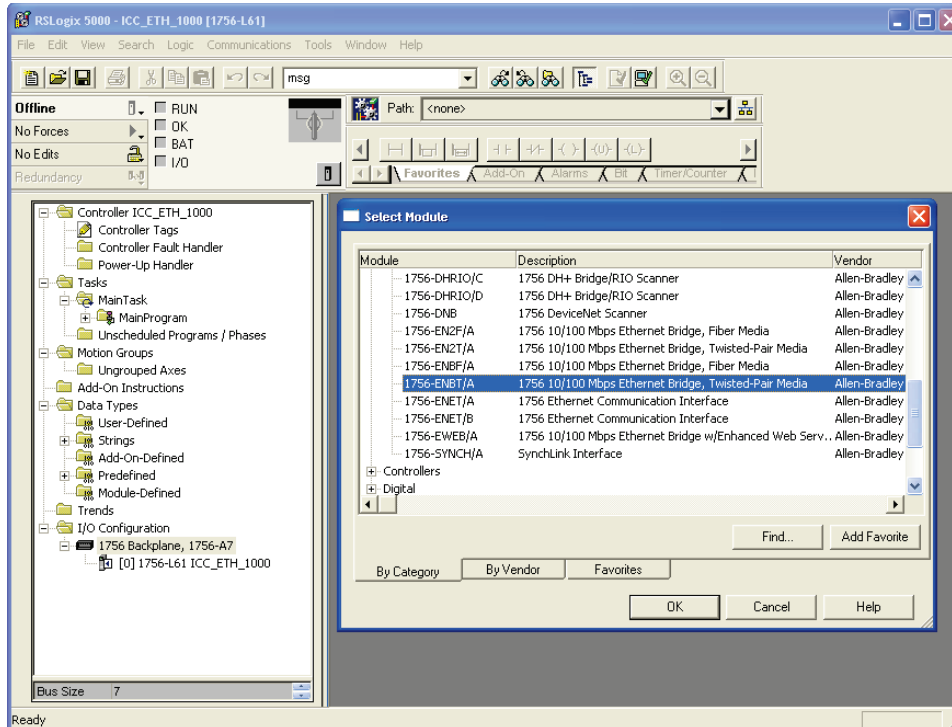
2. Right Click on the 1756 Backplane Selection and choose “New Module...”



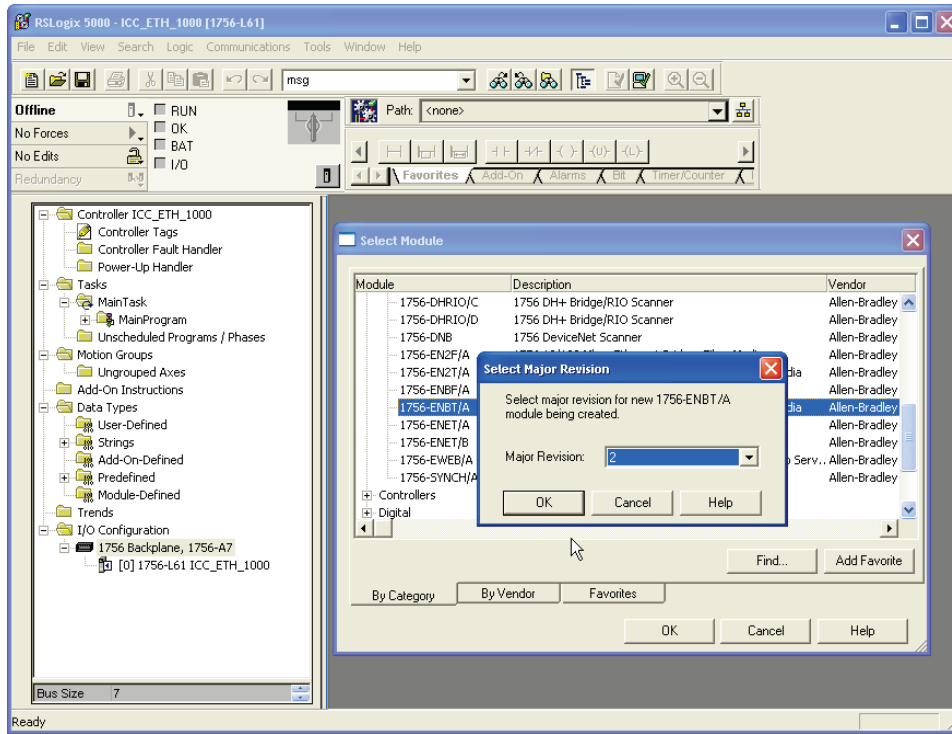
3. In the “Select Module” pop-up window, choose “Communications”.



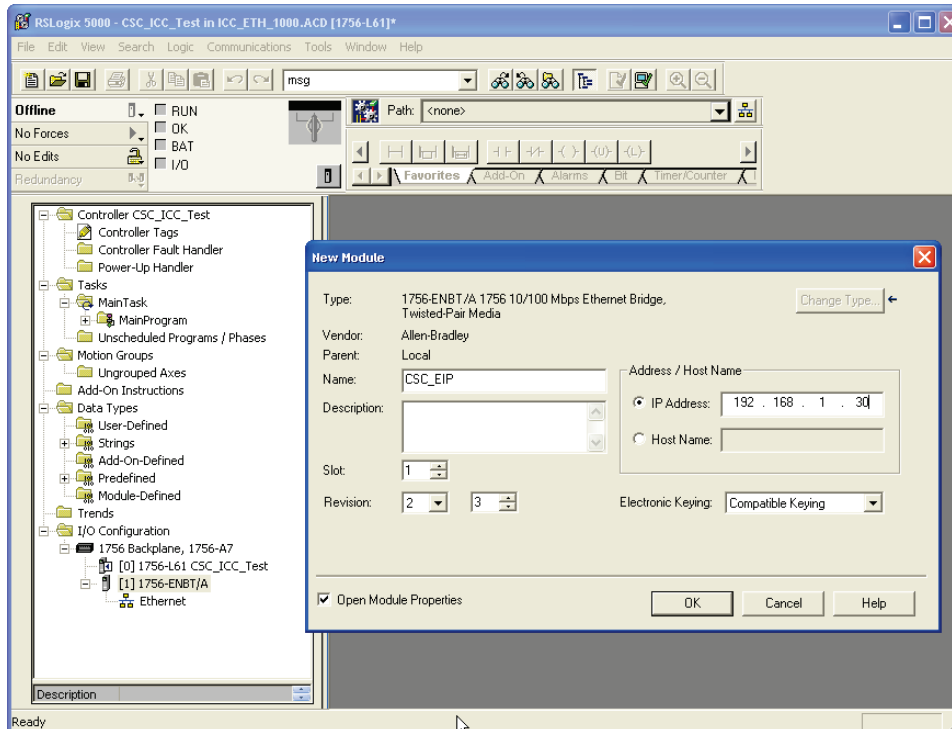
4. Select the 1756-ENBT module.



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



- 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.

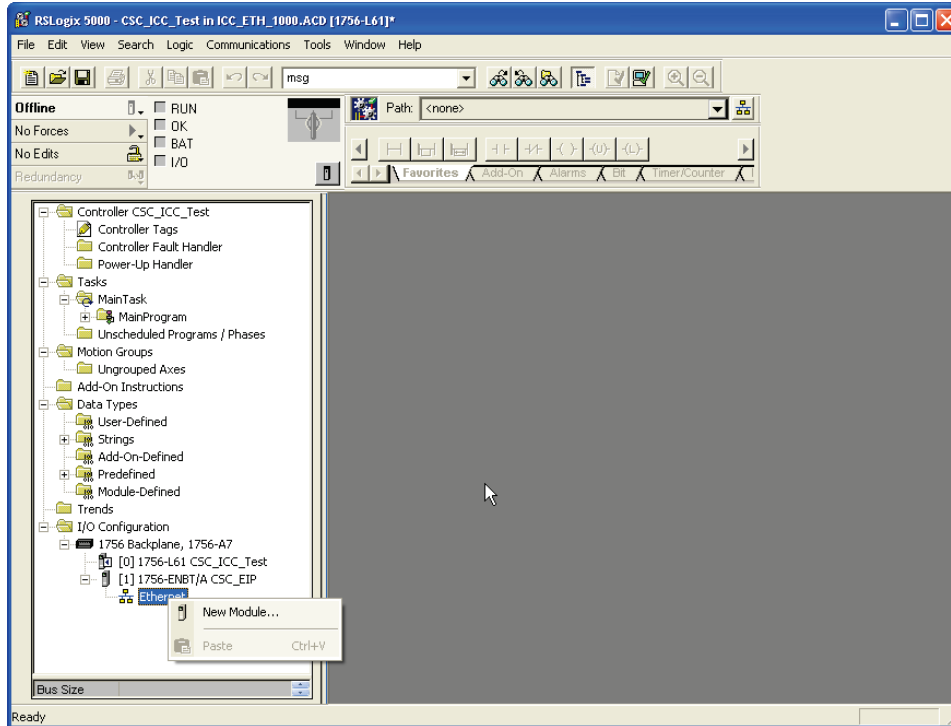


- Click “OK” to accept the configuration and make no additional configuration changes to the “Connection” tab. Click “OK” again to accept the configuration.

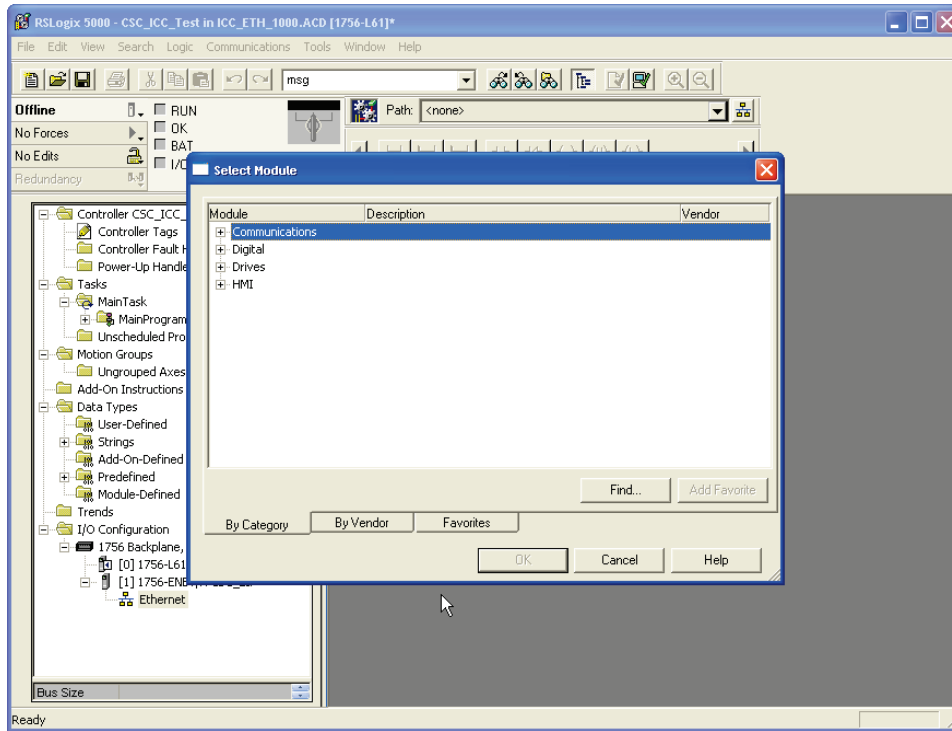
4.2 Adding the ETH-1000 Module

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

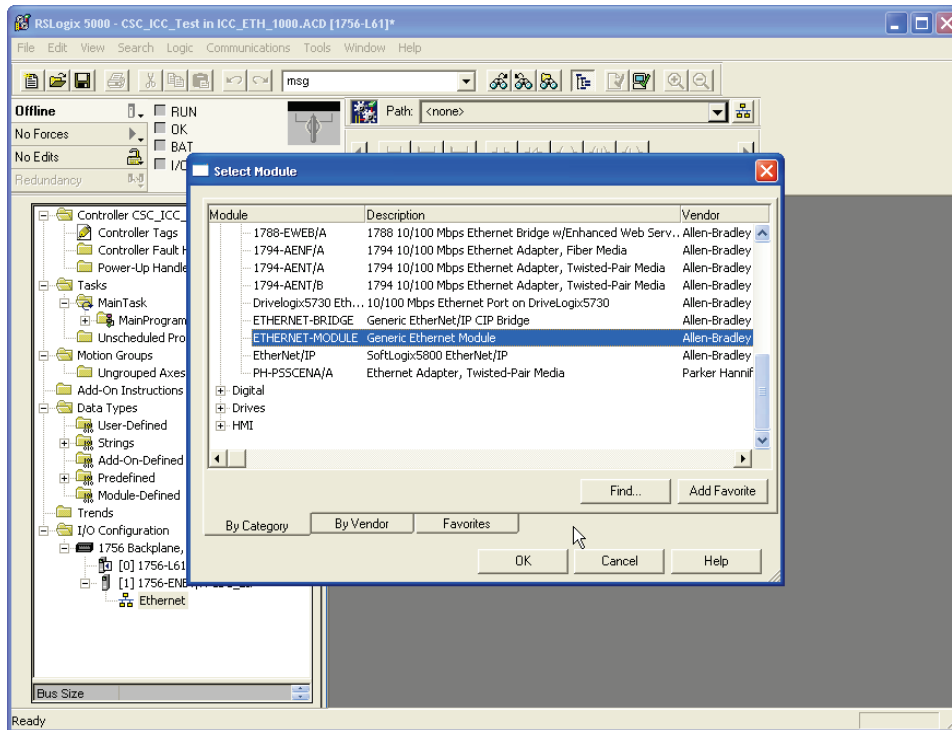
- Right click on the “Ethernet” icon under the ENBT module and select “New Module...”



2. Select "Communications" and expand the tree for additional selections.

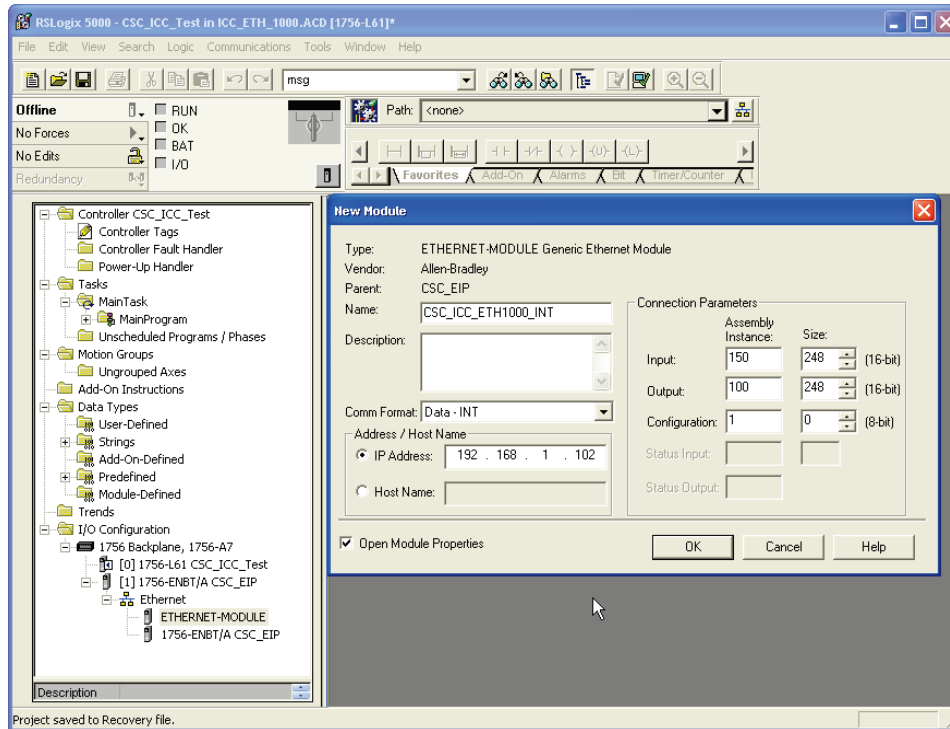


3. Choose "Ethernet-Module / Generic Ethernet Module".



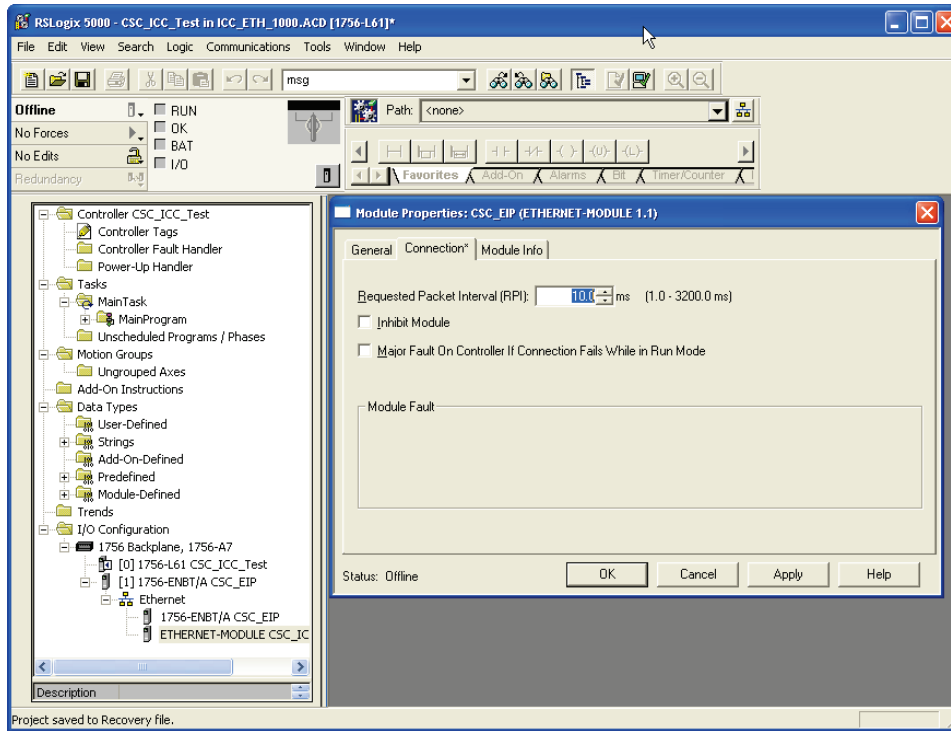
1	Introduction
2	System Overview
3	Connecting Devices to the Network
4	ControlLogix PLC Project Configuration
5	ETH-1000 Configuration
6	Using EtherNet/IP Explicit Messaging
7	Terminology

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.

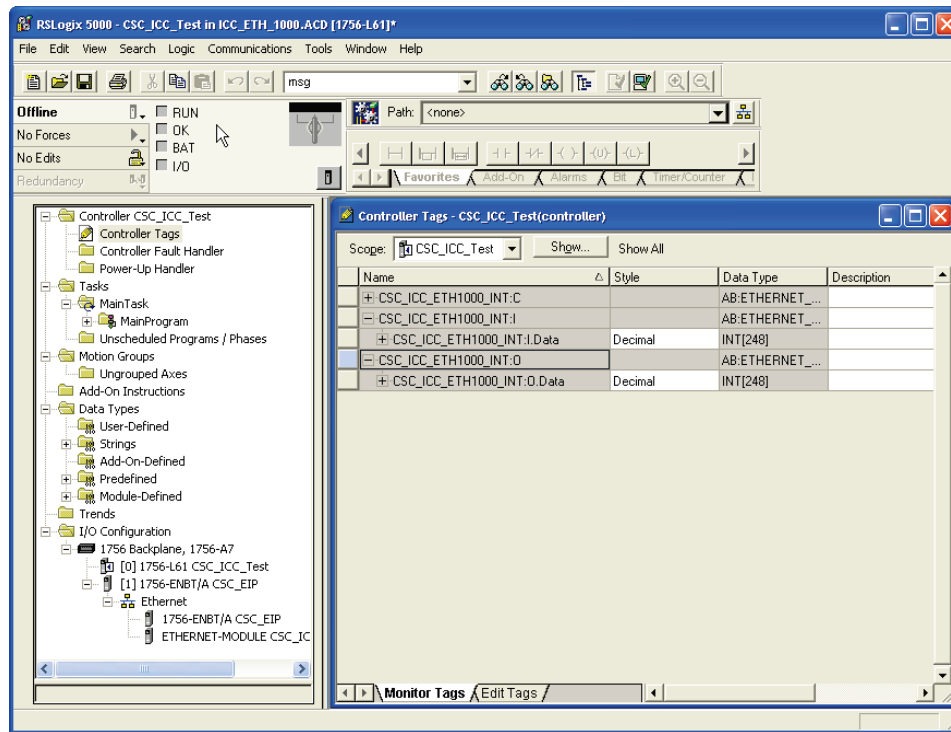


- a. Configure the “Comm Format” as “Data-INT” for the overall system to work best with the ETH-1000 and the motion control registers. This will allow the transfers to be done in 16 bit integers.
 - **Note: The Comm Format data type should be appropriately configured to match the requirements of each 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.

- Configure the RPI to 10.0ms.



- Select "OK" to accept and complete the ETH-1000 module configuration.
- Double-click the "Controller Tags" selection to view the automatically-created module tags:



- 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 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.

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 5 ETH-1000 Configuration

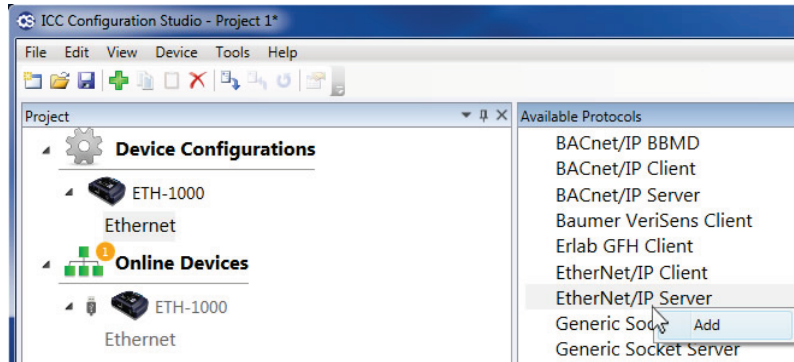
This chapter documents the steps to configure the ETH-1000 to allow communication between a ControlLogix PLC (using EtherNet/IP class 1 I/O messaging) and Mitsubishi motion controllers (using MC protocol). This configuration is performed with the ICC Configuration Studio software.

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.

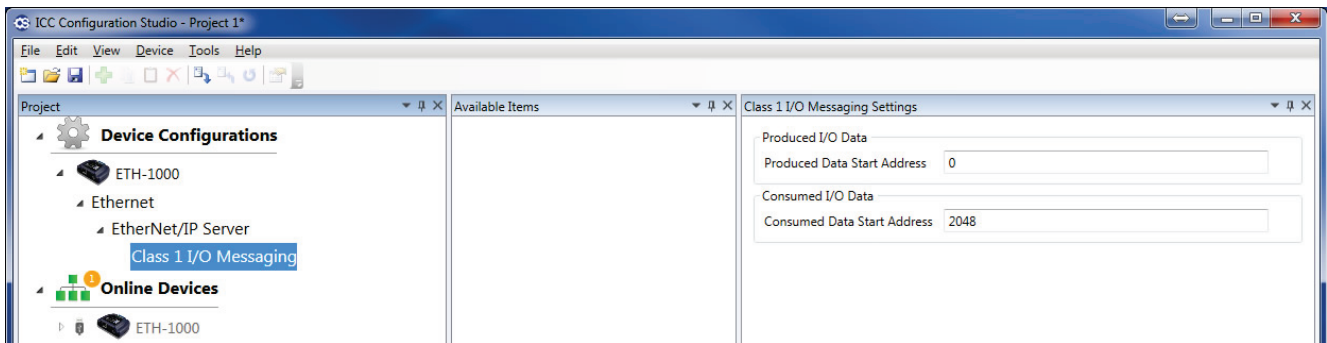
Prior to performing the following procedure, the configuration steps as shown in Section 3.2 of this document should first be completed.

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.



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.

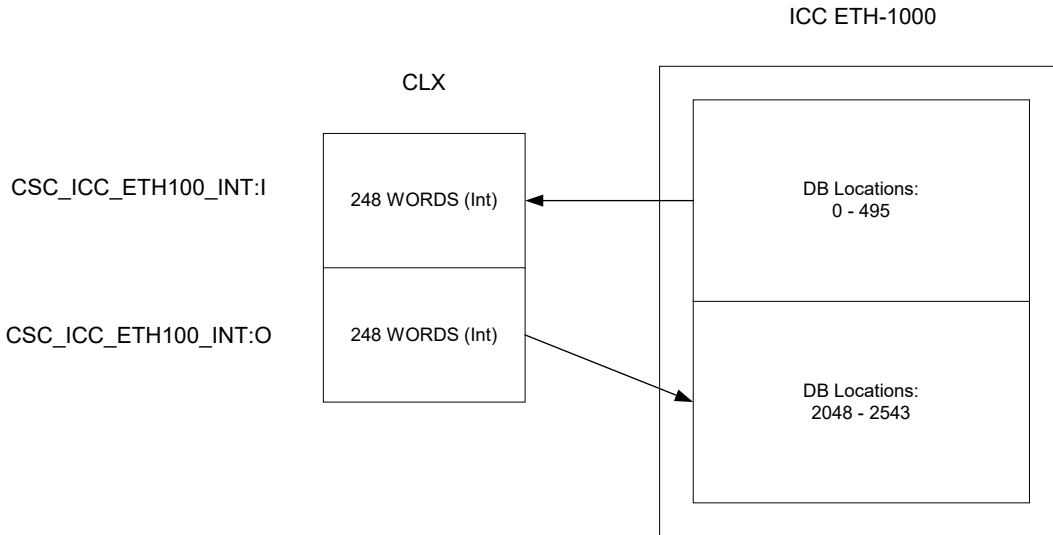


Figure 3: Mapping ControlLogix Tags to ETH-1000 Database Locations

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.

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

Figure 4: Produced Data Database Buffer

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	0	0	0	0	0
2080	0	0	0	0	0	0	0	0

Figure 5: Consumed Data Database Buffer

5.2 Configuring the MELSEC Protocol Driver

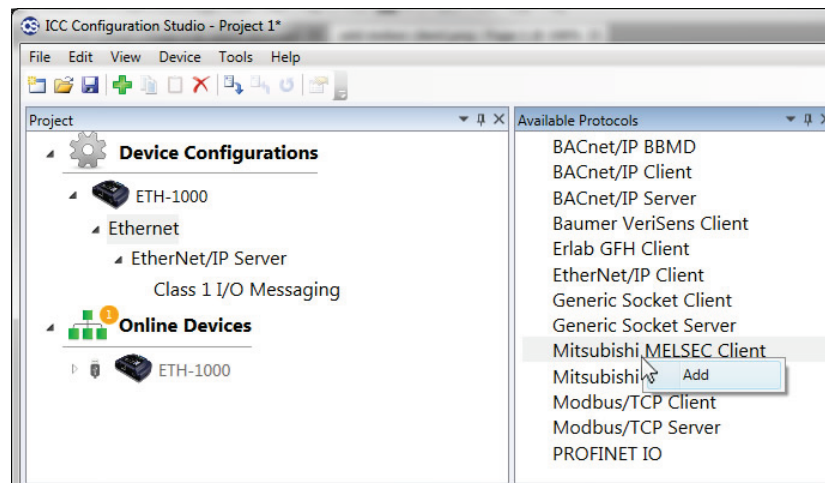
This section documents the steps to properly configure the MELSEC protocol driver. Note that this configuration is relevant only for the example Verification System architecture as shown in Figure 2: *the specific Connection Objects and Service Objects should be appropriately configured to match the requirements of each particular application.* Users should consult the ETH-1000 user's manual and understand the information and timing requirements required for transferring data to and from Mitsubishi motion controllers. For reference purposes, a technical report documenting the timing study of the Verification System architecture is available from Mitsubishi Electric Automation, Inc.

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

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 motion controller. 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 10 words of a motion controller’s D registers starting from D00, or to write 2 words to the Link Relay area.

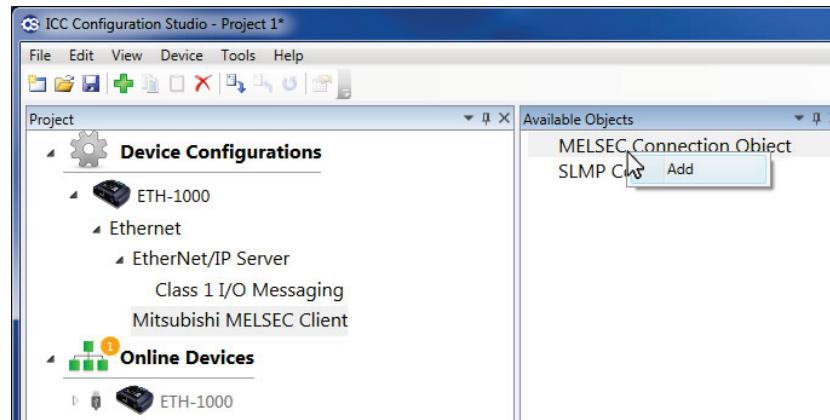
1. In the “Project” panel “Device Configurations” heading, expand the ETH-1000 configuration and select the “Ethernet” node. In the “Available Protocols” panel, right-click on “Mitsubishi MELSEC Client” and select “Add” to activate the MELSEC client driver. “Mitsubishi MELSEC Client” will then appear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

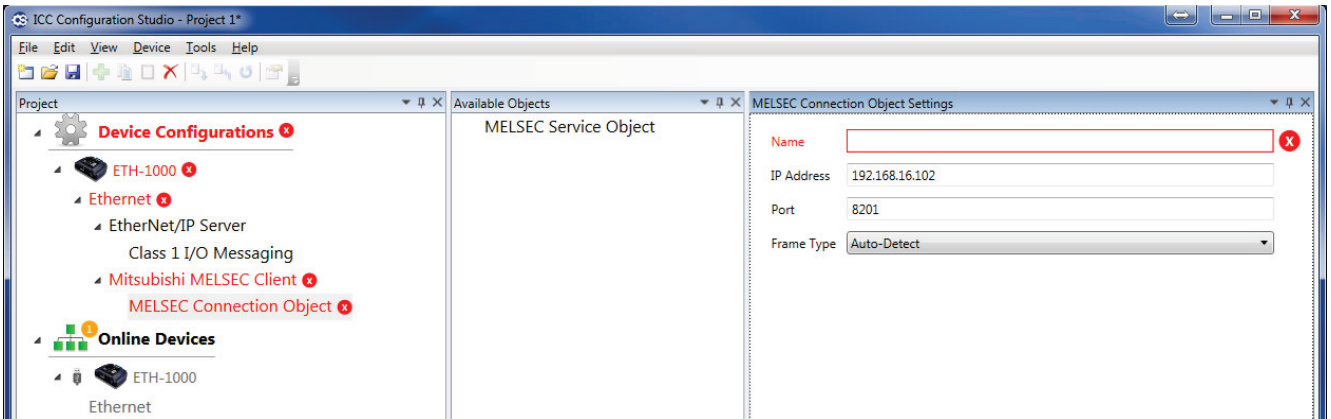


5.3 Configuring Connection Objects

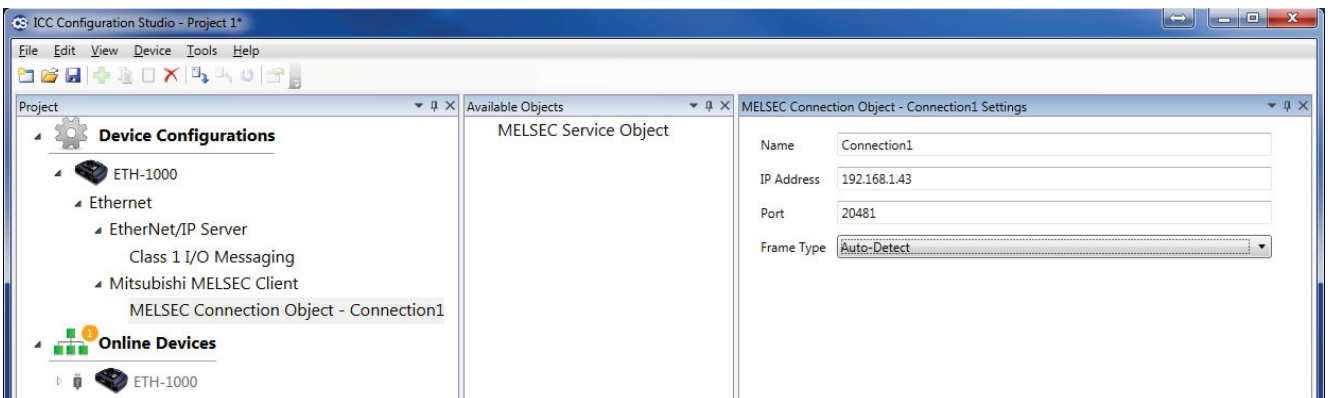
The following steps will create a Connection Object to communicate with a Q170 motion controller residing at IP address 192.168.1.43 using UDP port 0x5001 (20481 decimal).

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.

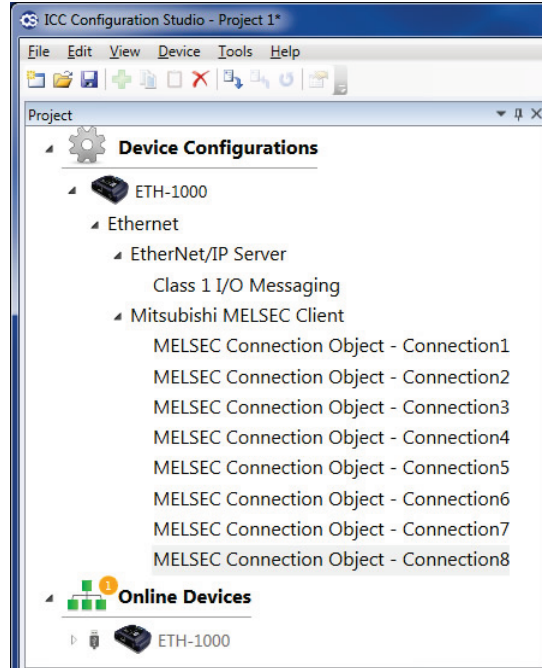




3. Provide a name for this Connection Object. The name “Connection1” has been used for this example.
4. Enter the IP Address of the targeted motion controller. The IP Address is 192.168.1.43 for this example.
5. Enter the Port to use on the targeted motion controller. The Port is 0x5001 (20481 decimal) for this example.
6. Set the Frame Type to “Auto-Detect” or “3E Frame”.



7. Repeat Steps 1-6 above to create the number of Connection Objects required for the application. For the Verification System architecture, eight Connection Objects are created as shown in the following image of the project tree.



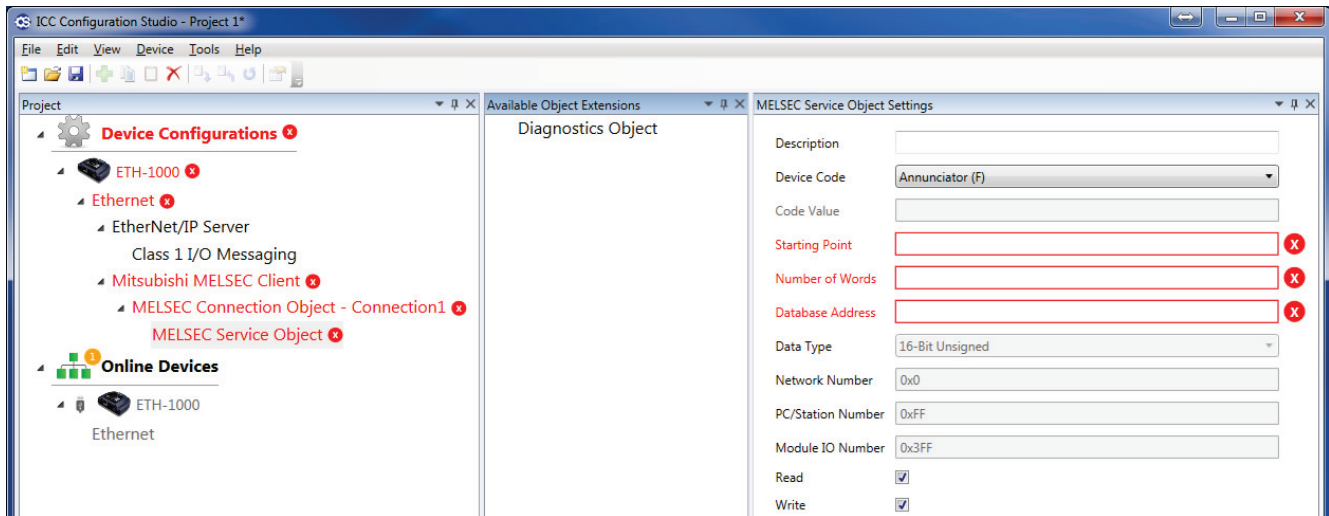
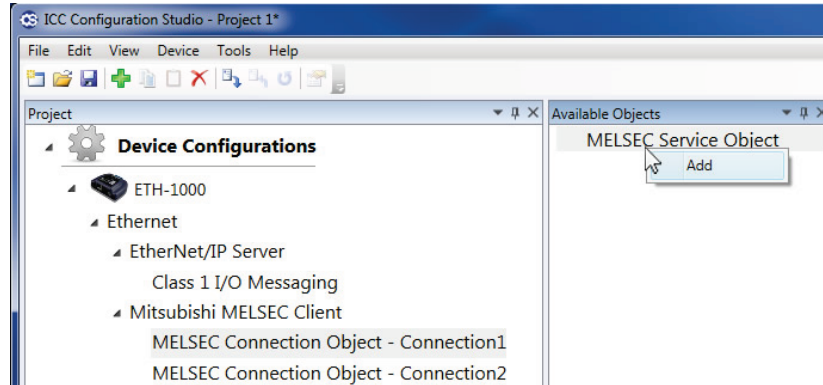
5.4 Configuring Service Objects

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

- Write 21 words from ETH-1000 to Motion Register #0...Motion Register #20
- Read 21 words from Motion Register #0...Motion Register #20 to ETH-1000
- Write 10 words (160 bits) from ETH-1000 to Link Relay B0...B159
- Read 10 words (160 bits) from Link Relay B0...B159 to ETH-1000

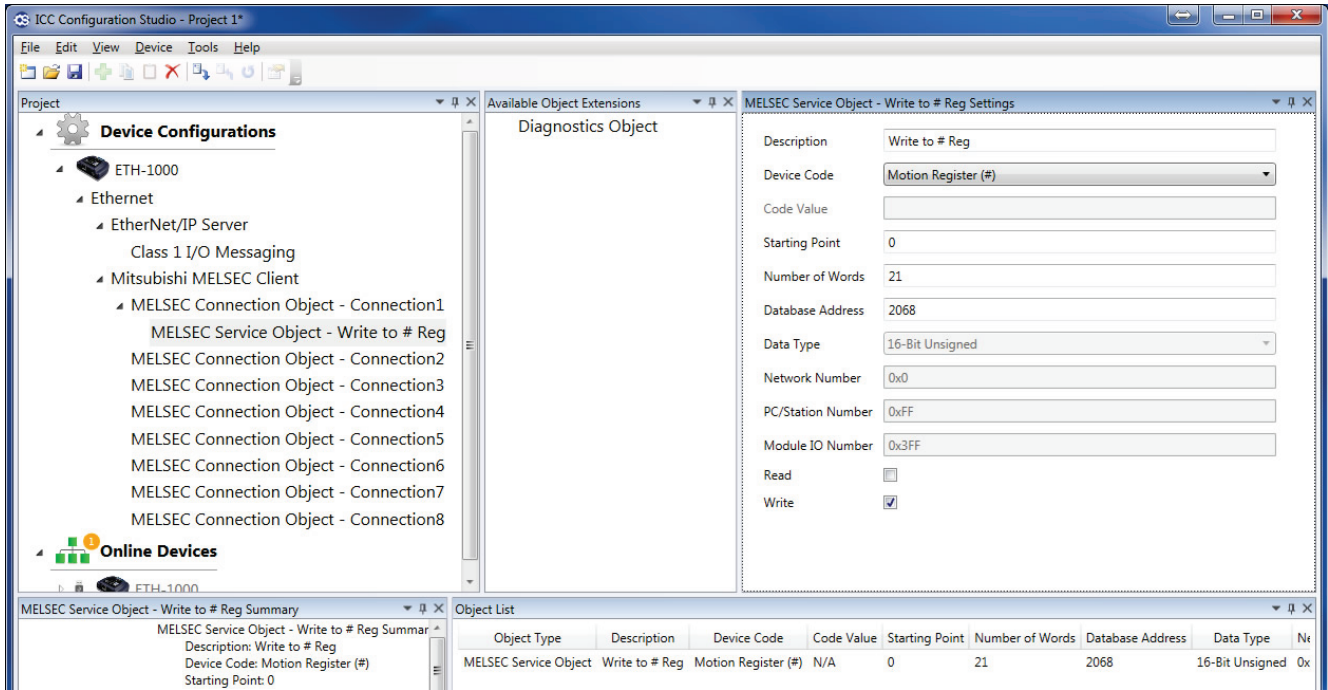
Perform the following steps to create and configure the Service Object that will write 21 words from the ETH-1000 to Motion Registers #0...#20:

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.

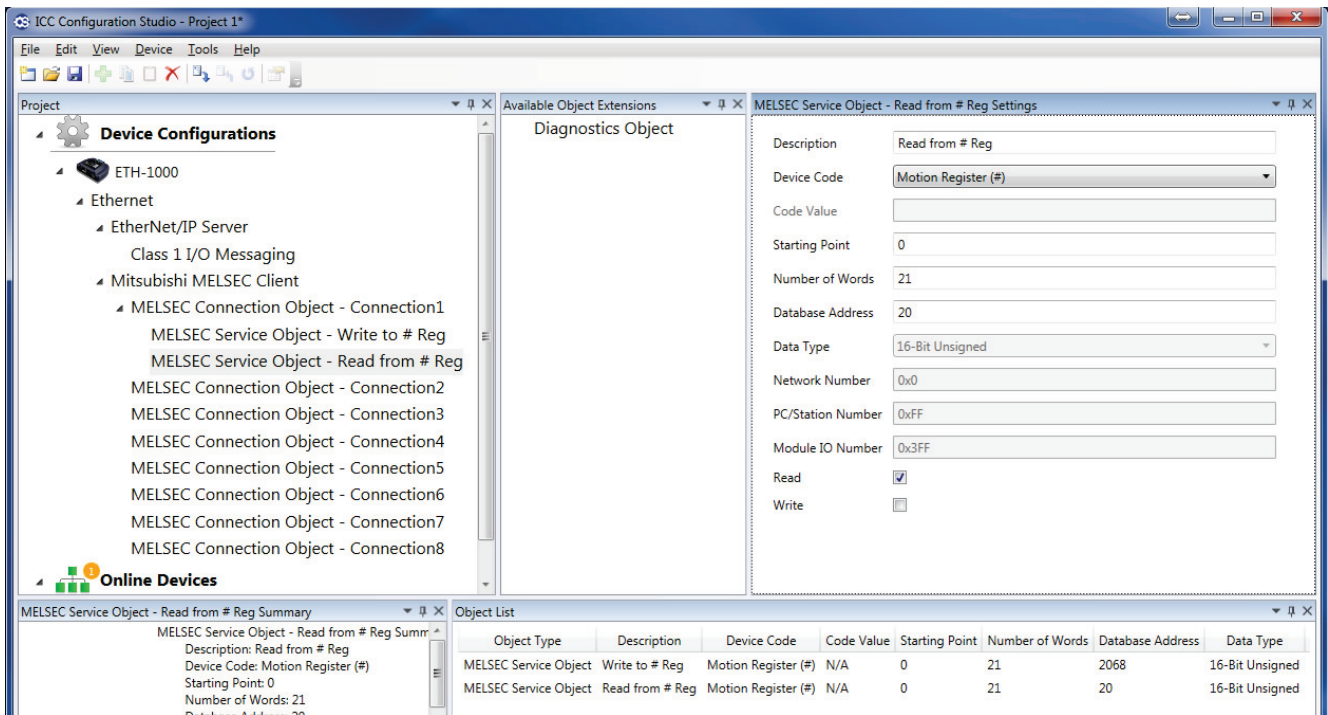


3. Enter an optional “Description” string to document this Service Object’s purpose.
4. Select the targeted “Device Code” on the motion controller to which data will be written. For this example, select “Motion Register (#).”
5. Enter the “Starting Point” of the Motion Register area where the data will be written to. In this example, the Starting Point is 0.
6. Enter the “Number of Words” to be transferred. In this example, the Number of Words is 21.
7. Enter the “Database Address” which designates the starting location where stored data will be written to the motion registers. In this example, the starting Database Address is 2068.
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 motion register locations on the motion controller. For this example, only the “Write” checkbox is selected.



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 Motion Registers #0...#20 to the ETH-1000:



b. Define the Service Object that will write 10 words (160 bits) from the ETH-1000 to Link Relay B0...B159:

The screenshot shows the ICC Configuration Studio interface. The 'Device Configurations' tree on the left is expanded to 'Mitsubishi MELSEC Client' > 'MELSEC Connection Object - Connection1' > 'MELSEC Service Object - Write to LR'. The 'MELSEC Service Object - Write to LR Settings' window is open, showing the following configuration:

- Description: Write to LR
- Device Code: Link Relay (B)
- Code Value: (empty)
- Starting Point: 0
- Number of Words: 10
- Database Address: 2048
- Data Type: 16-Bit Unsigned
- Network Number: 0x0
- PC/Station Number: 0xFF
- Module IO Number: 0x3FF
- Read:
- Write:

The 'Object List' table at the bottom shows the configured object:

Object Type	Description	Device Code	Code Value	Starting Point	Number of Words	Database Address	Data Type
MELSEC Service Object	Write to # Reg	Motion Register (#)	N/A	0	21	2068	16-Bit Unsigned
MELSEC Service Object	Read from # Reg	Motion Register (#)	N/A	0	21	20	16-Bit Unsigned
MELSEC Service Object	Write to LR	Link Relay (B)	N/A	0	10	2048	16-Bit Unsigned

c. Define the Service Object that will read Link Relay B0...B159 (160 bits/10 words) to the ETH-1000:

The screenshot shows the ICC Configuration Studio interface. The 'Device Configurations' tree on the left is expanded to 'Mitsubishi MELSEC Client' > 'MELSEC Connection Object - Connection1' > 'MELSEC Service Object - Read from LR'. The 'MELSEC Service Object - Read from LR Settings' window is open, showing the following configuration:

- Description: Read from LR
- Device Code: Link Relay (B)
- Code Value: (empty)
- Starting Point: 0
- Number of Words: 10
- Database Address: 0
- Data Type: 16-Bit Unsigned
- Network Number: 0x0
- PC/Station Number: 0xFF
- Module IO Number: 0x3FF
- Read:
- Write:

The 'Object List' table at the bottom shows the configured object:

Object Type	Description	Device Code	Code Value	Starting Point	Number of Words	Database Address	Data Type
MELSEC Service Object	Write to # Reg	Motion Register (#)	N/A	0	21	2068	16-Bit Unsigned
MELSEC Service Object	Read from # Reg	Motion Register (#)	N/A	0	21	20	16-Bit Unsigned
MELSEC Service Object	Write to LR	Link Relay (B)	N/A	0	10	2048	16-Bit Unsigned
MELSEC Service Object	Read from LR	Link Relay (B)	N/A	0	10	0	16-Bit Unsigned

10. Repeat Steps 1-9 above to create and configure the Service Objects for the other seven Connection Objects.

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 PLC 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 PLC to the motion controllers defined in the system. Figure 6 illustrates the complete mapping for the Validation System shown in Figure 2.

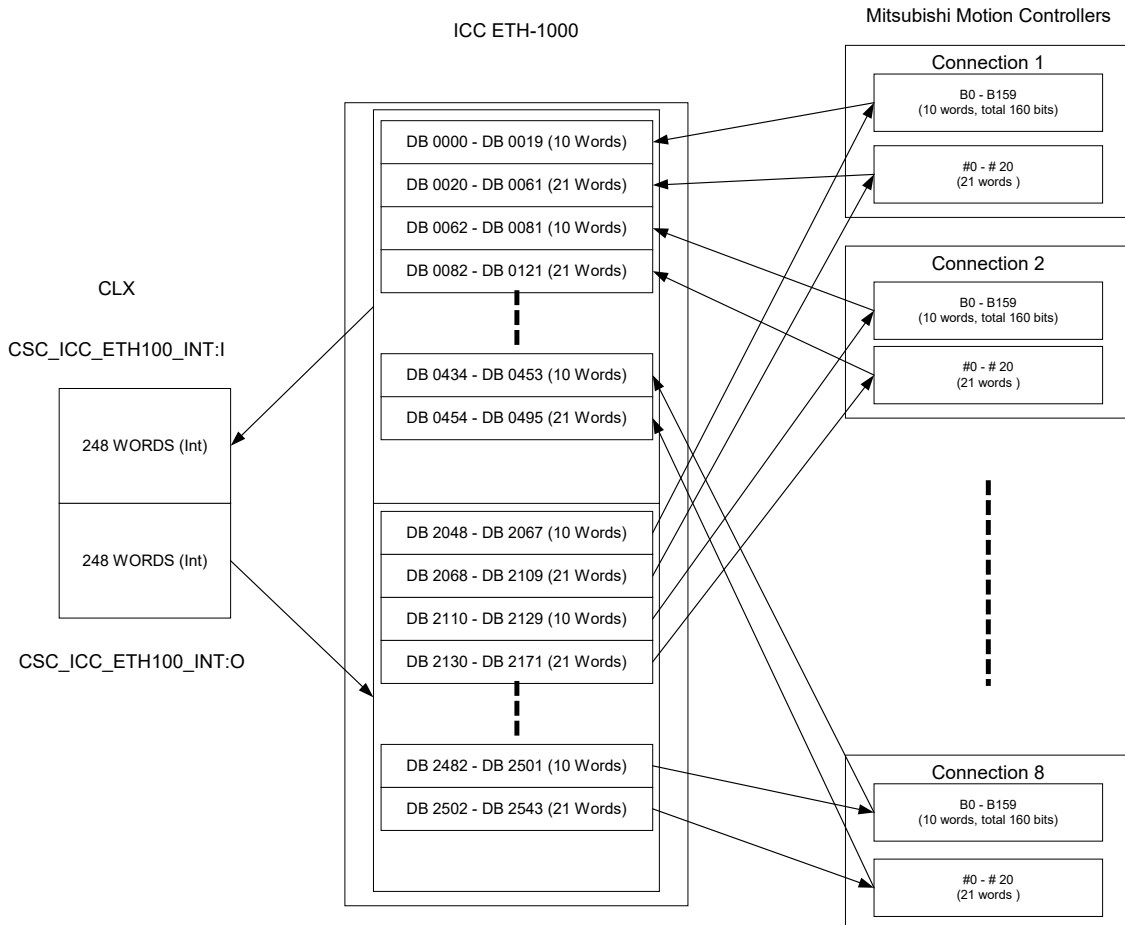


Figure 6: Complete Mapping of ControlLogix Tags to Motion Controller Elements

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.

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

6

Chapter 6 Using EtherNet/IP Explicit Messaging

The ControlLogix PLC can communicate with the Mitsubishi motion controllers 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.

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 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 (motion controller). 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 motion controller 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 1.2