ControlNet Router

(Enhanced)

Firmware Revision v1.001.010+

User Manual

A-CNR

Document No. D138-016
09/2026
Revision 1.15

aparian

CONTENTS

1. Preface 7
1.1. Introduction to the ControlNet Router (Enhanced) 7
1.2. Features 11
1.3. Architecture 12
1.3.1. Ethernet-Only PanelView interface 12
1.3.2. EtherNet/IP Target to ControlNet IO (Scheduled ControlNet) 14
1.3.3. Modbus TCP Slave to ControlNet IO (Scheduled ControlNet) 14
1.3.4. ControlNet Target (Scheduled) to EtherNet/IP IO (Class 1) 15
1.3.5. ControlNet Target (Scheduled) to Modbus TCP Slaves 16
1.4. Additional Information 17
1.5. Support 18
2. Installation 19
2.1. Module Layout 19
2.2. Module Mounting 20
2.3. Power 22
2.4. ControlNet 22
2.5. Ethernet Port 22
2.6. North American Hazardous Location Approval 22
3. Setup 24
3.1. Install Configuration Software 24
3.2. Network Parameters 24
3.3. Creating a New Project 28
3.4. General parameters 30
3.5. ControlNet Configuration 32
3.5.1. Target 34
3.5.2. Originator 45
3.5.3. Bridge 64
3.6. Ethernet Mode 69
3.6.1. PCCC Client 69
3.6.2. EtherNet/IP Target 77
3.6.3. Modbus TCP Slave 84

3.6.4. Modbus TCP Master....87
3.6.5. EtherNet/IP Originator....91
3.7. Internal Data Space Map....106
3.7.1. Copy From....107
3.7.2. Copy To....111
3.8. Advanced....115
3.9. Module Download....116
4. Logix Controller Programming Over ControlNet....119
4.1. RSLinx Setup....119
4.1.1. Uploading EDS File from Module....120
4.1.2. Manually Installing the EDS File Option....122
4.2. RSLogix 5000....125
5. Device Firmware Update....127
6. Operation....130
6.1. ControlNet Target....130
6.1.1. Scheduled....130
6.2. ControlNet Originator....131
6.2.1. Scheduled ControlNet Connections....131
6.2.2. Explicit Messaging....133
6.3. ControlNet Bridge....135
6.3.1. PLC5 IO Data Exchange....136
6.4. EtherNet/IP Target....139
6.4.1. Class 1 Connection....139
6.5. EtherNet/IP Originator....140
6.5.1. EtherNet/IP Class 1 Connections....140
6.5.2. Explicit Messaging....142
6.6. Modbus TCP Master....144
6.7. Modbus TCP Slave....145
6.8. FTView / PanelView Interfacing....146
6.8.1. PanelView Reading Data From Logix....147
6.8.2. PanelView Writing Data To Logix....148
6.8.3. PanelView Reading Diagnostic Data from ControlNet Router....149
6.9. Internal Data Space (IDS)....150

6.9.1. Internal Status 150
6.9.2. Internal Control 152
7. Diagnostics 153
7.1. LEDs 153
7.2. Module Status Monitoring in Slate 154
7.2.1. General 155
7.2.2. ControlNet Statistics 160
7.2.3. Bridge Connections 161
7.2.4. Bridge Failures 162
7.2.5. ControlNet Explicit 163
7.2.6. ControlNet Map 164
7.2.7. PCCC Statistics 165
7.2.8. EtherNet/IP Explicit 166
7.2.9. EtherNet/IP Map 167
7.2.10. EtherNet/IP Originator 168
7.2.11. Logix 169
7.2.12. Modbus 170
7.2.13. CIP Statistics 172
7.2.14. Ethernet Clients 173
7.2.15. TCP/ARP 173
7.3. Target Device Status Monitoring In Slate 174
7.3.1. EtherNet/IP 174
7.3.2. ControlNet 177
7.4. Module Event Log 180
7.5. Web Server 181
7.6. ControlNet Packet Capture 182
7.6.1. Operational Capture 183
7.6.2. Promiscuous Capture 185
7.7. Modbus Packet Capture 188
7.8. Module Status Report 190
7.9. Internal Register Viewer 191
8. Technical Specifications 193
8.1. Dimensions 193

8.2. Electrical 194
8.3. Ethernet 194
8.4. ControlNet 195
8.5. ControlNet Target 195
8.6. ControlNet Originator 195
8.7. ControlNet Bridge 196
8.8. PCCC 196
8.9. EtherNet/IP Target 196
8.10. EtherNet/IP Originator 197
8.11. Modbus TCP Master 197
8.12. Modbus TCP Slave 197
8.13. Certifications 198
9. Appendix 200
10. Index 201

Revision History

Revision	Date	Comment
1.0	31 January 2022	Initial document
1.1	16 February 2022	Fixed ControlNet Router catalog number A-CNR.
1.2	11 July 2022	Update information for Scheduled ControlNet Originator IO space.
1.3	4 August 2022	Added information required for UL regarding open type device enclosures.
1.4	17 Jan 2023	Update support contact details
1.5	8 November 2023	Added ATEX Conformance Mark Added UKCA Conformance Mark
1.6	28 November 2023	Updated EtherNet/IP Class 1 Connection count and Import options
1.7	19 February 2024	Updated Temperature Rating
1.8	7 June 2024	Added Bridge Mode Added A-CNR Logix Add-On Profile (AOP) Added Soft Schedule for ControlNet
1.9	19 March 2025	Added information regarding PLC5 IO integration into Logix via the A-CNR.
1.10	20 May 2025	Added duplicate IP address indication to LEDs in the Diagnostics section.
1.11	10 June 2025	Added Virtual Gateway Routing (SCADA) option.
1.12	28 January 2026	Added Internal Register Viewer section. Added note that current ControlNet Keeper modules can have the Keeper cleared with Factory Reset.
1.13	9 February 2026	Added note about ControlLogix Controller firmware issue with ControlNet Flex IO. Added note about RSLinx A_ETH driver issue.
1.14	20 April 2026	Added UL North American Hazardous Location Approval.
1.15	2 September 2026	Updated specifications.

1. PREFACE

1.1. INTRODUCTION TO THE CONTROLNET ROUTER (ENHANCED)

This manual describes the installation, operation, and diagnostics of the Aparian ControlNet Router (Enhanced). The ControlNet Router (Enhanced) is defined as the ControlNet Router with firmware revision 1.001.010 (or later) being used with Slate v1.057 (or later).

The ControlNet Router, (hereafter referred to as the **module**) provides intelligent data routing between either EtherNet/IP or Modbus TCP/RTU and a ControlNet network. This allows the user to integrate ControlNet devices into a EtherNet/IP based Rockwell Logix platform (e.g., ControlLogix or CompactLogix) or any Modbus Master or Slave device with minimal effort.

The ControlNet Router also provides intelligent data routing between ControlNet and Ethernet PCCC (CSP). This will allow the user to connect new PanelView Plus devices (with no ControlNet interface) to existing ControlNet networks.

The **ControlNet Mode** can be configured to be either of the following two interface modes:

Target
A Logix controller can own the ControlNet Router over ControlNet using a scheduled ControlNet connection when the ControlNet Router is operating as a ControlNet target. This will allow the ControlNet Router to exchange data with the Logix controller using the input and output assembly of the scheduled ControlNet connection. Data from EtherNet/IP or Modbus TCP devices can be mapped to the Logix controller over ControlNet.

Bridge
When the ControlNet Router is operating in Bridge Mode, it will allow the user to add, configure, and parameterize ControlNet IO via the ControlNet Add-on Profile (AOP) in Logix. ControlNet devices can be added under the A-CNR AOP which will operate as a ControlNet bridge. In this mode the ControlNet Router will operate as a ControlNet originator and Keeper, but the ControlNet network will be dynamically Scheduled as ControlNet Connections are established.



NOTE: When using a 1756-L8x Logix Controller (5580), firmware revision 37.11 has an anomaly that causes Flex IO over ControlNet to not connect. Please upgrade the Logix Controller to revision 37.12 to resolve this issue.

In Bridge Mode, the ControlNet Router will automatically become an EtherNet/IP

Target, allowing a newer Logix Controller, to own ControlNet IO via the ControlNet Router in the Studio5000 environment. This will allow for Rack Optimized connections in Logix. A maximum of 30 scheduled connections can be used in bridge mode (where rack optimized adapters will only use one connection for all devices in the local rack that support Rack Optimization).

The user will also be able to parameterize ControlNet devices (e.g., PowerFlex drive) from the Logix environment using the ControlNet device AOP.



NOTE: When the A-CNR is added in RSLinx under the AB_ETH driver (for legacy PCCC communication), it could cause the Add-On Profile for the A-CNR to not communicate correctly. Ensure that the A-CNR is under the AB-ETHIP driver (used for EtherNet/IP devices).

Originator

As a ControlNet originator, the module can use one of two methods to read and write data to and from ControlNet IO:

- **Scheduled ControlNet**

  The ControlNet Router can be configured to own ControlNet IO by using the Slate software to configure the IO connections and schedule the ControlNet network.

  Each ControlNet Router (when hardware scheduled) can own up to 10 ControlNet devices with up to 500 bytes of input data per connection and a total maximum of 205 to 277 bytes of output data (depending on the number of connections used). See the ControlNet Originator section for more details regarding the ControlNet originator connection sizes.

  Each ControlNet Router (when software scheduled) can own up to 30 ControlNet devices with up to 500 bytes of input and output data per connection.

  Multiple ControlNet Routers can exist on the same ControlNet network with the Slate software having the ability to schedule all of the ControlNet Routers on the network with originator IO connections and downloading the ControlNet Keeper configuration into one of the ControlNet Routers. Data from the ControlNet IO (via the IO input and output assemblies) can be exchanged with any of the Ethernet mode interfaces (e.g. EtherNet/IP Target or Modbus TCP Slave).

- **ControlNet Explicit Messaging**

  This allows the ControlNet Router to exchange data with up to 10 ControlNet devices using explicit messaging over ControlNet. The module can Get or Set data in the remote ControlNet devices with Custom Class, Instance, Attribute values.

The **Ethernet Mode** can be configure to be any of the following five interface modes:

PCCC Client

The support for Ethernet PCCC (CSP) allows the module to emulate a PLC5 or SLC providing a legacy interface for PanelViews and other devices over scheduled ControlNet. This will allow an Ethernet-only PanelView to exchange data with a Logix controller using an existing ControlNet connection.

EtherNet/IP Target

A Logix controller can own the ControlNet Router over EtherNet/IP using a class 1 connection when the ControlNet Router is operating as a EtherNet/IP Target. This will allow the ControlNet Router to exchange data with the Logix controller using the input and output assembly of the EtherNet/IP Class 1 connection. Data from ControlNet devices (when operating as a ControlNet Originator) or Controllers on ControlNet (when operating as a ControlNet Target) can be mapped to the Logix controller over EtherNet/IP.

EtherNet/IP Originator

As an EtherNet/IP originator, the module can use one of three methods to read and write data to and from the EtherNet/IP network:

- **EtherNet/IP Class 1 Connection**

  The ControlNet Router can be configured to own EtherNet/IP IO by using the Slate software to configure the Class 1 connections. Each ControlNet Router can own up to 16 EtherNet/IP devices. Data from the EtherNet/IP IO (via the input and output assemblies) can be exchanged with the ControlNet network (when operating as either a ControlNet Target or Originator).

- **EtherNet/IP Explicit Messaging**

  This allows the ControlNet Router to exchange data with up to 10 EtherNet/IP devices using explicit messaging over EtherNet/IP. The module can use either **Class 3**, Unconnected Messaging (**UCMM**), or **Logix Tag** to Get and Set data in the remote EtherNet/IP devices with configurable Class, Instance, Attribute values (when using UCMM or Class 3). Logix Tag messages are used to exchange data with a Logix controller by directly writing to or reading from Logix tags. The user can browse to the Logix controller (using the Slate Target Browser) as well as browse the Logix Controller Tag list (using the Slate Tag Browser) to select the desired destination Tag.

Modbus TCP Master

The data from either ControlNet IO (when operating as a ControlNet originator) or ControlNet input and output assemblies (when operating as a ControlNet Target) will

be written to, or read from, the module's internal Modbus Registers. The Modbus Auxiliary Map can then be used to configure the Modbus data exchange between multiple remote Modbus Slave devices and the module's internal Modbus registers. The Modbus communication utilizes Modbus TCP.

Modbus TCP Slave

The data from either ControlNet IO (when operating as a ControlNet originator) or ControlNet input and output assemblies (when operating as a ControlNet Target) will be written to, or read from, the module's internal Modbus Registers using the internal mapping functions. These Modbus registers can be accessed by a remote Modbus Master using Modbus TCP.

The ControlNet Router uses an internal mapping strategy allowing the user to map any data from any supported interface to any other supported interface. Slate provides an online import tool for existing Logix controllers with IO connections to assist with the setup and configure of originator connections (EtherNet/IP or ControlNet).

The ControlNet Router can also be used to go online and program a Logix controller over ControlNet using an Ethernet connection from the local PC.

The module provides a range of statistics on Ethernet and ControlNet to assist with fault finding. A built-in webserver provides detailed diagnostics of system configuration and operation, including the display of ControlNet operation and communication statistics, without the need for any additional software.

The ControlNet Router is configured using the Aparian Slate application. This program can be downloaded from www.aparian.com free of charge.

1.2. FEATURES

- The module has the following ControlNet operating modes:
    - **ControlNet Target**, where the ControlNet Router can be owned by a controller over scheduled ControlNet.
    - **ControlNet Bridge,** where the A-CNR owns ControlNet IO over a scheduled ControlNet connection, by using the A-CNR Add-on Profile (AOP) in Logix. The ControlNet IO will be added and configured in Logix while the ControlNet Router will dynamically schedule the ControlNet network as connections are established. This will allow for Rack Optimized connections in Logix.
    - **ControlNet Originator**, where the ControlNet Router owns ControlNet IO over a scheduled ControlNet connection.
- The module has the following Ethernet operating modes:
    - **PCCC Client,** allowing the ControlNet Router to emulate a PLC5 or SLC to exchange data with Ethernet-only PanelViews.
    - **EtherNet/IP Target,** where the ControlNet Router can be owned by a controller via a Class 1 EtherNet/IP connection.
    - **EtherNet/IP Originator,** where the ControlNet Router owns EtherNet/IP IO using a Class 1 EtherNet/IP connection.
    - **Modbus TCP Master,** where the ControlNet Router module can exchange data with Modbus TCP Slaves.
    - **Modbus TCP Slave,** where the ControlNet Router module can exchange data with a Modbus TCP Master.
- The ControlNet network can be scheduled using the Slate software environment.
- ControlNet Router can operate as **ControlNet Keeper**.
- Support for **PLC5 IO** (via the 1771-ACNR15 ControlNet adapter) when operating in Bridge Mode.
- Multiple ControlNet Routers can own ControlNet IO on the ControlNet network.
- Supports **Redundant ControlNet**.
- Supports **Promiscuous Capture** mode allowing the ControlNet Router to capture all traffic on the ControlNet network.
- Supports unscheduled ControlNet allowing **Logix programming** over ControlNet.
- Dual Ethernet ports which supports Device-Level-Ring (**DLR**).
- Network Time Protocol (**NTP**) supported for external time synchronization.
- ControlNet, EtherNet/IP, and Modbus **Statistics** and packet captures providing diagnostic data for fault finding.
- Built-in webserver providing (read-only) diagnostics.
- Small form factor – DIN rail mounted.

1.3. ARCHITECTURE

Various architectures can be supported by the ControlNet Router for interfacing various Ethernet protocols to ControlNet.

1.3.1. ETHERNET-ONLY PANELVIEW INTERFACE

When set to PCCC Client Mode, the ControlNet Router provides an interface between modern Rockwell Automation PanelView Plus HMIs (without any ControlNet interface) and existing ControlNet networks using either Unscheduled or Scheduled ControlNet.

When using unscheduled ControlNet, the ControlNet Router can be used as a drop-in replacement for existing legacy PanelView HMI's by simply changing the RSLinx path.

When using scheduled ControlNet, the ControlNet Router can be configured to allow for up to 400 bytes of input data and 400 bytes of output data to be exchanged with the Logix Controller. The ControlNet Router will emulate a PLC5 controller which is used by the PanelView HMI to map PLC5 Files (e.g. N7 or F8) to the Logix input and output assembly over scheduled ControlNet. The consumed (ControlNet) data can be mapped to a PLC5 type address file, N9 and F10, and then read by an Ethernet device e.g. a PanelView. Similarly, the produced (ControlNet) data can also be mapped to a PLC5 type address file, N7 and F8, to which an Ethernet device could write.

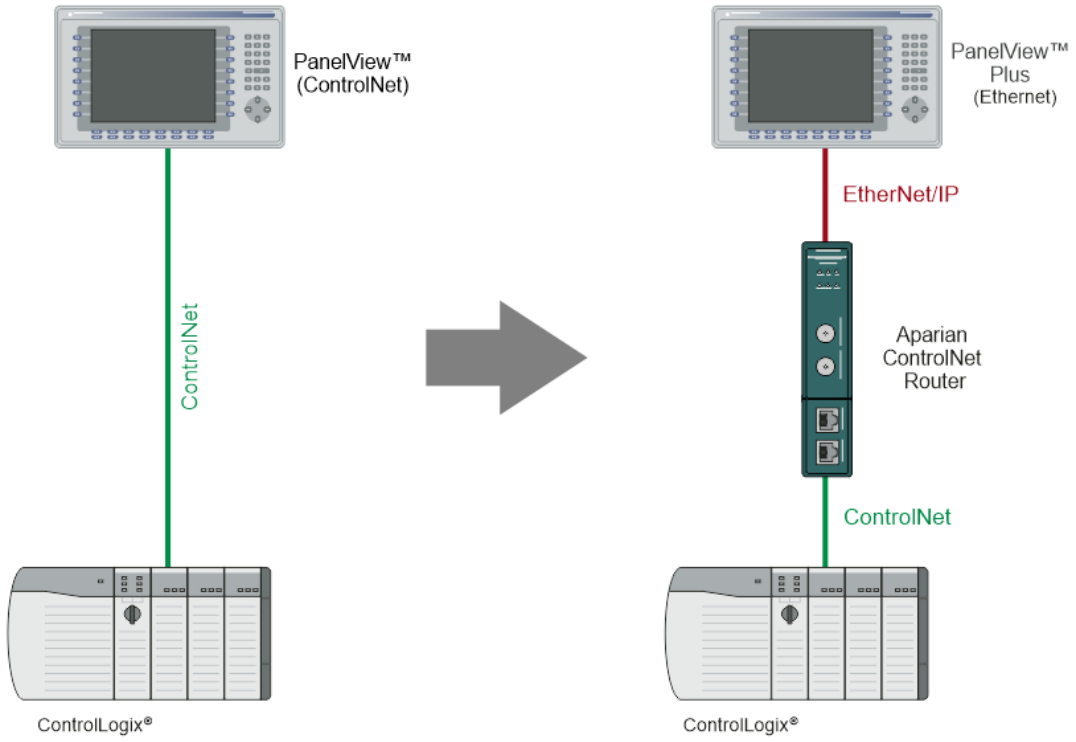


Figure 1.1 – Typical Setup for PanelViews

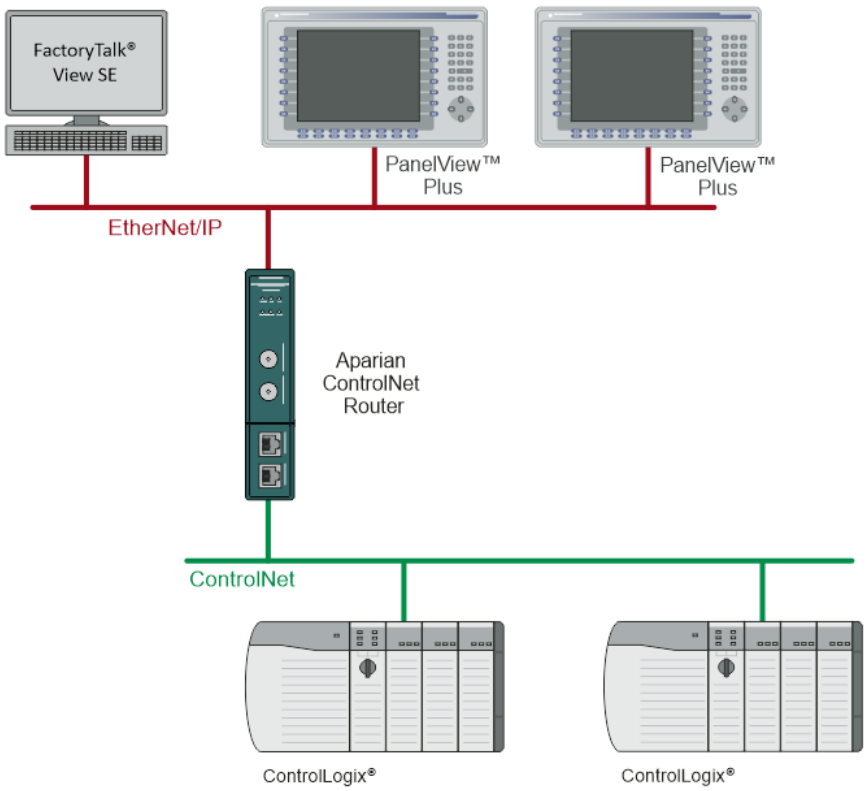


Figure 1.2. - Example of a typical network setup in PLC Emulation mode

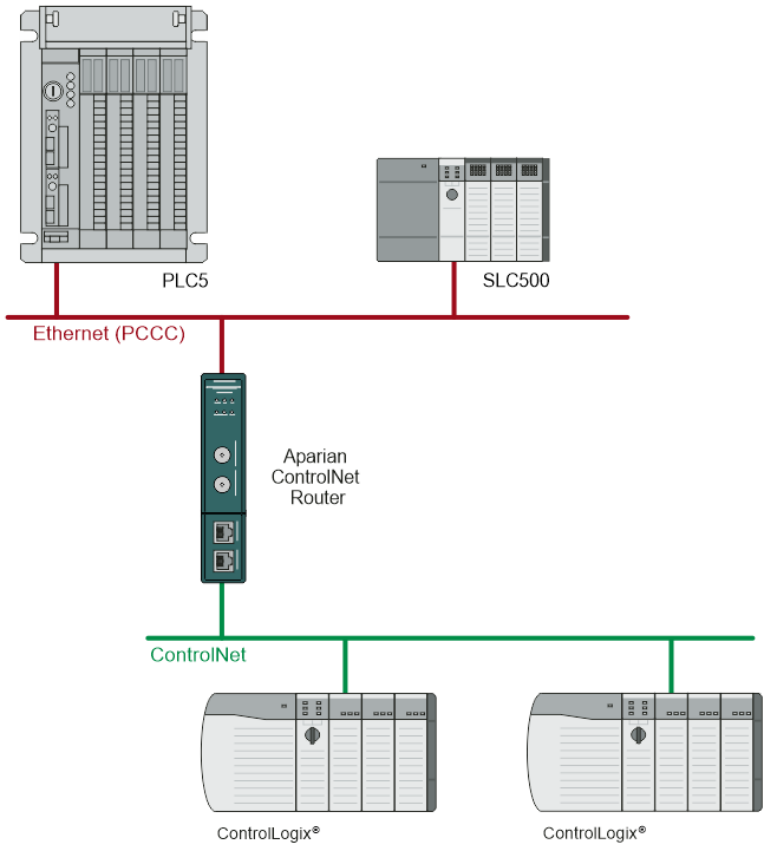


Figure 1.3. - Example of a PLC5/SLC500 Messaging in PLC Emulation mode

1.3.2. EtherNet/IP Target to ControlNet IO (Scheduled ControlNet)

When ControlNet Router is configured as a EtherNet/IP Target and ControlNet Originator, a controller (e.g. Logix Controller) can own the ControlNet Router via EtherNet/IP Class 1 while the ControlNet Router, in turn, owns ControlNet IO using Scheduled ControlNet.

The data from the controller (e.g., Logix Controller) can be exchanged with the ControlNet Router using the EtherNet/IP Class 1 input and output assembly. This data, in turn, can be mapped to any of the configured Scheduled ControlNet IO input and output assemblies being owned by the ControlNet Router.

The ControlNet Router can also exchange data with ControlNet devices using the Explicit Messaging Map. A CIP message with specific service, class, instance, and attribute can be configured to execute at a configured rate with the data being stored in the internal data space (IDS).

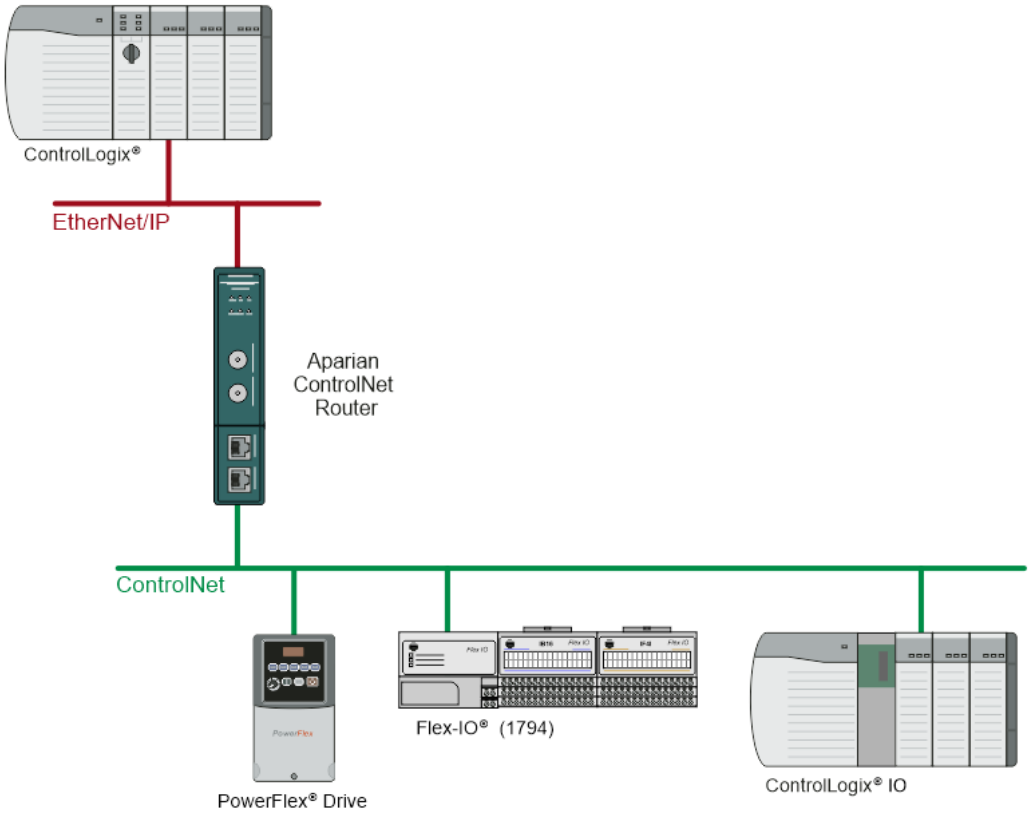


Figure 1.4 – Typical Setup for EtherNet/IP Target to ControlNet IO

1.3.3. Modbus TCP Slave to ControlNet IO (Scheduled ControlNet)

When ControlNet Router is configured as a Modbus TCP Slave and ControlNet Originator, Modbus TCP Master and read and write data to the ControlNet Router internal Modbus Registers while the ControlNet Router owns ControlNet IO using Scheduled ControlNet.

The data from the Modbus TCP Master can be mapped to any of the configured Scheduled ControlNet IO input and output assemblies being owned by the ControlNet Router.

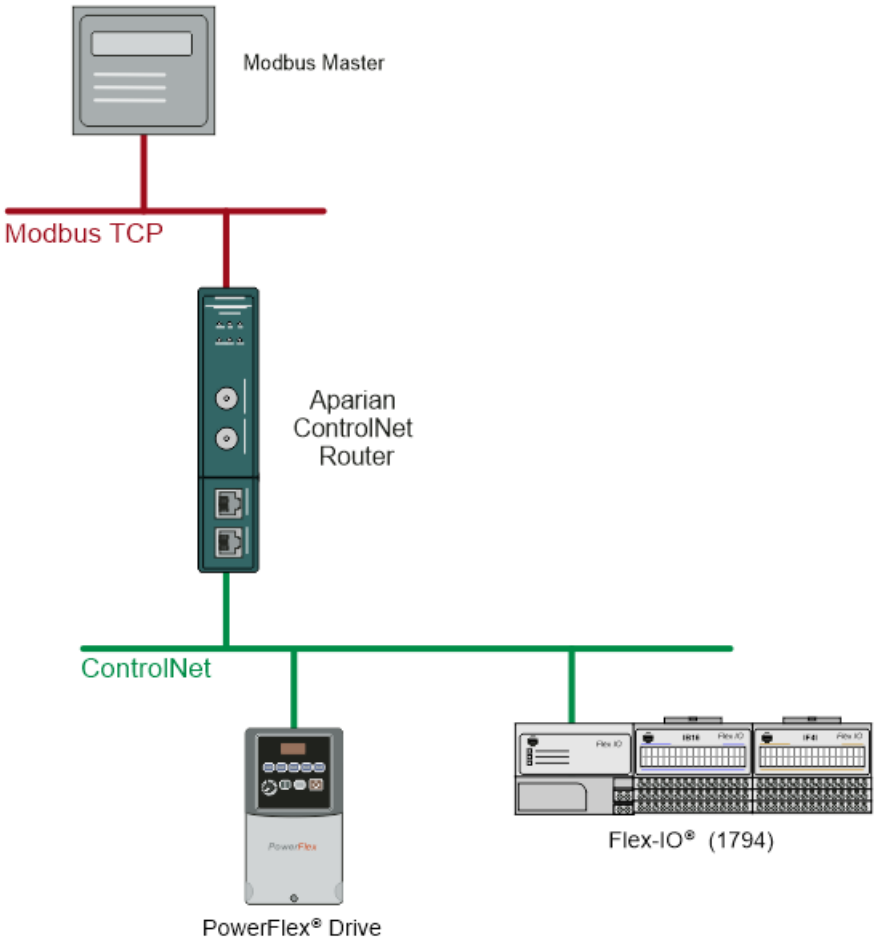


Figure 1.5 – Typical Setup for Modbus TCP Slave to ControlNet IO

1.3.4. ControlNet Target (Scheduled) to EtherNet/IP IO (Class 1)

When ControlNet Router is configured as a ControlNet Target and EtherNet/IP Originator, a controller (e.g. Logix Controller) can own the ControlNet Router via Scheduled ControlNet (e.g. via 1756-CNB/R) while the ControlNet Router owns EtherNet/IP IO using a class 1 connection.

The data from the controller (e.g., Logix Controller) can be exchanged with the ControlNet Router using the Scheduled ControlNet input and output assembly. This data, in turn, can be mapped to any of the configured Class 1 EtherNet/IP IO input and output assemblies being owned by the ControlNet Router.

The ControlNet Router can also exchange data with EtherNet/IP devices using the Explicit Messaging Map. A CIP message with specific service, class, instance, and attribute can be

configured to execute at a configured rate with the data being stored in the internal data space (IDS).

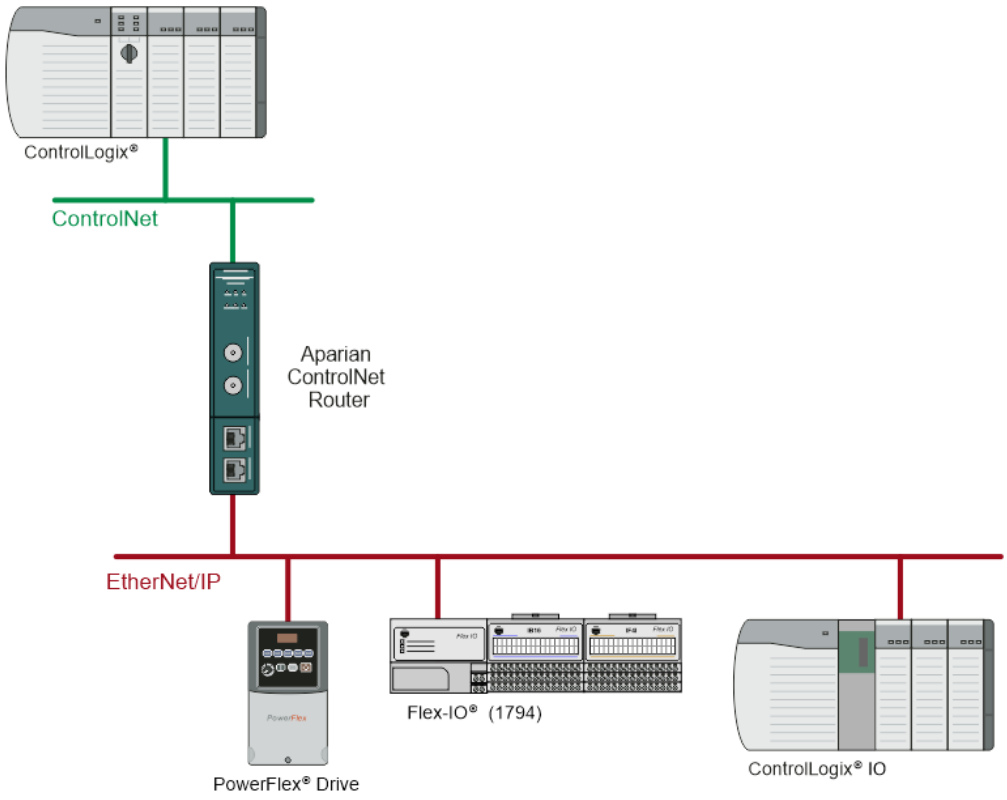


Figure 1.6 – Typical Setup for ControlNet Target to EtherNet/IP IO

1.3.5. CONTROLNET TARGET (SCHEDULED) TO MODBUS TCP SLAVES

When ControlNet Router is configured as a ControlNet Target and Modbus TCP Master, a controller (e.g. Logix Controller) can own the ControlNet Router via Scheduled ControlNet (e.g. via 1756-CNB/R) while the ControlNet Router exchanges data with multiple Modbus TCP Slaves.

The data from the controller (e.g., Logix Controller) can be exchanged with the ControlNet Router using the Scheduled ControlNet input and output assembly. This data, in turn, can be mapped to any of the Modbus TCP Slaves being written to or read from by the ControlNet Router.

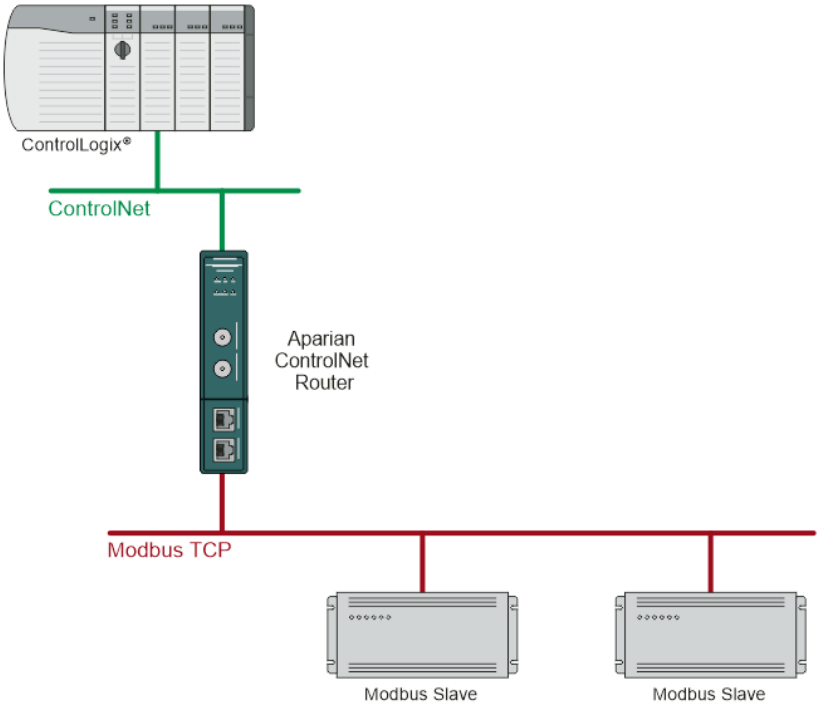


Figure 1.7 – Typical Setup for ControlNet Target to Modbus TCP Slaves

1.4. ADDITIONAL INFORMATION

The following documents contain additional information that can assist the user with the module installation and operation.

Resource	Link
Slate Installation	http://www.aparian.com/software/slate
ControlNet Router User Manual ControlNet Router Datasheet Example Code & UDTs	http://www.aparian.com/products/controlnetrouter
Ethernet wiring standard	www.cisco.com/c/en/us/td/docs/video/cds/cde/cde205_220_420/installation/guide/cde205_220_420_hig/Connectors.html
CIP Routing	The CIP Networks Library, Volume 1, Appendix C:Data Management
ControlNet	http://www.odva.org
ControlNet Cabling	ControlNet Coax Media Planning and Installation Guide https://literature.rockwellautomation.com/idc/groups/literature/documents/in/cnet-in002_-en-p.pdf

Table 1.1. - Additional Information

1.5. SUPPORT

Technical support is provided via the Web (in the form of user manuals, FAQ, datasheets etc.) to assist with installation, operation, and diagnostics.
For additional support the user can use either of the following:

Resource	Link
Contact Us web link	https://www.prosoft-technology.com/Services-Support/Customer-Support
Support email	support@prosoft-technology.com

Table 1.2. – Support Details

2. INSTALLATION

2.1. MODULE LAYOUT

The module has one 3-way power connector at the bottom of the enclosure as shown in the figure below.

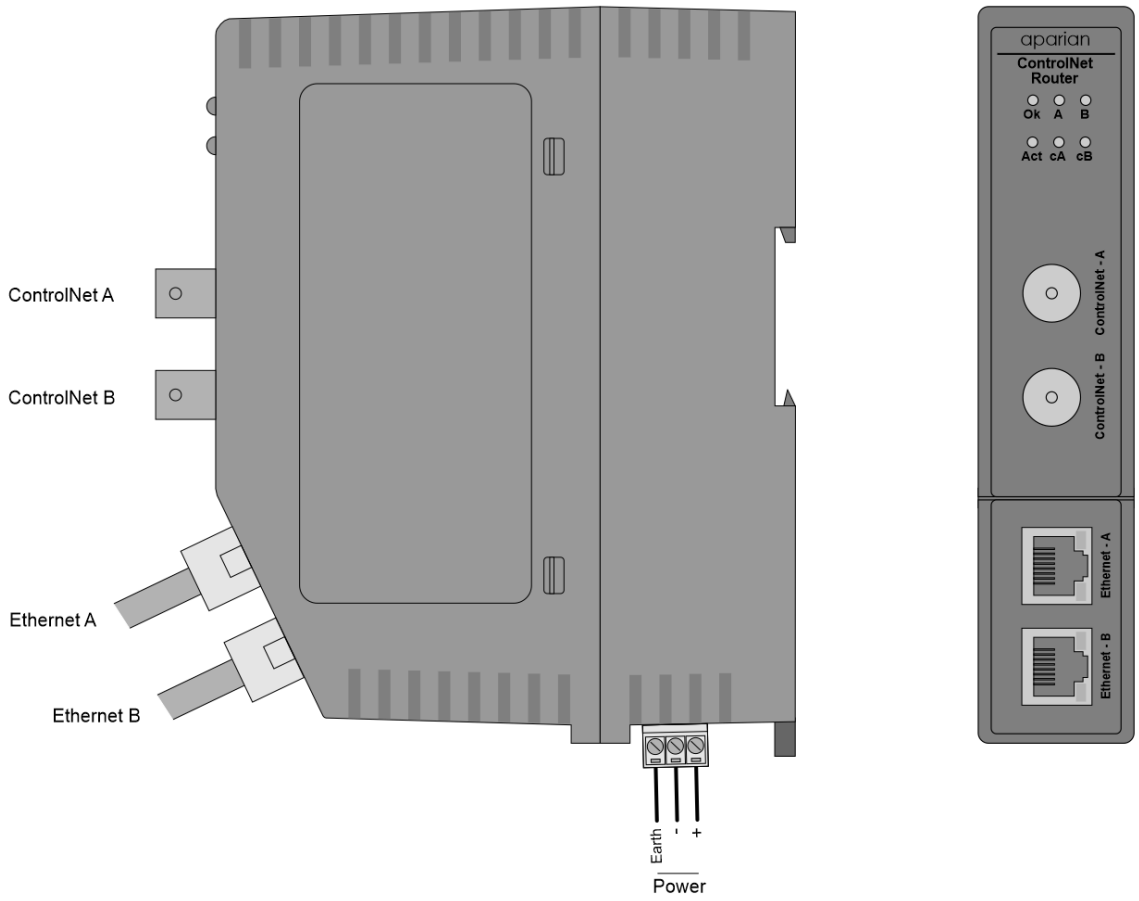


Figure 2.1. – ControlNet Router side and front view

There are two ControlNet BNC connectors in the front of the module for redundant ControlNet. There are two Ethernet RJ45 connectors on the front angle of the module which is used for the Ethernet connection. The Ethernet cable must be wired according to industry standards which can be found in the additional information section of this document.

The module provides six diagnostic LEDs on the front of the module. These LEDs are used to provide information regarding the module system operation, the Ethernet interface, and the ControlNet interface.

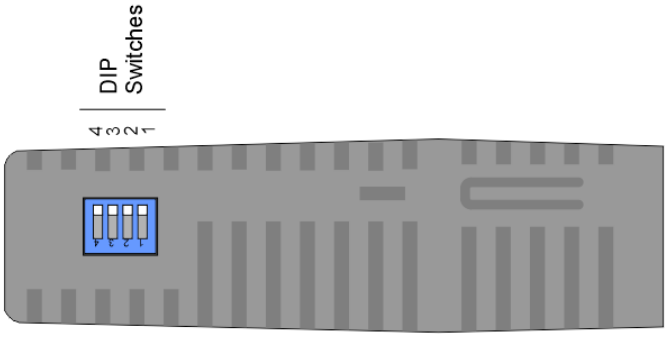


Figure 2.2 – ControlNet Router top view

The module provides four DIP switches at the top of the enclosure as shown in the top view figure above.

DIP Switch	Description
DIP Switch 1	Used to force the module into “Safe Mode”. When in “Safe Mode” the module will not load the application firmware and will wait for new firmware to be downloaded. This should only be used in the rare occasion when a firmware update was interrupted at a critical stage.
DIP Switch 2	This will force the module into DHCP mode which is useful when the user has forgotten the IP address of the module.
DIP Switch 3	This DIP Switch is used to lock the configuration from being overwritten by the Slate. When set Slate will not be able to download to the module.
DIP Switch 4	When this DIP Switch is set at bootup it will force the module Ethernet IP address to **192.168.1.100** and network mask 255.255.255.0. The user can then switch the DIP switch off and assign the module a static IP address if needed.

Table 2.1. - DIP Switch Settings

2.2. MODULE MOUNTING

NOTE: This module is an open-type device and is meant to be installed in an enclosure suitable for the environment such that the equipment is only accessible with the use of a tool.

The module provides a DIN rail clip to mount onto a 35mm DIN rail.

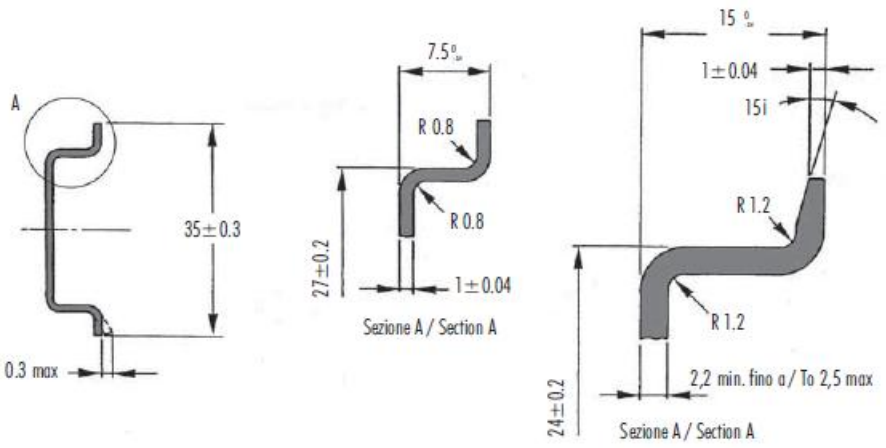


Figure 2.3 - DIN rail specification

The DIN rail clip is mounted on the bottom of the module at the back as shown in the figure below. Use a flat screwdriver to pull the clip downward. This will enable the user to mount the module onto the DIN rail. Once the module is mounted onto the DIN rail the clip must be pushed upwards to lock the module onto the DIN rail.

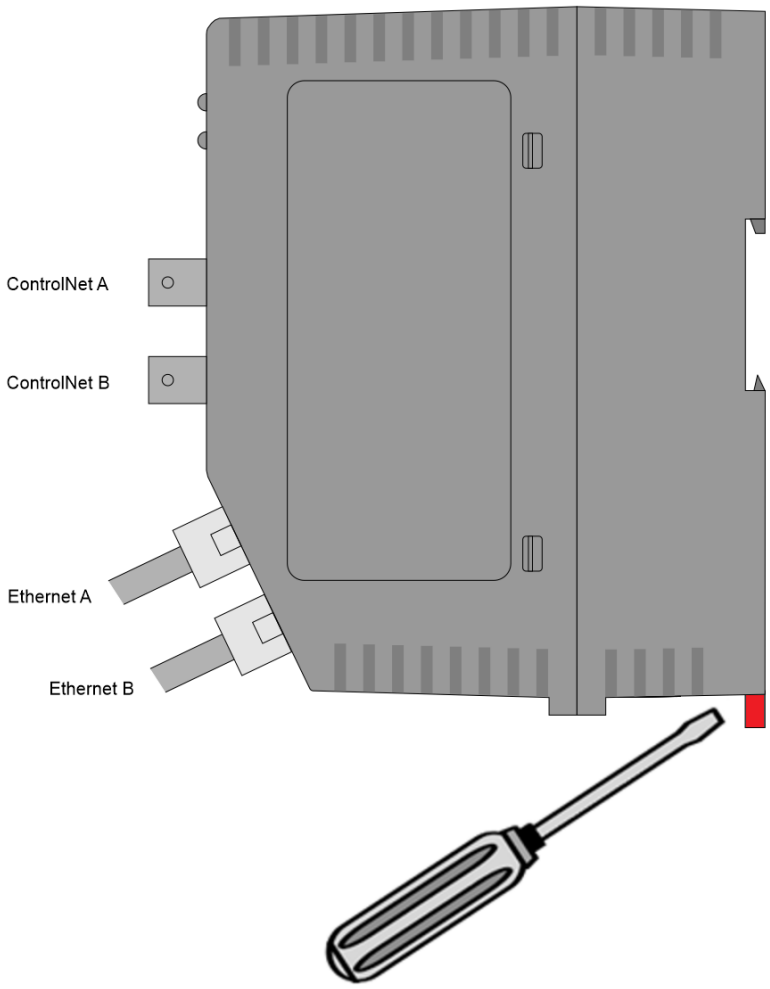


Figure 2.4 - DIN rail mouting

2.3. POWER

A 3-way power connector is used to connect Power+, Power– (GND), and earth. The module requires an input voltage of 10 – 36Vdc. **Refer** to the technical specifications section in this document.

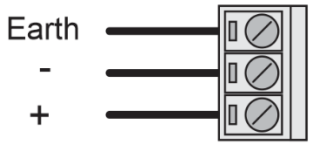


Figure 2.5 - Power connector

2.4. CONTROLNET

ControlNet uses a coax media system. The ControlNet connectors should be wired according to industry standards. **Refer** to the additional information section in this document for further details.

2.5. ETHERNET PORT

The Ethernet connector should be wired according to industry standards. **Refer** to the additional information section in this document for further details.

2.6. NORTH AMERICAN HAZARDOUS LOCATION APPROVAL

The following information applies when operating the product in hazardous locations:

SUITABLE FOR USE IN CLASS I, DIVISION 2, GROUPS A, B, C AND D HAZARDOUS LOCATIONS, OR NONHAZARDOUS LOCATIONS ONLY.

-20°C to +70°C, T5A
Class 1, Div 2 Gps A, B, C, D

THIS EQUIPMENT IS OPEN-TYPE DEVICE AND IS MEANT TO BE INSTALLED IN AN ENCLOSURE SUITABLE FOR THE ENVIRONMENT SUCH THAT THE EQUIPMENT IS ONLY ACCESSIBLE WITH THE USE OF A TOOL.

WARNING - EXPLOSION HAZARD - DO NOT DISCONNECT EQUIPMENT WHILE THE CIRCUIT IS

LIVE OR UNLESS THE AREA IS KNOW TO BE FREE OF IGNITABLE CONCENTRATIONS.

WARNING - EXPLOSION HAZARD - SUBSTITUTION OF ANY COMPONENT MAY IMPAIR SUITABILITY FOR CLASS I, DIVISION 2.

3. SETUP

3.1. INSTALL CONFIGURATION SOFTWARE

All the network setup and configuration of the module is achieved by means of the Aparian Slate device configuration environment. This software can be downloaded from http://www.aparian.com/software/slate.

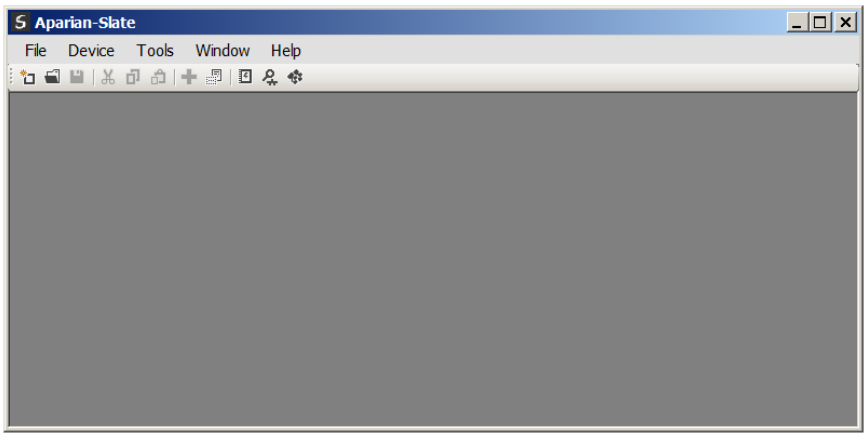


Figure 3.1. - Aparian Slate Environment

3.2. NETWORK PARAMETERS

The module will have DHCP (Dynamic Host Configuration Protocol) enabled as factory default. Thus, a DHCP server must be used to provide the module with the required network parameters (IP address, subnet mask, etc.). There are a number of DHCP utilities available, however it is recommended that the DHCP server in Slate be used. Within the Slate environment, the DHCP server can be found under the Tools menu.

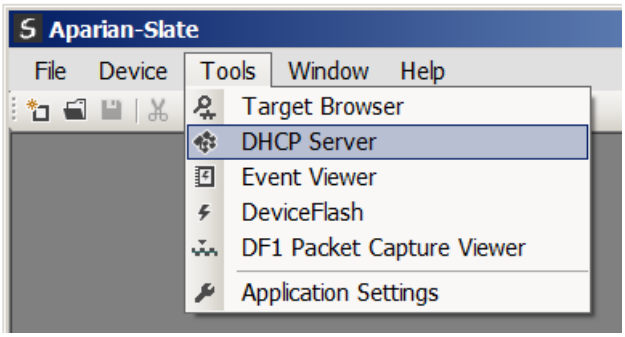


Figure 3.2. - Selecting DHCP Server

Once opened, the DHCP server will listen on all available network adapters for DHCP requests and display their corresponding MAC addresses.

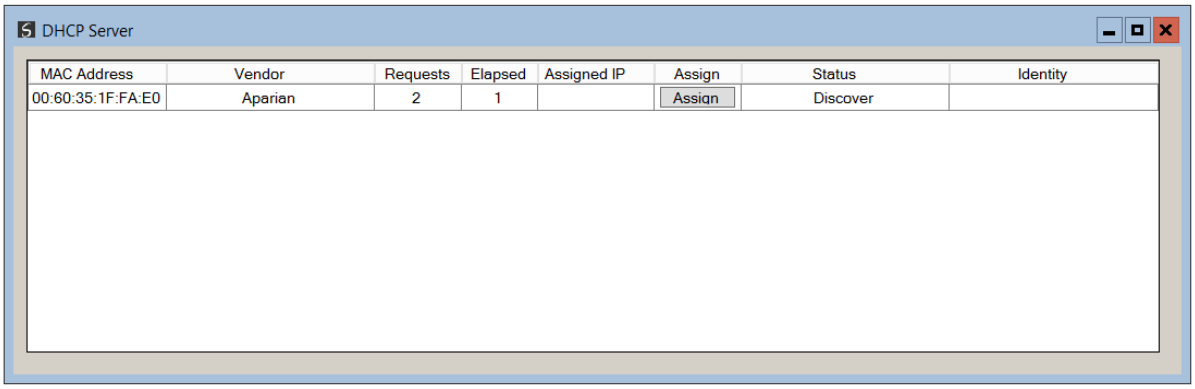


Figure 3.3. - DHCP Server

NOTE: If the DHCP requests are not displayed in the DHCP Server it may be due to the local PC's firewall. During installation the necessary firewall rules are automatically created for the Windows firewall. Another possibility is that another DHCP Server is operational on the network and it has assigned the IP address.

To assign an IP address, click on the corresponding "Assign" button. The IP Address Assignment window will open.

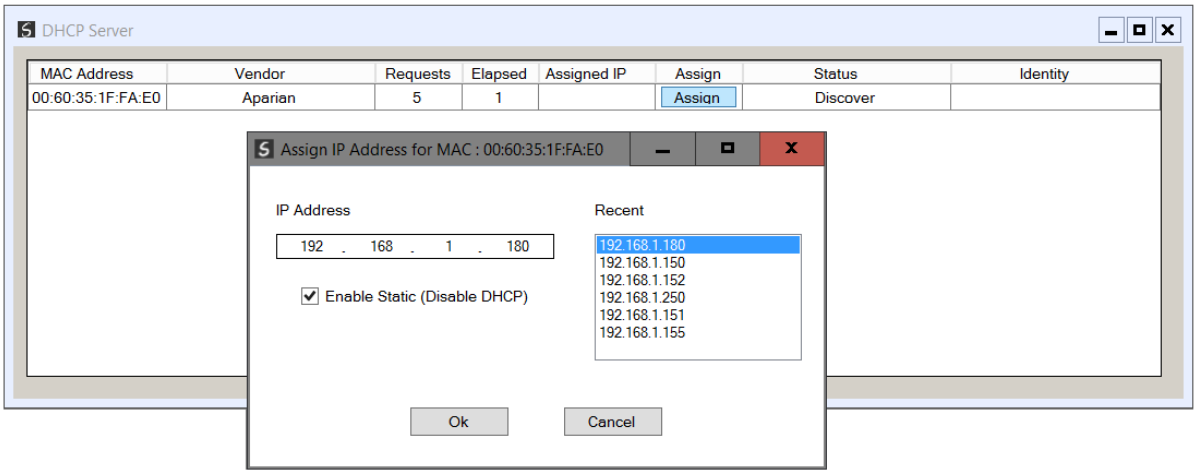


Figure 3.4. - Assigning IP Address

The required IP address can then be either entered, or a recently used IP address can be selected by clicking on an item in the Recent List. If the "Enable Static" checkbox is checked, then the IP address will be set to static after the IP assignment, thereby disabling future DHCP requests.

Once the IP address window has been accepted, the DHCP server will automatically assign the IP address to the module and then read the Identity object Product name from the device.
The successful assignment of the IP address by the device is indicated by the green

background of the associated row.

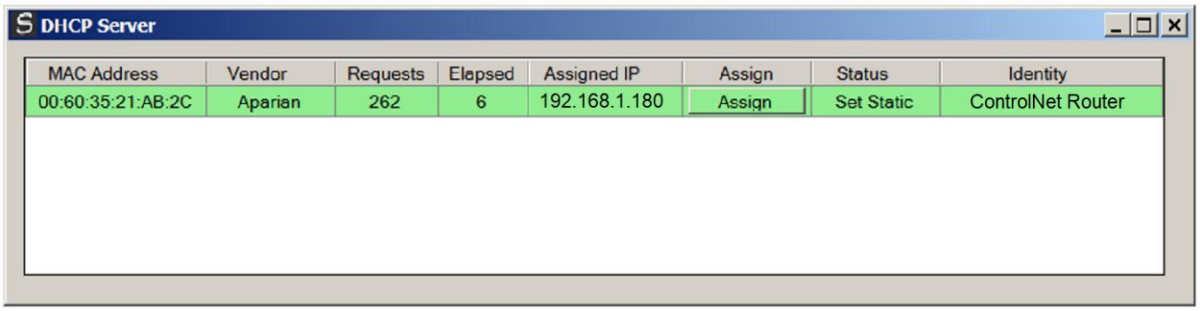


Figure 3.5. - Successful IP address assignment

It is possible to force the module back into DHCP mode by powering up the device with DIP switch 2 set to the On position. A new IP address can then be assigned by repeating the previous steps.

NOTE: It is important to return DIP switch 2 back to Off position, to avoid the module returning to a DHCP mode after the power is cycled again.

If the module's DIP switch 2 is in the On position during the address assignment, the user will be warned by the following message.



Figure 3.6. - Force DHCP warning

In addition to the setting the IP address, a number of other network parameters can be set during the DHCP process. These settings can be viewed and edited in Slate's Application Settings, in the DHCP Server tab. Once the DHCP process has been completed, the network settings can be set using the Ethernet Port Configuration via the Target Browser. The Target Browser can be accessed under the Tools menu.

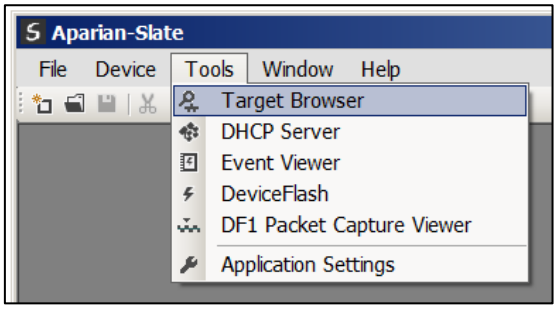


Figure 3.7. - Selecting the Target Browser

The ***Target Browser*** automatically scans the Ethernet network for EtherNet/IP devices.

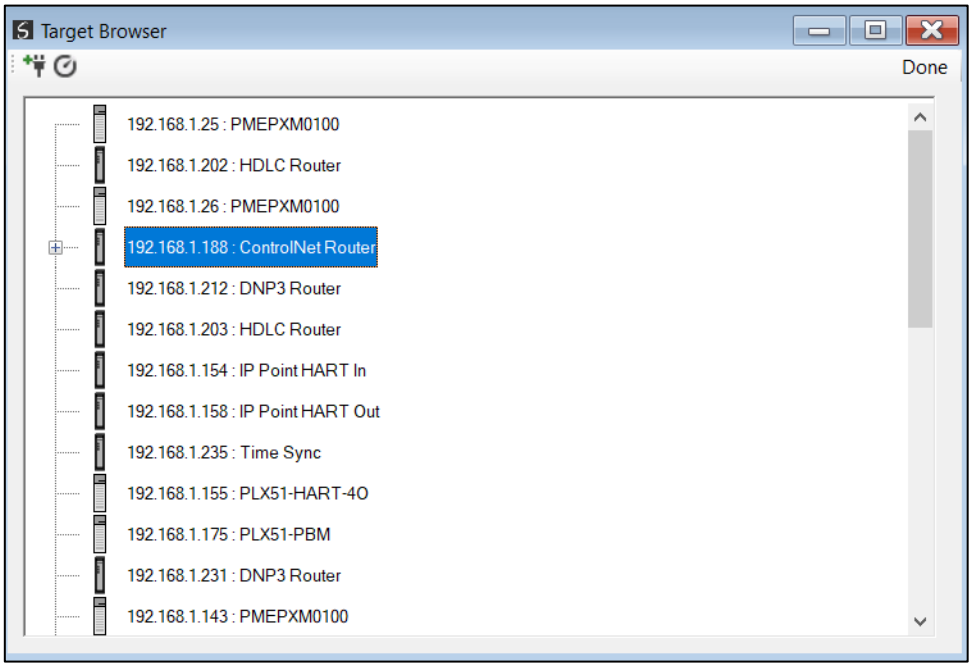


Figure 3.8. - Target Browser

Right-clicking on a device, reveals the context menu, including the ***Port Configuration*** option.

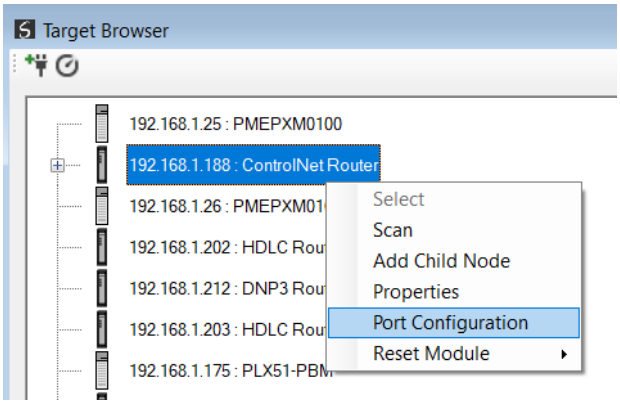


Figure 3.9. - Selecting Port Configuration

All the relevant Ethernet port configuration parameters can be modified using the ***Port Configuration*** window.

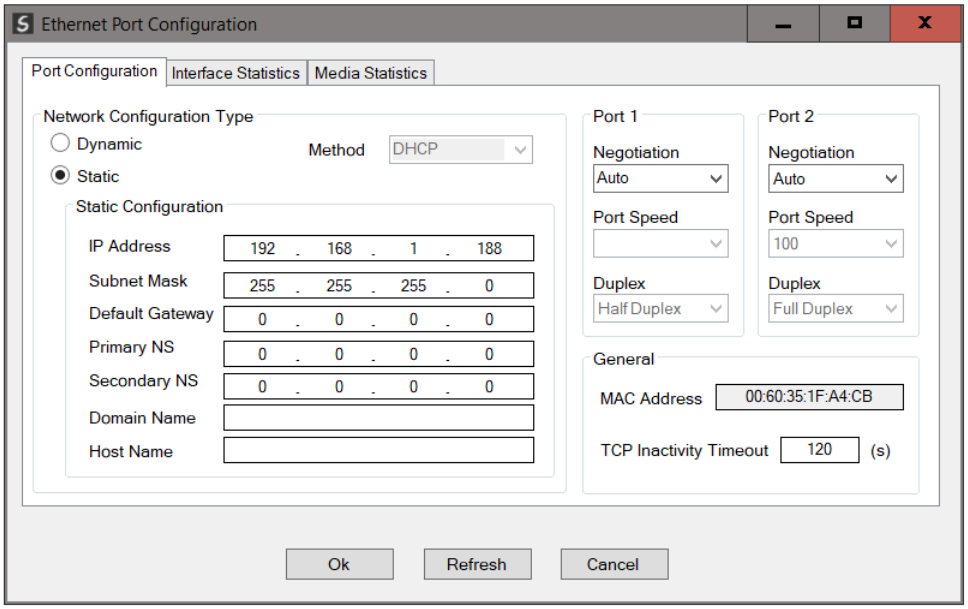


Figure 3.10. - Port Configuration

Alternatively, these parameters can be modified using Rockwell Automation's RSLinx software.

3.3. CREATING A NEW PROJECT

Before the user can configure the module, a new Slate project must be created. Under the ***File*** menu, select ***New***.

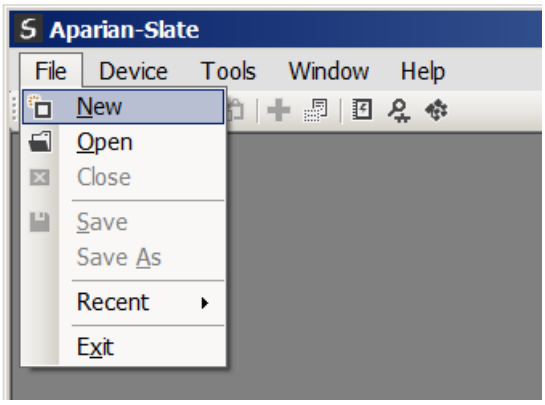


Figure 3.11. - Creating a new project

A Slate project will be created, showing the Project Explorer tree view. To save the project use the ***Save*** option under the ***File*** menu. A new device can now be added by selecting ***Add*** option under the ***Device*** menu.

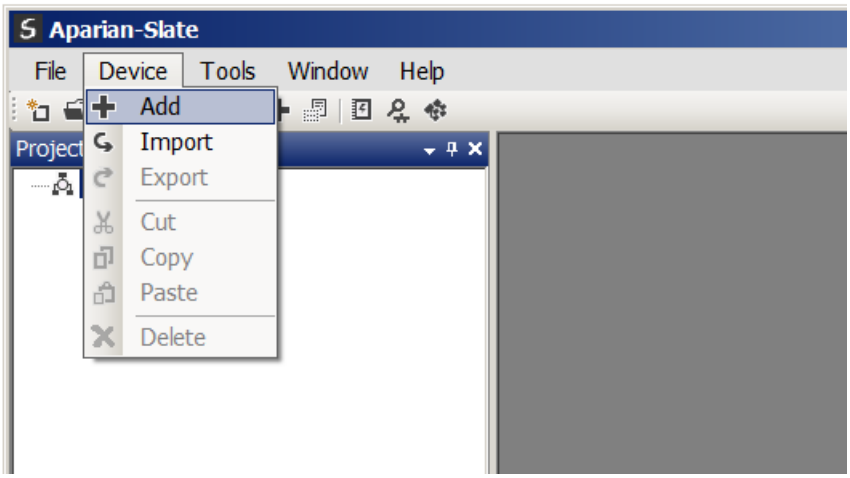


Figure 3.12. - Adding a new device

In the Add New Device window select the ***ControlNet Router***, and click the ***Ok*** button.

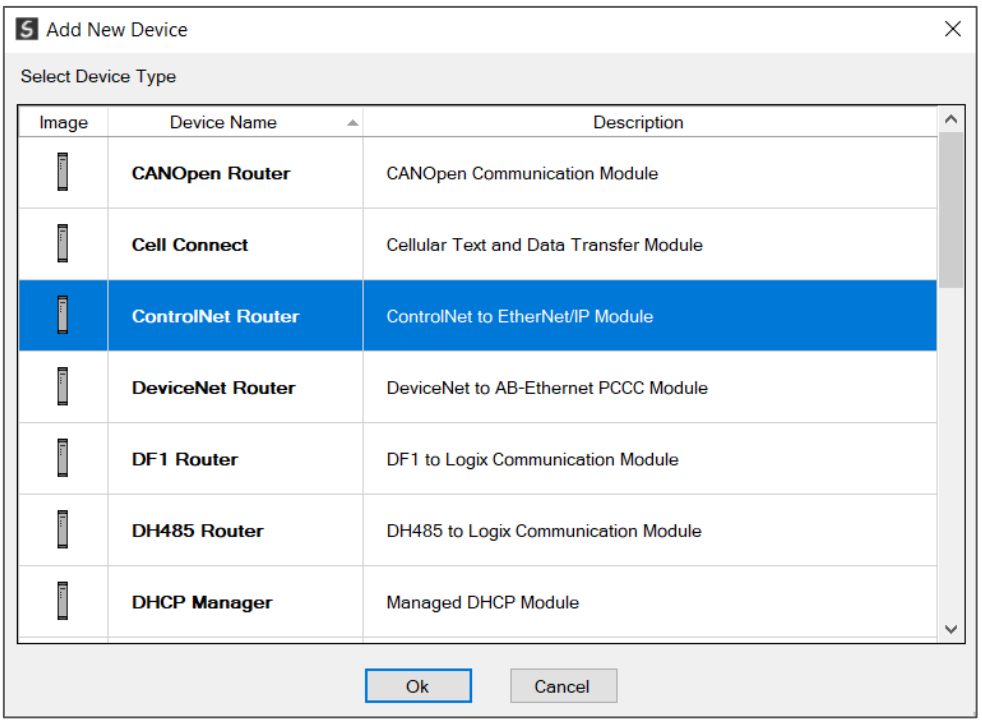


Figure 3.13 – Selecting a new ControlNet Router

The software will provide the user with the option to select either the *Legacy* or *Enhanced* ControlNet Router. The ***ControlNet Router Enhanced*** must be selected.

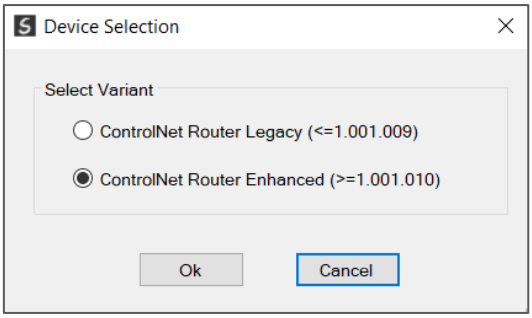


Figure 3.14 – Selecting a Legacy or Enhanced ControlNet Router

The device will appear in the Project Explorer tree as shown below, and its configuration window opened. The device configuration window can be reopened by either double clicking the module in the Project Explorer tree or right clicking the module and selecting ***Configuration***.

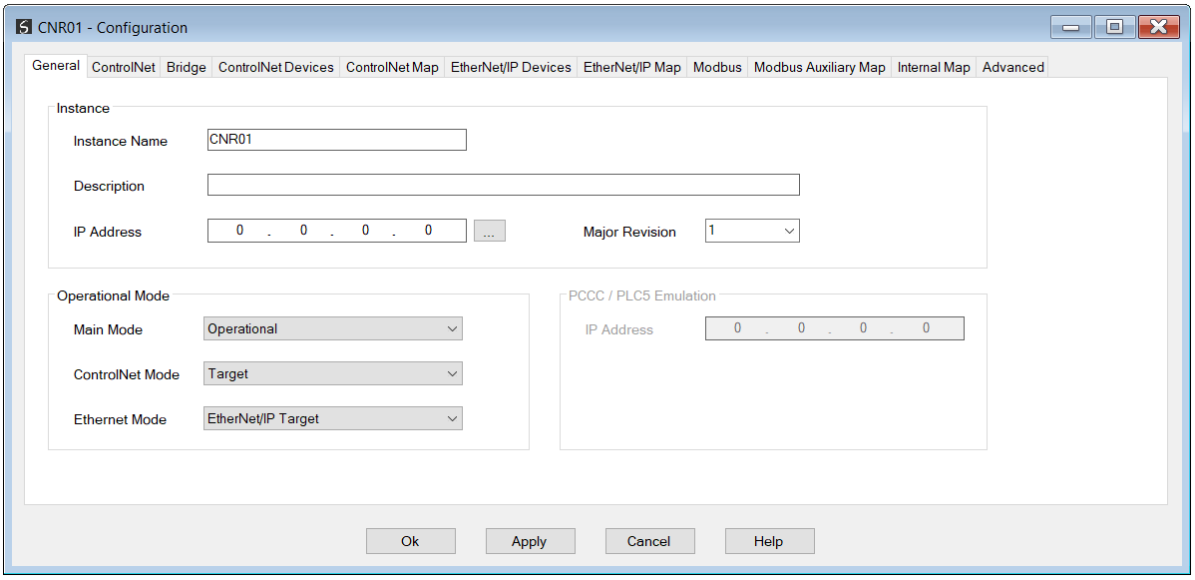


Figure 3.15 - ControlNet Router configuration

3.4. GENERAL PARAMETERS

The ControlNet parameters will be configured by Slate. When downloading this configuration into the module it will be saved in non-volatile memory that persists when the module is powered down.

NOTE: When a firmware upgrade is performed the module will clear all the module configuration stored in non-volatile storage.

The general configuration is shown in the figure below. The ControlNet general configuration window is opened by either double clicking on the module in the tree or right-clicking the module and selecting ***Configuration***.

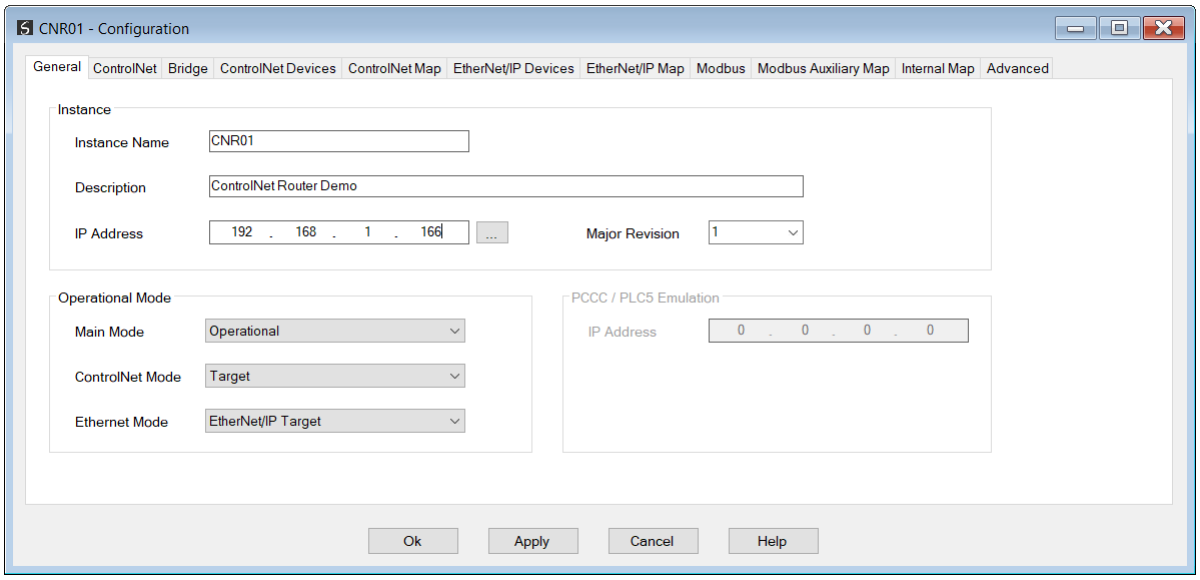


Figure 3.16 - General Configuration

The general configuration consists of the following parameters:

Parameter	Description
Instance Name	This parameter is a user defined name to identify between various ControlNet Routers.
Description	This parameter is used to provide a more detailed description of the application for the module.
Major Revision	The major revision of the module
IP Address	The IP address of the target module. The user can use the target browse button to launch the target browser to the select the ControlNet Router on the network.
Main Mode	**Operational** The module will operate as a ControlNet gateway exchanging data between the configured Ethernet network and ControlNet network. **Promiscuous Capture** In this mode the module will operate as a ControlNet network analyser by capturing all data being transmitted on the ControlNet network.
ControlNet Mode	**Target** A Logix controller can own the ControlNet Router over ControlNet using a scheduled ControlNet connection (e.g. via a 1756-CNB/R) **Originator** As a ControlNet originator, the module can use two methods to exchange data with one or more ControlNet devices: *Scheduled ControlNet* The ControlNet Router can own ControlNet IO by using the Slate software to configure the IO connections and schedule the ControlNet network. *ControlNet Explicit Messaging*

	This ControlNet Router can exchange data with up to 10 ControlNet devices using explicit messaging over ControlNet. **Bridge** The ControlNet Router owns ControlNet IO over a scheduled ControlNet connection, by using the A-CNR AOP in Logix. The ControlNet IO will be added and configured in Logix while the ControlNet Router will dynamically schedule the ControlNet network as connections are established.
Ethernet Mode	**PCCC Client** This will allow the module to emulate a PLC5 or SLC, providing a legacy interface for PanelViews and other devices over scheduled ControlNet. **EtherNet/IP Target** A Logix controller can own the ControlNet Router over EtherNet/IP using a class 1 connection. **EtherNet/IP Originator** As an EtherNet/IP originator, the module can use two methods to read and write data to and from EtherNet/IP IO: *EtherNet/IP Class 1 Connection* The ControlNet Router can own EtherNet/IP IO by using the Slate software to configure the IO connections. *EtherNet/IP Explicit Messaging* The ControlNet Router can exchange data with up to 10 EtherNet/IP devices using explicit messaging. **Modbus TCP Slave** A Modbus TCP Master can read and write data to the module which can then be mapped to one or more ControlNet devices. **Modbus TCP Master** A module can read and write data from various Modbus TCP devices which can then be mapped to one or more ControlNet devices.
PLC5 Emulation IP Address	The IP Address used to emulate a PLC5 controller. This IP address will be seen as the IP address of the PLC5 controller that the ControlNet Router is emulating. Therefore, there will be two IP addresses on the network when in PLC5 Emulation mode. One for the actual target module and one for the emulated PLC5 controller. **NOTE:** These two IP addresses must **not** be the same.

Table 3.1 - General configuration parameters

3.5. CONTROLNET CONFIGURATION

The ControlNet configuration is shown in the figure below. The ControlNet configuration window is opened by either double clicking on the module in the tree, or right-clicking the module and selecting ***Configuration***.

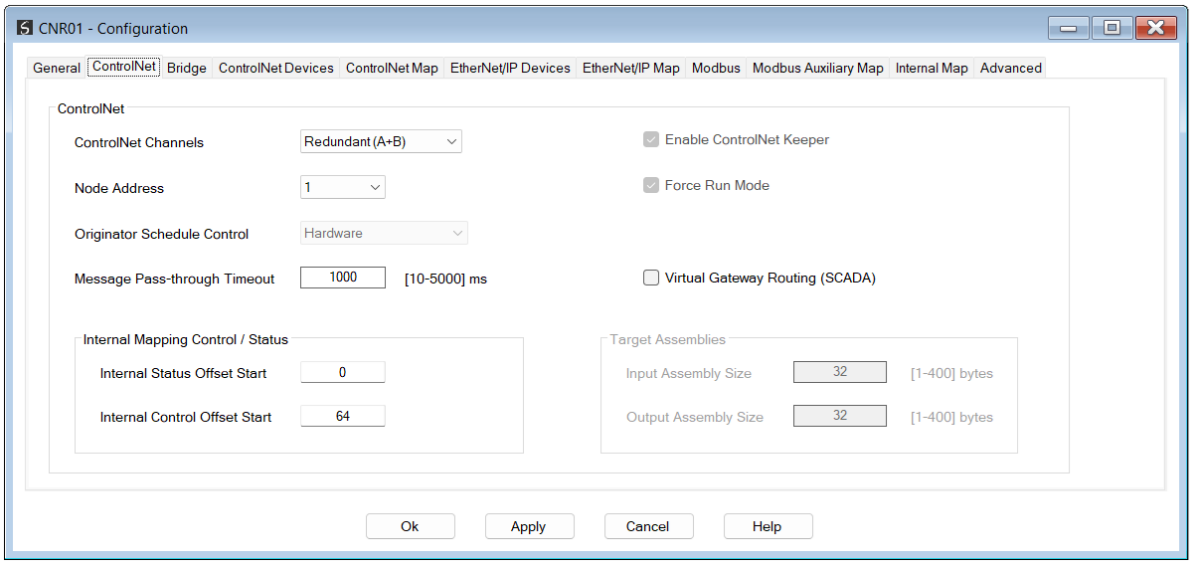


Figure 3.17 - General Configuration

The ControlNet configuration consists of the following parameters:

Parameter	Description
ControlNet Channels	The ControlNet Router module has two ControlNet ports and supports redundant ControlNet operation. The ControlNet Channels can be selected in this parameter to communicate on Channel A Only, Channel B Only, or A+B Redundant.
Node Address	This is the node address of the ControlNet Router on the ControlNet network.
Originator Schedule Control	This option will become available when the ControlNet mode has been set to *Originator*. The Originator Schedule Control allows the user to select between Hardware and Software control. **Hardware** Provides hardware triggers for scheduled packet sending (most accurate), but can only allow up to 10 scheduled connections with limited data being sent from the A-CNR to the ControlNet IO device (see the Specification Appendix for more information). **Software** Provides software triggers for scheduled packet sending, but can have up to 30 scheduled connections, each with the maximum input and output packet size.
Message Pass-Through Timeout	This is the ControlNet message pass-through timeout which specifies how long the ControlNet Router will wait for a response from a ControlNet node before the response will be seen as a failed no-response.
Internal Status Offset Start	The internal status information of the module can be saved at a specific Internal Data Space (IDS) address. This parameter is the start address for the module status in the IDS.
Internal Control Offset Start	The internal control of the module can be obtained from a specific Internal Data Space (IDS) address. This parameter is the start address for the module control in the IDS.
Enable ControlNet	When this parameter is set **and** valid Keeper parameters have been downloaded to

Keeper	the module during the ControlNet network scheduling in Slate, then the module will operate as the ControlNet Keeper if it has the lowest address of the active Keepers on the network. **NOTE: If an existing Control Keeper is on the ControlNet network, it might be required to disable the Keeper on that ControlNet module. This is done by resetting the existing Keeper to factory defaults using the Slate Target Browser.**
Force Run Mode	When this parameter is set, it will force all the originator connections (where the module owns the IO) to have the connection operation set to **Run**. If this parameter is not set, then the user will need to write to the IDS control (see the parameter in the table) to set the originator connection to **Run** or **Prog** mode.
Virtual Gateway Routing (SCADA)	When enabled, the ControlNet Router emulates an intermediate Logix chassis (Backplane + ControlNet bridge) by consuming (removing) a superfluous [Port + Address] segment in the routed CIP path, and adjusting the ControlNet port from 2 to 3. This may be required by some 3rd party HMI/SCADA drivers where any manual (free form) CIP path cannot be entered. For example, the routed path: 192.168.1.28, 1, 0, **2**, 6, 1, 0 will effectively be reduced to: 192.168.1.28, **3**, 6, 1, 0 **NOTE**: This option is applicable only in **Bridge** mode.
Input Assembly Size	When using Scheduled ControlNet this is the size of the input data (being read from the ControlNet Router) that can be mapped for the PLC5 emulation. See the *Scheduled ControlNet* section.
Output Assembly Size	When using Scheduled ControlNet this is the size of the output data (being sent from the ControlNet Router) that can be mapped for the PLC5 emulation. See the *Scheduled ControlNet* section.

Table 3.2 - General configuration parameters

The module can operate as either a ControlNet Target or a ControlNet Originator. The below sections will provide more information regarding these operational modes.

3.5.1. TARGET

A Logix controller can own the ControlNet Router over ControlNet using a scheduled ControlNet connection when the ControlNet Router is operating as a ControlNet target. This will allow the ControlNet Router to exchange data with the Logix controller using the input and output assembly of the scheduled ControlNet connection. Data from EtherNet/IP or Modbus TCP devices can be mapped to the Logix controller over ControlNet.

The user will need to add the ControlNet Router to the Logix IO tree under a ControlNet bridge (e.g. 1756-CNB/R). After the module has been added, the ControlNet connection will need to be scheduled with RSNetworx for ControlNet.

3.5.1.1. STUDIO / LOGIX 5000 CONFIGURATION

A. ADD MODULE TO CONTROLNET I/O CONFIGURATION (USING SCHEDULED CONTROLNET)

The ControlNet Router can be added to the Logix 5000 I/O tree under a ControlNet bridge (e.g. 1756-CNB). The module will need to be added as a generic ControlNet module. This is done by right clicking on the ControlNet Bridge in the Logix 5000 and selecting *New Module* after which the *CONTROLNET-MODULE* is selected to be added as shown in the figure below.

NOTE: See the next section for importing the configuration (L5X).

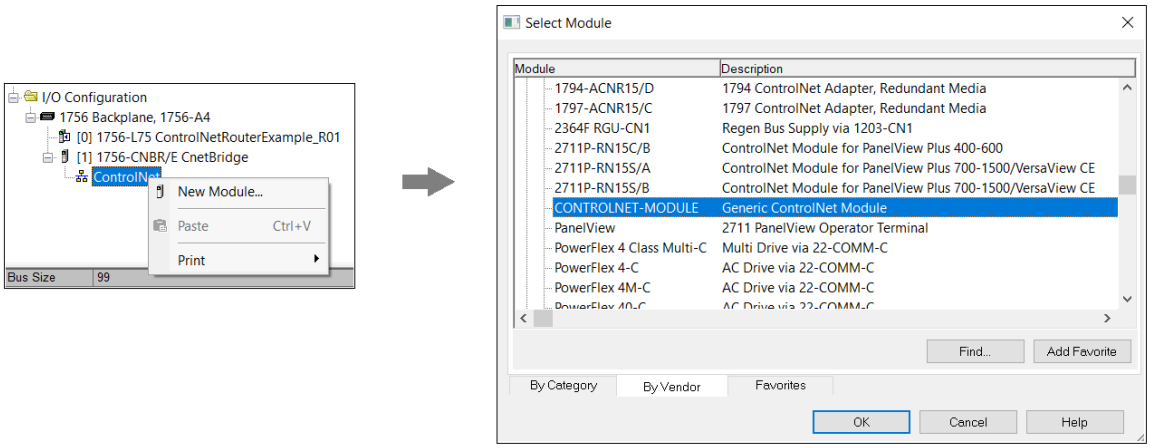


Figure 3.18 - Add a Generic ControlNet Module in Logix 5000

The user must enter the ControlNet node number of the ControlNet Router that will be used. The assembly instance and size must also be added for the input, output, and configuration in the connection parameters section. Below are the required connection parameters.

NOTE: The input and output assembly connection sizes will be dynamic and based on the sizes of the Mapped Data for the input and output (as shown below).

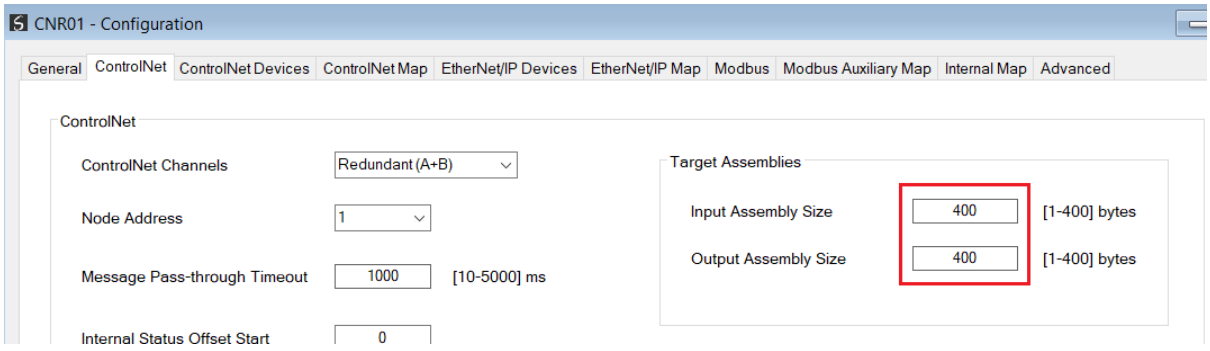


Figure 3.19 – Mapped data for input and output

Connection Parameter	Assembly Instance	Size
Input	130	8 + Input Mapped Data Size (8-bit) *For example: if the input mapped data is 400 then the input assembly size will be 408.*
Output	131	4 + Output Mapped Data Size (8-bit) *For example: if the output mapped data is 400 then the output assembly size will be 404.*
Configuration	102	0 (8-bit)

Table 3.3 – ControlNet Logix class 1 connection parameters for the ControlNet Router

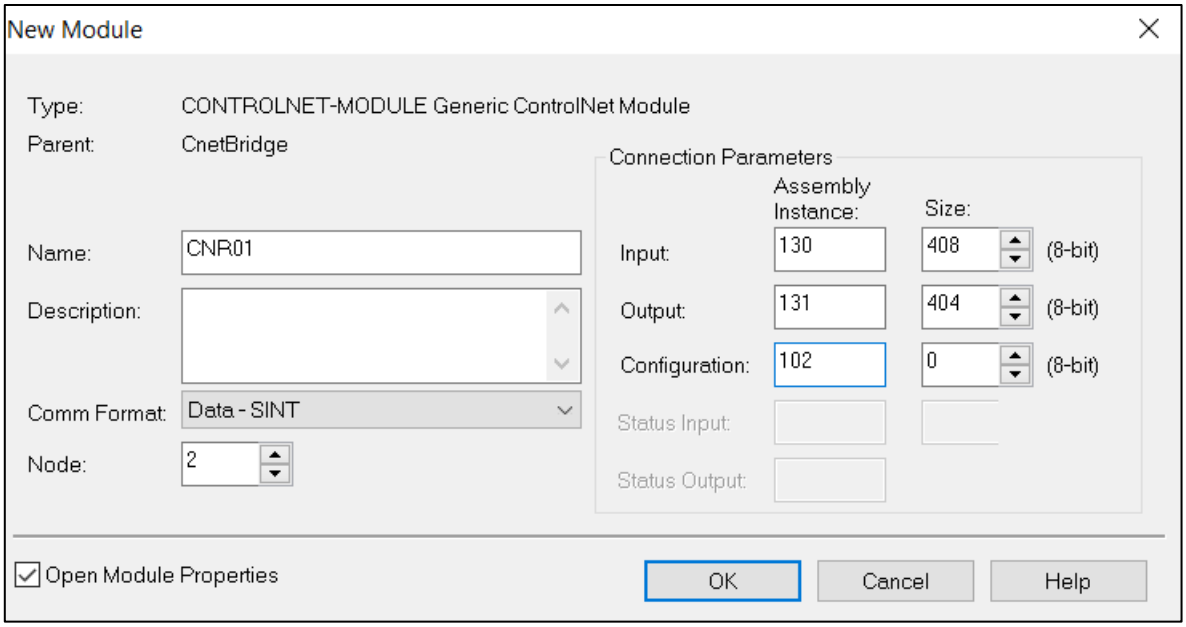


Figure 3.20 - Logix General module properties in Logix 5000

NOTE: The user will need to enter the exact connection parameters before the module will establish a class 1 connection with the Logix controller.

Next the user needs to configure the connection requested packet interval (RPI). This is the rate at which the input and output assemblies are exchanged. Refer to the technical specification section in this document for further details on the limits of the RPI.

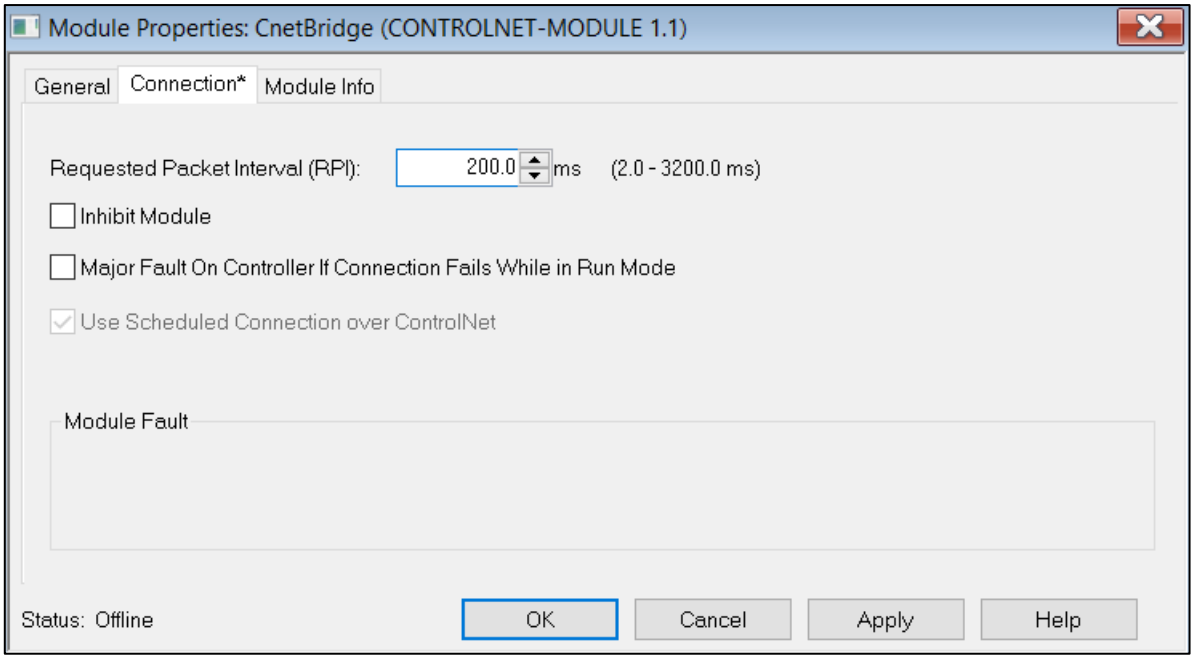


Figure 3.21 - Connection module properties in Logix 5000

Once the module has been added to the Logix 5000 I/O tree the user must assign the User Defined Types (UDTs) to the input and output assemblies. The user can import the required UDTs by right-clicking on *User-Defined* sub-folder in the *Data Types* folder of the IO tree and selecting *Import Data Type*. The assemblies are then assigned to the UDTs with a ladder copy instruction (COP) as shown in the figure below.

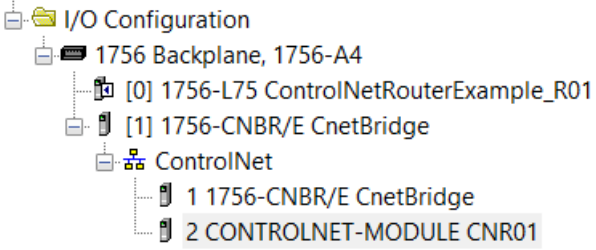


Figure 3.22 – Logix 5000 I/O module tree

NOTE: At this stage the module will not connect to the Logix controller because the connection has not been scheduled. See the *RSNetworx Configuration* section for more details on how to schedule the network.

B. IMPORTING UDTS AND MAPPING ROUTINES

To simplify the mapping of the input image, a Logix 5000 Routine Partial Import (L5X) file is provided. This file can be imported by right-clicking on the required Program and selecting the Import Routine option.

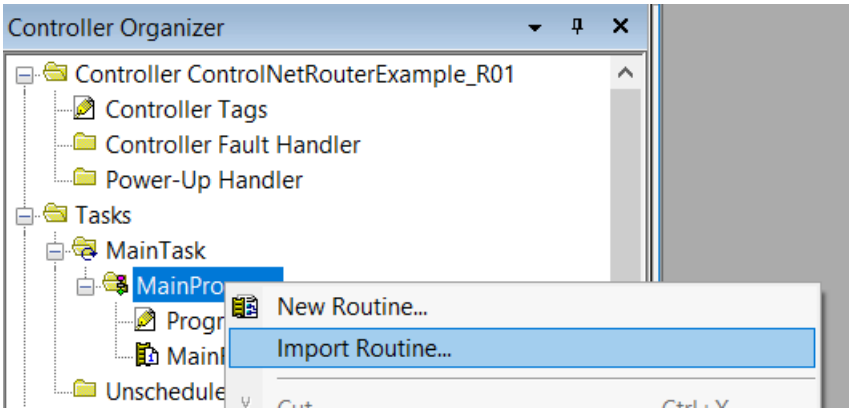


Figure 3.23 – Logix 5000 Importing ControlNet Router specific routine and UDTs

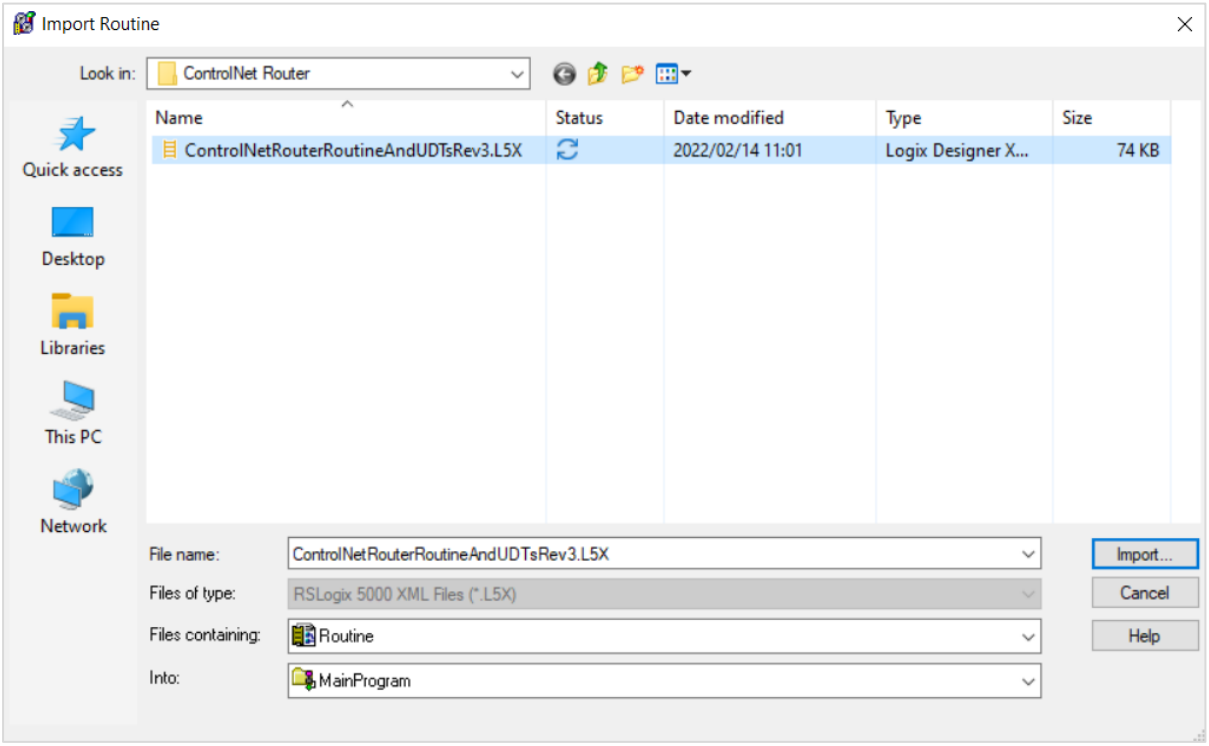


Figure 3.24 - Selecting partial import file

The import will create the following:

- The required UDTs (user defined data types)
- Two controller tags representing the Input and Output assemblies.
- A routine mapping the ControlNet Router module to the aforementioned tags.

The user may need to change the routine to map to the correct ControlNet Router module instance name, and make sure that the mapping routine is called by the Program's Main Routine.

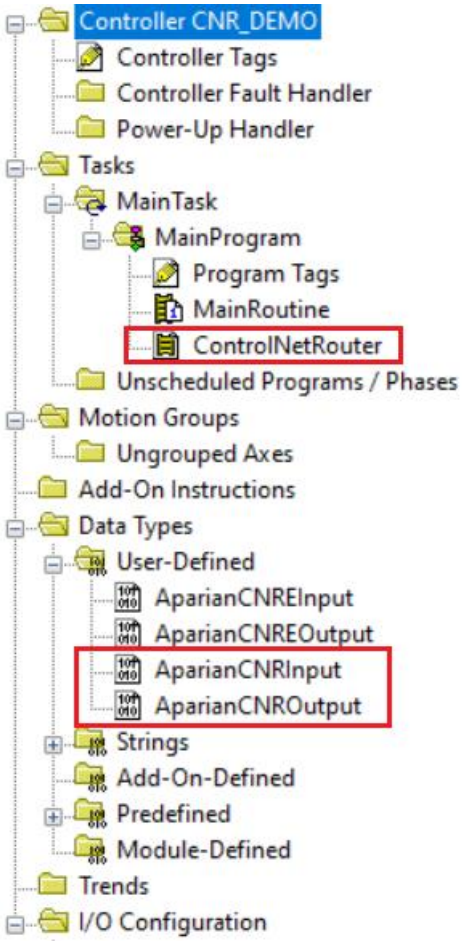


Figure 3.25 - Imported Logix 5000 objects

Refer to the additional information section of this document for an example Logix 5000 project as well as the required UDTs.

3.5.1.2. RSNETWORX CONFIGURATION

The ControlNet scheduling is typically done using ***RSNetworx for ControlNet*** and is required when the module is operating in *ControlNet Target Mode*.

Open RSNetWorx, create a new project and browse to the ControlNet network.

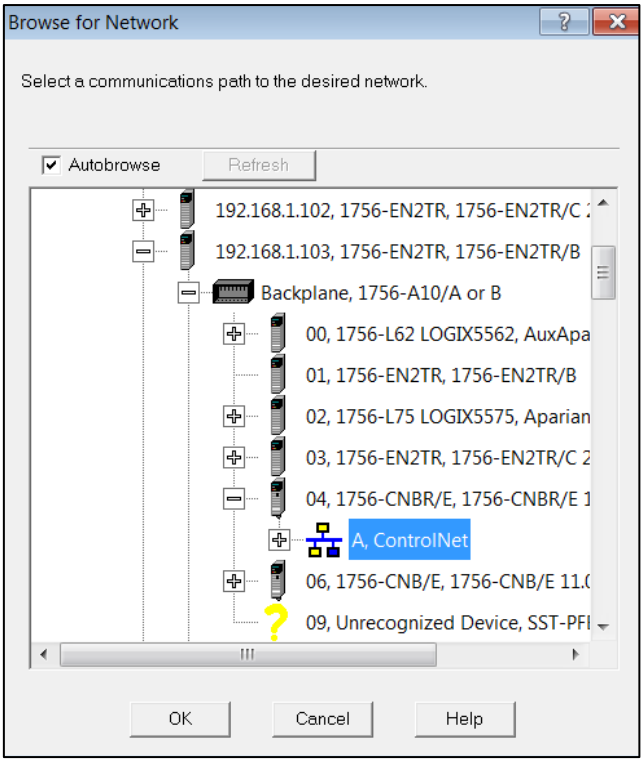


Figure 3.26 – RSNetWorx browse for ControlNet network

Click the online button and the software will scan the network for all the devices. Additional devices can be added (offline) if required.

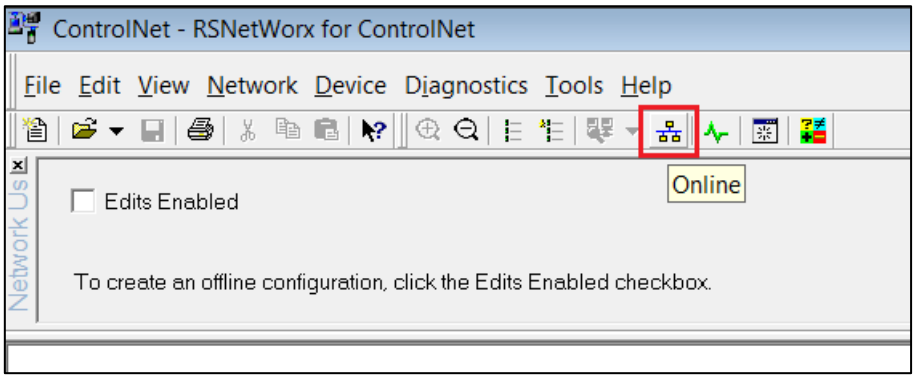


Figure 3.27 – RSNetWorx for ControlNet Online

NOTE: The EDS file of the ControlNet Router will need to be installed to schedule the ControlNet network. The EDS file can be either manually installed or uploaded from the module in RSLinx (as shown below).

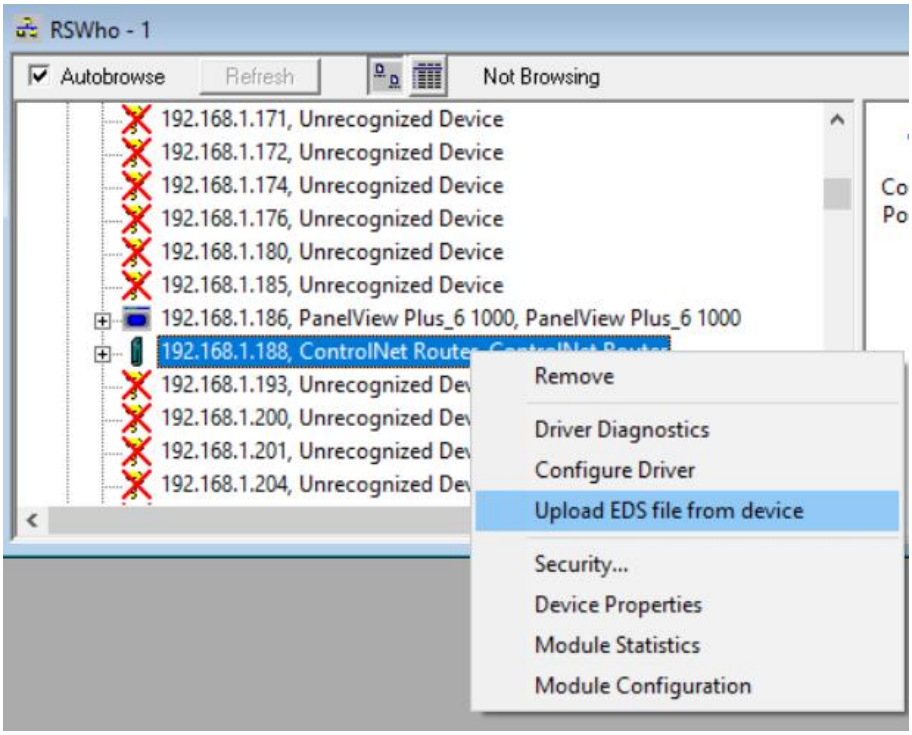


Figure 3.28 – RSLinx EDS upload

Once the scan is completed the online devices will be displayed.

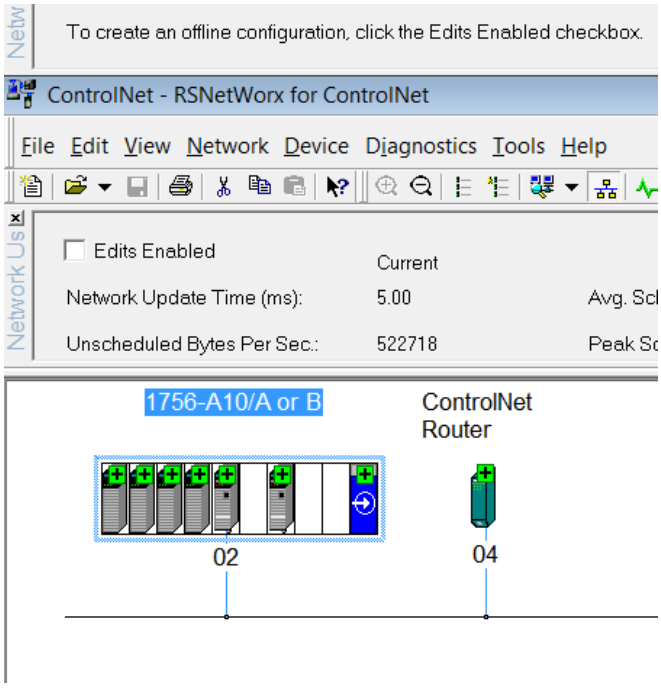


Figure 3.29 – RSNetWorx found devices on the ControlNet network

The user can then make changes (e.g. Redundant ControlNet selected, Max scan address, etc.) to the ControlNet network by pressing the edit button.

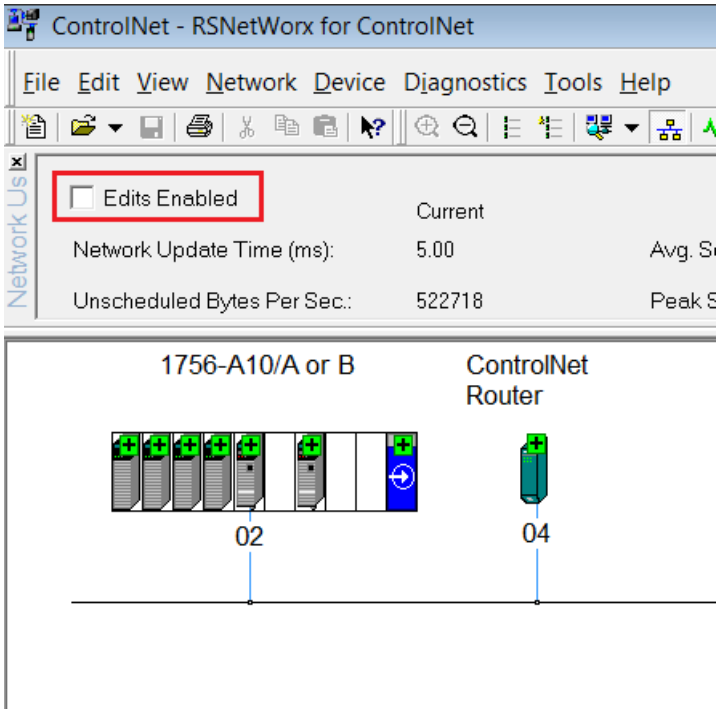


Figure 3.30 – RSNetWorx network edits.

Once done the user must download the ControlNet network configuration to the network by right-clicking on the background and selecting *Download to Network*.

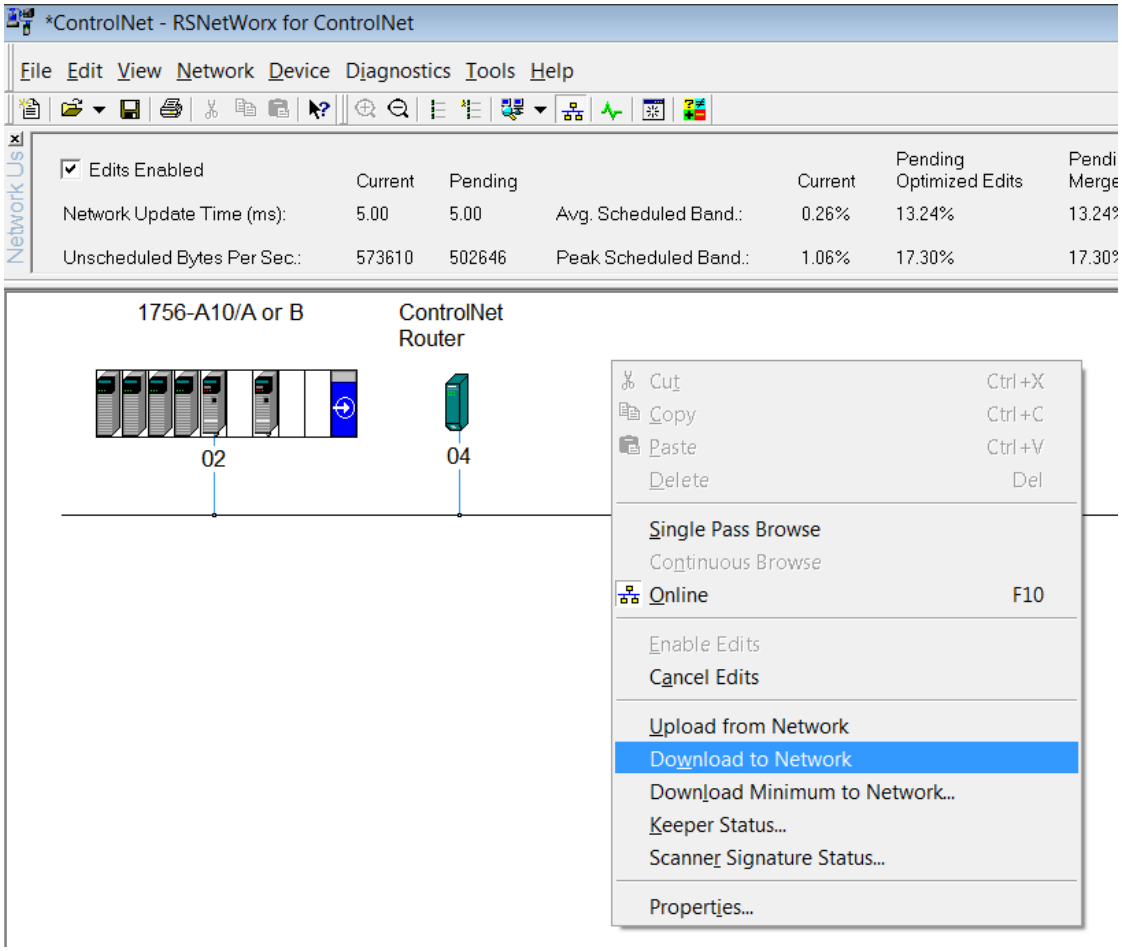


Figure 3.31 – RSNetWorx download

Once this is done the ControlNet Router will connect to the Logix controller over the scheduled ControlNet network (see below):

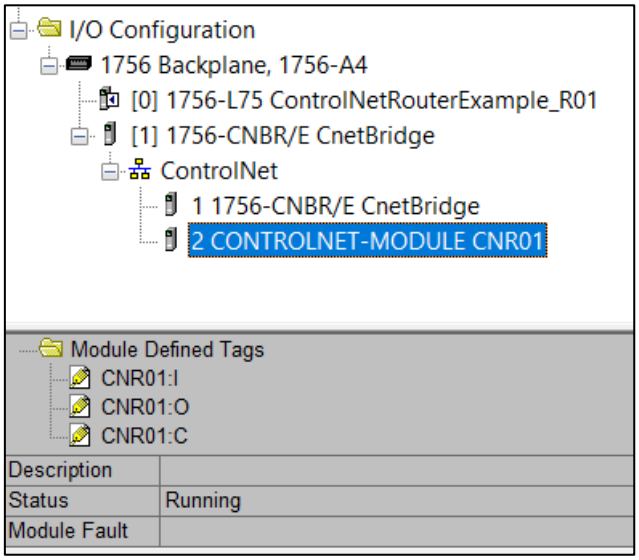


Figure 3.32 – ControlNet Router connected over Scheduled ControlNet

3.5.1.3. INTERNAL DATA SPACE MAPPING

When the module is operating as a ControlNet Target, the data from the originator device (e.g. Logix Controller) can be mapped to the Ethernet interface using the Internal Map. The Internal Map configuration window is opened by either double clicking on the module in the tree or right-clicking the module and selecting *Configuration* and selecting the *Internal Map* tab.

A. IDS COPY – CONTROLNET TARGET SOURCE

When copying data from a connection originator (e.g. the output assembly from the Logix Controller) to the Ethernet interface, the source type needs to be CNet Target.

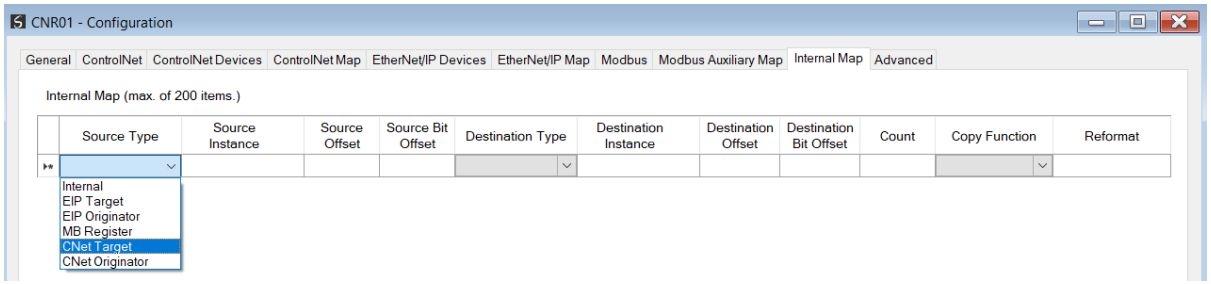


Figure 3.33 – IDS Copy – ControlNet Target Source Type

The source instance will be the connection number, which in this case is always *Connection 0*. The Source Offset is the offset in the *Mapped Data* section of the ControlNet output assembly from where the data must be copied. The Count is the number of **bytes** that will be copied. The *Mapped Data* section will be available when using the provided UDT for the Output Assembly.

CNR01Output	{...}	{...}		AparianCNROutput
CNR01Output.GeneralCtrl	0		Decimal	DINT
CNR01Output.MappedData	{...}	{...}	Decimal	SINT[400]
CNR01Output.MappedData[0]	16#aa		Hex	SINT
CNR01Output.MappedData[1]	0		Decimal	SINT
CNR01Output.MappedData[2]	0		Decimal	SINT

Figure 3.34 – ControlNet Output Mapped Data UDT

See the Internal Data Space Mapping section for more information regarding the operation.

B. IDS COPY – CONTROLNET TARGET DESTINATION

When copying data from the Ethernet interface to the ControlNet Target input assembly, the destination type needs to be CNet Target.

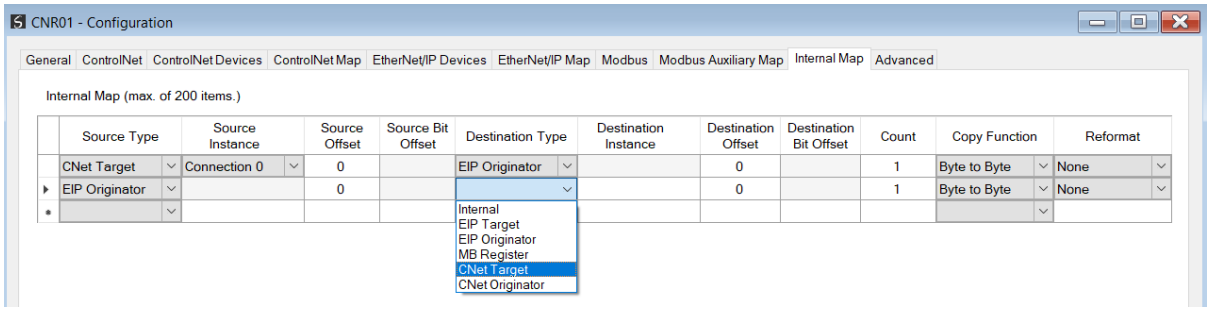


Figure 3.35 – IDS Copy – ControlNet Target Destination Type

The destination instance will be the connection number, which in this case is always *Connection 0*. The Destination Offset is the offset in the *Mapped Data* section of the ControlNet input assembly from where the data must be copied. The Count is the number of **bytes** that will be copied. The *Mapped Data* section will be available when using the provided UDT for the Input Assembly.

CNR01Input	{...}	{...}		AparianCNRInput
CNR01Input.GeneralStatus	-79		Decimal	SINT
CNR01Input.CNetNodeNumber	1		Decimal	SINT
CNR01Input.CNetLedState	1		Decimal	SINT
CNR01Input.CNetMode	6		Decimal	SINT
CNR01Input.Temperature	37.30043		Float	REAL
CNR01Input.MappedData	{...}	{...}	Decimal	SINT[400]
CNR01Input.MappedData[0]	2		Decimal	SINT
CNR01Input.MappedData[1]	0		Decimal	SINT

Figure 3.36 – ControlNet Input Mapped Data UDT

3.5.2. ORIGINATOR

The ControlNet Router can operate as a ControlNet connection originator for cyclic (Class 1) or explicit UCMM data exchange. The explicit messaging can be setup in the *ControlNet Devices* and *ControlNet Map* in the Master configuration while the scheduled ControlNet connections are added to the *ControlNet Connections* node under the module in the Slate project tree.

The Originator Schedule Control allows the user to select between Hardware and Software control.

Hardware

Provides hardware triggers for scheduled packet sending (most accurate), but can only allow up to 10 scheduled connections with limited data being sent from the A-CNR to the ControlNet IO device (see the Specification Appendix for more information).

Software

Provides software triggers for scheduled packet sending, but can have up to 30 scheduled connections.

3.5.2.1. CONTROLNET SCHEDULED DEVICE CONNECTIONS

Hardware Schedule Control

The ControlNet Router can establish up to **10** scheduled ControlNet connections to ControlNet IO devices. This can be done by either manually entering the connection data into the Connection Parameter window or by using the Import Tool. The Import Tool will automatically extract the required connection and configuration parameters from an existing Logix configuration.

A maximum of 500 bytes input data is allowed per scheduled ControlNet connection and a maximum total of 205-277 bytes of output data for all the scheduled ControlNet connections. See the below table for the various total byte counts of output data for the number of connections configured.

Scheduled ControlNet Connection Count	Maximum total byte count for Scheduled ControlNet output data (for all connections)
1	277 bytes
2	269 bytes
3	261 bytes
4	253 bytes
5	245 bytes
6	237 bytes
7	229 bytes
8	221 bytes
9	213 bytes
10	205 bytes

Table 3.4 – Scheduled ControlNet Originator – Maximum total output bytes

Software Schedule Control

The ControlNet Router can establish up to **30** scheduled ControlNet connections to ControlNet IO devices. This can be done by either manually entering the connection data into the Connection Parameter window or by using the Import Tool. The Import Tool will automatically extract the required connection and configuration parameters from an existing Logix configuration.

A maximum of 500 bytes input and output data is allowed per scheduled ControlNet connection.

A. MANUAL CONFIGURATION

A class 1 connection can be added to the *ControlNet Connections* tree by right-clicking on the tree in Slate and selecting *Add ControlNet Connection.*

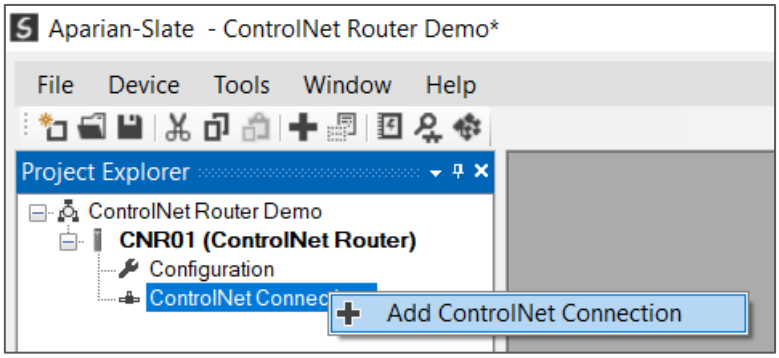


Figure 3.37 – Adding ControlNet Scheduled IO Device Connection

Next the user will need to enter the connection parameters for the Scheduled connection.

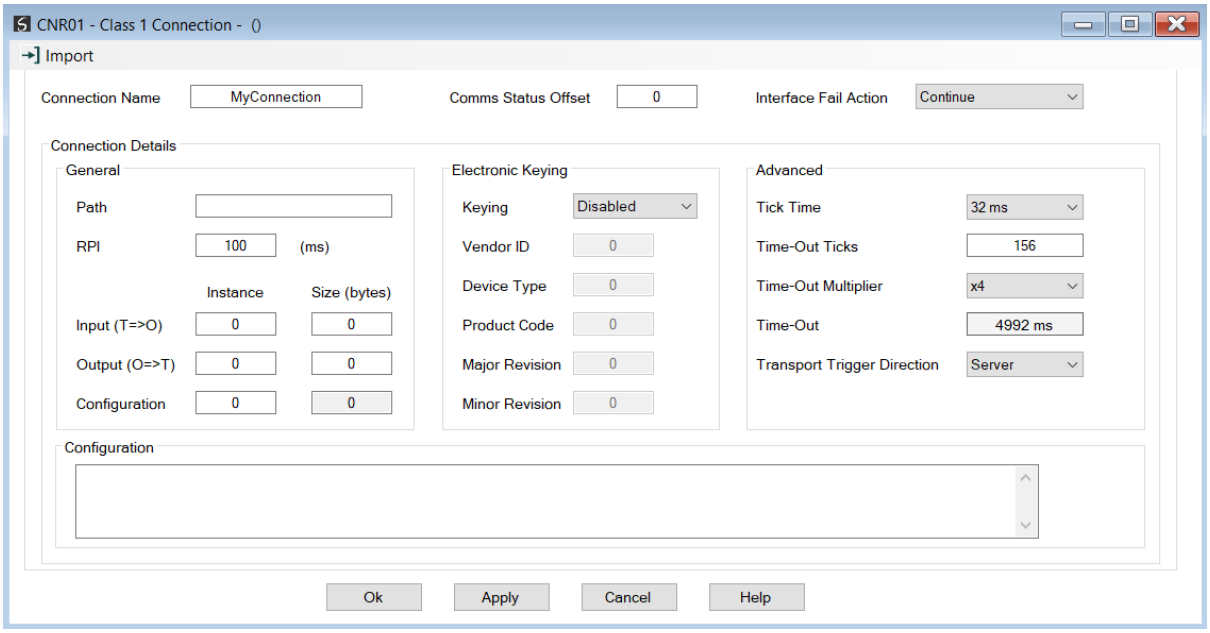


Figure 3.38 – IO Device Scheduled ControlNet Connection Parameters

NOTE: It is recommended that the user not change the values in the *Advanced* frame of the connection parameters.

Parameter	Description
Connection Name	The instance name given to the Scheduled ControlNet Connection.
Comm Status Offset	This is the offset in the Internal Data Space (IDS) which provides the communication status of each ControlNet device. The Communication Status is as shown below:

	Bit 0 - (1: Device online / 0:Device offline) Bit 1 to 7 – Reserved.
Interface Fail Action	When the Ethernet interface communication has failed, the Scheduled ControlNet IO can be configured to either keep the connection running as is, change the connection status to program mode, or force the connection offline. This will allow the ControlNet IO to go into a pre-determined state when the communication to the controller (i.e., connection originator on EtherNet/IP) is lost.
General	
Path	The path to the target device. If the device is a ControlNet end-point device then this will just be the ControlNet node number of the IO device. If the device is, for example, a module in a backplane or via an adapter, then the user will need to enter the ControlNet node number of the bridge or adapter followed by the backplane port (for example 1) and the slot number of the device. Each item is separated by a comma. As an example; to connect to an Allen-Bradley Flex module (via the Flex ControlNet Adapter at ControlNet Node 5) that is in slot 2 of the Flex backplane, the user will need to enter the following path: 5,1,2 (ControlNet node number, port (backplane), slot).
RPI	The requested packet interval (RPI) is the rate in milliseconds at which the data will be sent from the originator to the target and vice versa.
Input (T=>O) – Instance	The instance of the input assembly.
Input (T=>O) – Size (bytes)	The size in bytes of the input assembly.
Output (O=>T) – Instance	The instance of the output assembly.
Output (O=>T) – Size (bytes)	The size in bytes of the output assembly.
Configuration – Instance	The instance of the configuration assembly.
Configuration – Size (bytes)	The size in bytes of the configuration assembly. **NOTE:** This is a read-only value and will be equal to the number of bytes entered into the configuration window below.
Electronic Keying	
Keying	Electronic Keying can be used to ensure that the target device is the correct device type. **Disabled** Keying is not enabled and no key information will be sent in the connection establishment. **Compatible** Keying has been enabled with compatibility enabled. This will allow devices with older firmware to also establish a connection. **Exact** Keying has been enabled and the exact device with specific firmware revision

	will allow the establishment of the connection.
Vendor ID	The Vendor ID of the target device.
Device Type	The Device Type of the target device.
Product Code	The Product Code of the target device.
Major Revision	The Major Revision of the target device.
Minor Revision	The Minor Revision of the target device.
Advanced (Note: Changing these values is not recommended)	
Tick Time	For unconnected messages, this is the time for each tick to calculate the unconnected Time-Out time.
Time-Out Ticks	The number of ticks before the unconnected message is set for timeout.
Time-Out Multiplier	This is the multiplier of the RPI to define the connection timeout time.
Time-Out	The unconnected message timeout time (read-only)
Transport Trigger Direction	The Transport Trigger direction; **Server** or **Client**.
Configuration	
Data	The configuration data that is sent with the forward open connection establishment. The data will need to be entered as a space-delimited, hexadecimal string. For example: 0A 0D 12 EE The configuration size will increase by one, each time a byte is added to the configuration.

Table 3.5 – Scheduled ControlNet Connection Parameters

B. IMPORT TOOL

PREPARATION

Before the connection information can be imported some preparation is required using Studio5000 and a Logix controller:

1. In Studio5000 create a new project and add the required ControlNet device in the IO tree. If the device's profile supports configuration, then configure the device as required.
2. Download the project to a Logix controller.



NOTE: When instantiating modules in Studio5000 do not make use of the "Rack Optimization" communication format.



NOTE: Some versions Logix (V32+) do not support the reading of the module's configuration. Where possible use an earlier version (e.g. V24).

IMPORT CONNECTION PARAMETERS

The connection parameters can be imported from the Logix controller by selecting the ***Import Connection Parameters from Online Controller*** option under the ***Import*** menu of the Class 1 Connection form.

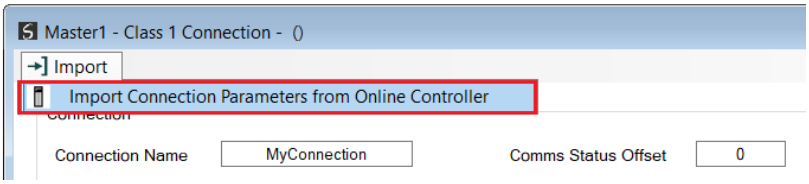


Figure 3.39 – Import Connection Parameters from Online Controller

The Import Connection Parameters form will open.

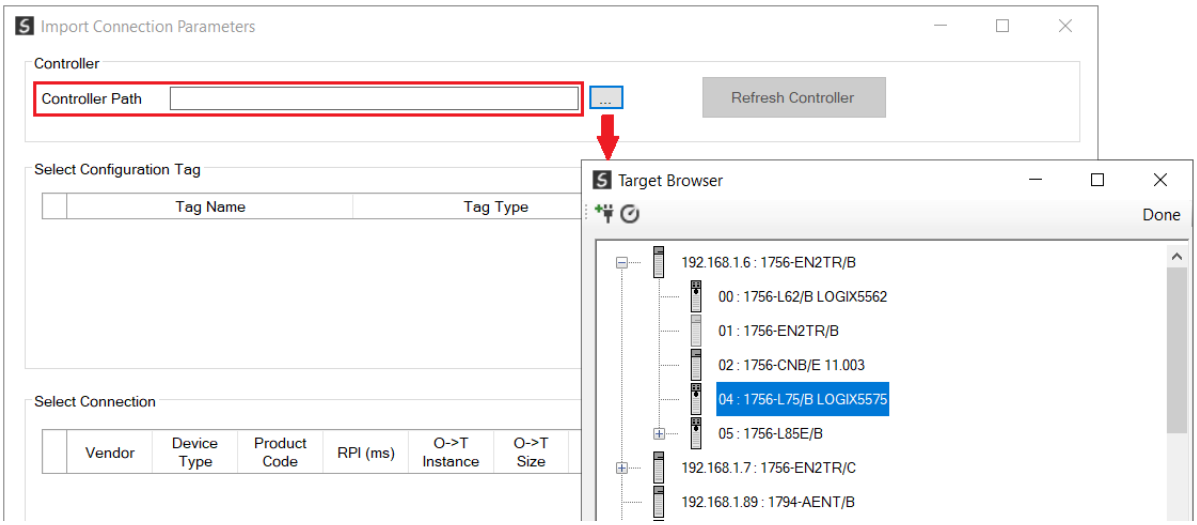


Figure 3.40 – Import Connection Parameters – Controller Path

Enter the path to the Logix controller. This can be either entered manually, or the Browse button "…", can be selected to launch the Target Browser, where the Logix controller can be selected.

Once the Logix controller path has been selected, all the device configuration tags and device connections will be read from the controller and displayed in the Configuration Tag grid and Connection grid respectively.

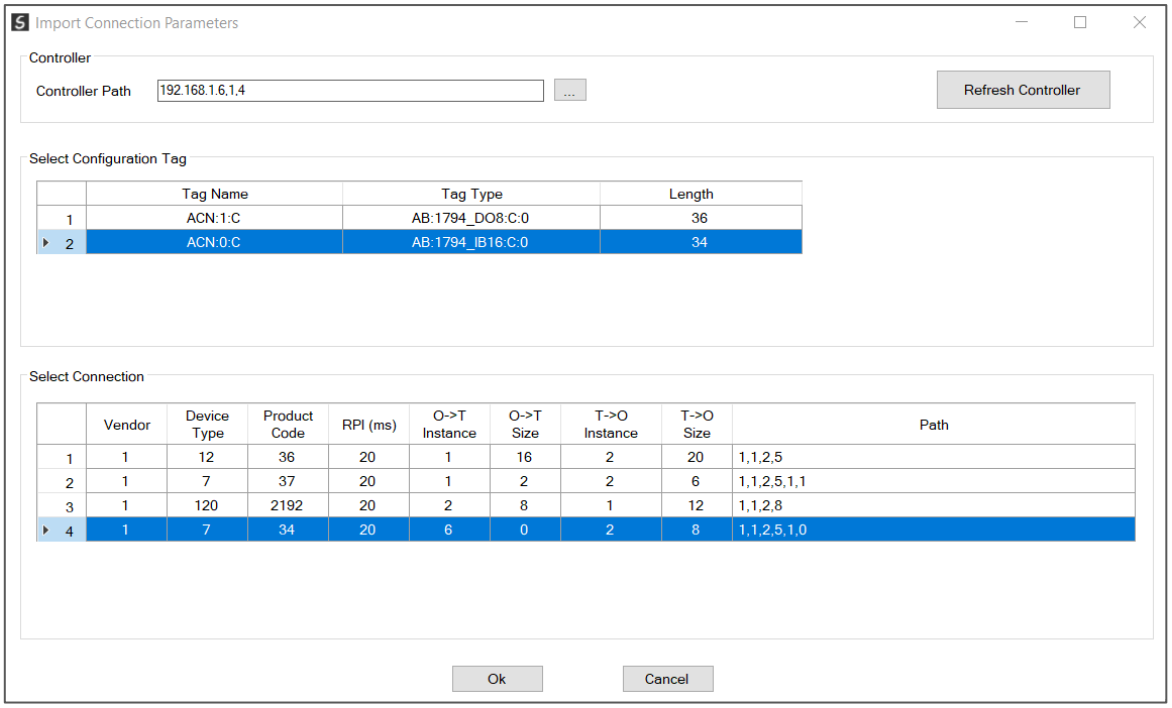


Figure 3.41 – Import Connection Parameters – Select Connection

In order to import all the necessary connection information, the user will need to select both the appropriate ***Configuration Tag***, and the matching ***Connection***.

The new connection's configuration data is derived from the selected ***Configuration Tag***, when the new connection's parameters are derived from the selected ***Connection***.

Once the appropriate selections have been made, press ***Ok***. The imported data will be populated into the Connection form.

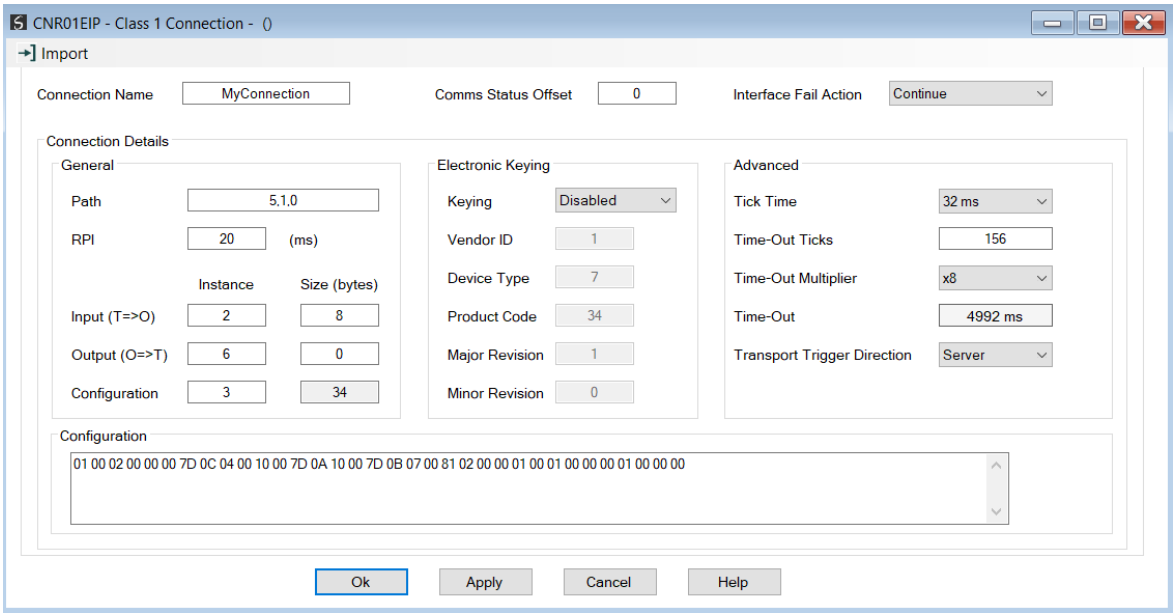


Figure 3.42 – Import Connection Parameters – Controller Path

The user can then modify the ***Connection Name***, ***Path*** and ***RPI*** as required.

3.5.2.2. CONTROLNET EXPLICIT MESSAGE DEVICE CONNECTIONS

Up to 10 ControlNet devices can be added for explicit unscheduled messaging. The user will need to add each device as explained in the ControlNet Devices section below. Once the ControlNet devices have been added the user can then configure the required mapping for the ControlNet Explicit messaging as shown in ControlNet Map section below.

A. CONTROLNET DEVICES

This tab is enabled when the ControlNet Mode is set to *ControlNet Originator*.

The ControlNet Devices configuration is shown in the figure below. Up to 10 ControlNet devices can be configured with up to 50 ControlNet mapped items allowing for explicit Unscheduled Unconnected Messaging (UCMM) to any of the 10 configured devices. The data from each ControlNet device is written to, or read from, an Internal Data Space with a size of 100Kbytes. See the *Explicit ControlNet Messaging* section for more details.

The ControlNet Devices configuration window is opened by either double clicking on the module in the tree or right-clicking the module and selecting *Configuration*.

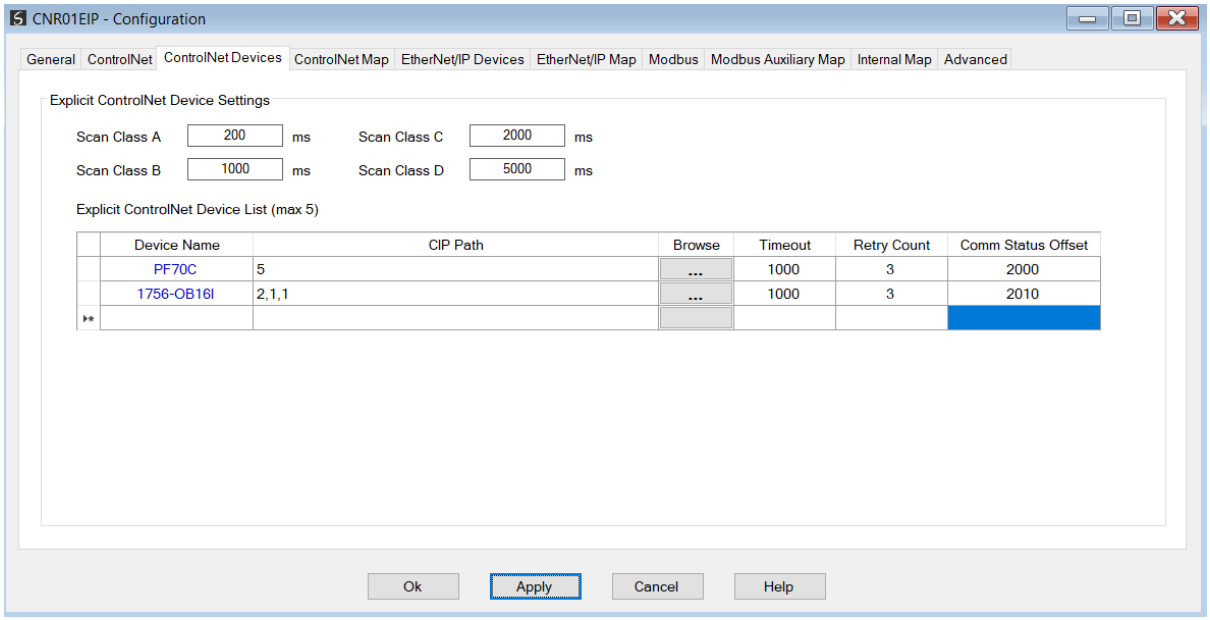


Figure 3.43 – ControlNet Device configuration

The ControlNet Devices configuration consists of the following parameters:

Parameter	Description

Scan Class A, B, C, D	The configurable update rates (in milliseconds) for each Scan Class in the ControlNet Map.
Device List (per device)	
Device Name	The user assigned instance name for the specific device.
CIP Path	The CIP Path to the target device. This can either be entered manually or the user can browse to them by clicking the ***Browse*** button. The Target Browser will open and automatically scan for all available ControlNet devices. If the ControlNet module is a bridge module, it can be expanded by right-clicking on the module and selecting the ***Scan*** option. The required ControlNet device can then be chosen by selecting it and clicking the ***Ok*** button, or by double-clicking on the target module.
Timeout	The amount of time (in milliseconds) the module will wait for a response from the target ControlNet device.
Retry Count	The number of message retries before the target ControlNet device is considered offline.
Comm Status Offset	This is the offset in the Internal Data Space (used to map ControlNet device data) which provides the communication status of each ControlNet device. The Communication Status is as shown below: Bit 0 - (1:Device Online , 0:Device Offline) Bit 1 to 7 – Reserved.

Table 3.6 – ControlNet Devices - Configuration Parameters

B. CONTROLNET MAP

This tab is enabled when the ControlNet Mode is set to *ControlNet Originator*.

The ControlNet Map configuration is shown in the figure below. Up to 50 ControlNet mapped items (unscheduled UCMM messages) can be configured to any of the 10 pre-configured ControlNet devices. The data from each ControlNet device is written to, or read from, the Internal Data Space with a size of 100Kbytes. See the *Explicit ControlNet Messaging* section for more details.

The ControlNet Map configuration window is opened by either double clicking on the module in the tree or right-clicking the module and selecting *Configuration*.

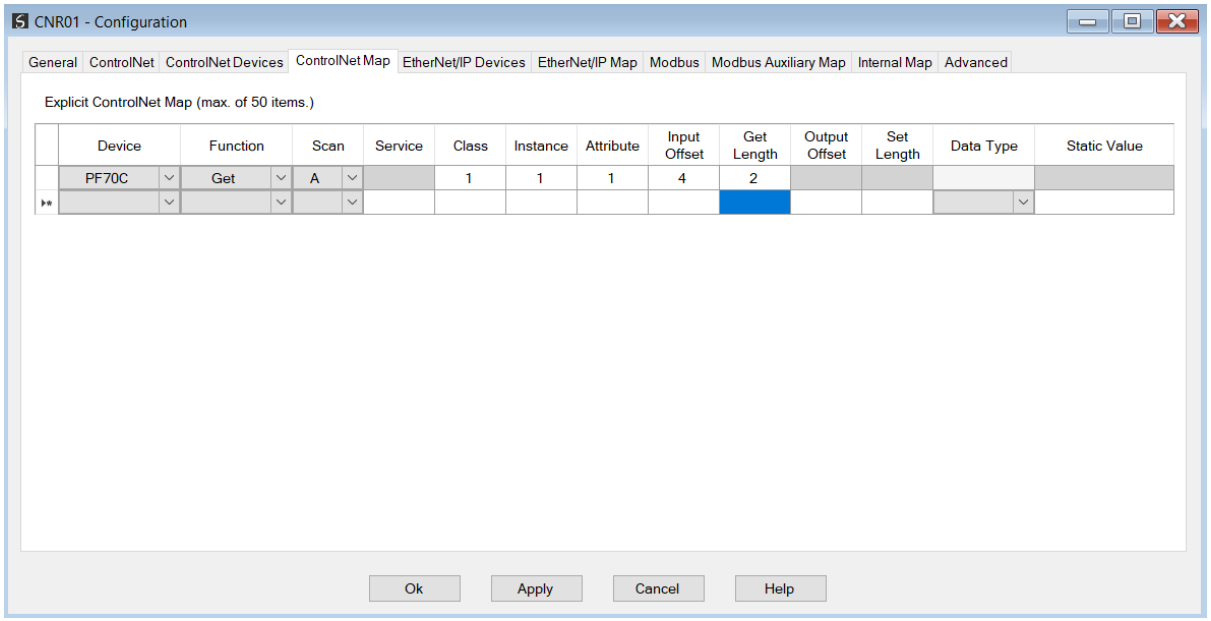


Figure 3.44 – ControlNet Map configuration

The ControlNet Map configuration consists of the following parameters:

Parameter	Description
Device	The device instance name configured in the previous ControlNet Devices tab. The selected device will be used for executing the communication function.
Function	The user can select one of four functions. **Get** The module will read data from the target ControlNet device by using the Get Single Attribute CIP function. The received data will be placed into the Internal Data Space at the ***Input Offset*** location configured in this tab. **Set** The module will write data to the target ControlNet device by using the Set Single Attribute CIP function. The data to be written will be retrieved from the Internal Data Space at the ***Output Offset*** location configured in this tab. **Set Static** Similar to the Set function above, but the data to be written will be fixed (equal to the *Static Value*) parameter in this configuration window. This function will typically be used with the single (**S**) Scan class which means the ControlNet Router module can be setup to write the fixed value only once when the target device communication has been established. **Custom** This function allows the user to use a custom Service to write and read data in the same transaction. The user will need to see which custom services that target device

	supports in that device's user manual.
Scan	The user can select Scan Class **A**, **B**, **C** or **D** (which was configured in the ControlNet Devices tab). The specific mapped item will then be executed at that configured scan class rate. The user can also select the **S** class which means that the mapped item will only execute once when communication to the target device is established. If the target device goes offline, then the mapped items with this class will be re-armed, and resent when communication is re-established.
Service	The custom CIP service/function which is only available when the ***Custom*** function has been selected.
Class, Instance, Attribute	The CIP class, instance, and attribute of the request message to be sent.
Input Offset	The location in the Internal Data Space where the received data will be written. This will only be available for ***Get*** and ***Custom*** functions.
Get Length	The length of the data to be received. If the number of bytes received is more than the Get Length, then the data will **not** be written to the Internal Data Space. This will only be available for ***Get*** and ***Custom*** functions.
Output Offset	The location in the Internal Data Space from where the data to be written to the target device will be read. This will only be available for ***Set*** and ***Custom*** functions.
Set Length	The length of the data to be written. This will only be available for ***Set*** and ***Custom*** functions.
Data Type	The data type of the Static Value. This will only be available for ***Set Static*** function.
Static Value	The value to be written to the target device when the ***Set Static*** function has been selected. **Note**: When using the SINT Array data type, the values must be entered as space-delimited hex values. For example: 05 34 2E A1

Table 3.7 – ControlNet Map configuration parameters

3.5.2.3. SCHEDULING THE CONTROLNET NETWORK

Once all the ControlNet scheduled connections have been added to the ControlNet Router IO tree in Slate (when the module is operating as a ControlNet Originator), the ControlNet network will need to be scheduled. The scheduling of the ControlNet network will allow for each connection originator and connection target to exchange data at a predetermined time in the Network Update Time (NUT).

The ControlNet Scheduler is opened by right-clicking on the module when online in Slate and selecting *ControlNet Scheduler.*

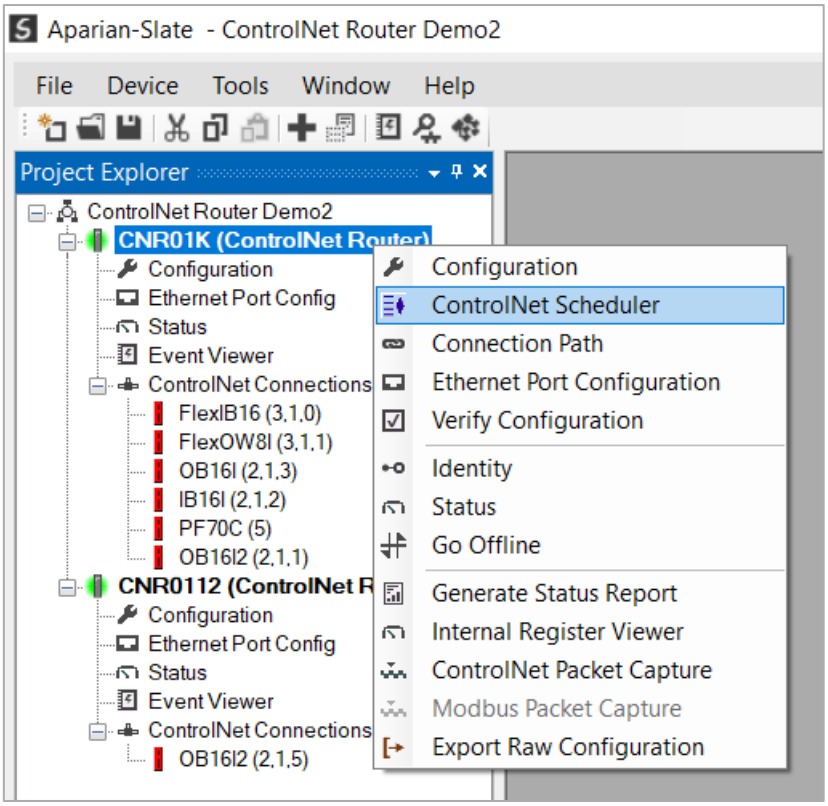


Figure 3.45 – ControlNet Scheduler select

From the ControlNet Scheduler the ControlNet network can be scanned for connection originator devices, the network schedule can be calculated, and the scheduled and Keeper data can be downloaded to a ControlNet Router.

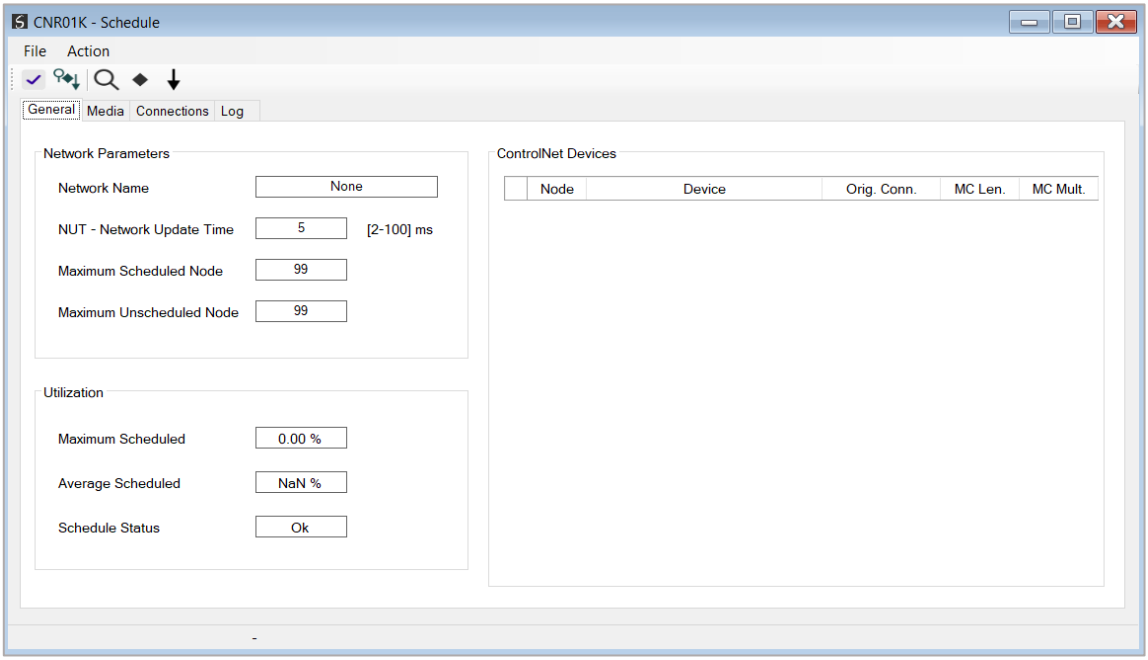


Figure 3.46 – ControlNet Scheduler

The user can update the Network Parameters in the Scheduler.

NOTE: Unless the user is experienced in ControlNet networks, it is recommended to leave the Network Update Time (NUT) at its default value of 5ms.

Parameter	Description
Network Name	The name assigned to the ControlNet network. This can be any name selected by the user.
NUT – Network Update Time	The Network Update Time (NUT) that will determine the schedule rate.
Maximum Scheduled Node	The maximum ControlNet node number that can be used for scheduled communication.
Maximum Unscheduled Node	The maximum ControlNet node number that can be used for unscheduled communication.

Table 3.8 – Scheduler Network parameters

A. SCHEDULING THE CONTROLNET NETWORK

Once all the ControlNet Routers that are operating as ControlNet Originators (Control Mode – Originator) have been downloaded, the ControlNet network can be scheduled. At least one of the ControlNet Routers on the network will need to be the ControlNet Keeper which distributes network information to the rest of the all the ControlNet Nodes on the network. When Slate downloads the Schedule to the module, it will automatically also download the Keeper data as well.

NOTE: The ControlNet Router will only become a Keeper if the *Enable ControlNet Keeper* parameter has been enabled in the general ControlNet configuration in Slate.

NOTE: It is recommended for the ControlNet Router with the lowest node number to be the active Keeper.

The entire scheduling process can be done automatically or step-by-step as described below.

SCAN-SCHEDULE-DOWNLOAD

When pressing the  button in the Slate Scheduler, the software will scan the ControlNet network up to the Maximum Scheduled Node (as configured in the *Network Parameters*). The ControlNet network will be scheduled once all the ControlNet devices (with configuration with connection origination) have been found. When the scheduled has finished compiling, it will be downloaded to the local module as well as all the other ControlNet Router connection

originators (via the ControlNet network). The schedule will be stored in Non-Volatile memory for each of the ControlNet Routers having connection origination.

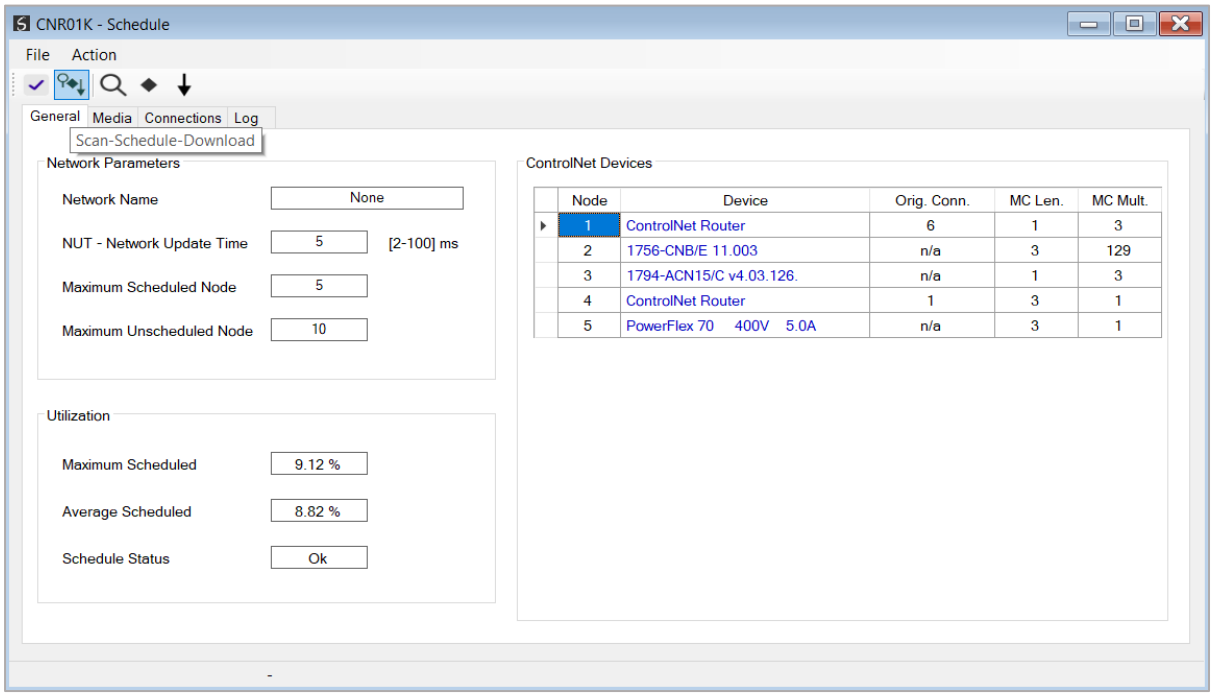


Figure 3.47 – ControlNet Scheduler – Scan-Schedule-Download

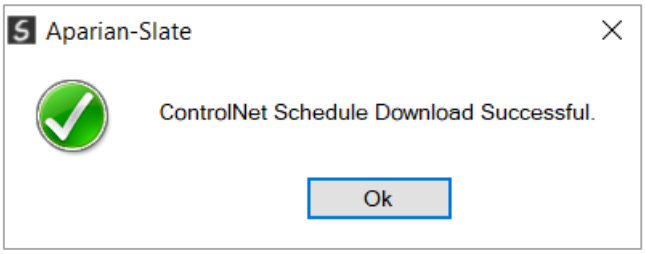


Figure 3.48 – ControlNet Scheduler – Scheduling Complete

Once done the ControlNet Router will establish scheduled connections to the devices in the ControlNet IO tree of the module (in Slate) and ControlNet IO devices will go green in the local and other ControlNet Routers.

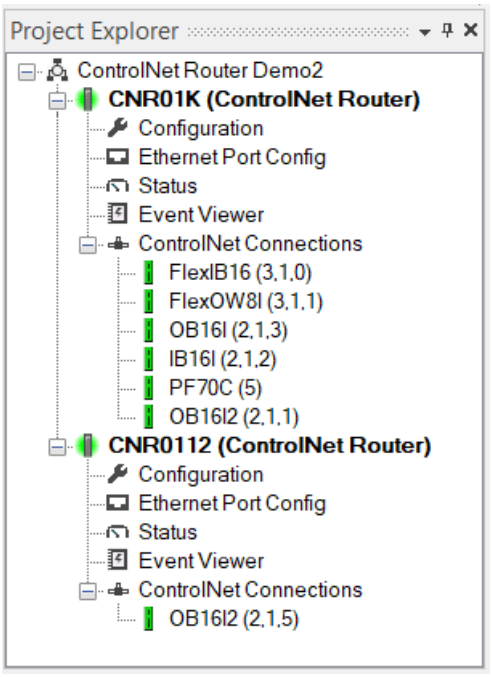


Figure 3.49 – ControlNet Scheduler – ControlNet IO devices online

SCAN ONLY

When pressing the  button in the Slate Scheduler, the software will only scan the ControlNet network up to the Maximum Scheduled Node (as configured in the *Network Parameters*).

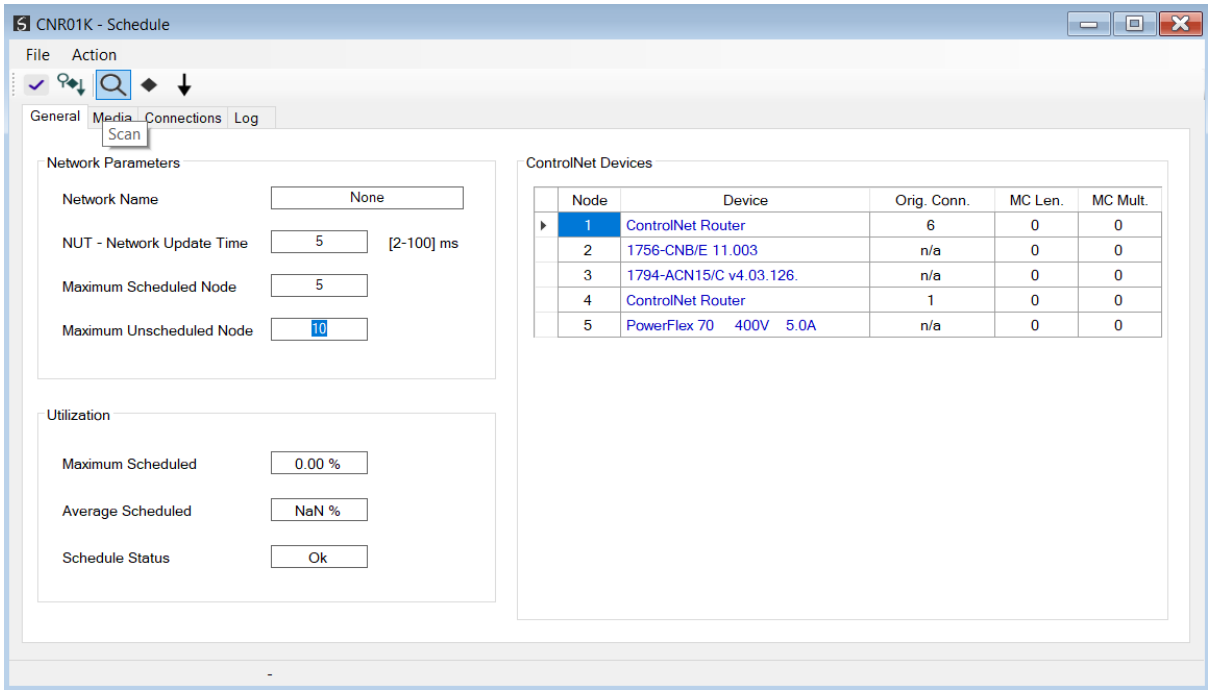


Figure 3.50 – ControlNet Scheduler – Scan

SCHEDULE ONLY

When pressing the button in the Slate Scheduler, the ControlNet network will be scheduled once all the ControlNet devices (with configuration with connection origination) have been found.

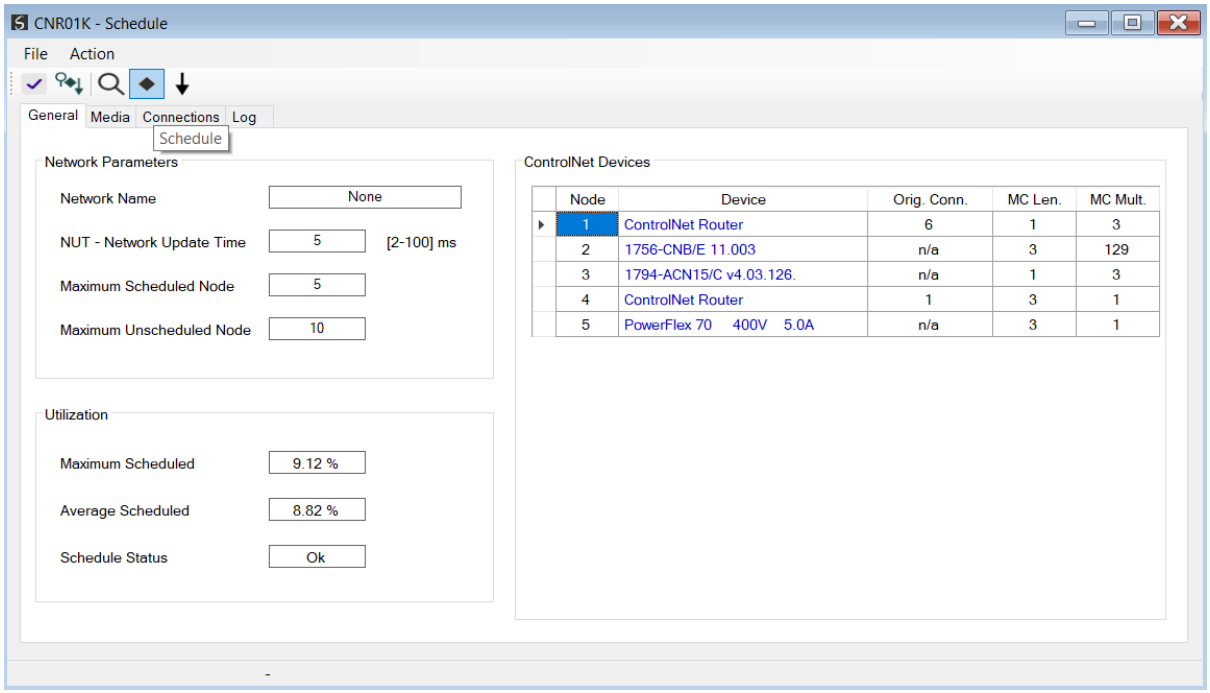


Figure 3.51 – ControlNet Scheduler – Schedule

DOWNLOAD ONLY

When pressing the button in the Slate Scheduler (and 2hen the scheduled has finished compiling), it will download the schedule to the local module as well as all the other ControlNet Router connection originators (via the ControlNet network). The schedule will stored in Non-Volatile memory for each of the ControlNet Routers having connection origination.

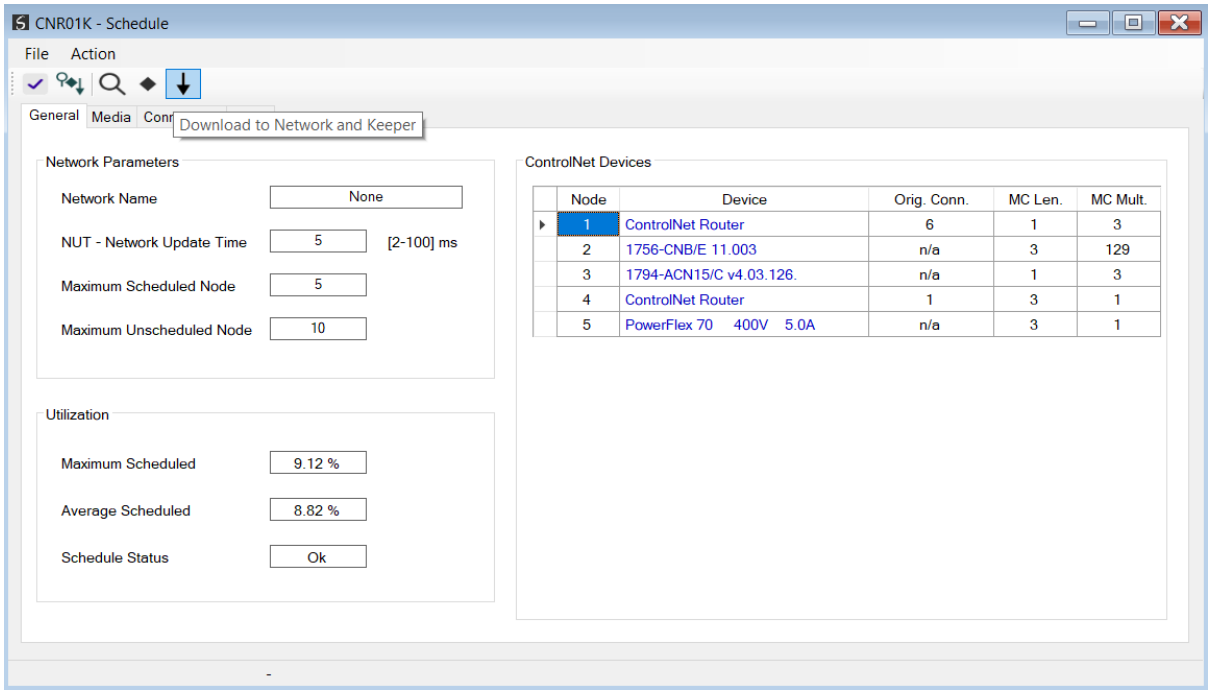


Figure 3.52 – ControlNet Scheduler – Download

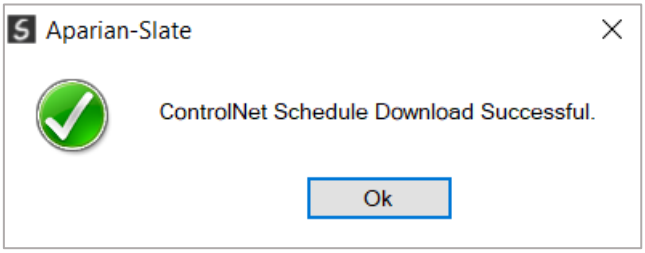


Figure 3.53 – ControlNet Scheduler – Download Complete

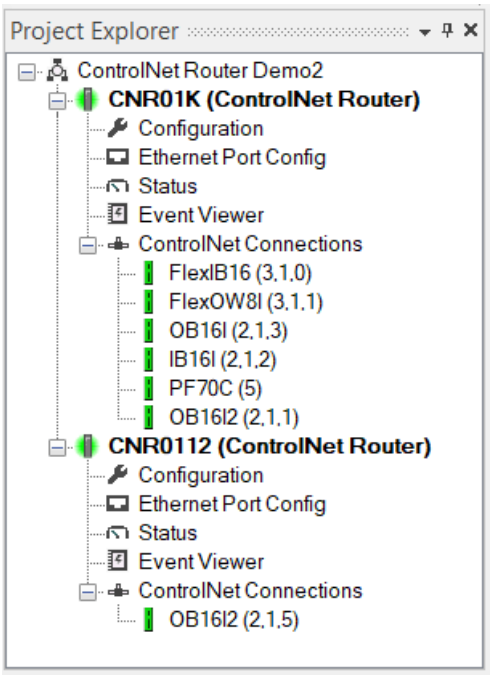


Figure 3.54 – ControlNet Scheduler – ControlNet IO devices online

B. MEDIA

To allow for a more accurate ControlNet schedule, the user can enter the media type and cable distance being used in the ControlNet network. This will add the appropriate delays to the ControlNet schedule calculation. The Media configuration is opened by selecting the *Media* tab in the Slate Scheduler.

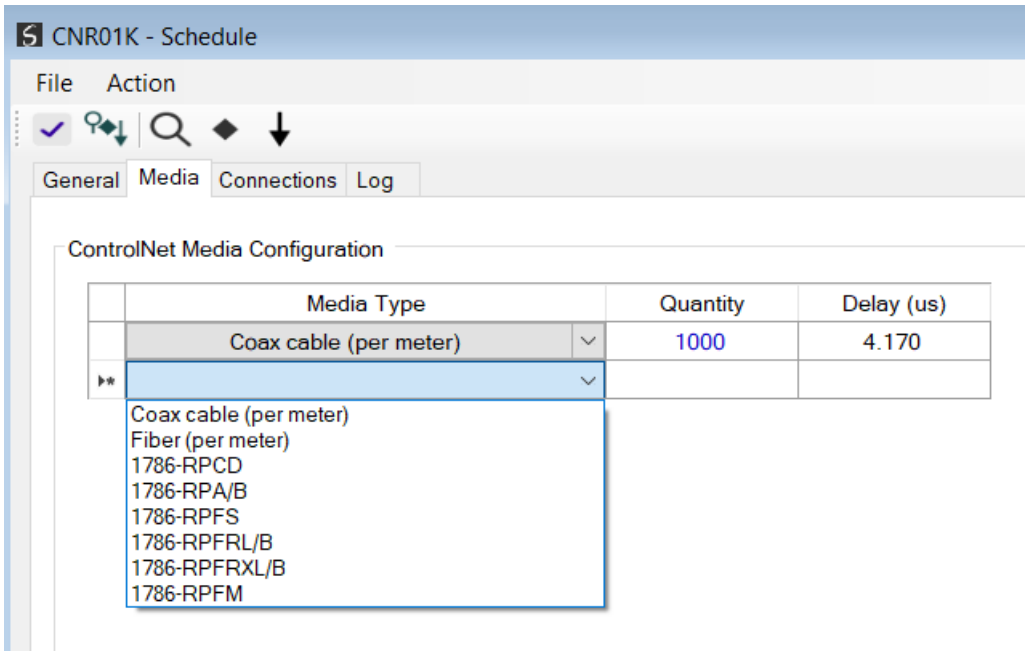


Figure 3.55 – ControlNet Scheduler – Media Configuration

C. ERASE SCHEDULE AND KEEPER

If needed, the Schedule and Keeper parameters can be removed from the non-volatile storage of the device. In the *Action* tab of the Slate Scheduler, select either to *Erase Keeper, Erase CO Schedule,* or *Erase Keeper and CO Schedule.*

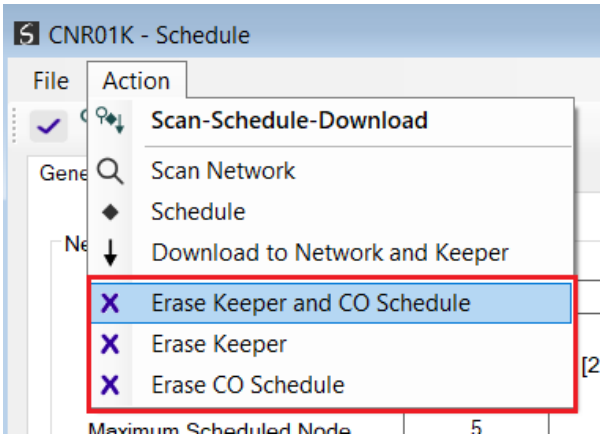


Figure 3.56 – ControlNet Scheduler – Erase Keeper and CO Schedule

3.5.2.4. INTERNAL DATA SPACE MAPPING

When the module is operating as a ControlNet Originator, the data from the ControlNet IO devices can be mapped to the Ethernet interface using the Internal Map. The Internal Map configuration window is opened by either double clicking on the module in the tree or right-clicking the module and selecting *Configuration* and selecting the *Internal Map* tab.

A. IDS COPY – CONTROLNET ORIGINATOR SOURCE

When copying data from a ControlNet IO to the Ethernet interface, the source type needs to be CNet Originator.

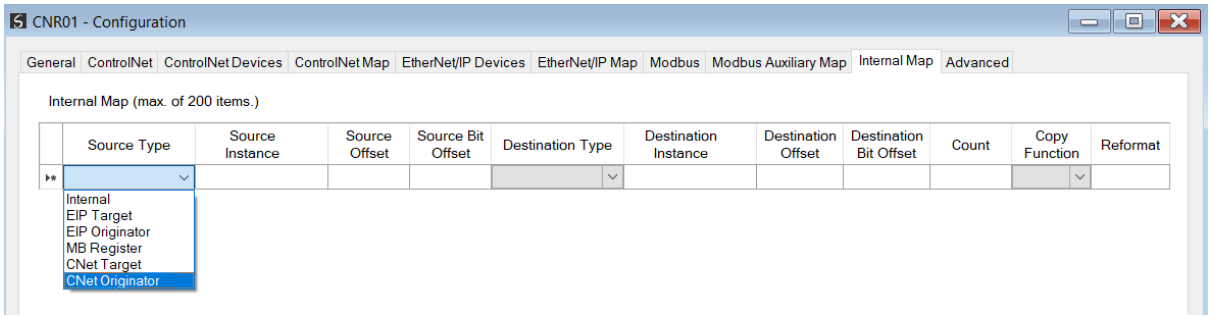


Figure 3.57 – IDS Copy – ControlNet Originator Source Type

The source instance will be one of the ControlNet IO devices added to the ControlNet IO tree in Slate.

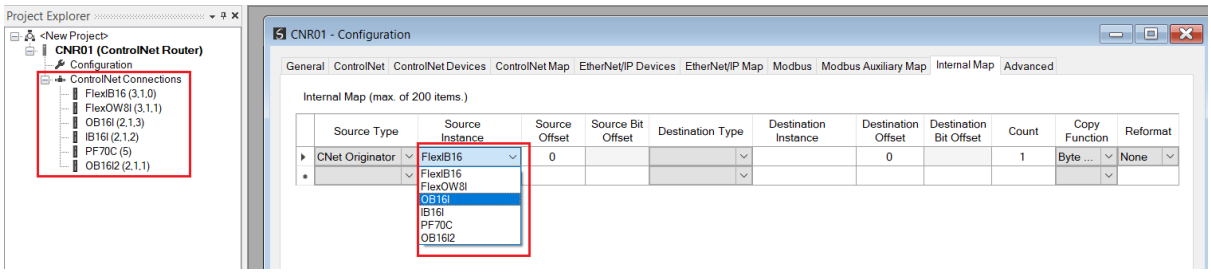


Figure 3.58 – IDS Copy – ControlNet Originator Source Instance

The Source Offset is the offset in the selected ControlNet device Scheduled **Input** Assembly. The Count is the number of **bytes** that will be copied. See the Internal Data Space Mapping section for more information regarding the operation.

B. IDS COPY – ETHERNET/IP TARGET DESTINATION

When copying data from the Ethernet interface to a ControlNet IO device **Output** Assembly, the destination type needs to be CNet Originator.

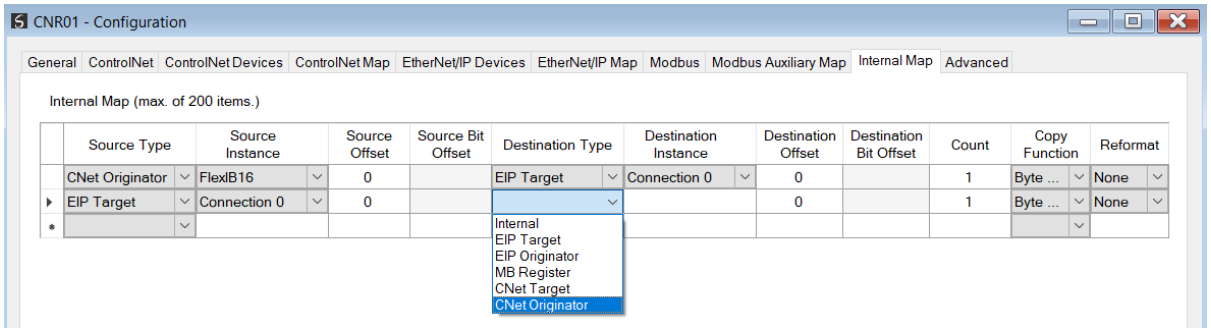


Figure 3.59 – IDS Copy – ControlNet Originator Destination Type

The destination instance will be one of the ControlNet IO devices added to the ControlNet IO tree in Slate.

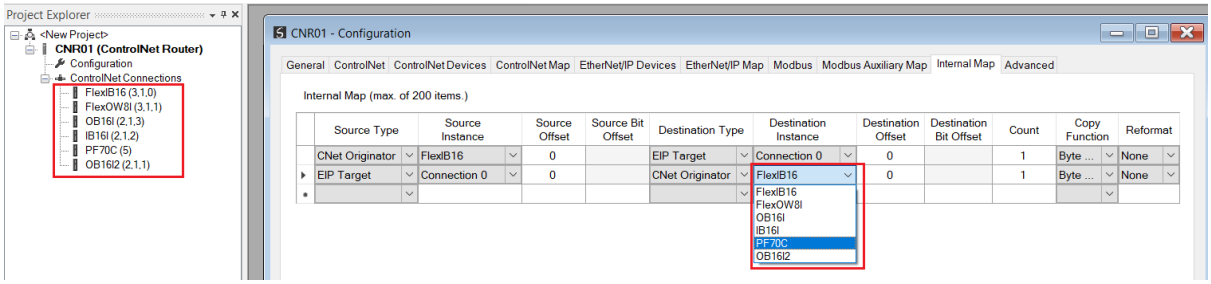


Figure 3.60 – IDS Copy – ControlNet Originator Destination Instance

The Destination Offset is the offset in the selected ControlNet device Scheduled **Output** Assembly. The Count is the number of **bytes** that will be copied. See the Internal Data Space Mapping section for more information regarding the operation.

3.5.3. BRIDGE

In Bridge Mode, it will allow the user to add, configure, and parameterize ControlNet IO via the ControlNet Add-on Profile (AOP) in Logix. ControlNet devices can be added under the A-CNR AOP which will operate as a ControlNet bridge. In this mode the ControlNet Router will operate as a ControlNet originator and Keeper, but the ControlNet network will be dynamically Scheduled as ControlNet Connections are established.



NOTE: When using a 1756-L8x Logix Controller (5580), firmware revision 37.11 has an anomaly that causes Flex IO over ControlNet to not connect. Please upgrade the Logix Controller to revision 37.12 to resolve this issue.

The Bridge configuration is shown in the figure below. The Bridge configuration window is opened by either double clicking on the module in the tree, or right-clicking the module and selecting ***Configuration***.

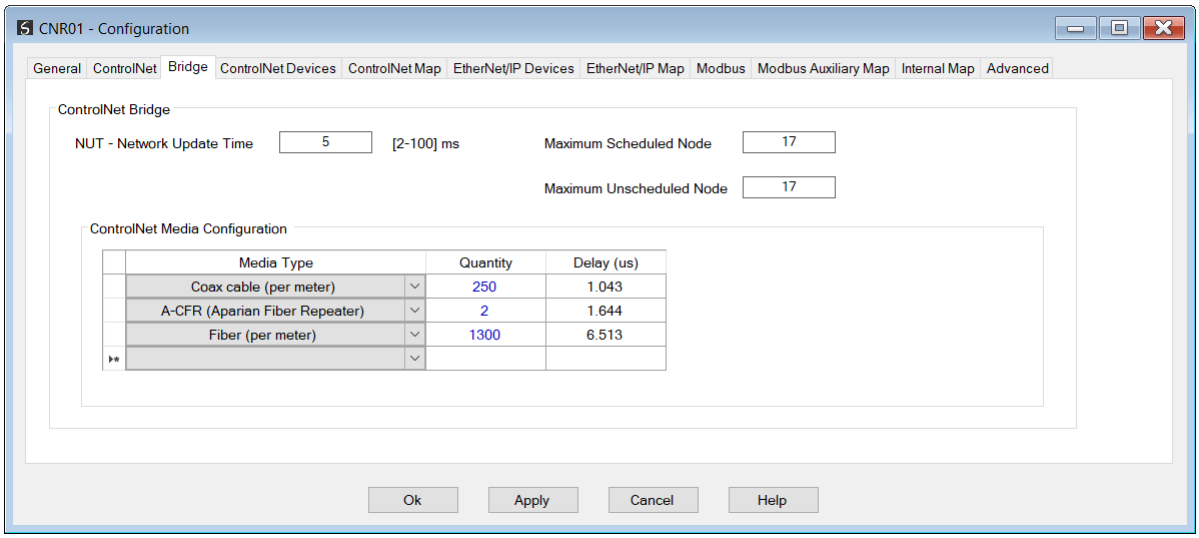


Figure 3.61 - Bridge Configuration

The Bridge configuration consists of the following parameters:

Parameter	Description
NUT – Network Update Time	The Network Update Time (NUT) that will determine the schedule rate.
Maximum Scheduled Node	The maximum ControlNet node number that can be used for scheduled communication.
Maximum Unscheduled Node	The maximum ControlNet node number that can be used for unscheduled communication.
ControlNet Media Configuration	
Media Type	The media type or network accessory type.
Quantity	The quantity of the above. Either in meters for Media, or quantity for accessories.
Delay	The calculated delay in microseconds.

Table 3.9 – Bridge parameters

To operate as a ControlNet bridge in Studio 5000, the A-CNR AOP will need to be installed.

3.5.3.1. ADD-ON PROFILE (AOP) INSTALLATION

The user will first need to install the A-CNR AOP before the module can be added to the Logix I/O tree. The AOP for the ControlNet Router (*AOP_AP_ACNR_Rel1.01.08_Signed.zip*) can be downloaded from the Aparian website:
https://aparian.com/products/controlnetrouter#downloads

Once downloaded extract the zip file, run the *MPSetup.exe* file, and follow the on-screen instructions.

3.5.3.2. ADDING A-CNR IN LOGIX

Integration with the Logix family in Studio5000 makes use of an AOP. Under a Logix Ethernet bridge (e.g., 1756-EN2TR), right-click and select the New Module option.

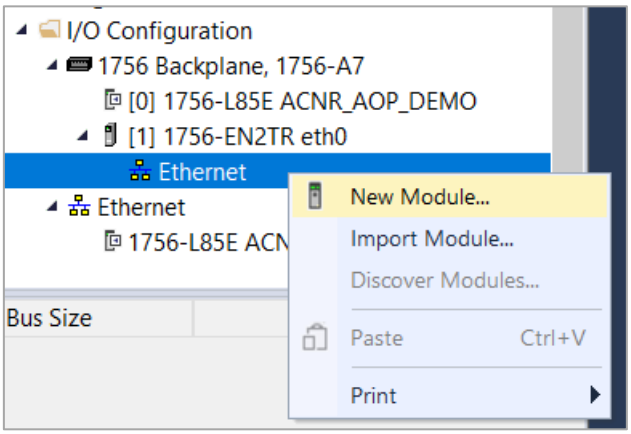


Figure 3.62 – Adding a module

The module selection dialog will open. To find the module more easily, use the Vendor filter to select only the Aparian modules as shown in the figure below.

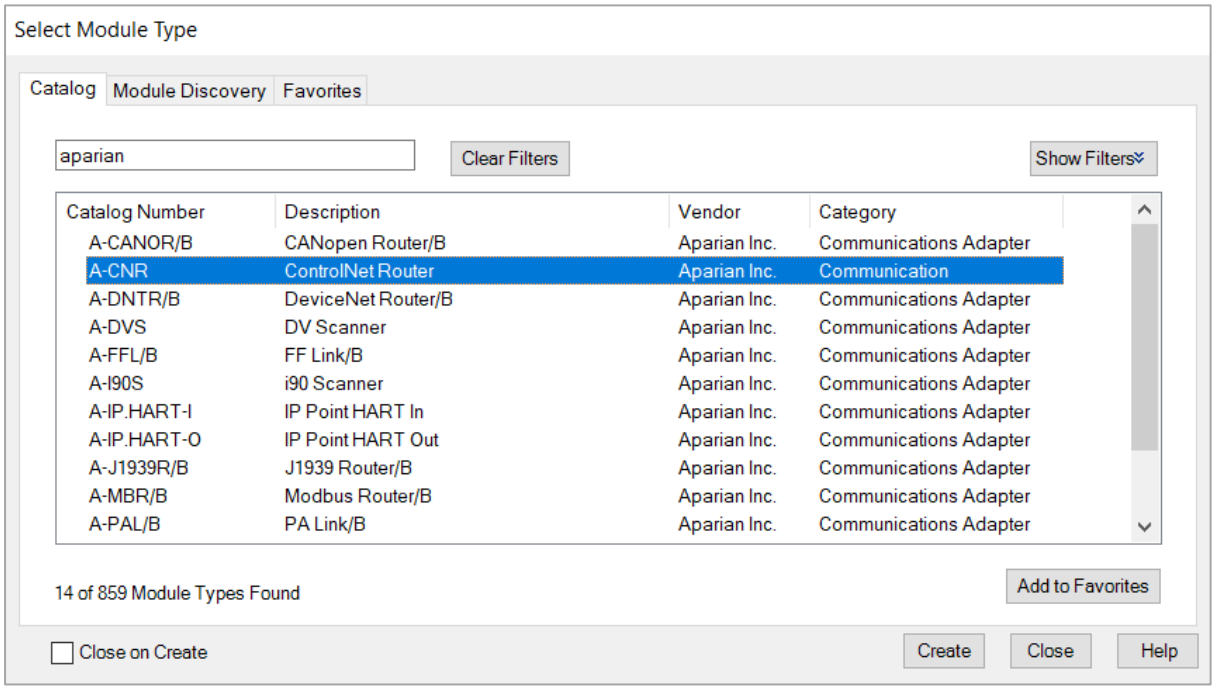


Figure 3.63 – Selecting the module

Locate and select the A-CNR module and select the *Create* option. The module configuration dialog will open, where the user must specify the Name, IP Address, and ControlNet node

number to complete the instantiation.

NOTE: The ControlNet Node must match that of the *Node Address* selected under the ControlNet configuration tab in Slate.

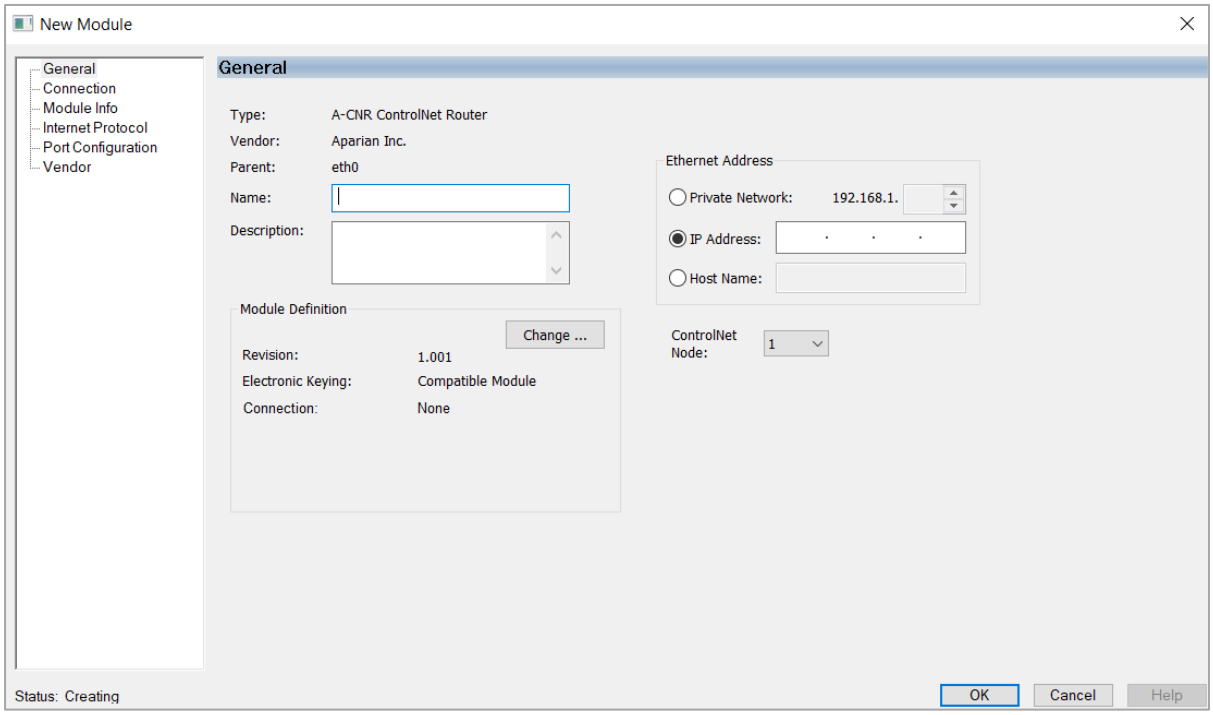


Figure 3.64 – Module instantiation

Once the instantiation is complete the module will appear in the Logix IO tree.

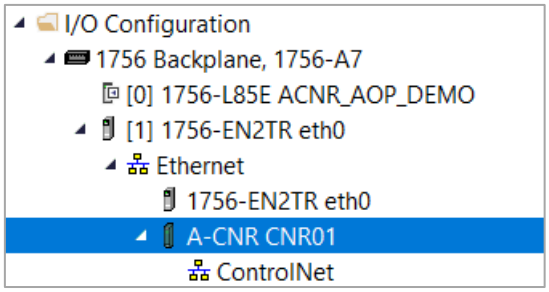


Figure 3.65 – Logix IO tree

NOTE: When the A-CNR is added in RSLinx under the AB_ETH driver (for legacy PCCC communication), it could cause the Add-On Profile for the A-CNR to not communicate correctly. Ensure that the A-CNR is under the AB-ETHIP driver (used for EtherNet/IP devices).

3.5.3.3. ADDING CONTROLNET IO IN LOGIX

Once the ControlNet Router has been added in the Logix IO tree, the user can then add

ControlNet IO under the A-CNR AOP.

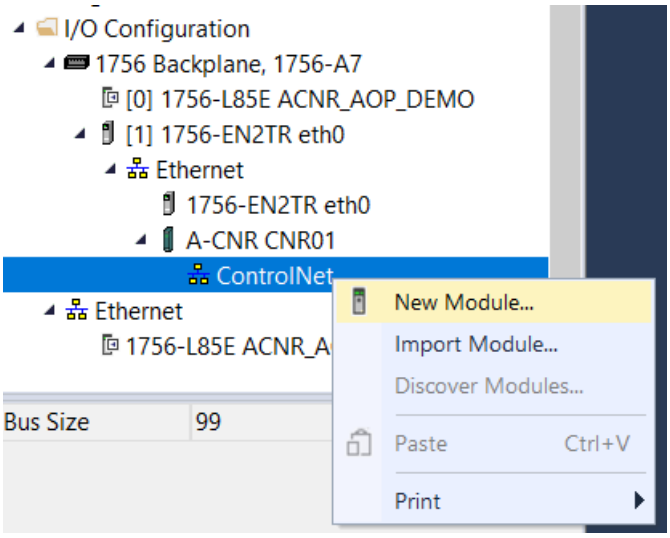


Figure 3.66 – Logix ControlNet IO tree

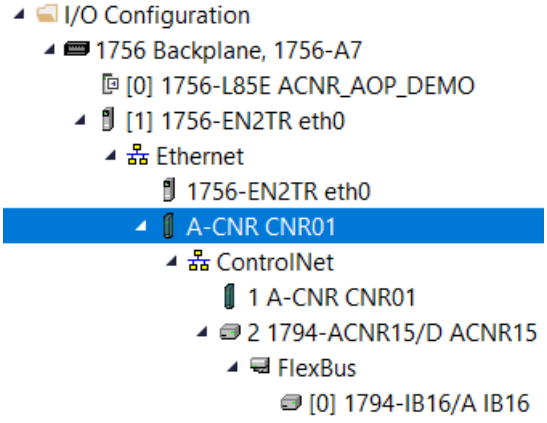


Figure 3.67 – Logix ControlNet IO devices

NOTE: When operating in Bridge mode, the ControlNet network will be scheduled dynamically as the connections are established. The user does not need to schedule the ControlNet network as it is done when using ControlNet in Originator mode.

When the Logix Controller is trying to establish a connection to ControlNet IO, the error code received by the Logix controller (when an error code is present) will come from the ControlNet IO device, except in certain circumstances when the A-CNR replies due to a network error. For example;

- *Error Code 0x01 – Connection Failure with Extended Error code 0x0302 – Bandwidth not available*; this is returned when another connection is received by the A-CNR for cyclic data exchange, but there is no more bandwidth on the ControlNet network.
- *Error Code 0x01 – Connection Failure with Extended Error code 0x0113 – Out of Connections*; this is returned when a ControlNet connection establish request is received, but there are already 30 scheduled ControlNet connections running.

- *Error Code 0x10 – Device State Conflict*; this is returned when the requests being received by the Logix controller is faster than what should be sent out to the ControlNet IO devices (which can cause ControlNet scheduling bandwidth conflicts). There will be a 10 second break between each connection establishment request for a specific ControlNet node.

3.6. ETHERNET MODE

The ControlNet Router module supports five different modes for the Ethernet interface.



NOTE: When the Bridge is selected for the ControlNet mode, the Ethernet Mode will automatically be set to Target.

3.6.1. PCCC CLIENT

The module will enable Ethernet PCCC (AB-ETH) communication that allows the module to emulate a PLC5 or SLC providing a legacy interface for PanelViews and other legacy devices over scheduled ControlNet.



NOTE: When the PCCC Client mode is selected, the ControlNet Mode will automatically be set to Target.

The PCCC IP Address (the IP Address used to emulate a PLC5 controller) will need to be configured to allow the legacy device to communicate with the ControlNet Router using the PCCC Ethernet protocol.

This IP address will be seen as the IP address of the PLC5 controller that the ControlNet Router is emulating. Therefore, there will be two IP addresses on the network when in PLC5 Emulation mode. One for the actual target module and one for the emulated PLC5 controller.



NOTE: These two IP addresses must **not** be the same.



NOTE: The legacy device will need to use the PCCC IP Address as the target IP address and **not** the module IP address.

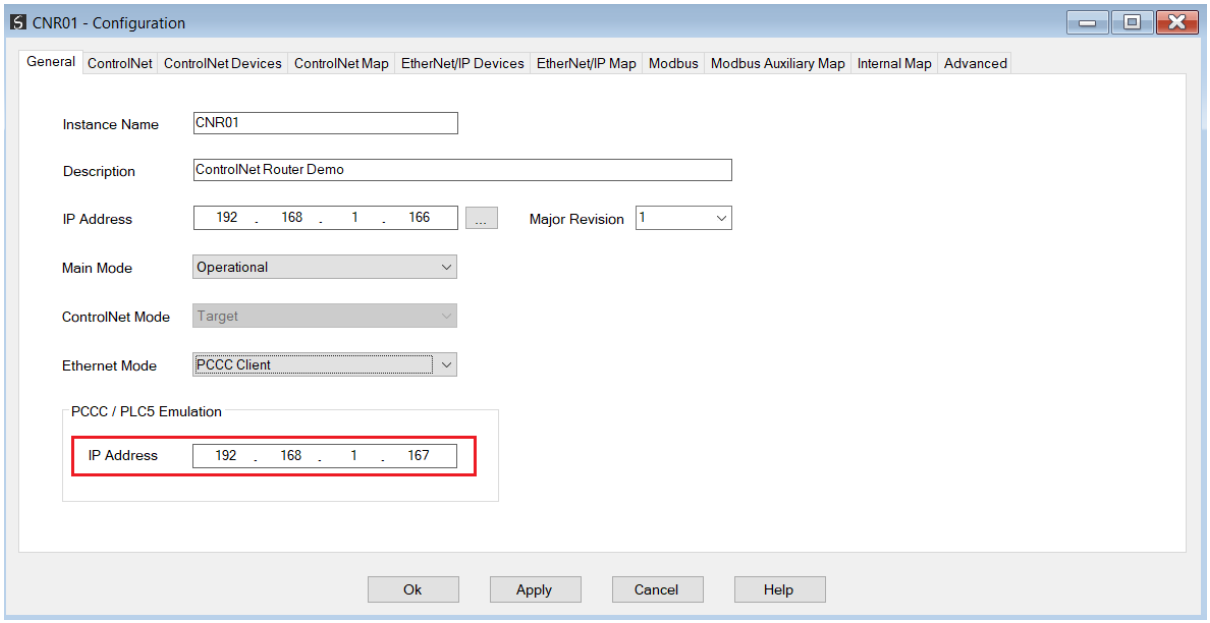


Figure 3.68 – PCCC Client Mode – IP Address Configuration

3.6.1.1. UNSCHEDULED CONTROLNET

The ControlNet Router can also be used to communicate with a Logix controller over ControlNet using unscheduled communication. This is done by browsing to the target Logix controller in RSLinx Enterprise and selecting it as the target. The user will now have direct access to all the Logix tags via the ControlNet Router (without the need for any PLC5 emulation). See the *FTView Configuration* section for more details regarding the communication path setup.

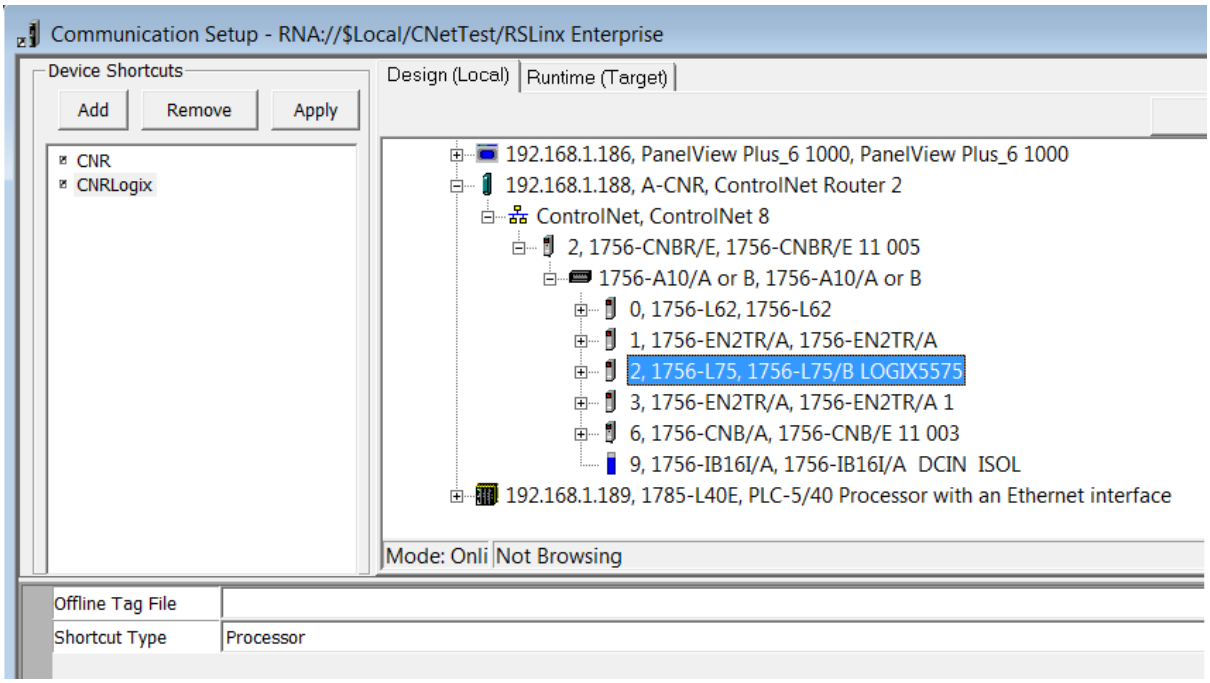


Figure 3.69 - Selecting Connection Path in RSLinx Enterprise

3.6.1.2. FTVIEW CONFIGURATION

The ControlNet Router can be interfaced directly to FTView using PCCC (PLC5 Ethernet emulation) or Unscheduled ControlNet. This is illustrated in the following examples where a PanelView is configured to read data from the ControlNet Router.

Using FTView Studio (Machine Edition) create a new project.

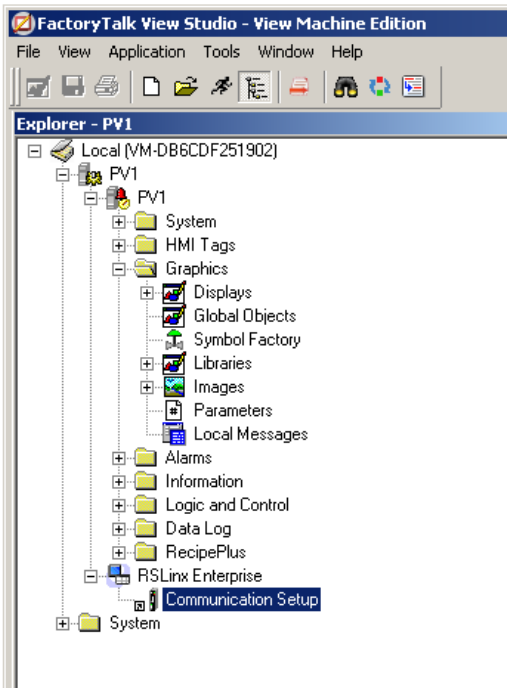


Figure 3.70 - FTView project

A. COMMUNICATION – SCHEDULED CONTROLNET

To configure the communication link to the ControlNet Router (when using Scheduled ControlNet), select the ***Communication Setup*** under the ***RSLinx Enterprise*** section. If the RSLinx Enterprise heading does not appear, then it should be added by right-clicking on the project and selecting ***Add New Server***.

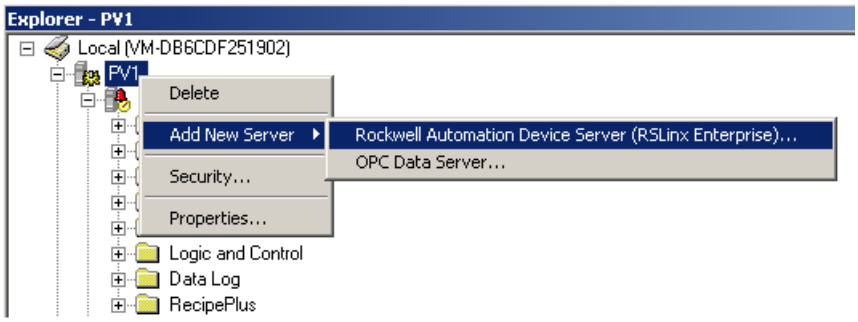


Figure 3.71 – Adding RSLinx Enterprise Server

The ***Communication Setup*** dialog will open. Under the ***Device Shortcuts*** group box, click on the ***Add*** button to create a new shortcut. Rename the shortcut as required. In this example the shortcut is renamed to “CNR”.

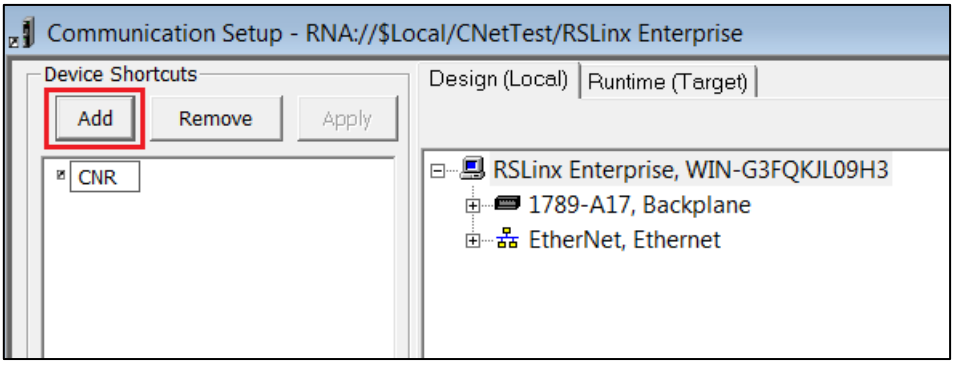


Figure 3.72 – Adding Device Shortcut

With the newly created device shortcut selected, right-click on the ***Ethernet*** network and select the ***Add Device*** option.

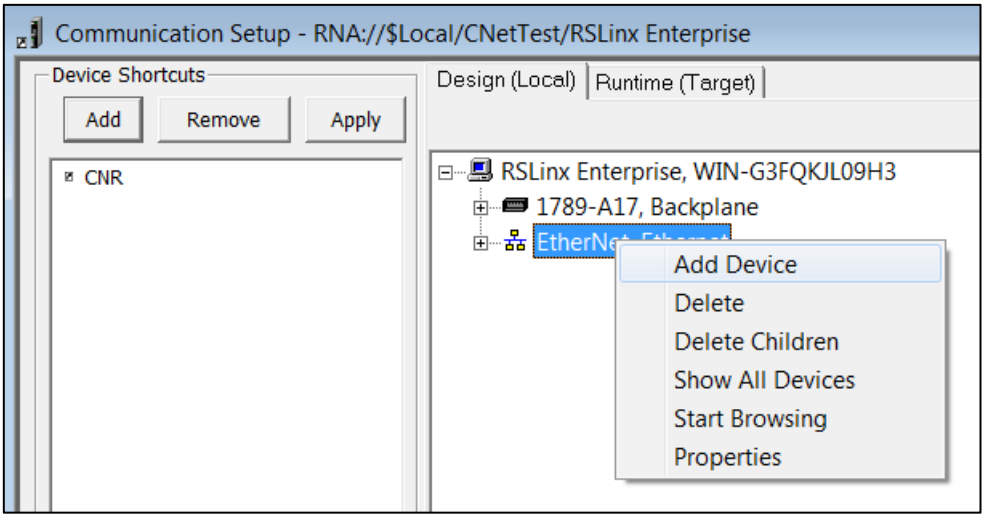


Figure 3.73 – Adding Ethernet Device

The ***Add Device*** dialog will open. Under the ***Ethernet PLC devices*** section, select the ***1785-L40E PLC-5/40 Processor with an Ethernet interface*** option.

NOTE: The ControlNet Router supports a PLC5 emulation mode, allowing it to be accessible by RSLinx Enterprise.

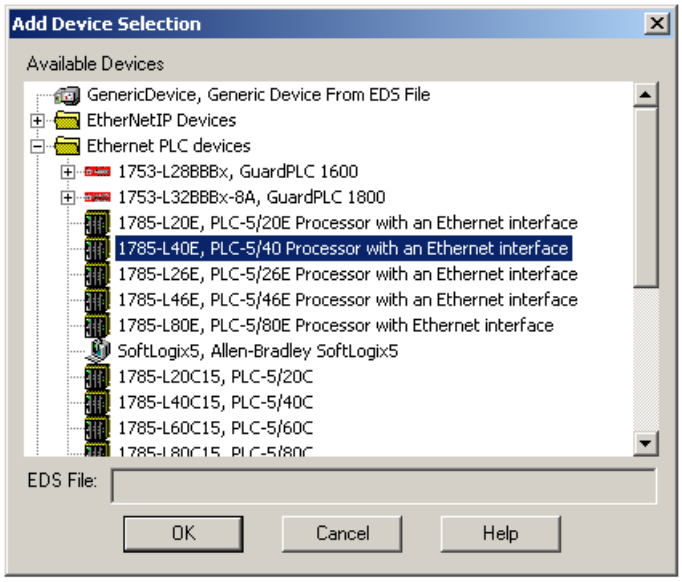


Figure 3.74 – Ethernet Device Selection

In the ***Device Properties*** page, enter the ControlNet Router's IP address, and then click on the Ok button.

NOTE: The user will need to enter the PLC5 emulation IP address for the PLC5 shown below, and **not** the ControlNet Router's primary (EtherNet/IP) IP address.

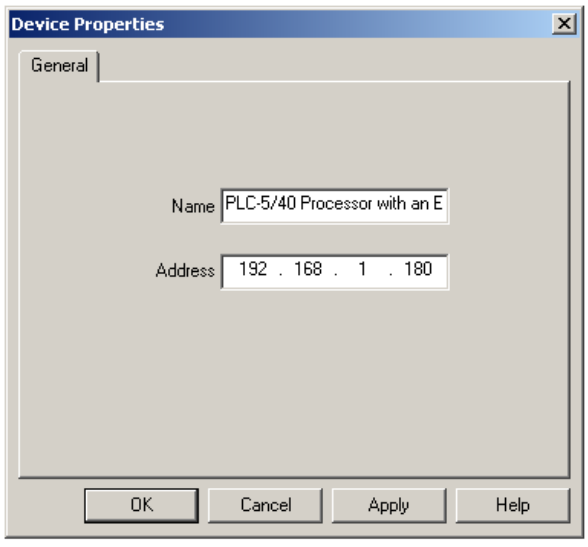


Figure 3.75 – Device Properties

At the top of the ***Communication Setup*** dialog, select the ***Copy from Design to Runtime*** button. As the name implies, this copies the configuration to be used by the PanelView at runtime. Select the ***Ok*** button to close the ***Communication Setup*** dialog.

B. COMMUNICATION – UNSCHEDULED CONTROLNET

To configure the communication link to the ControlNet Router (when using Unscheduled ControlNet), select the ***Communication Setup*** under the ***RSLinx Enterprise*** section. If the RSLinx Enterprise heading does not appear, then it should be added by right-clicking on the project and selecting ***Add New Server***.

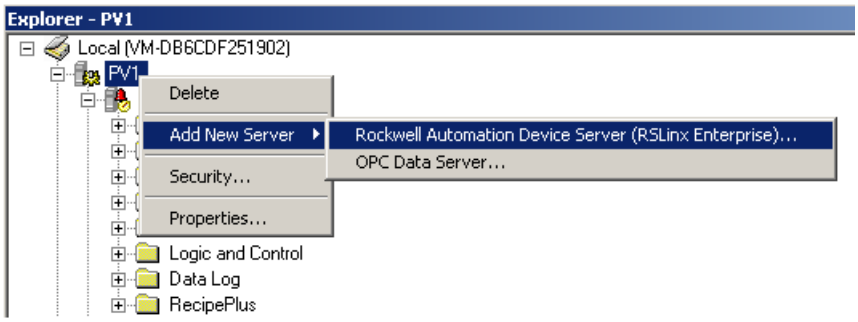


Figure 3.76 – Adding RSLinx Enterprise Server

The ***Communication Setup*** dialog will open. Under the ***Device Shortcuts*** group box, click on the ***Add*** button to create a new shortcut. Rename the shortcut as required. In this example the shortcut is renamed to "CNRLogix". Next browse to the Logix Controller using the ControlNet Router and select it.

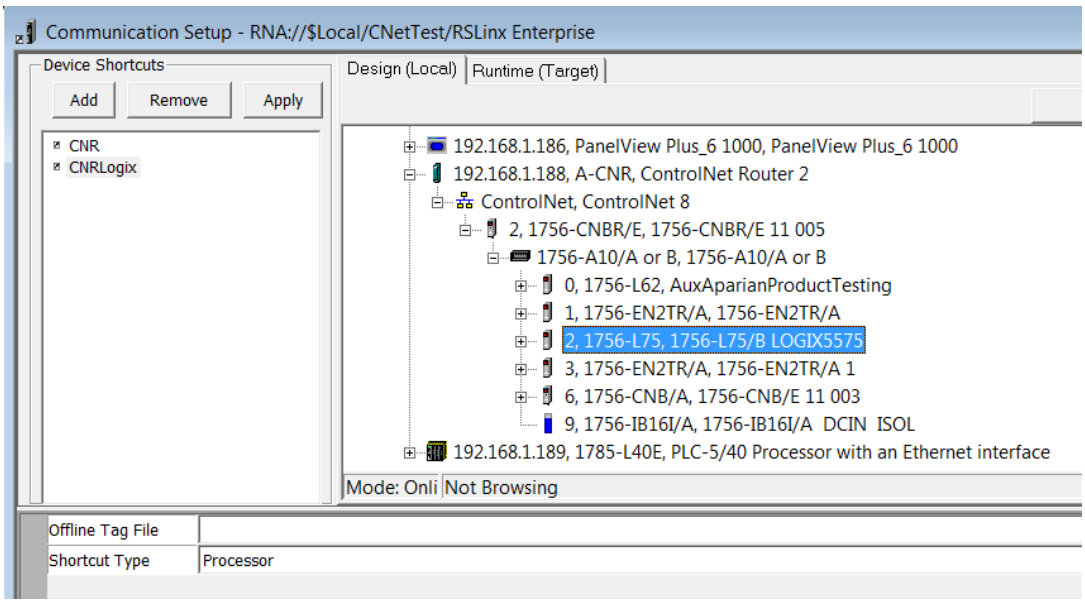


Figure 3.77 – Adding RSLinx Enterprise Server

C. ANIMATION

Once the communication has been correctly configured, objects can be linked to the ControlNet Router data points. Create a new graphic Display by right-clicking on the ***Display***

item, under the ***Graphics*** section.

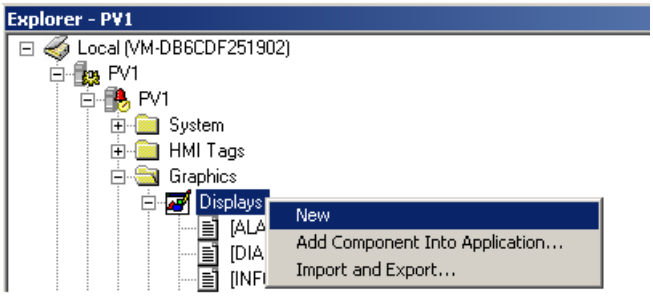


Figure 3.78 – Adding a Graphical Display

A blank Display dialog will be created. To display a number, select the ***Numeric Display*** object, from either the toolbar or from the ***Numeric and String*** menu, located under in ***Objects*** menu.

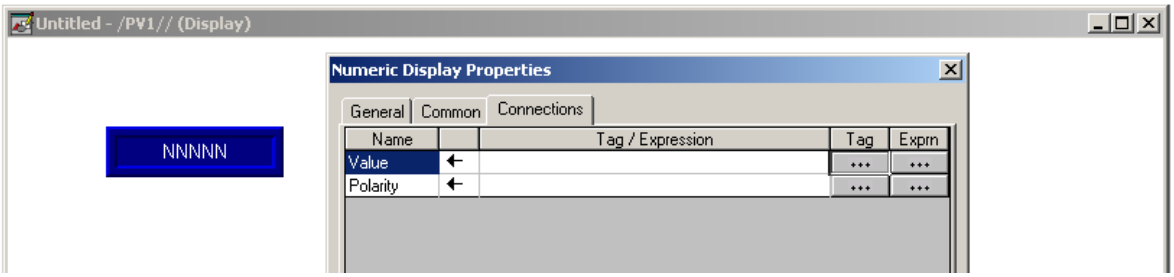


Figure 3.79 – Numeric Display Connections

The Numeric Display can now be linked to a ControlNet Router data point using the ***Connections*** tab. Select the ***Tag*** (…) button adjacent to the ***Value*** item. The FTView Tag Browser dialog will open. To view all the available data points, select the ***Refresh All Folders*** button.

NOTE: The ControlNet Router must be online for the tag browsing option to work.

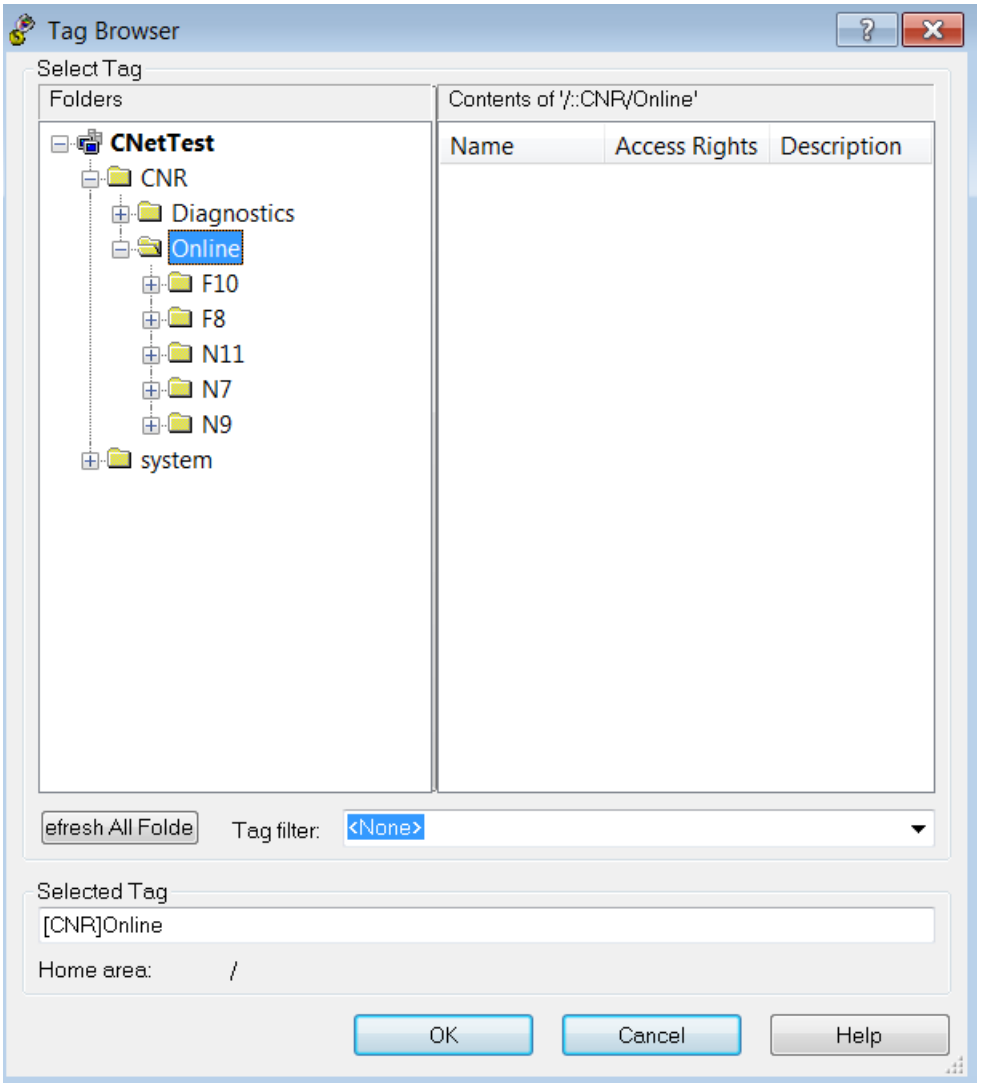


Figure 3.80 – Browsing Data Points

The data files, as configured in the ControlNet Router, will appear under the ***Online*** section. Select the required data point to be connected to the Numeric Display.

In this example, N9 and F10 represent the Input data, N7 and F8 represent the Output data, and N11 represents the diagnostic data.

D. DIAGNOSTIC DATA

Various diagnostic items can be displayed in the FTView using the Diagnostic File. The diagnostic file number is N11. The description of each diagnostic data point is tabulated below.

Offset	Group	Description
0	General	ControlNet Router Status

		Bit 0 – Config Valid Bit 1 – Routing Inhibited
1		ControlNet Node Number
2		ControlNet LED State
3		ControlNet Net Mode
4		ControlNet Router Internal Temperature
5	ControlNet Stats	Rx Packet Count
6		Tx Packet Count
7		Scheduled Tx Packet Count
8		Scheduled Rx Packet Count
9		Unscheduled Tx Packet Count
10		Unscheduled Rx Packet Count
11		Routed Class 3 Forward Open
12		Routed Class 3 Forward Close
13		Routed Class 3 Message
14		Routed UCMM Message
15	PCCC - Ethernet	PCCC Connection Requests
16		PCCC Read Requests
17		PCCC Write Requests
18		PCCC Unsupported Command
19		PCCC Unsupported FNC Code
20		PCCC Client Not Found
21		PCCC Client Max Reached
22		PCCC File Not Found
23		Current Connections

Table 3.10 - Diagnostic File

3.6.2. ETHERNET/IP TARGET

A controller (e.g. Logix controller) can own the ControlNet Router over EtherNet/IP using a Class 1 EtherNet/IP connection when the ControlNet Router is operating as an EtherNet/IP target. This will allow the ControlNet Router to exchange data with the controller using the input and output assembly of the Class 1 EtherNet/IP connection. Data from ControlNet devices can be mapped to the Logix controller over EtherNet/IP.

The user will need to add the ControlNet Router to the Logix IO tree under a EtherNet/IP bridge (e.g. 1756-EN2TR).

3.6.2.1. STUDIO / LOGIX 5000 CONFIGURATION

A. ADD MODULE TO ETHERNET/IP I/O CONFIGURATION

When operating in *EtherNet/IP Target* mode, the ControlNet Router can be added to the Logix 5000 I/O tree under an Ethernet bridge (e.g. EN2TR). The module will need to be added as a generic Ethernet module. This is done by right clicking on the Ethernet Bridge in the Logix 5000 and selecting *New Module* after which the *ETHERNET-MODULE* is selected to be added

as shown in the figure below.

NOTE: See the next section for importing the configuration (L5X).

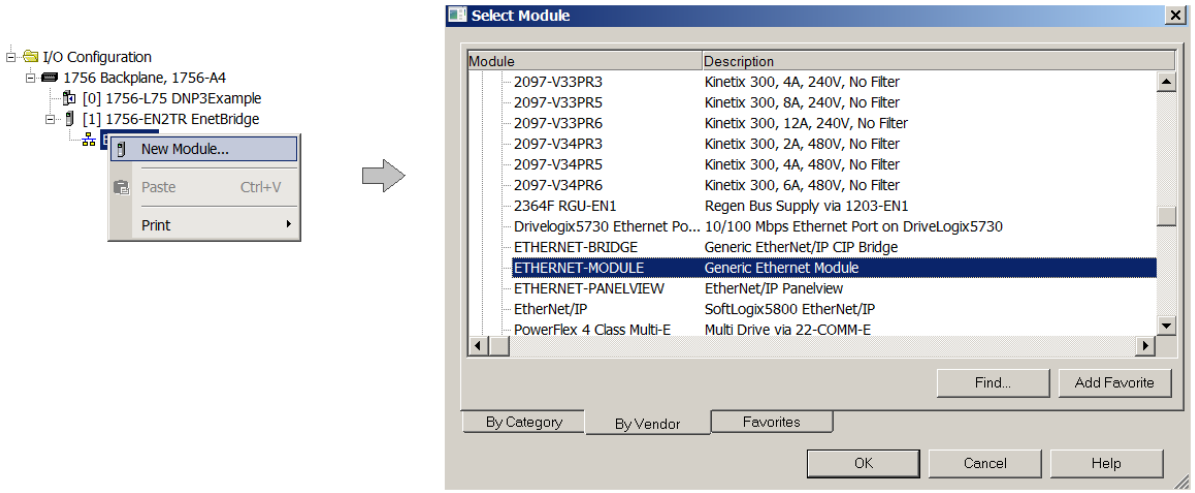


Figure 3.81 - Add a Generic Ethernet Module in Logix 5000

The user must enter the ***IP Address*** of the ControlNet Router that will be used. The assembly instance and size must also be added for the input, output, and configuration in the connection parameters section. Below are the required connection parameters.

Connection Parameter	Assembly Instance	Size
Input	166	500
Output	167	496
Configuration	102	0 (8-bit)

Table 3.11 – EtherNet/IP Logix class 1 connection parameters for the ControlNet Router

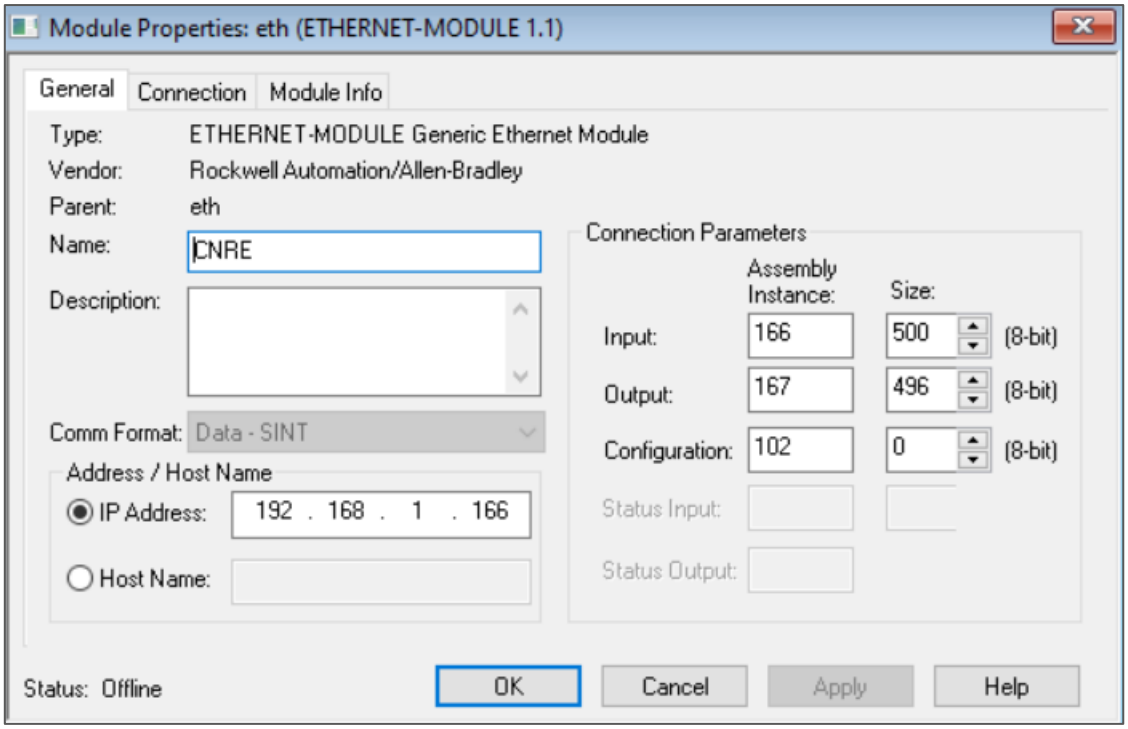


Figure 3.82 - Logix General module properties in Logix 5000

NOTE: The user will need to enter the exact connection parameters before the module will establish a class 1 connection with the Logix controller.

Next the user needs to configure the connection requested packet interval (RPI). This is the rate at which the input and output assemblies are exchanged. Refer to the technical specification section in this document for further details on the limits of the RPI.

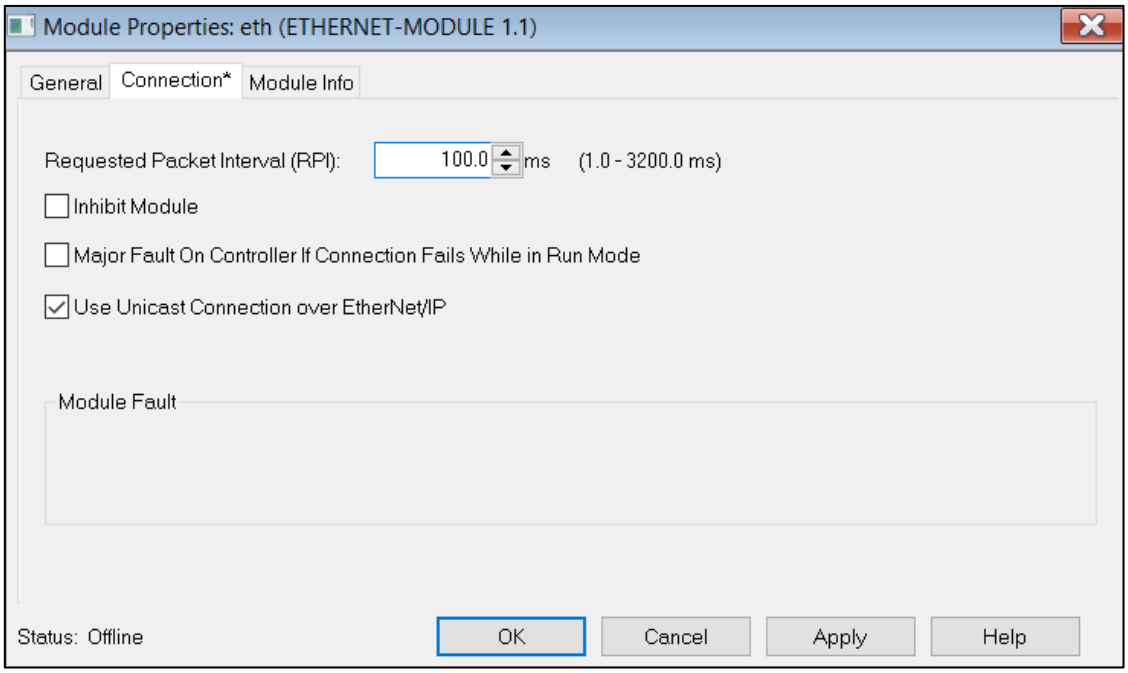


Figure 3.83 - Connection module properties in Logix 5000

Once the module has been added to the Logix 5000 I/O tree the user must assign the User Defined Types (UDTs) to the input and output assemblies. The user can import the required UDTs by right-clicking on ***User-Defined*** sub-folder in the ***Data Types*** folder of the IO tree and selecting ***Import Data Type***. The assemblies are then assigned to the UDTs with a ladder copy instruction (COP) as shown in the figure below.

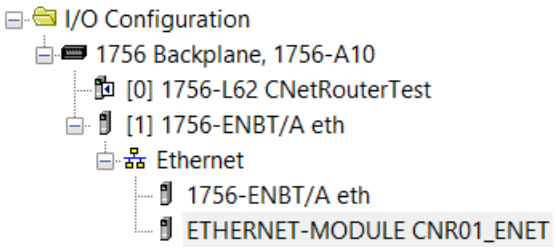


Figure 3.84 – Logix 5000 I/O module tree

B. IMPORTING UDTS AND MAPPING ROUTINES

To simplify the mapping of the input image, a Logix 5000 Routine Partial Import (L5X) file is provided. This file can be imported by right-clicking on the required Program and selecting the Import Routine option.

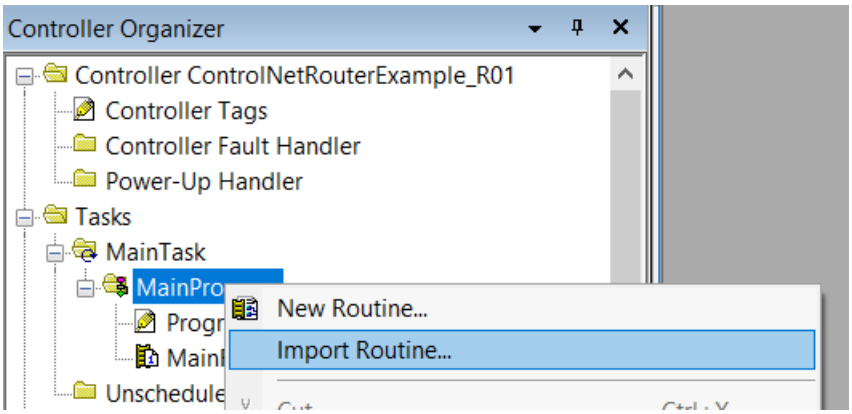


Figure 3.85 – Logix 5000 Importing ControlNet Router specific routine and UDTs

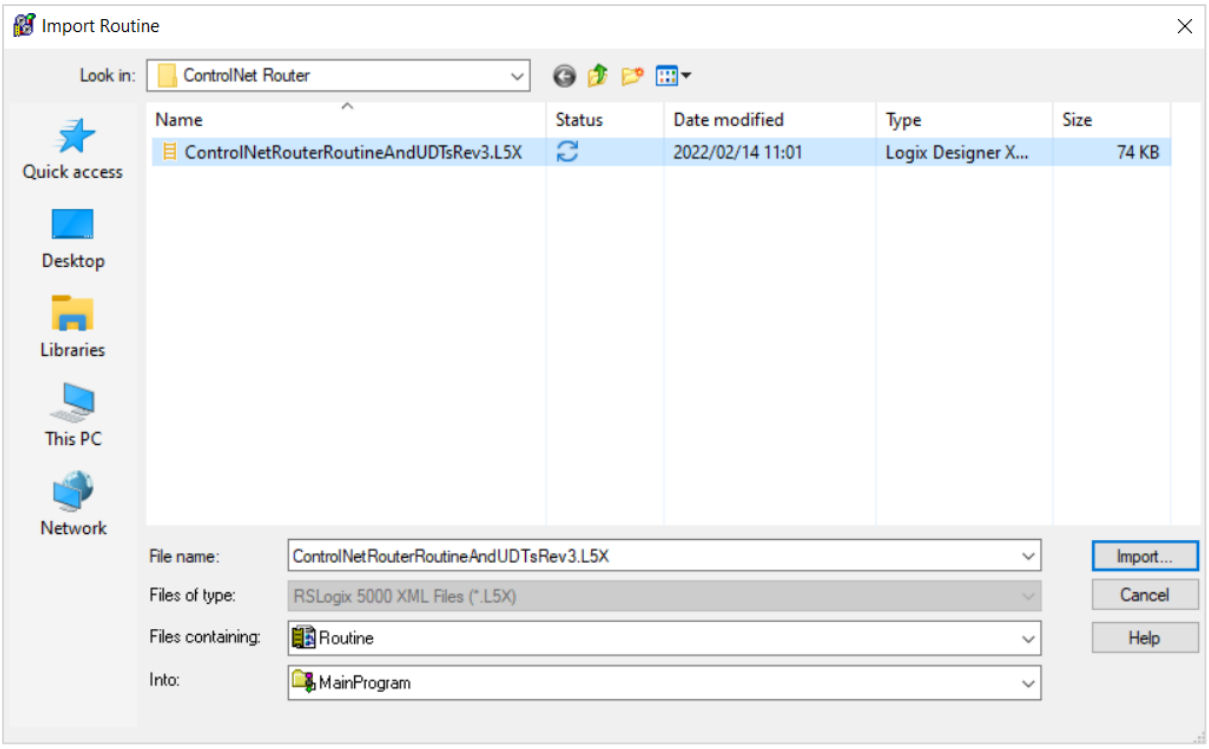


Figure 3.86 - Selecting partial import file

The import will create the following:

- The required UDTs (user defined data types)
- Two controller tags representing the Input and Output assemblies.
- A routine mapping the ControlNet Router module to the aforementioned tags.

The user may need to change the routine to map to the correct ControlNet Router module instance name, and make sure that the mapping routine is called by the Program's Main Routine.

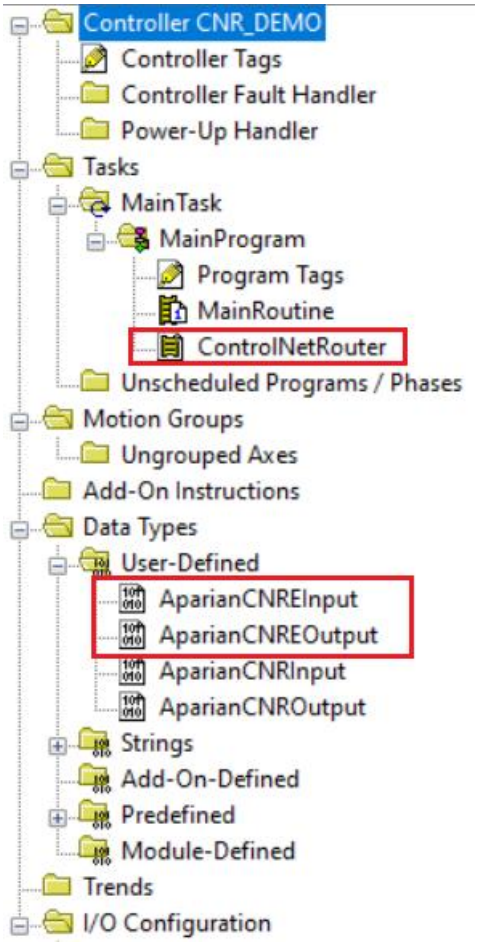


Figure 3.87 - Imported Logix 5000 objects

Refer to the additional information section of this document for an example Logix 5000 project as well as the required UDTs.

3.6.2.2. INTERNAL DATA SPACE MAPPING

When the module is operating as a EtherNet/IP Target, the data from the originator device (e.g. Logix Controller) can be mapped to the ControlNet interface using the Internal Map. The Internal Map configuration window is opened by either double clicking on the module in the tree or right-clicking the module and selecting *Configuration* and selecting the *Internal Map* tab.

A. IDS COPY – ETHERNET/IP TARGET SOURCE

When copying data from a connection originator (e.g. the output assembly from the Logix Controller) to the ControlNet interface, the source type needs to be EIP Target.

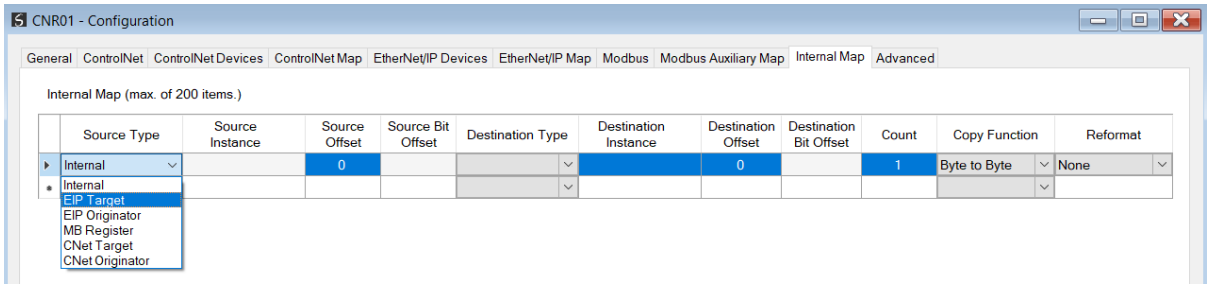


Figure 3.88 – IDS Copy – EtherNet/IP Target Source Type

The source instance will be the connection number, which in this case is always *Connection 0*. The Source Offset is the offset in the *Mapped Data* section of the EtherNet/IP output assembly from where the data must be copied. The Count is the number of **bytes** that will be copied. The *Mapped Data* section will be available when using the provided UDT for the Output Assembly.

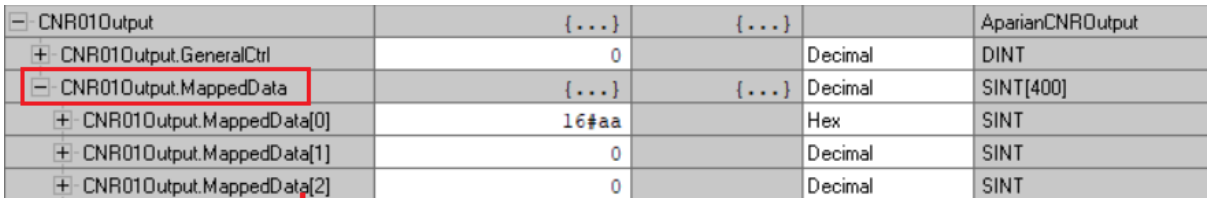


Figure 3.89 – EtherNet/IP Output Mapped Data UDT

See the Internal Data Space Mapping section for more information regarding the operation.

B. IDS COPY – ETHERNET/IP TARGET DESTINATION

When copying data from the ControlNet interface to the EtherNet/IP Target input assembly, the destination type needs to be EIP Target.

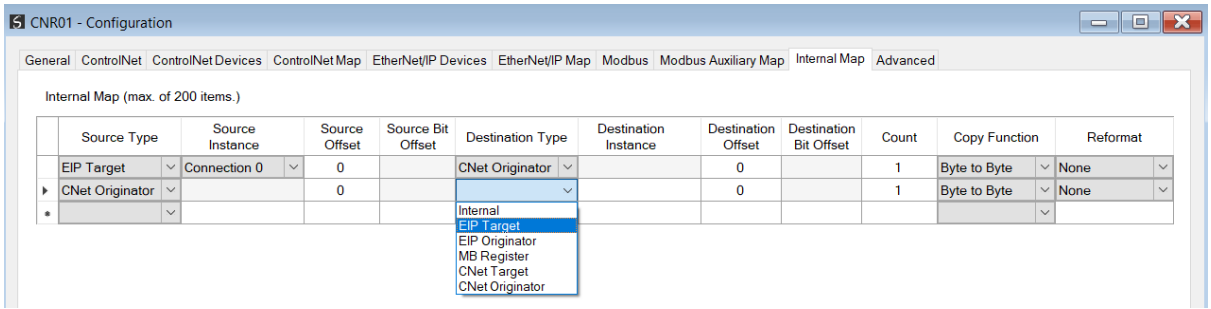


Figure 3.90 – IDS Copy – EtherNet/IP Target Destination Type

The destination instance will be the connection number, which in this case is always *Connection 0*. The Destination Offset is the offset in the *Mapped Data* section of the EtherNet/IP input assembly from where the data must be copied. The Count is the number of **bytes** that will be copied. The *Mapped Data* section will be available when using the provided UDT for the Input Assembly.

CNR01Input	{...}	{...}		AparianCNRInput
CNR01Input.GeneralStatus	-79		Decimal	SINT
CNR01Input.CNetNodeNumber	1		Decimal	SINT
CNR01Input.CNetLedState	1		Decimal	SINT
CNR01Input.CNetMode	6		Decimal	SINT
CNR01Input.Temperature	37.30043		Float	REAL
CNR01Input.MappedData	{...}	{...}	Decimal	SINT[400]
CNR01Input.MappedData[0]	2		Decimal	SINT
CNR01Input.MappedData[1]	0		Decimal	SINT

Figure 3.91 – EtherNet/IP Input Mapped Data UDT

3.6.3. Modbus TCP Slave

The ControlNet Router can operate as a Modbus Slave for Modbus TCP. A Modbus TCP Master can read and write to the full Modbus Register range in the ControlNet Router. The user will need to configure the relevant Modbus Parameters as shown below.

The Modbus configuration window is opened by either double clicking on the module in the tree or right-clicking the module and selecting *Configuration* and selecting the *Modbus* tab.

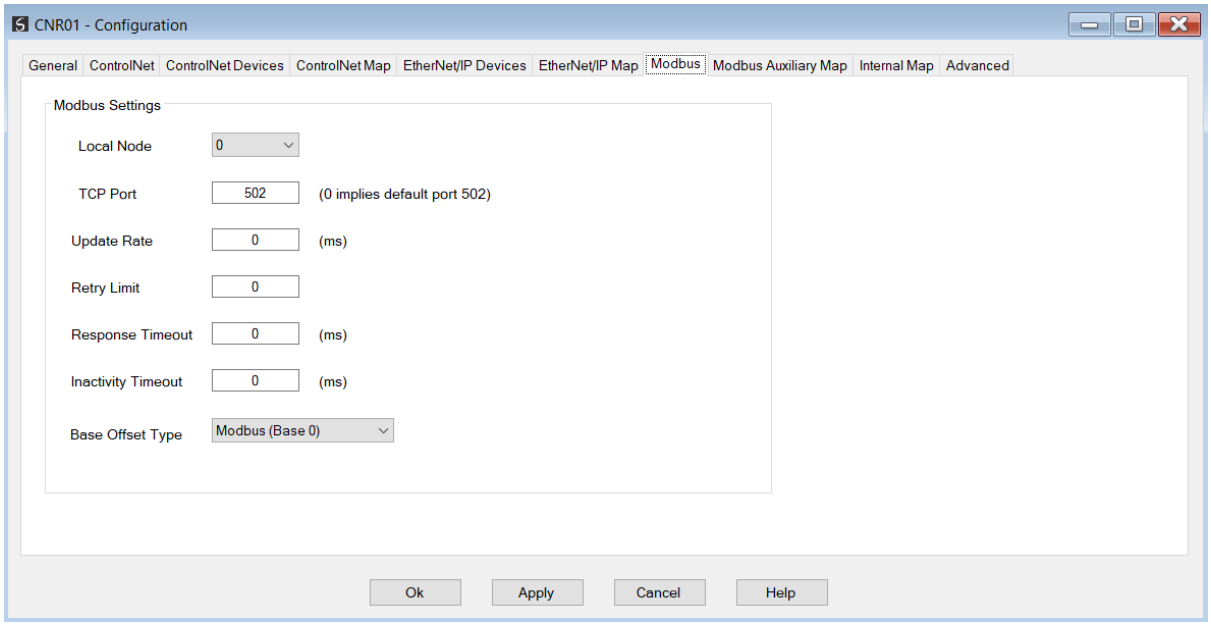


Figure 3.92 – Modbus Configuration

The Modbus Communication configuration consists of the following parameters:

Parameter	Description
Local Node	The Modbus Node address assigned to the ControlNet Router.
Modbus TCP Port	The TCP port to be used for the Modbus communication. By default the module will use the standard TCP port 502.

Update Rate	The period (in milliseconds) between master requests to the target Modbus device. **(Modbus TCP Master mode only)**
Retry Limit	The number of successive Modbus request retries before the request is set to have failed. **(Modbus TCP Master mode only)**
Response Timeout	The time (in milliseconds) the module will wait for a valid Modbus response. **(Modbus TCP Master mode only)**
Inactivity Timeout	The amount of time during which no Modbus requests have been received before the ControlNet Router indicates that the connection to the Modbus Master is no longer active. **(Modbus TCP Slave mode only)**
Base Offset Type	**Modbus (Base 0)** Conventional Modbus addressing where the first address is 0. **PLC (Base 1)** PLC addressing, where the first address is 1.

Table 3.12 – Modbus parameters

The Modbus Node Number will need to be configured in the parameters above to allow a Modbus Master to access the ControlNet Router as a Modbus Slave device.

3.6.3.1. INTERNAL DATA SPACE MAPPING

When the module is operating as a Modbus TCP Slave, the data from the Modbus Registers (used to exchange data with the Modbus TCP Master) can be mapped to the ControlNet interface using the Internal Map. The Internal Map configuration window is opened by either double clicking on the module in the tree or right-clicking the module and selecting *Configuration* and selecting the *Internal Map* tab.

A. IDS COPY – MODBUS SOURCE

When copying Modbus data to the ControlNet interface, the source type needs to be MB Register.

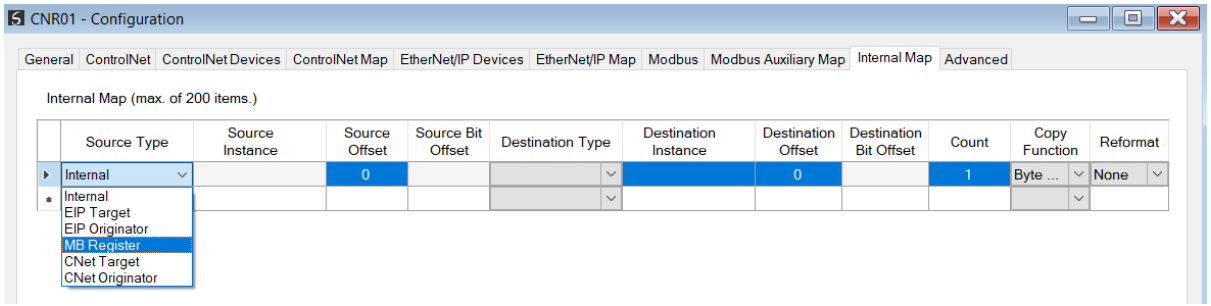


Figure 3.93 – IDS Copy - Modbus Source Type

The source instance will be the Modbus register type required.

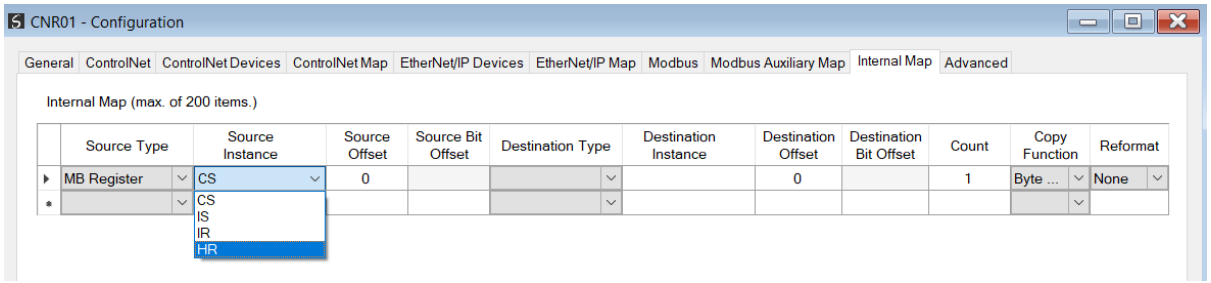


Figure 3.94 – IDS Copy - Modbus Source Instance

The Source Offset is the Modbus Register offset from where the data must be copied. The Count is the number of **bytes** that will be copied. See the Internal Data Space Mapping section for more information regarding the operation.

B. IDS COPY – MODBUS DESTINATION

When copying data from the ControlNet interface to a Modbus Register, the destination type needs to be MB Register.

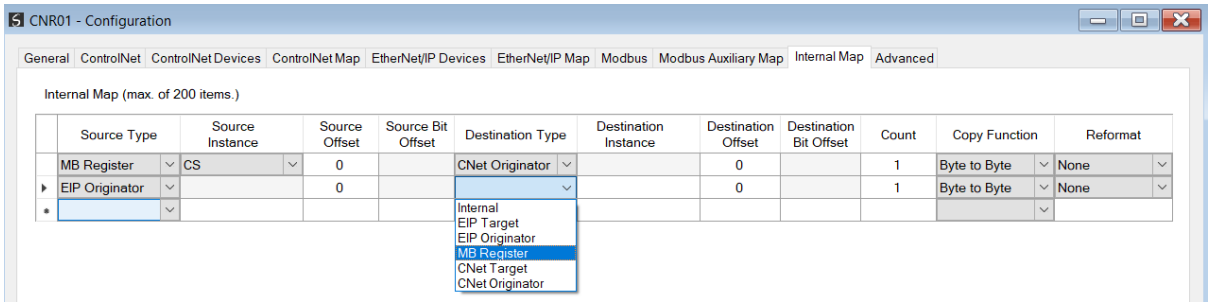


Figure 3.95 – IDS Copy - Modbus Destination Type

The destination instance will be the Modbus register type required.

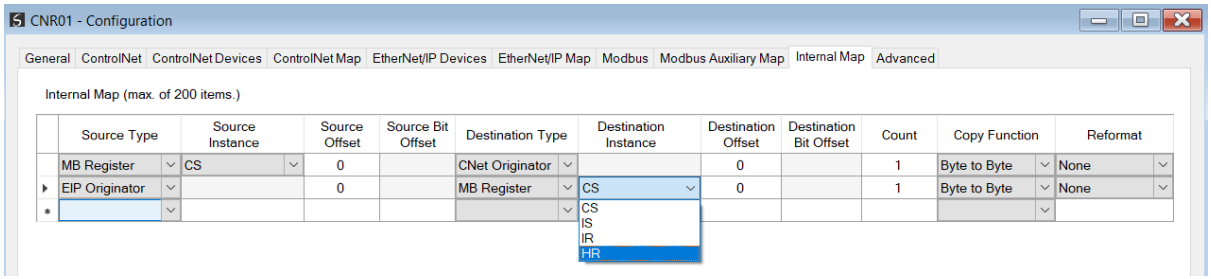


Figure 3.96 – IDS Copy - Modbus Destination Instance

The Destination Offset is the Modbus Register offset to where the data must be copied. The Count is the number of **bytes** that will be copied. See the Internal Data Space Mapping section for more information regarding the operation.

3.6.4. MODBUS TCP MASTER

The ControlNet Router can operate as a Modbus Master for Modbus TCP. The user will need to configure the relevant Modbus Parameters as shown below followed by the configuration of the Modbus Auxiliary Map. This map will allow the user configure various read and write functions to external Modbus Registers to and from the internal Modbus registers.

The Modbus configuration window is opened by either double clicking on the module in the tree or right-clicking the module and selecting *Configuration* and selecting the *Modbus* tab.

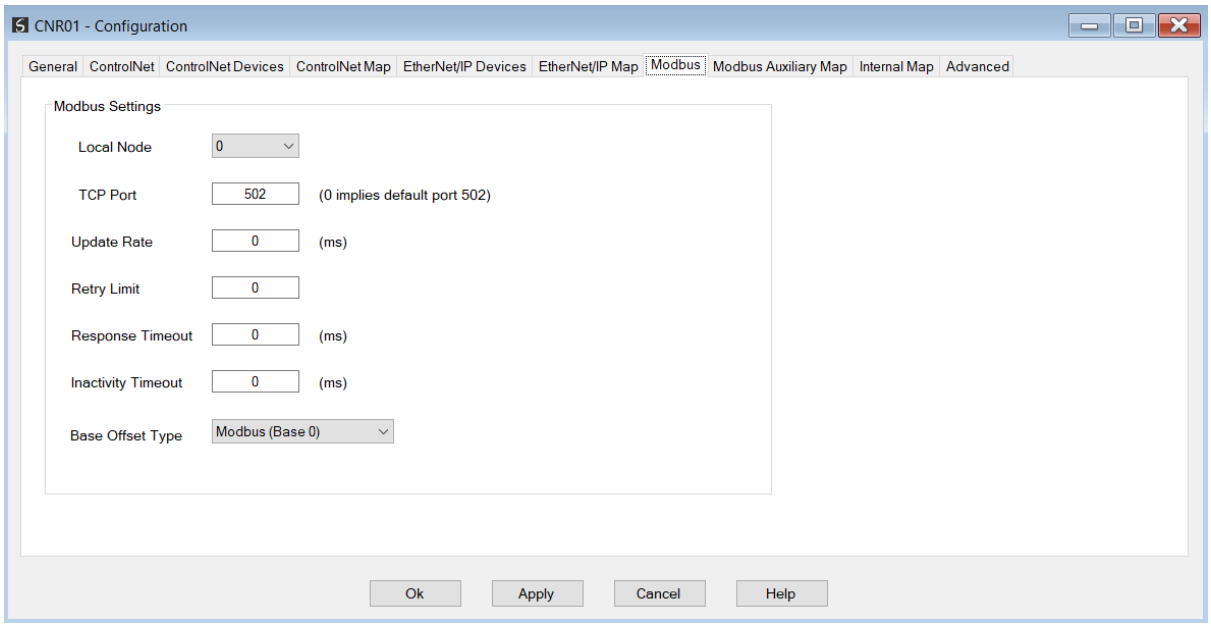


Figure 3.97 – Modbus Configuration

The Modbus Communication configuration consists of the following parameters:

Parameter	Description
Local Node	The Modbus Node address assigned to the ControlNet Router.
Modbus TCP Port	The TCP port to be used for the Modbus communication. By default the module will use the standard TCP port 502.
Update Rate	The period (in milliseconds) between master requests to the target Modbus device. **(Modbus TCP Master mode only)**
Retry Limit	The number of successive Modbus request retries before the request is set to have failed. **(Modbus TCP Master mode only)**
Response Timeout	The time (in milliseconds) the module will wait for a valid Modbus response. **(Modbus TCP Master mode only)**
Inactivity Timeout	The amount of time during which no Modbus requests have been received before the ControlNet Router indicates that the connection to the Modbus Master is no longer active. **(Modbus TCP Slave mode only)**
Base Offset Type	**Modbus (Base 0)** Conventional Modbus addressing where the first address is 0. **PLC (Base 1)** PLC addressing, where the first address is 1.

Table 3.13 – Modbus parameters

3.6.4.1. MODBUS AUXILIARY MAP

The Modbus Auxiliary Map configuration is shown in the figure below. The Modbus configuration is only applicable when the module has a Modbus Master operating interface. Up to 100 mapping items can be configured while communicating to up to 20 Modbus Slave devices.

The Modbus Aux Map will be executed in a sequential manner and a mapped item will be executed at the *Update Rate* in the Modbus parameters. That is, the *Update Rate* is the time between two successive mapped item executions.

The Modbus Auxiliary Map configuration window is opened by either double clicking on the module in the tree or right-clicking the module and selecting *Configuration*.

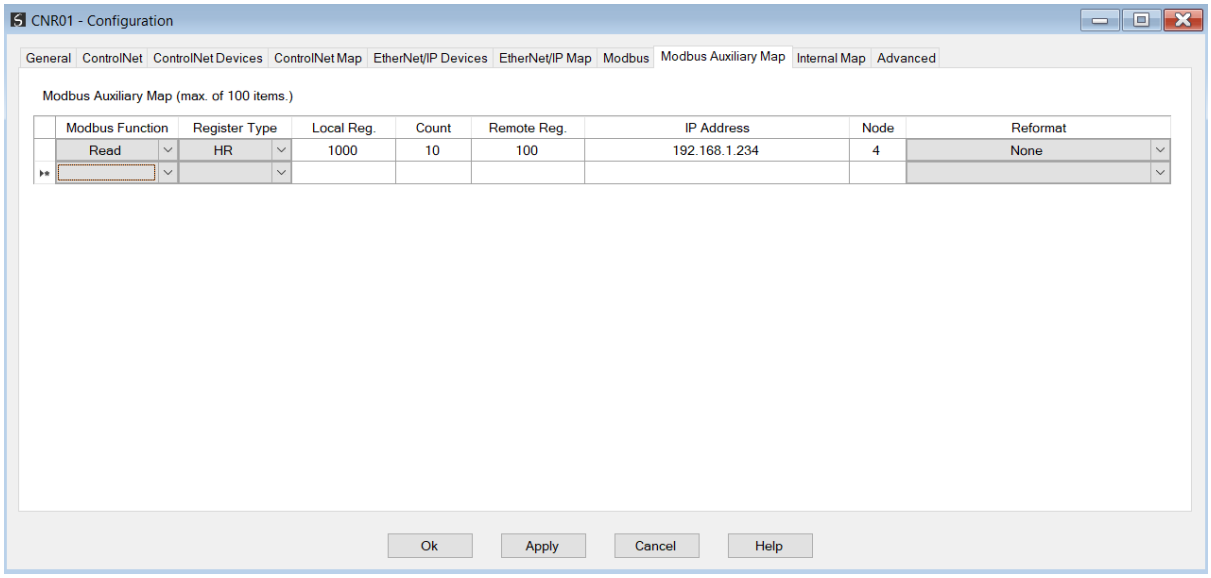


Figure 3.98 – Modbus Auxiliary Map Configuration

The Modbus Auxiliary Map configuration consists of the following parameters:

Parameter	Description
Modbus Function	This is the Modbus function that is sent to the Modbus Slave. **Read** – Read a Modbus Register (e.g. HR, IR, CS, or IS) from a Modbus Slave. **Write** – Write a Modbus Register (e.g. HR or CS) to a Modbus Slave.
Register Type	Modbus Register Type: **CS** – Coil Status **IS** – Input Status **IR** – Input Register **HR** – Holding Register
Local Reg.	The local (internal) ControlNet Router Modbus register address.
Count	The number of Modbus elements to read or write.
Remote Reg.	The remote slave Modbus address register.
IP Address	The IP address of the remote Modbus TCP slave.
Node	The Modbus Node address of the remote Modbus slave.
Reformat	Used to specify how the data is formatted before writing to, or after reading from, the Modbus slave. **None** – No reformatting applied. (AA BB CC DD). **BB AA** – 16bit Byte swap **BB AA DD CC** – 32bit Byte Pair Swap **CC DD AA BB** – Word Swap **DD CC BB AA** – Word and Byte Pair Swap

Table 3.14 – Modbus Auxiliary Map parameters

3.6.4.2. INTERNAL DATA SPACE MAPPING

When the module is operating as a Modbus TCP Master, the data from the Modbus Registers (used to exchange data with various Modbus TCP Slaves) can be mapped to the ControlNet interface using the Internal Map. The Internal Map configuration window is opened by either double clicking on the module in the tree or right-clicking the module and selecting *Configuration* and selecting the *Internal Map* tab.

A. IDS COPY – MODBUS SOURCE

When copying Modbus data to the ControlNet interface, the source type needs to be MB Register.

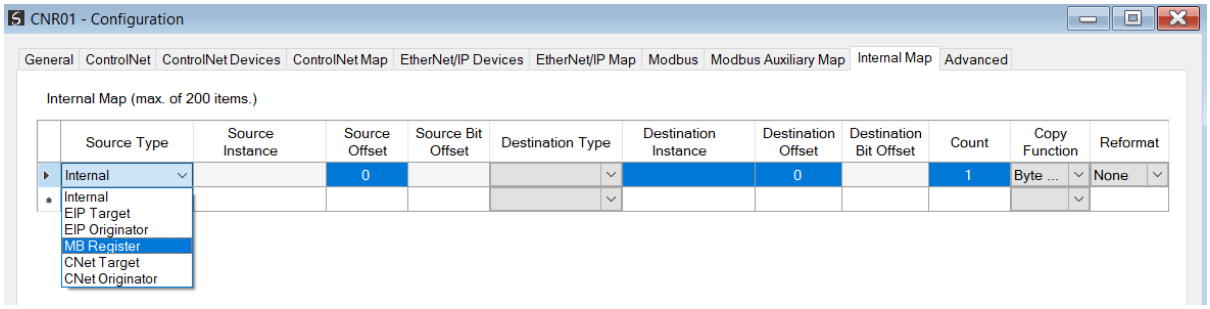


Figure 3.99 – IDS Copy - Modbus Source Type

The source instance will be the Modbus register type required.

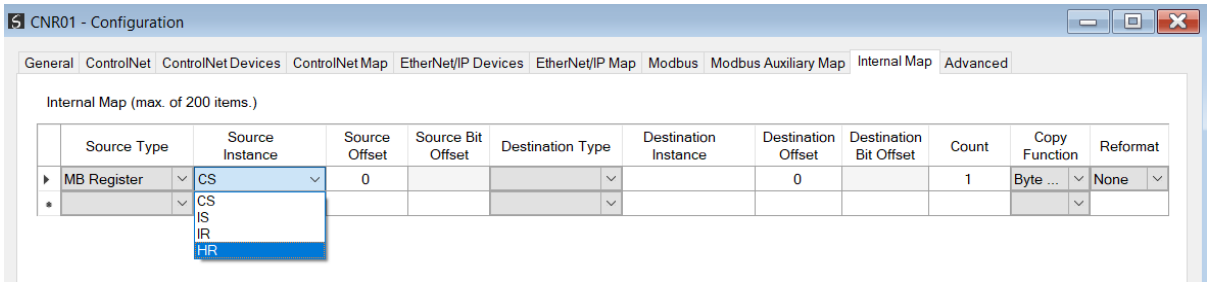


Figure 3.100 – IDS Copy - Modbus Source Instance

The Source Offset is the Modbus Register offset from where the data must be copied. The Count is the number of **bytes** that will be copied. See the Internal Data Space Mapping section for more information regarding the operation.

B. IDS COPY – MODBUS DESTINATION

When copying data from the ControlNet interface to a Modbus Register, the destination type needs to be MB Register.

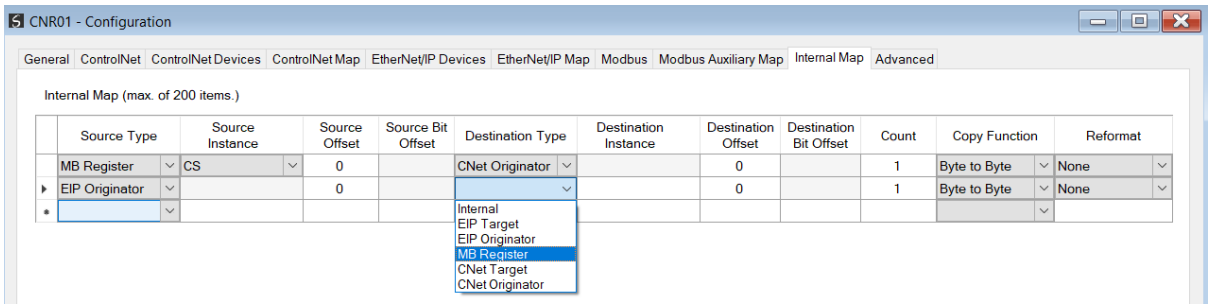


Figure 3.101 – IDS Copy - Modbus Destination Type

The destination instance will be the Modbus register type required.

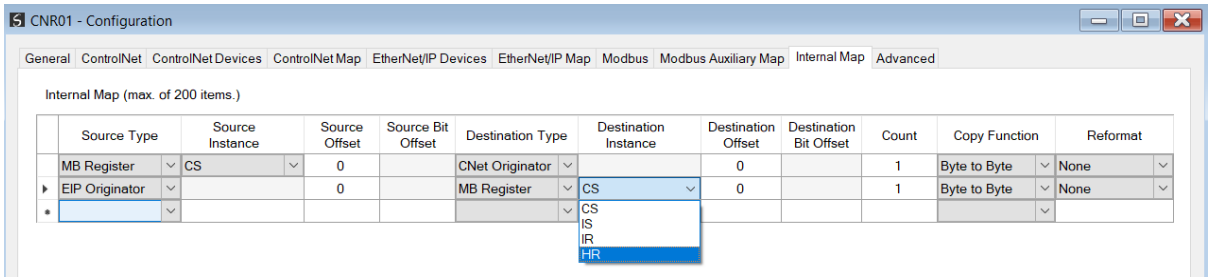


Figure 3.102 – IDS Copy - Modbus Destination Instance

The Destination Offset is the Modbus Register offset to where the data must be copied. The Count is the number of **bytes** that will be copied. See the Internal Data Space Mapping section for more information regarding the operation.

3.6.5. EtherNet/IP Originator

The ControlNet Router can operate as a EtherNet/IP connection originator for cyclic (Class 1) or explicit (Class 3 or UCMM) data exchange. The explicit messaging can be setup in the *EtherNet/IP Devices* and *EtherNet/IP Map* in the Master configuration while the cyclic class 1 connections are added to the *EtherNet/IP Connections* node under the module in the Slate project tree.

3.6.5.1. ETHERNET/IP CLASS 1 DEVICE CONNECTIONS

The ControlNet Router can establish up to 16 cyclic Class 1 EtherNet/IP connections to EtherNet/IP devices. This can be done by either manually entering the connection data into the Connection Parameter window, or by importing the configuration from one or more of the following sources:

- Online Logix Controller
- Logix Controller L5X
- EDS File

- Connection Library

A. MANUAL CONFIGURATION

A class 1 connection can be added to the *EtherNet/IP Connections* tree by right-clicking on the tree in Slate and selecting *Add EtherNet/IP Connection.*

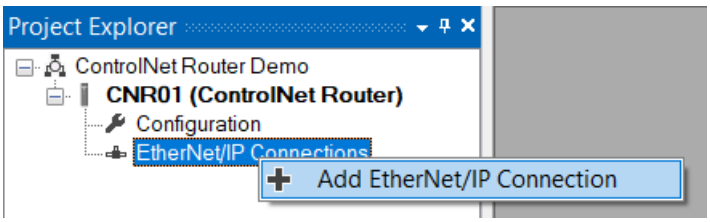


Figure 3.103 – Adding EtherNet/IP Class 1 Connection

Next the user will need to enter the connection parameters for the Class 1 connection.

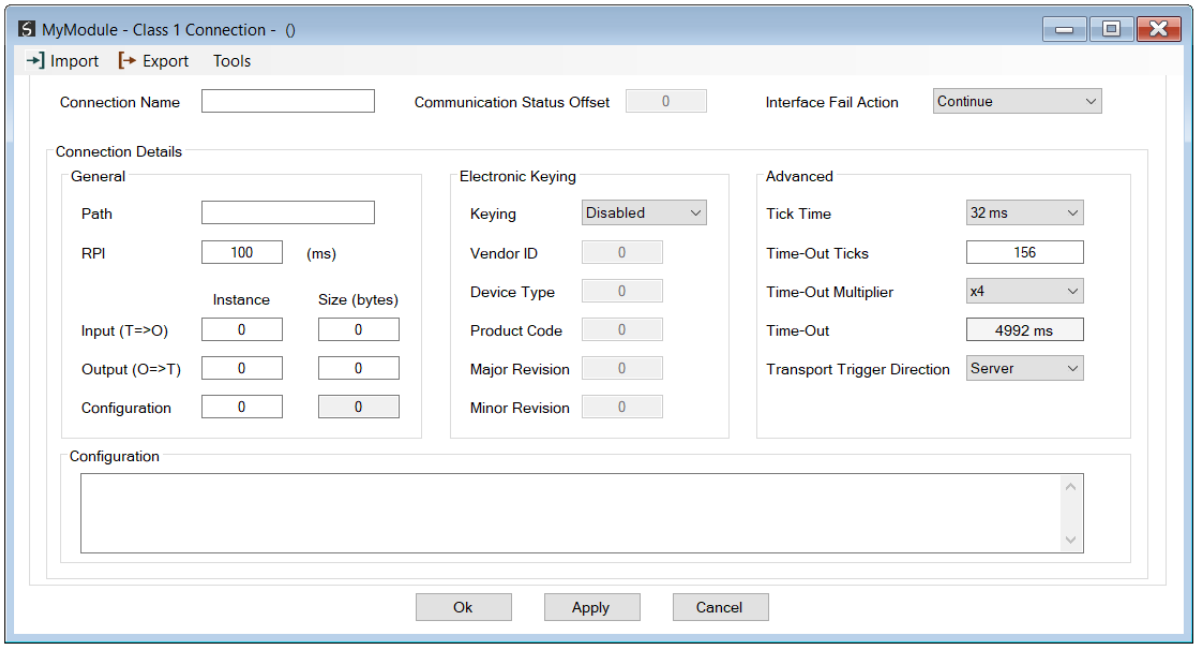


Figure 3.104 – EtherNet/IP Class 1 Connection Parameters

NOTE: It is recommended that the user not change the values in the *Advanced* frame of the connection parameters.

Parameter	Description
Connection Name	The instance name given to the Class 1 Connection.
Comm Status Offset	This is the offset in the Internal Data Space (IDS) which provides the communication status of each EtherNet/IP device. The Communication Status is as shown below:

	Bit 0 - (1: Device online / 0:Device offline) Bit 1 to 7 – Reserved.
Interface Fail Action	When the ControlNet communication has failed, the EtherNet/IP IO can be configured to either keep the connection running as is, change the connection status to program mode, or force the connection offline. This will allow the EtherNet/IP device to go into a pre-determined state when the communication to the controller (i.e., connection originator on ControlNet) is lost.
General	
Path	The path to the target device. If the device is an Ethernet device then this will just be the IP address of the module. If the device is, for example, a module in a backplane or via an adapter, then the user will need to enter the IP address of the bridge or adapter followed by the backplane port (for example 1) and the slot number of the device. Each item is separated by a comma. As an example; to connect to an Allen-Bradley Flex module (via the Flex Adapter at IP address 192.168.1.100) that is in slot 2 of the Flex backplane, the user will need to enter the following path: 192.168.1.100,1,2 (IP address, port (backplane), slot).
RPI	The requested packet interval (RPI) is the rate in milliseconds at which the data will be sent from the originator to the target and vice versa.
Input (T=>O) – Instance	The instance of the input assembly.
Input (T=>O) – Size (bytes)	The size in bytes of the input assembly.
Output (O=>T) – Instance	The instance of the output assembly.
Output (O=>T) – Size (bytes)	The size in bytes of the output assembly.
Configuration – Instance	The instance of the configuration assembly.
Configuration – Size (bytes)	The size in bytes of the configuration assembly. **NOTE:** This is a read-only value and will be equal to the number of bytes entered into the configuration window below.
Electronic Keying	
Keying	Electronic Keying can be used to ensure that the target device is the correct device type. **Disabled** Keying is not enabled and no key information will be sent in the connection establishment. **Compatible** Keying has been enabled with compatibility enabled. This will allow devices with older firmware to also establish a connection. **Exact** Keying has been enabled and the exact device with specific firmware revision

	will allow the establishment of the connection.
Vendor ID	The Vendor ID of the target device.
Device Type	The Device Type of the target device.
Product Code	The Product Code of the target device.
Major Revision	The Major Revision of the target device.
Minor Revision	The Minor Revision of the target device.
Advanced (Note: Changing these values is not recommended)	
Tick Time	For unconnected messages, this is the time for each tick to calculate the unconnected Time-Out time.
Time-Out Ticks	The number of ticks before the unconnected message is set for timeout.
Time-Out Multiplier	This is the multiplier of the RPI to define the connection timeout time.
Time-Out	The unconnected message timeout time (read-only)
Transport Trigger Direction	The Transport Trigger direction; **Server** or **Client**.
Configuration	
Data	The configuration data that is sent with the forward open connection establishment. The data will need to be entered as a space-delimited, hexadecimal string. For example: 0A 0D 12 EE The configuration size will increase by one each time a byte is added to the configuration.

Table 3.15 – EtherNet/IP Class 1 Connection Parameters

B. IMPORT FROM ONLINE CONTROLLER

Here the EtherNet/IP connection parameters are imported directly from an online Logix controller.

PREPARATION

Before the connection information can be imported, some preparation is required using Studio5000 and a Logix controller:

1. In Studio5000 create a new project and add the required EtherNet/IP device in the IO tree. If the device's profile supports configuration, then configure the device as required.
2. Download the project to a Logix controller.



NOTE: When instantiating modules in Studio5000 do not make use of the "Rack Optimization" communication format.

NOTE: Some versions Logix (V32+) do not support the reading of the module's configuration. Where possible use an earlier version (e.g. V24).

NOTE: It is possible that not all the connection information will be imported as it may not be available due to the type of device and Logix version.

IMPORT CONNECTION PARAMETERS

The connection parameters can be imported from the Logix controller by selecting the ***Import from Online Controller*** option located under the ***Import*** menu of the Class 1 Connection form.

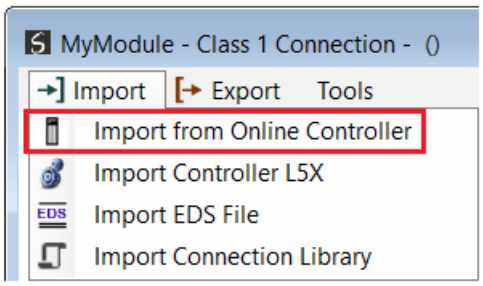


Figure 3.105 – Import from Online Controller

The Import Connection Parameters form will open.

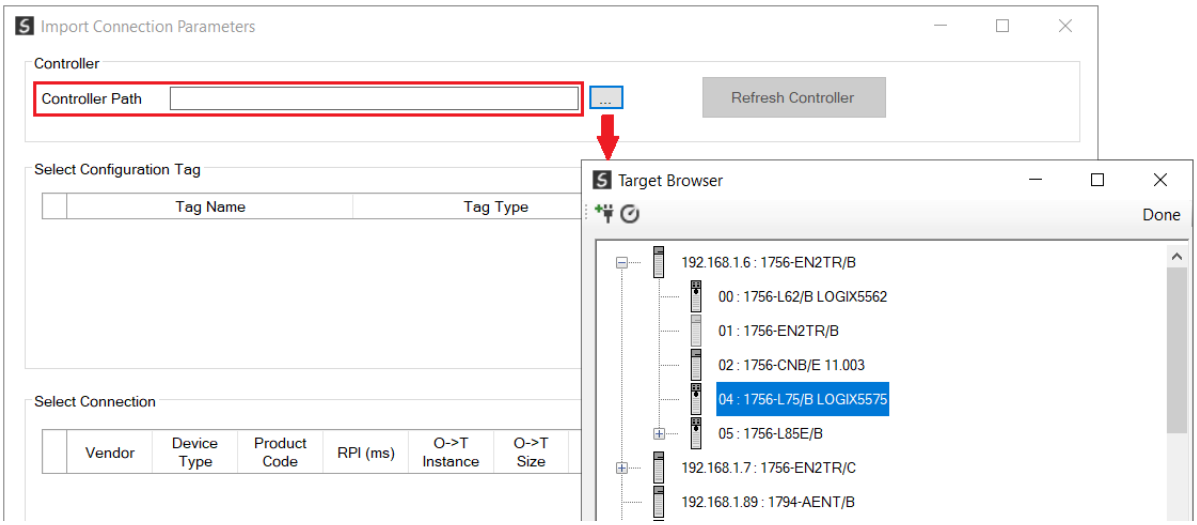


Figure 3.106 – Import Connection Parameters – Controller Path

Enter the path to the Logix controller. This can be either entered manually, or the Browse button "…", can be selected to launch the Target Browser, where the Logix controller can be selected.

Once the Logix controller path has been selected, all the device configuration tags and device connections will be read from the controller and displayed in the Configuration Tag grid and Connection grid respectively.

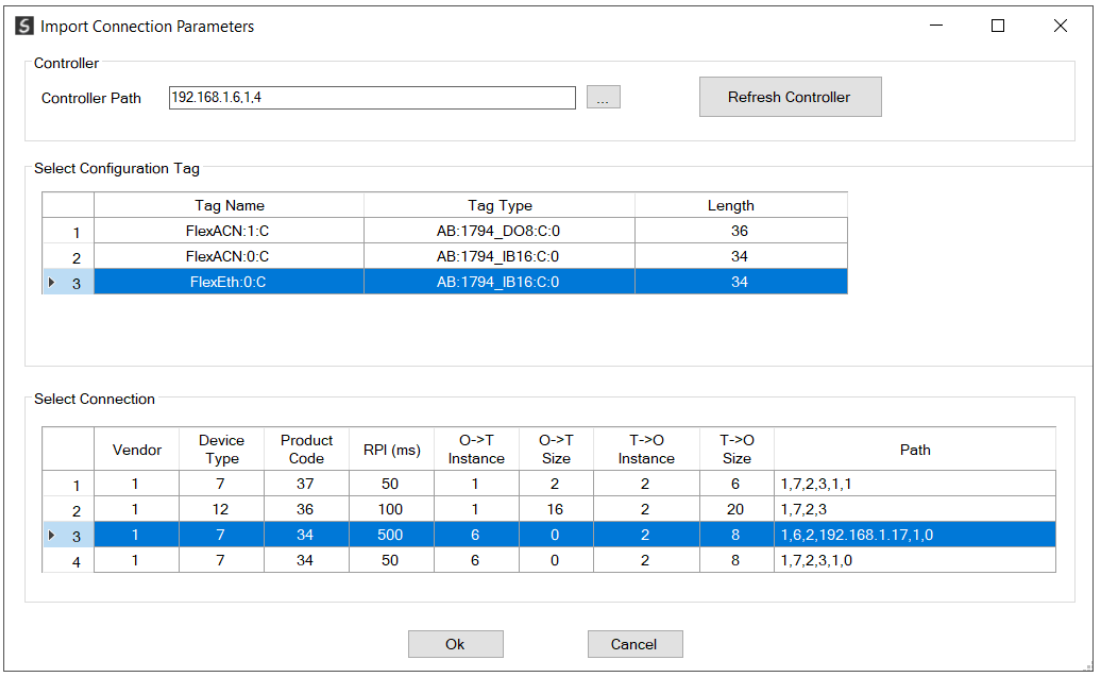


Figure 3.107 – Import Connection Parameters – Select Connection

In order to import all the necessary connection information, the user will need to select both the appropriate ***Configuration Tag***, and the matching ***Connection***.

The new connection's configuration data is derived from the selected ***Configuration Tag***, when the new connection's parameters are derived from the selected ***Connection***.

Once the appropriate selections have been made, press ***Ok***. The imported data will be populated into the Connection form.

The user can then modify the ***Connection Name***, ***Path*** and ***RPI*** as required.

C. IMPORT FROM CONTROLLER L5X FILE

Here the EtherNet/IP connection parameters are imported from a Logix controller's L5X file.

PREPARATION

Before the connection information can be imported some preparation is required using Studio5000:

1. In Studio5000 create a new project and add the required EtherNet/IP device in the IO tree. If the device's profile supports configuration, then configure the device as required.

2. Save the Studio5000 project as an L5X file.

NOTE: When instantiating modules in Studio5000 do not make use of the "Rack Optimization" communication format.

NOTE: It is possible that not all the connection information will be imported as it may not be available in the L5X file due to the type of device and Logix version.

IMPORT L5X FILE

The connection parameters can be imported from the L5X file by selecting the ***Import Controller L5X*** option located under the ***Import*** menu of the Class 1 Connection form.

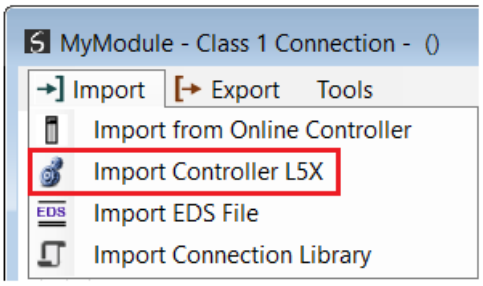


Figure 3.108 – Import from Controller L5X

The Import Connection Parameters form will open.

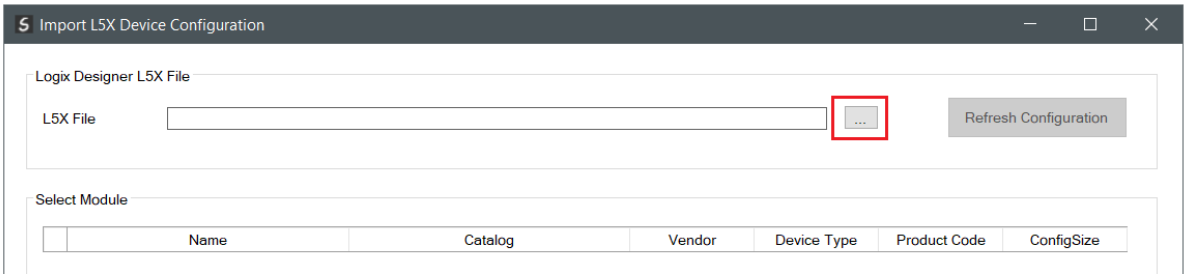


Figure 3.109 – Import L5X Device Configuration – Select L5X

Click on the Browse ("…") button to select the previously generated L5X file.

The modules found in the selected L5X file will then be displayed in the Module List.

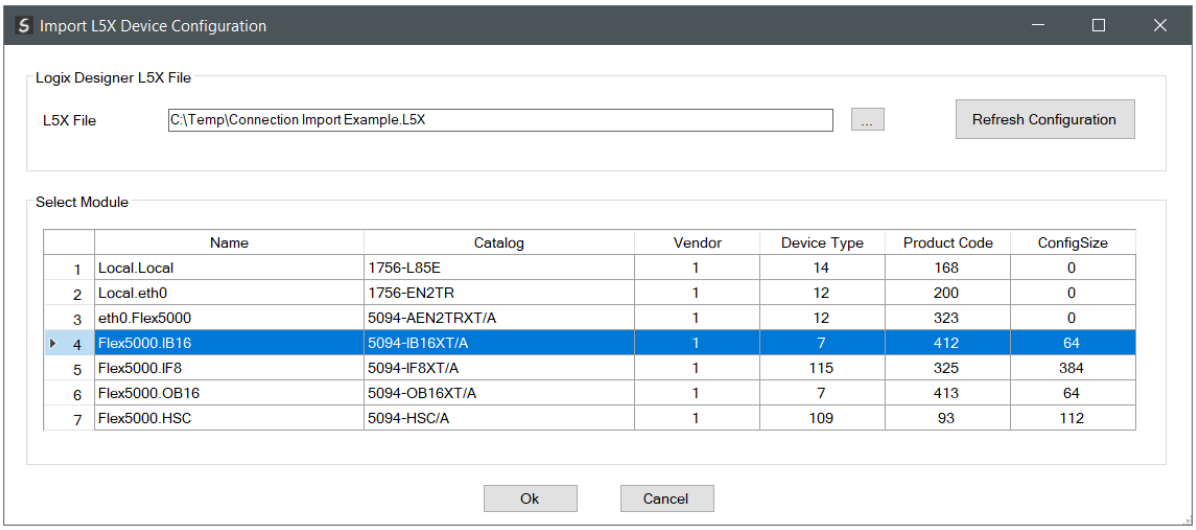


Figure 3.110 – Import L5X Device Configuration

Select the required module and click ***Ok***. The imported data will be populated into the Connection form.

The user can then modify the ***Connection Name***, ***Path*** and ***RPI*** as required.

D. IMPORT EDS FILE

The connection parameters can be imported from a suitable EDS file. Typically, this approach is preferred for devices that do not require configuration data.

To import the connection parameters from a device EDS file, select the ***Import EDS File*** option located under the ***Import*** menu of the Class 1 Connection form.

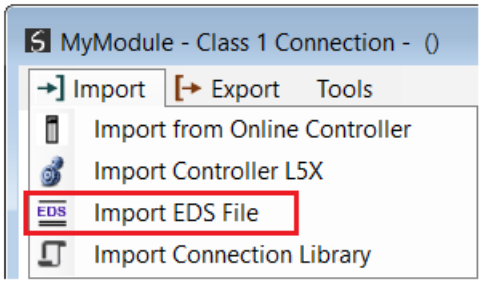


Figure 3.111 – Import EDS File

A File Open dialog will open allowing the user to select the EDS file.

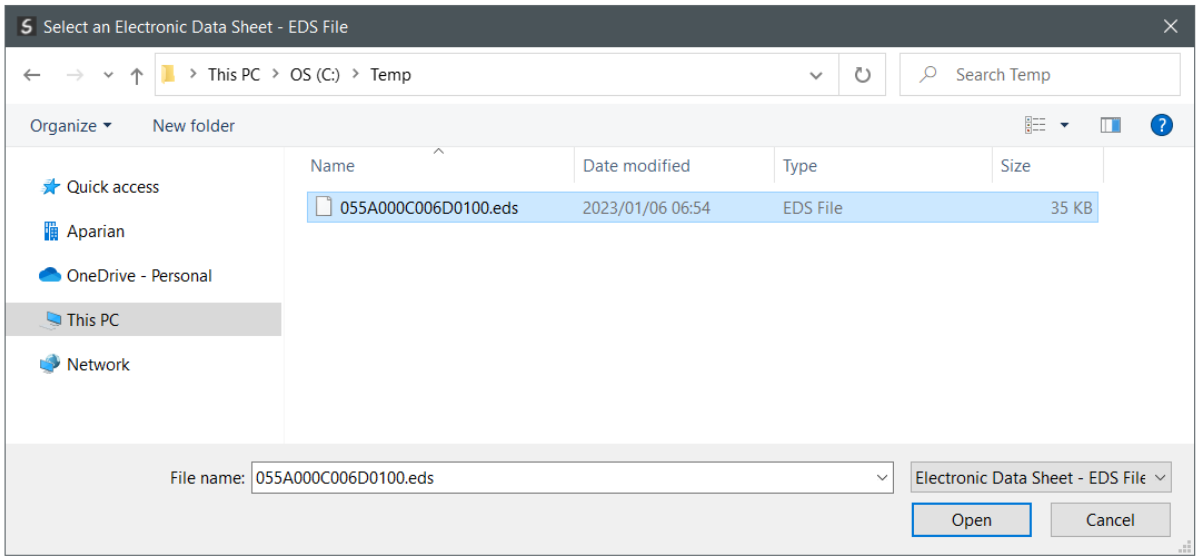


Figure 3.112 – Browse to EDS File

The selected EDS file will be imported, and a summary of the connections displayed. The user will need to select one of the IO connections.

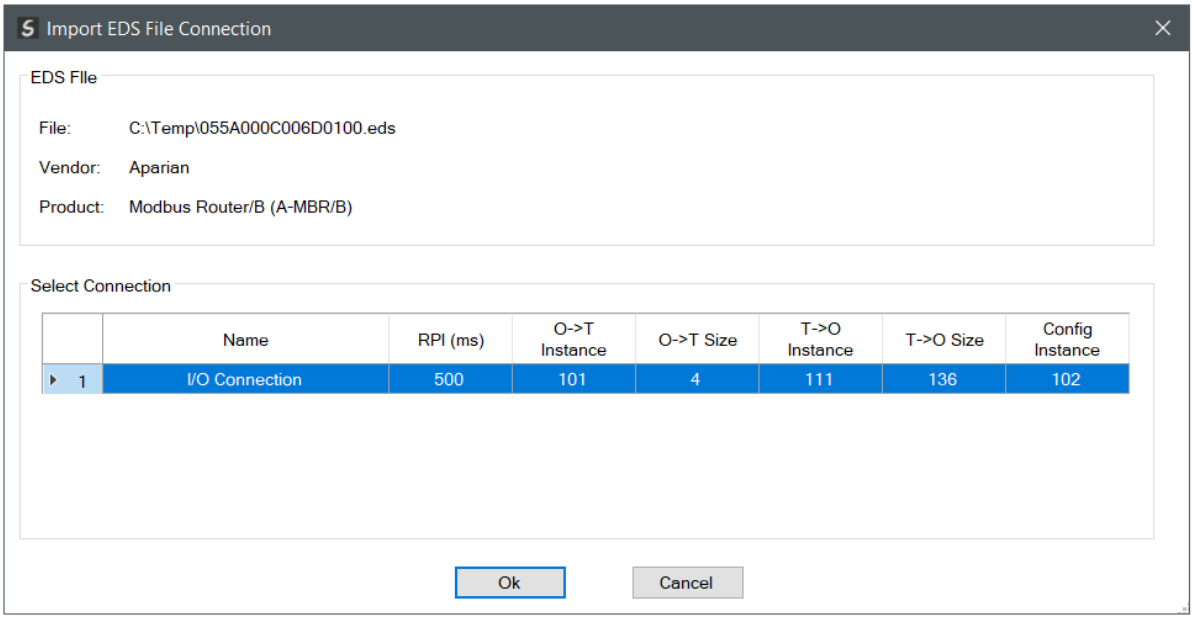


Figure 3.113 – Select Connection

The selected connection within the EDS file will be used to populate the Connection parameters.

The user can then modify the ***Connection Name***, ***Path*** and ***RPI*** as required.

E. IMPORT CONNECTION LIBRARY

The connection parameters can be imported from a previously created Connection Library (.EIPCNX) file.

NOTE: Please contact support to receive a pack of the latest Connection Library files, for commonly used devices.

To import the connection parameters from a Library file, select the ***Import Connection Library File*** option located under the ***Import*** menu of the Class 1 Connection form.

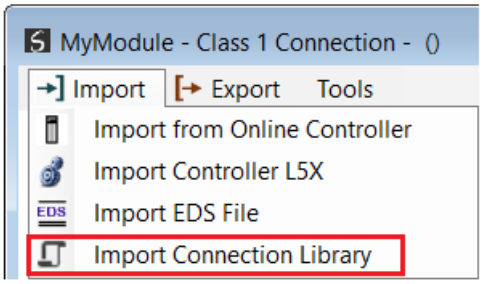


Figure 3.114 – Import Connection Library File

A File Open dialog will open allowing the user to select the Library (.EIPCNX) file.
The selected Library file will be used to populate the Connection parameters.

The user can then modify the ***Connection Name***, ***Path*** and ***RPI*** as required.

EXPORT LIBRARY FILE

In order to create a Library file for future use, select the ***Export Connection Library*** option located under the ***Export*** menu.

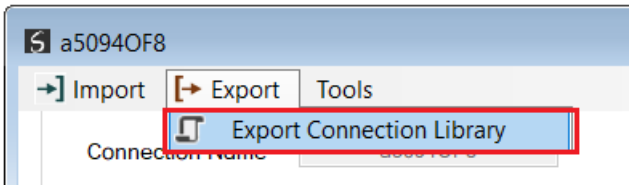


Figure 3.115 – Export Connection Library File

3.6.5.2. ETHERNET/IP EXPLICIT MESSAGE DEVICE CONNECTIONS

Up to 10 EtherNet/IP devices can be added for explicit messaging. The user will need to add each device as explained in the EtherNet/IP Devices section below. Once the EtherNet/IP devices have been added the user can then configure the required mapping for the

EtherNet/IP Explicit messaging as shown in EtherNet/IP Map section below.

A. ETHERNET/IP DEVICES

This tab is enabled when the Ethernet Mode is set to *EtherNet/IP Originator*.

The EtherNet/IP Devices configuration is shown in the figure below. Up to 10 EtherNet/IP devices can be configured with up to 50 EtherNet/IP mapped items allowing for either explicit EtherNet/IP Class 3 or Unconnected Messaging (UCMM) to any of the 10 configured devices. The data from each EtherNet/IP device is written to, or read from, an Internal Data Space with a size of 100Kbytes. See the *Explicit EtherNet/IP Messaging* section for more details.

The EtherNet/IP Devices configuration window is opened by either double clicking on the module in the tree or right-clicking the module and selecting *Configuration*.

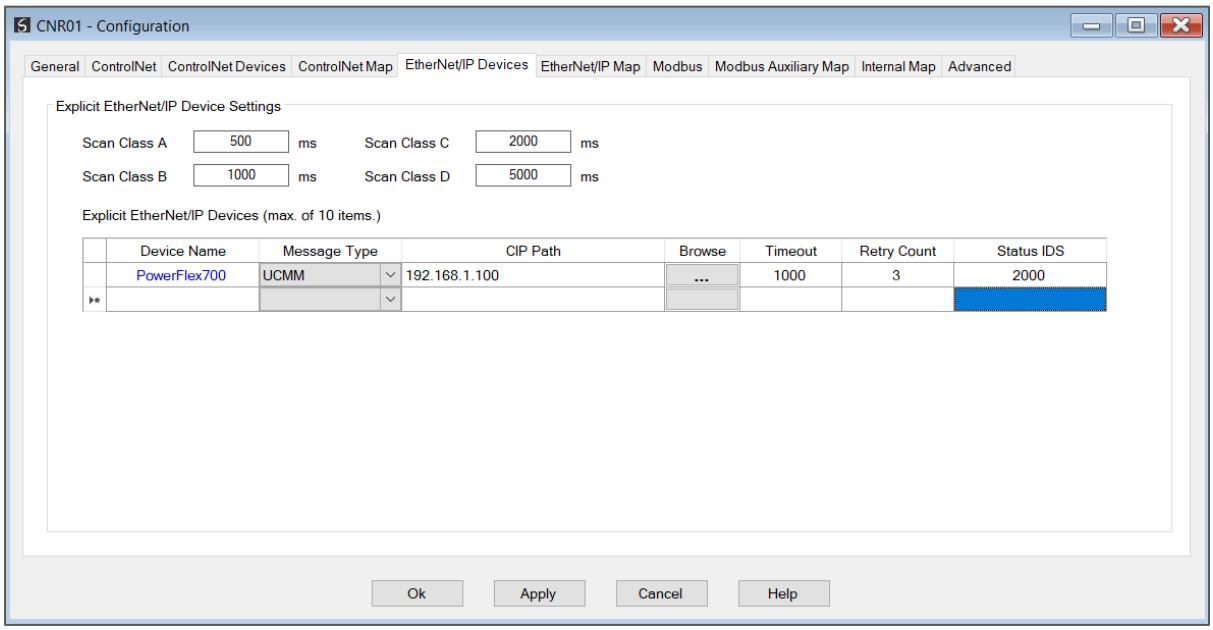


Figure 3.116 – EtherNet/IP Devices - Configuration

The EtherNet/IP Devices configuration consists of the following parameters:

Parameter	Description
Scan Class A, B, C, D	The configurable update rates (in milliseconds) for each scan class in the EtherNet/IP Map.
Device List (per device)	
Device Name	The user assigned instance name for the specific device.

Message Type	The module can use either ***Class 3*** or ***Unconnected Messaging*** when communicating to the target EtherNet/IP device.
CIP Path	The CIP Path to the target device. This can either be entered manually or the user can browse to them by clicking the ***Browse*** button. The Target Browser will open and automatically scan for all available EtherNet/IP devices. If the Ethernet/IP module is a bridge module, it can be expanded by right-clicking on the module and selecting the ***Scan*** option. The required EtherNet/IP device can then be chosen by selecting it and clicking the ***Ok*** button, or by double-clicking on the target module.
Timeout	The amount of time (in milliseconds) the module will wait for a response from the target EtherNet/IP device.
Retry Count	The number of message retries before the target EtherNet/IP device is considered offline.
Comm Status Offset	This is the offset in the Internal Data Space (used to map EtherNet/IP device data) which provides the communication status of each EtherNet/IP device. The Communication Status is as shown below: Bit 0 - (1:Device Online , 0:Device Offline) Bit 1 to 7 – Reserved.

Table 3.16 – EtherNet/IP Devices configuration parameters

B. ETHERNET/IP MAP

This tab is enabled when the Ethernet Mode is set to *EtherNet/IP Originator*.

The EtherNet/IP Map configuration is shown in the figure below. Up to 50 EtherNet/IP mapped items, either explicit EtherNet/IP Class 3 or Unconnected Messaging (UCMM) to any of the 10 pre-configured devices can be configured. The data from each EtherNet/IP device is written to or read from a Internal Data Space with a size of 100Kbytes. See the *Explicit EtherNet/IP Messaging* section for more details.

The EtherNet/IP Map configuration window is opened by either double clicking on the module in the tree or right-clicking the module and selecting *Configuration*.

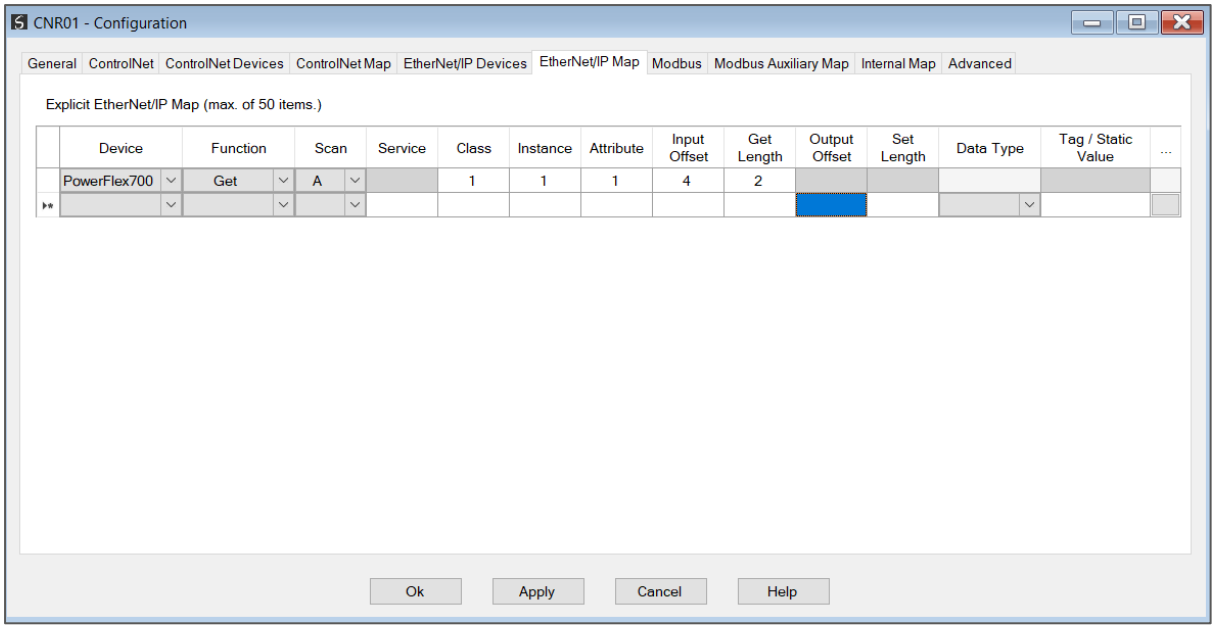


Figure 3.117 – EtherNet/IP Map configuration

The EtherNet/IP Map configuration consists of the following parameters:

Parameter	Description
Device	The device instance name configured in the previous EtherNet/IP Devices tab. The selected device will be used for executing the communication function.
Function	The user can select one of four functions. **Get** The module will read data from the target EtherNet/IP device by using the Get Single Attribute CIP function. The received data will be placed into the Internal Data Space at the ***Input Offset*** location configured in this tab. **Set** The module will write data to the target EtherNet/IP device by using the Set Single Attribute CIP function. The data to be written will be retrieved from the Internal Data Space at the ***Output Offset*** location configured in this tab. **Set Static** Similar to the Set function above, but the data to be written will be fixed (equal to the *Static Value*) parameter in this configuration window. This function will typically be used with the single (**S**) Scan class which means the ControlNet Router module can be setup to write the fixed value only once when the target device communication has been established. **Custom** This function allows the user to use a custom Service to write and read data in the same transaction. The user will need to see which custom services that target device

	supports in that device's user manual. **Read Tag** When using a Logix controller as a EtherNet/IP Device, the ControlNet Router module can read a Logix tag from the target Logix controller using Class 3 or UCMM messaging. The value from the tag will be saved at the configured Input Offset. **Write Tag** When using a Logix controller as a EtherNet/IP Device, the ControlNet Router module can write to Logix tag from the target Logix controller using Class 3 or UCMM messaging. The value from the tag will be read from the configured Output Offset.
Scan	The user can select Scan Class **A**, **B**, **C** or **D** (which was configured in the EtherNet/IP Devices tab). The specific mapped item will then be executed at that configured scan class rate. The user can also select the **S** class which means that the mapped item will only execute once when communication to the target device is established. If the target device goes offline, then the mapped items with this class will be re-armed, and resent when communication is re-established.
Service	The custom CIP service/function which is only available when the ***Custom*** function has been selected.
Class, Instance, Attribute	The CIP class, instance, and attribute of the request message to be sent.
Input Offset	The location in the Internal Data Space where the received data will be written. This will only be available for ***Get*** and ***Custom*** functions.
Get Length	The length of the data to be received. If the number of bytes received is more than the Get Length, then the data will **not** be written to the Internal Data Space. This will only be available for ***Get*** and ***Custom*** functions.
Output Offset	The location in the Internal Data Space from where the data to be written to the target device will be read. This will only be available for ***Set*** and ***Custom*** functions.
Set Length	The length of the data to be written. This will only be available for ***Set*** and ***Custom*** functions.
Data Type	The data type of the Static Value. This will only be available for ***Set Static*** function.
Tag / Static Value	The value to be written to the target device when the ***Set Static*** function has been selected. **Note**: When using the SINT Array data type, the values must be entered as space-delimited hex values. For example: 05 34 2E A1

Table 3.17 – EtherNet/IP Map configuration parameters

3.6.5.3. INTERNAL DATA SPACE MAPPING

When the module is operating as a EtherNet/IP Originator, the data from the EtherNet/IP IO devices can be mapped to the ControlNet interface using the Internal Map. The Internal Map configuration window is opened by either double clicking on the module in the tree or right-clicking the module and selecting *Configuration* and selecting the *Internal Map* tab.

A. IDS COPY – ETHERNET/IP ORIGINATOR SOURCE

When copying data from a EtherNet/IP IO to the ControlNet interface, the source type needs to be EIP Originator.

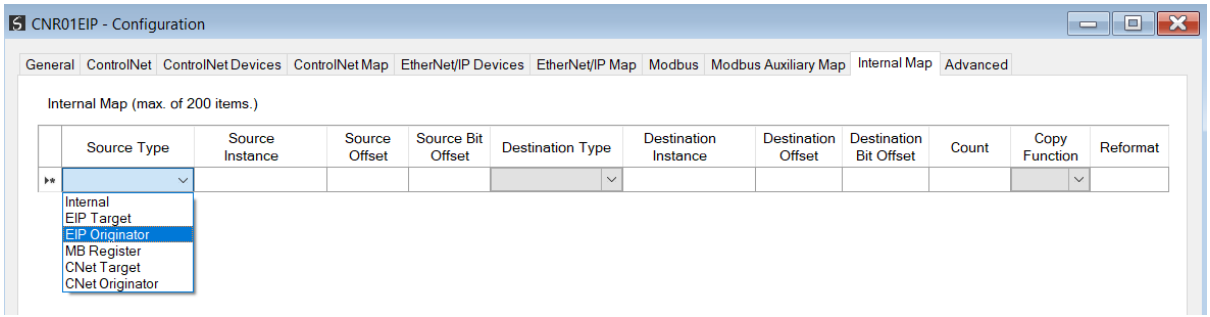


Figure 3.118 – IDS Copy – EtherNet/IP Originator Source Type

The source instance will be one of the EtherNet/IP IO devices added to the EtherNet/IP IO tree in Slate.

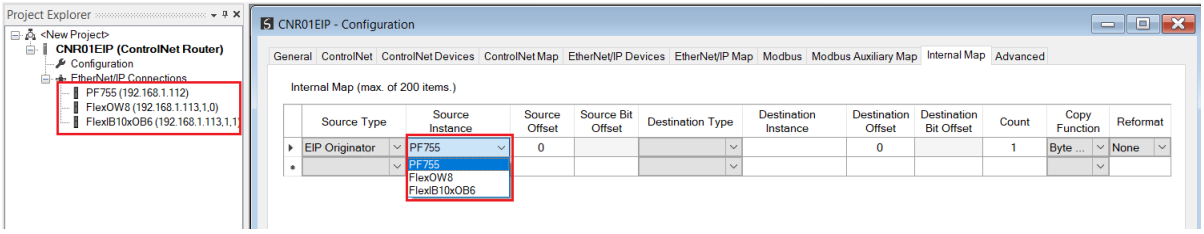


Figure 3.119 – IDS Copy – EtherNet/IP Originator Source Instance

The Source Offset is the offset in the selected EtherNet/IP device Class 1 **Input** Assembly. The Count is the number of **bytes** that will be copied. See the Internal Data Space Mapping section for more information regarding the operation.

B. IDS COPY – ETHERNET/IP TARGET DESTINATION

When copying data from the ControlNet interface to a EtherNet/IP IO device **Output** Assembly, the destination type needs to be EIP Originator.

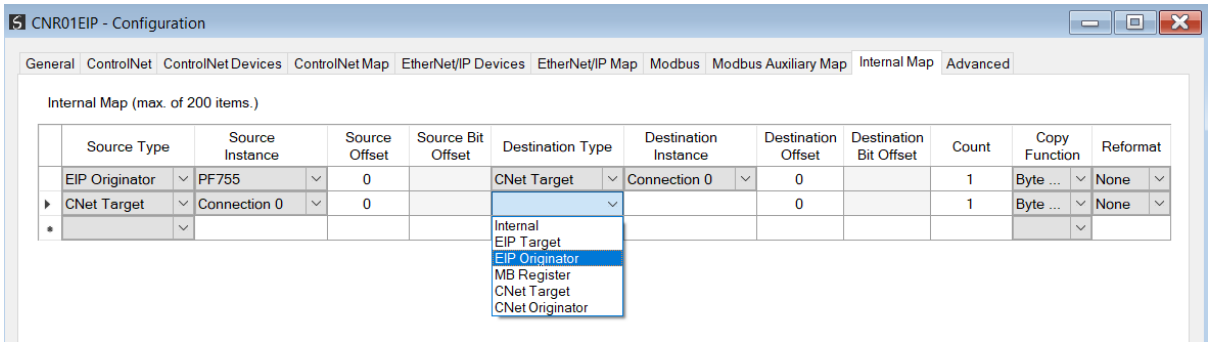


Figure 3.120 – IDS Copy – EtherNet/IP Originator Destination Type

The destination instance will be one of the EtherNet/IP IO devices added to the EtherNet/IP IO tree in Slate.

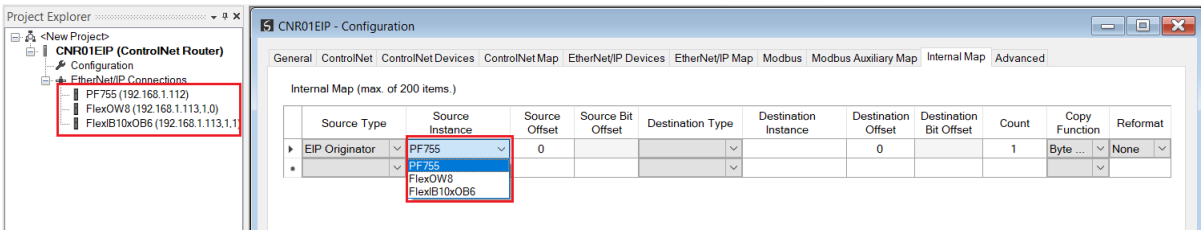


Figure 3.121 – IDS Copy – EtherNet/IP Originator Destination Instance

The Destination Offset is the offset in the selected EtherNet/IP device Class 1 **Output** Assembly. The Count is the number of **bytes** that will be copied. See the Internal Data Space Mapping section for more information regarding the operation.

3.7. INTERNAL DATA SPACE MAP

The internal data map is used to exchange data from the Ethernet interface to the ControlNet interface and vice versa. Up to 200 items can be mapped. The Internal Map configuration window is opened by either double clicking on the module in the tree or right-clicking the module and selecting *Configuration* and selecting the *Internal Map* tab.



NOTE: When the Bridge is selected for the ControlNet mode, Internal Data Space Map will be disabled.

The Count is the number of bytes that will be copied from the source to the destination. There are four different Copy Functions that can be used.

Function	Description
Byte to Byte	Each **byte** from the source will be directly copied to each **byte** in the destination.

Byte to Bit	Each byte from the source will be copied to each bit in the destination. If a value greater than zero is read from the **source byte** then a 1 will be written to the **destination bit** address. If a value of zero is read from the **source byte** then a 0 will be written to the **destination bit** address. The destination offset will be the bit offset and the destination address will be incremented by one bit each time.
Bit to Bit	Each **bit** from the source will be directly copied to each **bit** in the destination.
Bit to Byte	Each bit from the source will be copied to each byte in the destination. If a value of one is read from the **source bit** then a 1 will be written to the **destination byte** address. If a value of zero is read from the **source bit** then a 0 will be written to the **destination byte** address. The source offset will be the bit offset and the source address will be incremented by one bit each time.

Table 3.18 – Internal Map Copy functions

The data in the destination source can also be reformatted. The reformat option provides five different reformat options.
NOTE: The reformat option is only available for *Byte to Byte* Copy Functions.

Function	Description
None	No reformatting applied (AA BB CC DD)
BB AA	16bit Byte swap
BB AA DD CC	32bit Byte Pair Swap
CC DD AA BB	Word Swap
DD CC BB AA	Word and Byte Pair Swap

Table 3.19 – Internal Map Reformat Options

3.7.1. COPY FROM

One of six sources can be selected to copy from; Internal, EIP Target, EIP Originator, CNet Target, and CNet Originator.

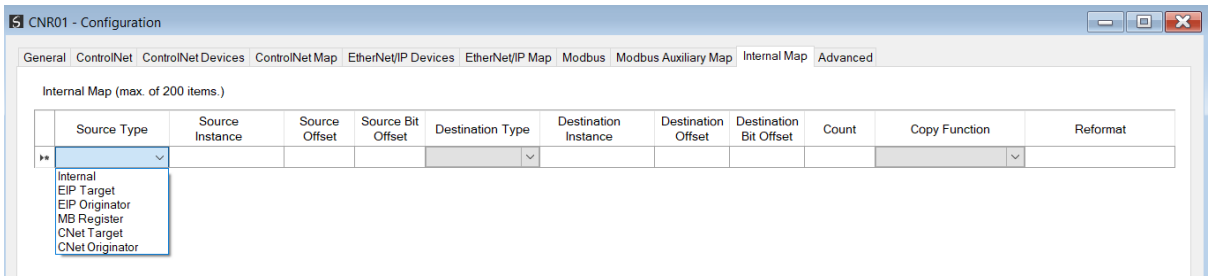


Figure 3.122 – Internal Map – Source Type

3.7.1.1. INTERNAL

When copying data from the internal data space (IDS), the source type needs to be Internal.

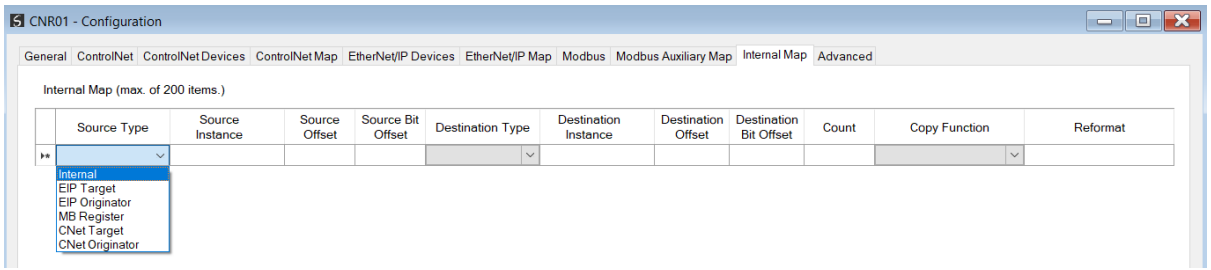


Figure 3.123 – IDS Copy – Internal Source Type

The source instance is Not Applicable for the internal data space. The Source Offset is the offset in the *Internal Data Space (IDS)* which has a max of 100,000 bytes. The Count is the number of **bytes** that will be copied.

3.7.1.2. EIP TARGET

When copying data from a connection originator (e.g. the output assembly from the Logix Controller) to the ControlNet interface, the source type needs to be EIP Target.

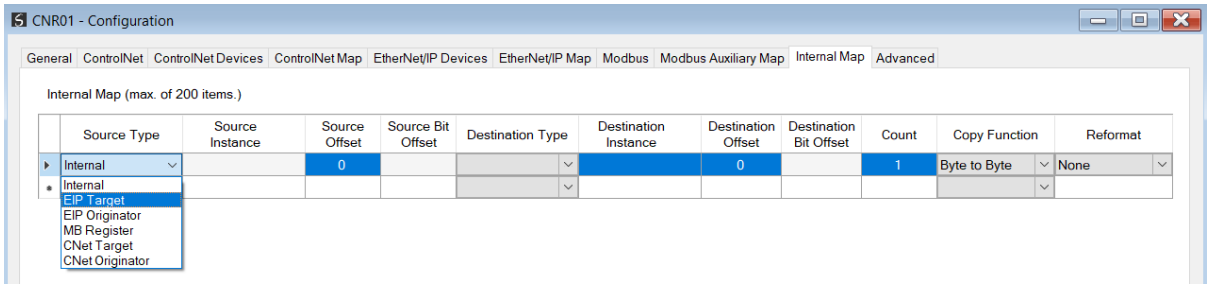


Figure 3.124 – IDS Copy – EtherNet/IP Target Source Type

The source instance will be the connection number, which in this case is always *Connection 0*. The Source Offset is the offset in the *Mapped Data* section of the EtherNet/IP output assembly from where the data must be copied. The Count is the number of **bytes** that will be copied. The *Mapped Data* section will be available when using the provided UDT for the Output Assembly.

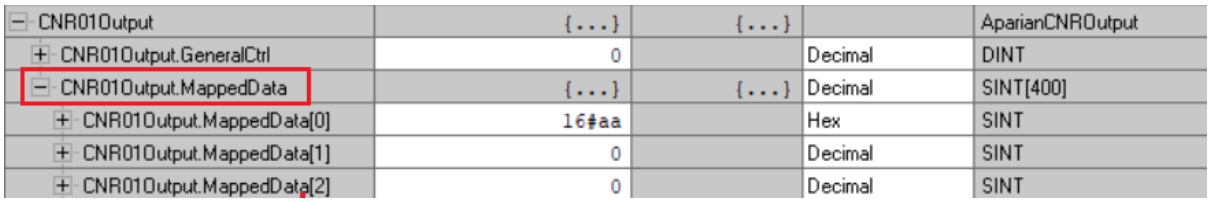


Figure 3.125 – EtherNet/IP Output Mapped Data UDT

3.7.1.3. EIP ORIGINATOR

When copying data from a EtherNet/IP IO to the ControlNet interface, the source type needs to be EIP Originator.

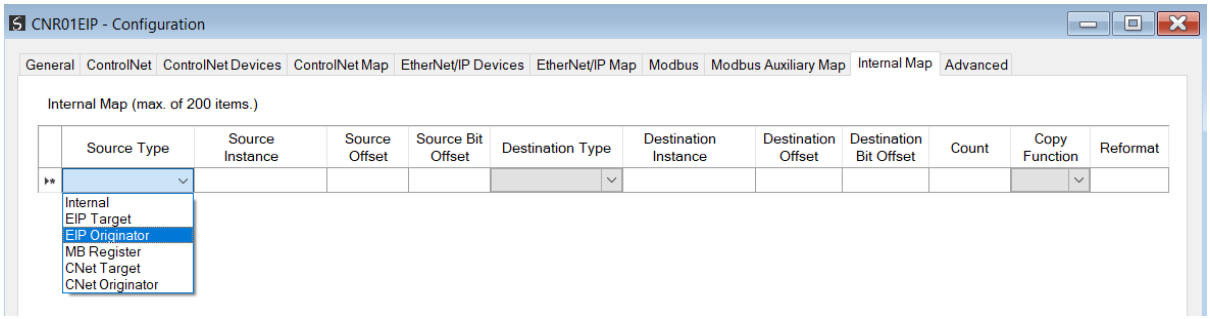


Figure 3.126 – IDS Copy – EtherNet/IP Originator Source Type

The source instance will be one of the EtherNet/IP IO devices added to the EtherNet/IP IO tree in Slate.

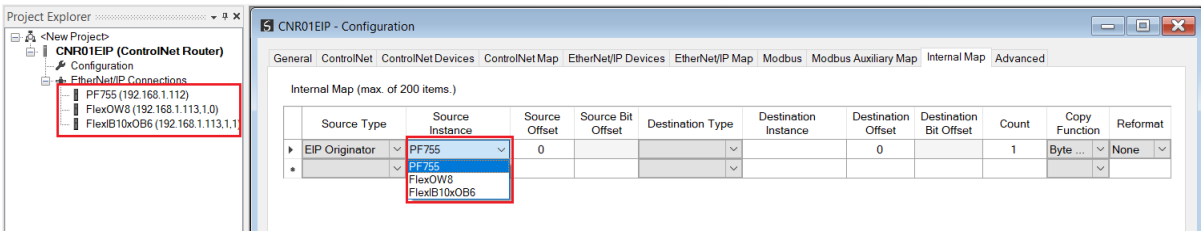


Figure 3.127 – IDS Copy – EtherNet/IP Originator Source Instance

The Source Offset is the offset in the selected EtherNet/IP device Class 1 **Input** Assembly. The Count is the number of **bytes** that will be copied.

3.7.1.4. MODBUS REGISTER

When copying Modbus data to the ControlNet interface, the source type needs to be MB Register.

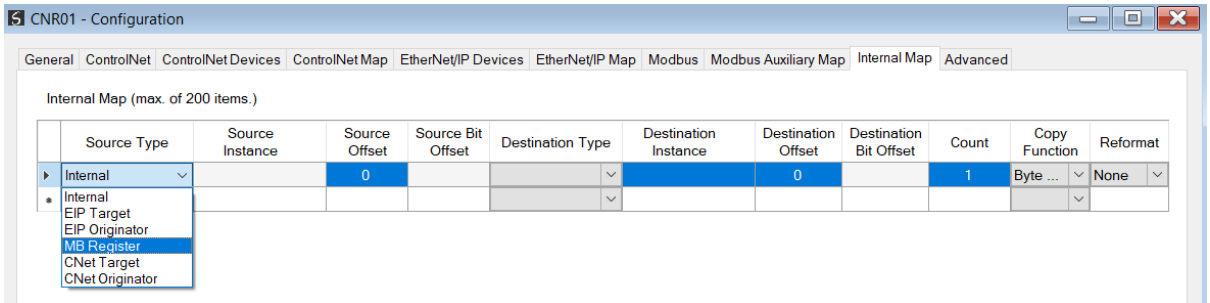


Figure 3.128 – IDS Copy - Modbus Source Type

The source instance will be the Modbus register type required.

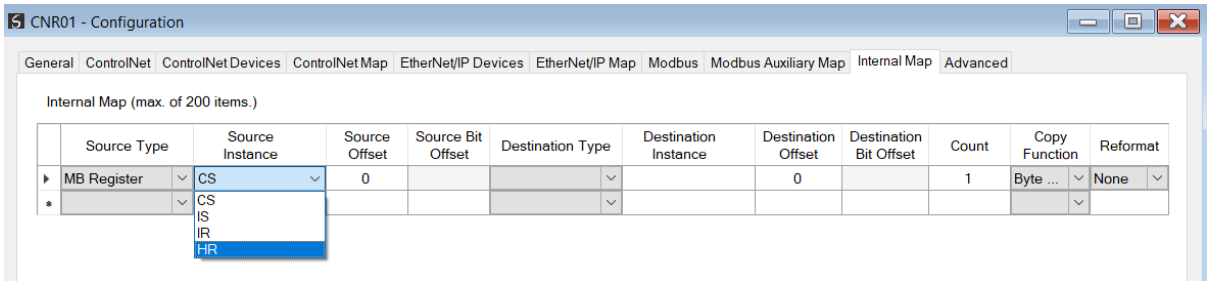


Figure 3.129 – IDS Copy - Modbus Source Instance

The Source Offset is the Modbus Register offset from where the data must be copied. The Count is the number of **bytes** that will be copied.

3.7.1.5. CNET TARGET

When copying data from a connection originator (e.g. the output assembly from the Logix Controller) to the Ethernet interface, the source type needs to be CNet Target.

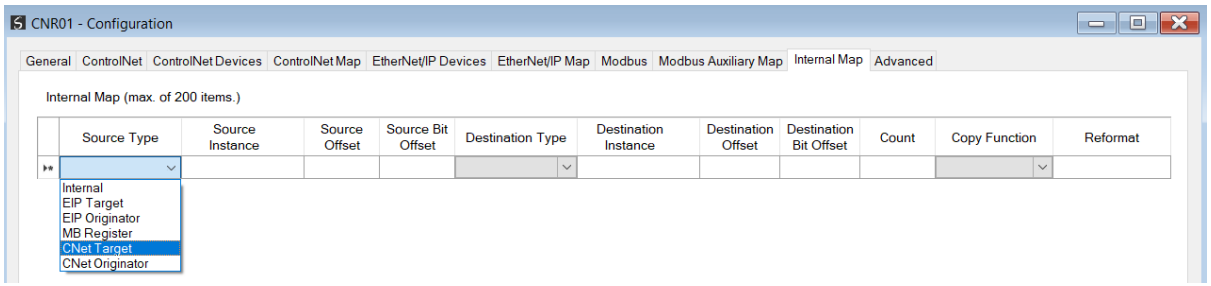


Figure 3.130 – IDS Copy – ControlNet Target Source Type

The source instance will be the connection number, which in this case is always *Connection 0*. The Source Offset is the offset in the *Mapped Data* section of the ControlNet output assembly from where the data must be copied. The Count is the number of **bytes** that will be copied. The *Mapped Data* section will be available when using the provided UDT for the Output Assembly.

CNR01Output	{...}	{...}		AparianCNROutput
CNR01Output.GeneralCtrl	0		Decimal	DINT
CNR01Output.MappedData	{...}	{...}	Decimal	SINT[400]
CNR01Output.MappedData[0]	16#aa		Hex	SINT
CNR01Output.MappedData[1]	0		Decimal	SINT
CNR01Output.MappedData[2]	0		Decimal	SINT

Figure 3.131 – ControlNet Output Mapped Data UDT

3.7.1.6. CNET ORIGINATOR

When copying data from a ControlNet IO to the Ethernet interface, the source type needs to be CNet Originator.

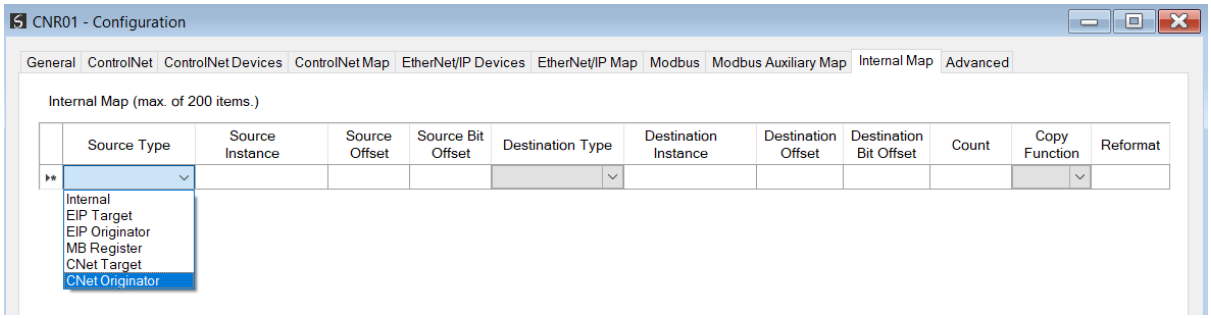


Figure 3.132 – IDS Copy – ControlNet Originator Source Type

The source instance will be one of the ControlNet IO devices added to the ControlNet IO tree in Slate.

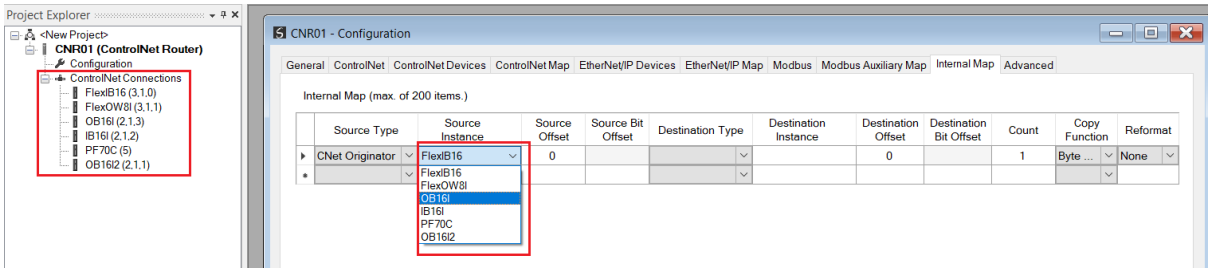


Figure 3.133 – IDS Copy – ControlNet Originator Source Instance

The Source Offset is the offset in the selected ControlNet device Scheduled **Input** Assembly. The Count is the number of **bytes** that will be copied.

3.7.2. COPY TO

One of six destinations can be selected to copy to; Internal, EIP Target, EIP Originator, CNet Target, and CNet Originator.

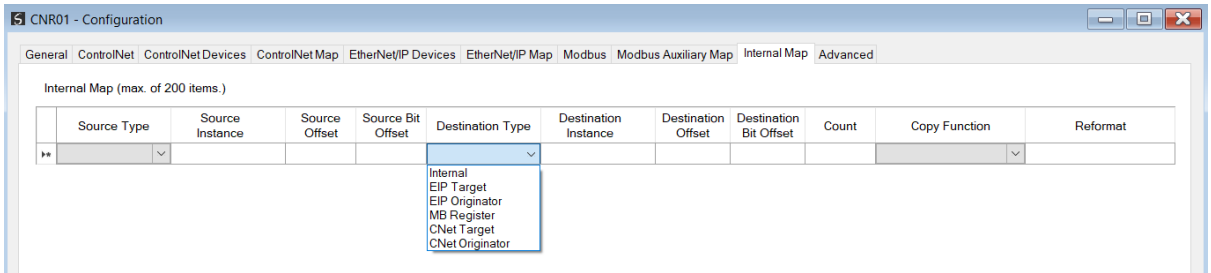


Figure 3.134 – Internal Map – Destination Type

3.7.2.1. INTERNAL

When copying data to the internal data space (IDS), the destination type needs to be Internal.

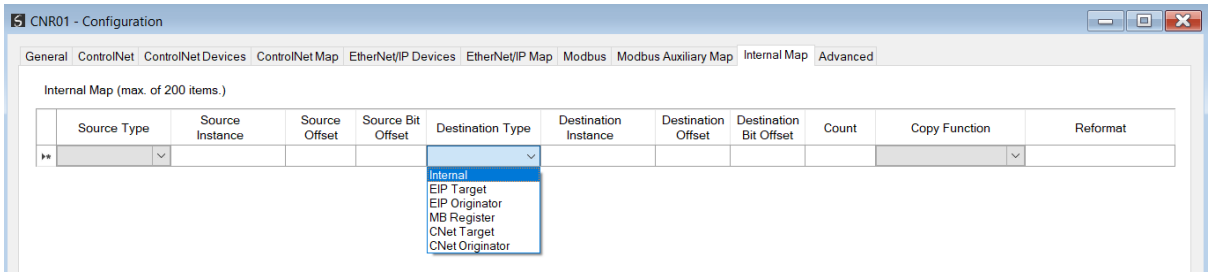


Figure 3.135 – IDS Copy – Internal Source Type

The destination instance is Not Applicable for the internal data space. The Destination Offset is the offset in the *Internal Data Space (IDS)* which has a max of 100,000 bytes. The Count is the number of **bytes** that will be copied.

3.7.2.2. EIP TARGET

When copying data from the ControlNet interface to the EtherNet/IP Target input assembly, the destination type needs to be EIP Target.

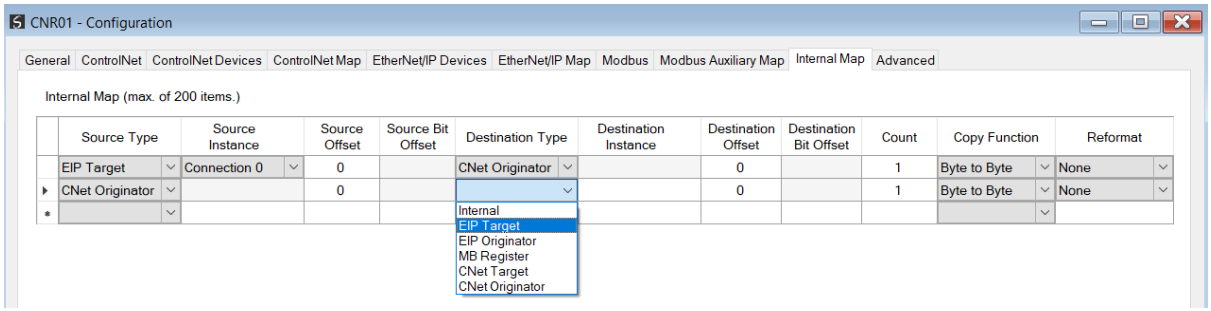


Figure 3.136 – IDS Copy – EtherNet/IP Target Destination Type

The destination instance will be the connection number, which in this case is always *Connection 0*. The Destination Offset is the offset in the *Mapped Data* section of the EtherNet/IP input assembly from where the data must be copied. The Count is the number of **bytes** that will be copied. The *Mapped Data* section will be available when using the provided UDT for the Input Assembly.

CNR01Input	{...}	{...}		AparianCNRInput
CNR01Input.GeneralStatus	-79		Decimal	SINT
CNR01Input.CNetNodeNumber	1		Decimal	SINT
CNR01Input.CNetLedState	1		Decimal	SINT
CNR01Input.CNetMode	6		Decimal	SINT
CNR01Input.Temperature	37.30043		Float	REAL
CNR01Input.MappedData	{...}	{...}	Decimal	SINT[400]
CNR01Input.MappedData[0]	2		Decimal	SINT
CNR01Input.MappedData[1]	0		Decimal	SINT

Figure 3.137 – EtherNet/IP Input Mapped Data UDT

3.7.2.3. EIP ORIGINATOR

When copying data from the ControlNet interface to a EtherNet/IP IO device **Output** Assembly, the destination type needs to be EIP Originator.

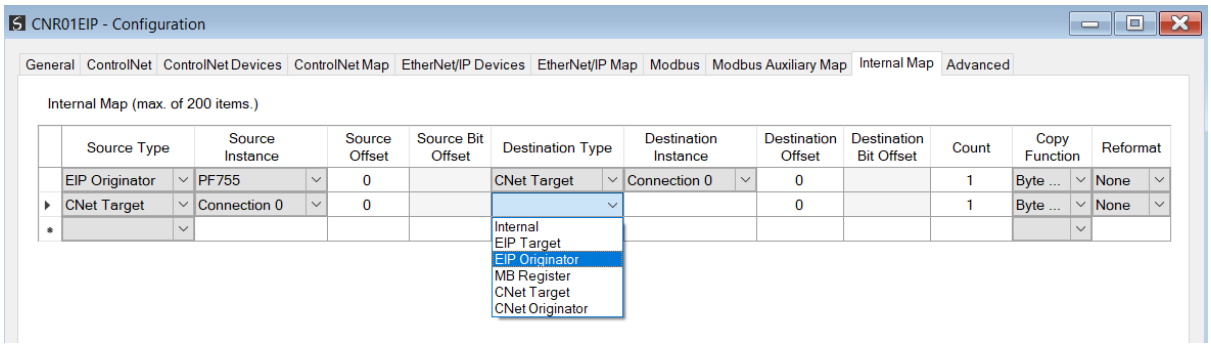


Figure 3.138 – IDS Copy – EtherNet/IP Originator Destination Type

The destination instance will be one of the EtherNet/IP IO devices added to the EtherNet/IP IO tree in Slate.

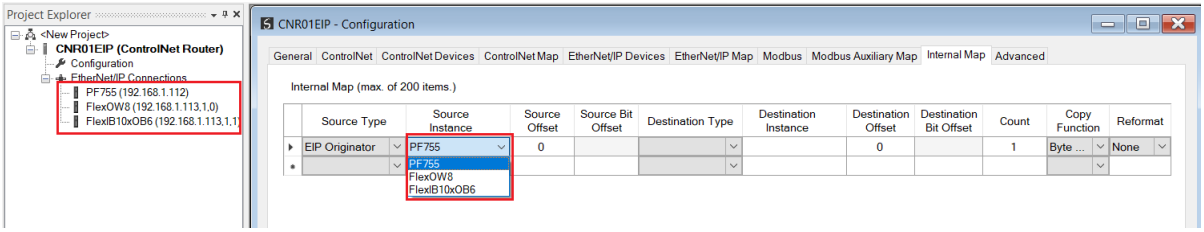


Figure 3.139 – IDS Copy – EtherNet/IP Originator Destination Instance

The Destination Offset is the offset in the selected EtherNet/IP device Class 1 **Output** Assembly. The Count is the number of **bytes** that will be copied.

3.7.2.4. MODBUS REGISTER

When copying data from the ControlNet interface to a Modbus Register, the destination type needs to be MB Register.

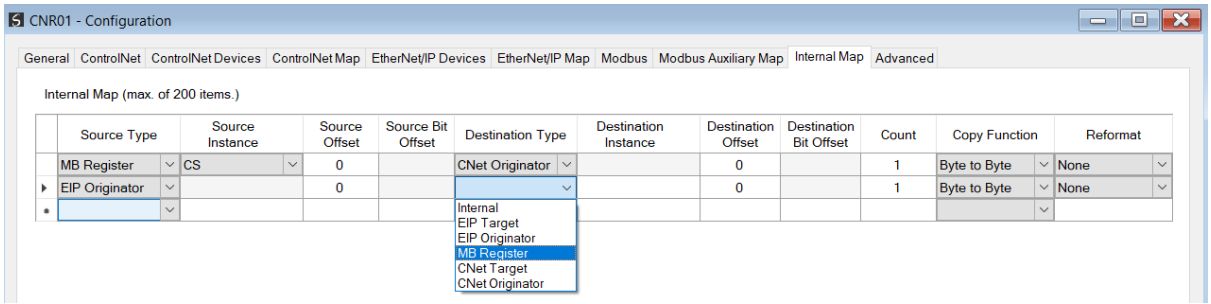


Figure 3.140 – IDS Copy - Modbus Destination Type

The destination instance will be the Modbus register type required.

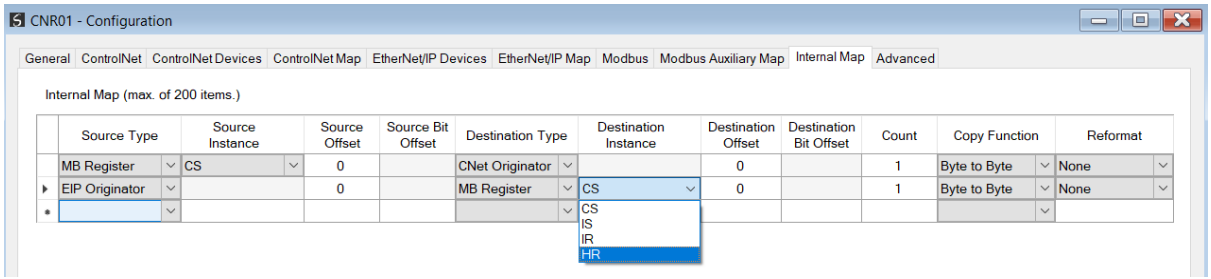


Figure 3.141 – IDS Copy - Modbus Destination Instance

The Destination Offset is the Modbus Register offset to where the data must be copied. The Count is the number of **bytes** that will be copied.

3.7.2.5. CNET TARGET

When copying data from the Ethernet interface to the ControlNet Target input assembly, the destination type needs to be CNet Target.

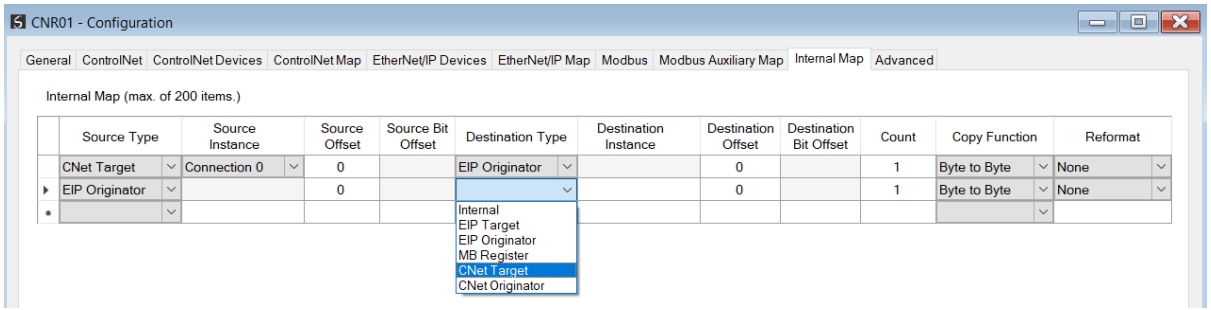


Figure 3.142 – IDS Copy – ControlNet Target Destination Type

The destination instance will be the connection number, which in this case is always *Connection 0*. The Destination Offset is the offset in the *Mapped Data* section of the ControlNet input assembly from where the data must be copied. The Count is the number of **bytes** that will be copied. The *Mapped Data* section will be available when using the provided UDT for the Input Assembly.

CNR01Input	{...}	{...}		AparianCNRInput
CNR01Input.GeneralStatus	-79		Decimal	SINT
CNR01Input.CNetNodeNumber	1		Decimal	SINT
CNR01Input.CNetLedState	1		Decimal	SINT
CNR01Input.CNetMode	6		Decimal	SINT
CNR01Input.Temperature	37.30043		Float	REAL
CNR01Input.MappedData	{...}	{...}	Decimal	SINT[400]
CNR01Input.MappedData[0]	2		Decimal	SINT
CNR01Input.MappedData[1]	0		Decimal	SINT

Figure 3.143 – ControlNet Input Mapped Data UDT

3.7.2.6. CNET ORIGINATOR

When copying data from the Ethernet interface to a ControlNet IO device **Output** Assembly, the destination type needs to be CNet Originator.

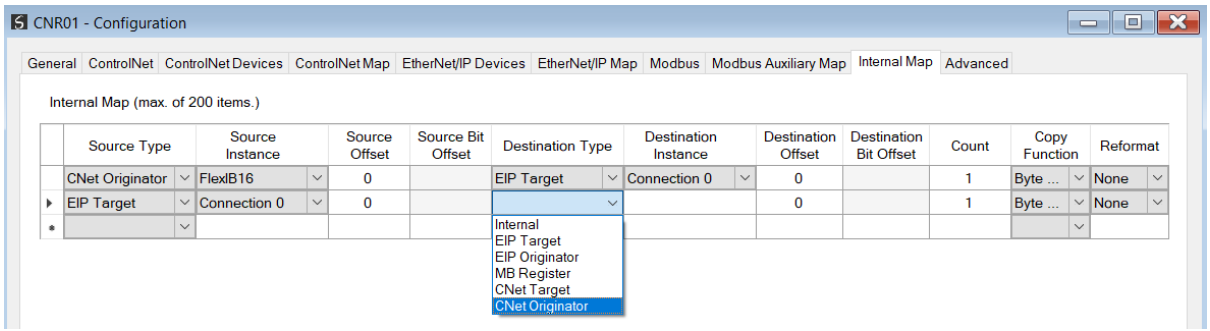


Figure 3.144 – IDS Copy – ControlNet Originator Destination Type

The destination instance will be one of the ControlNet IO devices added to the ControlNet IO tree in Slate.

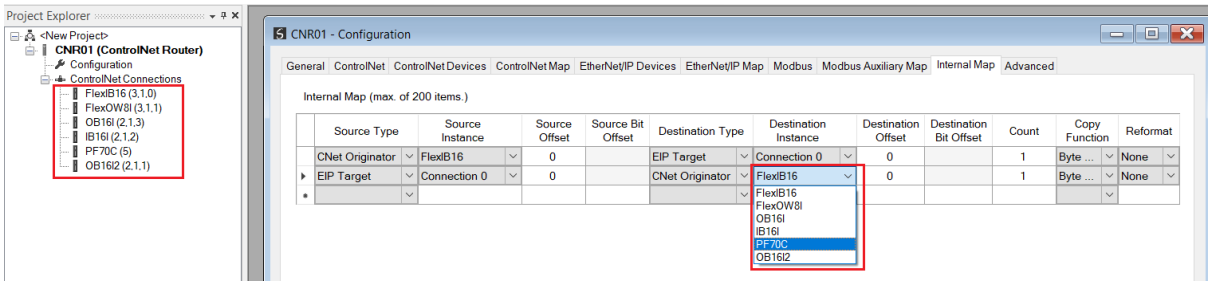


Figure 3.145 – IDS Copy – ControlNet Originator Destination Instance

The Destination Offset is the offset in the selected ControlNet device Scheduled **Output** Assembly. The Count is the number of **bytes** that will be copied. See the Internal Data Space Mapping section for more information regarding the operation.

NOTE: For more information regarding the specific Internal Map operation for specific interfaces, see the setup and configuration sections for the various ControlNet and Ethernet modes.

3.8. ADVANCED

The Advanced configuration window is opened by either double clicking on the module in the tree or right-clicking the module and selecting *Configuration*.

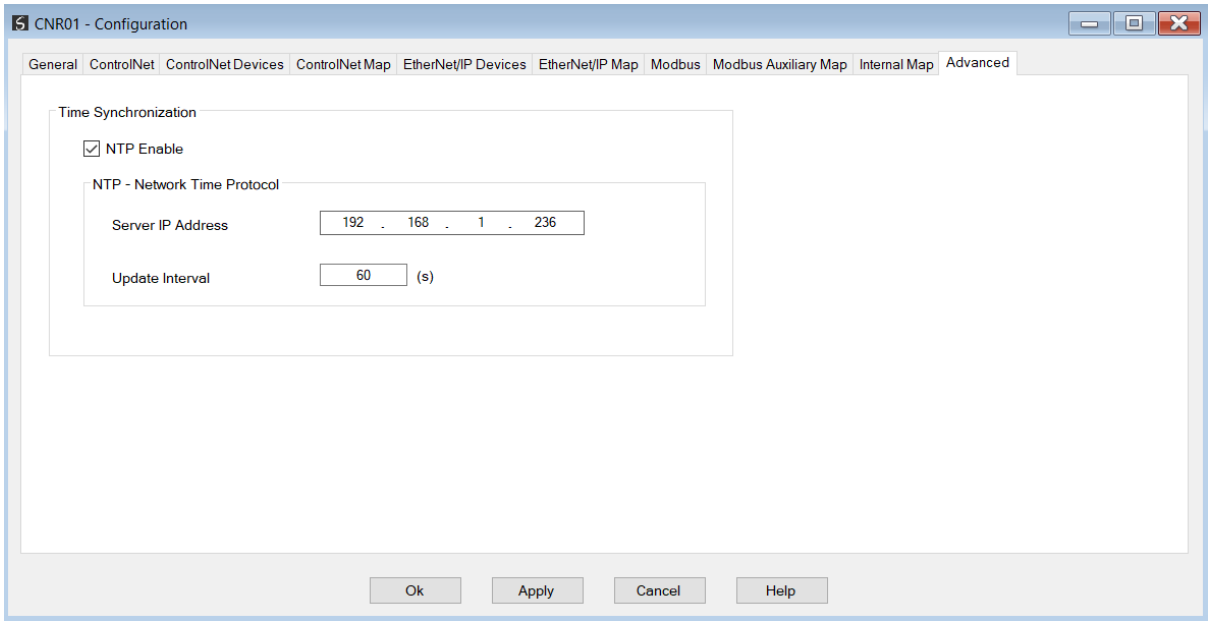


Figure 3.146 – Advanced configuration

The Advanced configuration consists of the following parameters:

Parameter	Description
NTP Enable	The ControlNet Router can synchronize its onboard clock to an NTP Server by enabling NTP.
NTP – Server IP Address	This setting is the IP address of the NTP Server which will be used as a time source.
NTP – Update Interval	This setting is the updated interval (in seconds) that the ControlNet Router will request time from the NTP Server.

Table 3.20 – Advanced configuration parameters

3.9. MODULE DOWNLOAD

Once the ControlNet Router configuration has been completed, it must be downloaded to the module. Before downloading the ***Connection Path*** of the module should be set. This path will automatically default to the IP address of the module, as set in the module configuration. It can however be modified, if the ControlNet Router is not on a local network.

The connection path can be set by right-clicking on the module and selecting the ***Connection Path*** option.

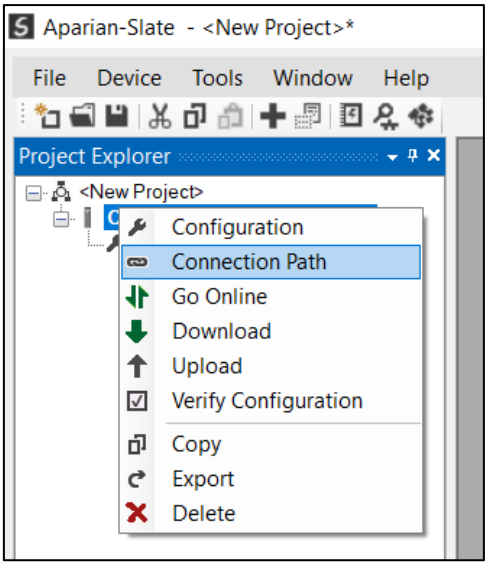


Figure 3.147 - Selecting Connection Path

The new connection path can then be either entered manually or selected by means of the ***Target Browser***.

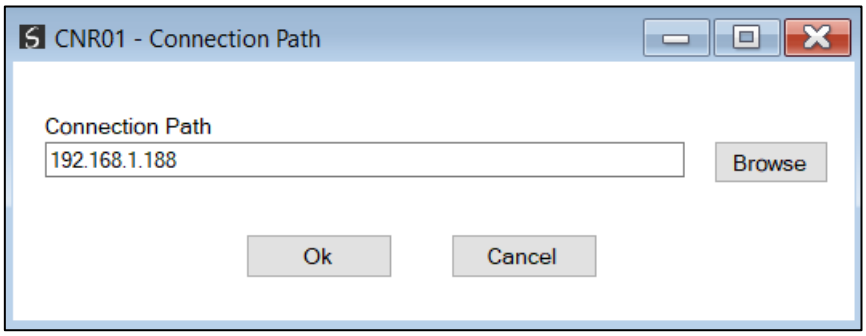


Figure 3.148 - Connection Path

To initiate the download, right-click on the module and select the ***Download*** option.

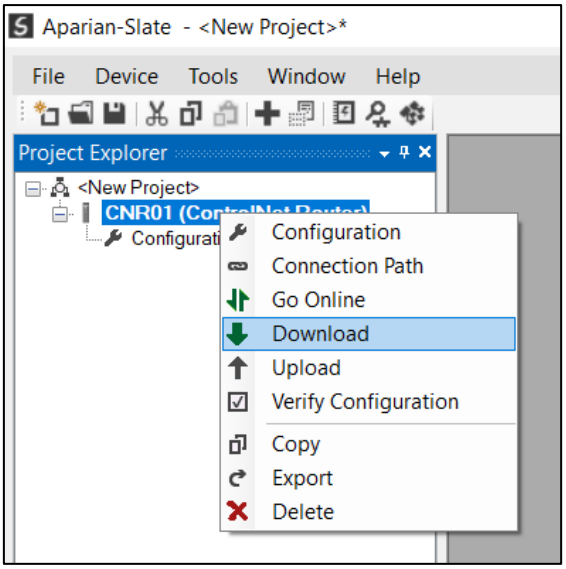


Figure 3.149 - Selecting Download

Once complete, the user will be notified that the download was successful.

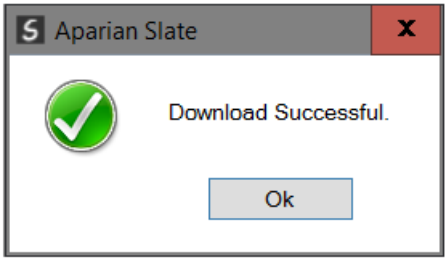


Figure 3.150 - Successful download

Within the Slate environment the module will be in the Online state, indicated by the green circle around the module. The module is now configured and will start operating immediately.

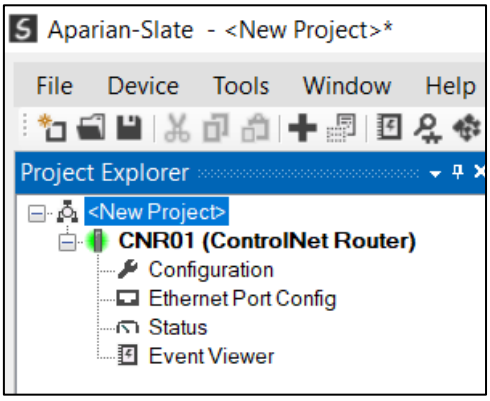


Figure 3.151 - Module online

4. LOGIX CONTROLLER PROGRAMMING OVER CONTROLNET

The ControlNet Router can be used to Program or Go-Online with a Logix Controller on a legacy ControlNet network from an EtherNet/IP network via the ControlNet Router. The ControlNet Router will use the unscheduled ControlNet bandwidth to access the other nodes on the ControlNet network (e.g. programming a Logix controller).

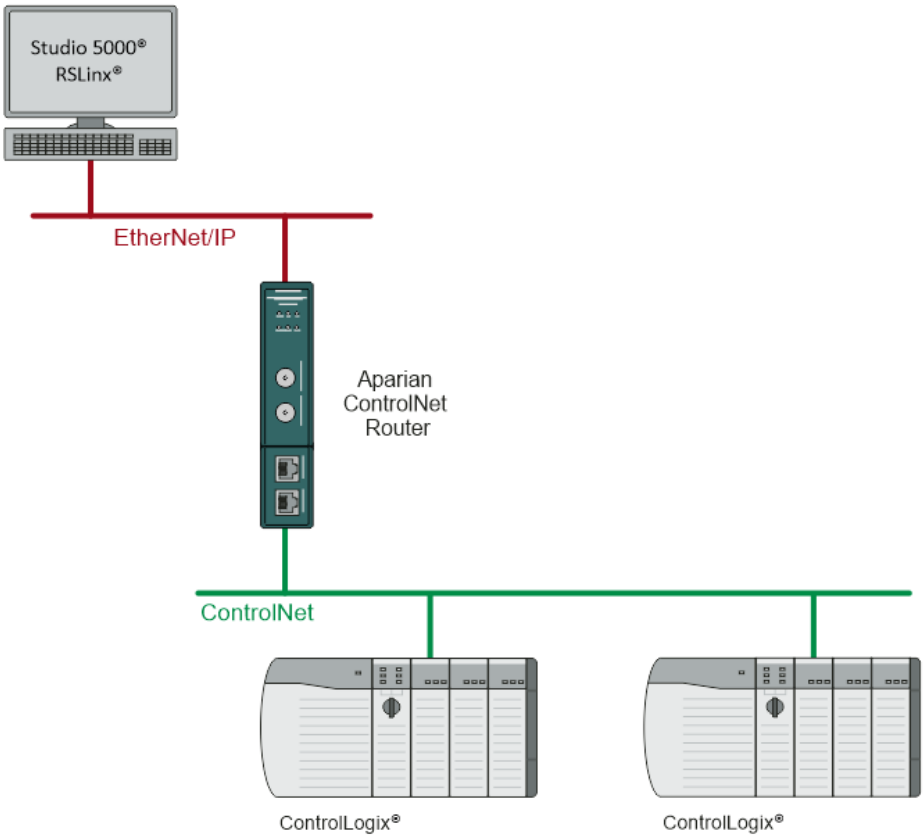


Figure 4.1 – Programming a Logix Controller over ControlNet via the ControlNet Router

4.1. RSLINX SETUP

The user will need to setup the path to the Logix Controller to access the Logix controller in RSLogix 5000 / Studio 5000. The ControlNet Router's EDS file must be registered on the programming PC. This can be achieved in two ways:

- Uploading the EDS file from the ControlNet Router online, or
- Downloading the EDS file from the Aparian website and manually registering it.

NOTE: Older revision of RSLinx may not support the uploading of the EDS file option, in which case, the second (manual) option should be used.

4.1.1. Uploading EDS File from Module

First add the ControlNet Router module in RSLinx. Under the Communications menu select the Configure Drivers menu item.

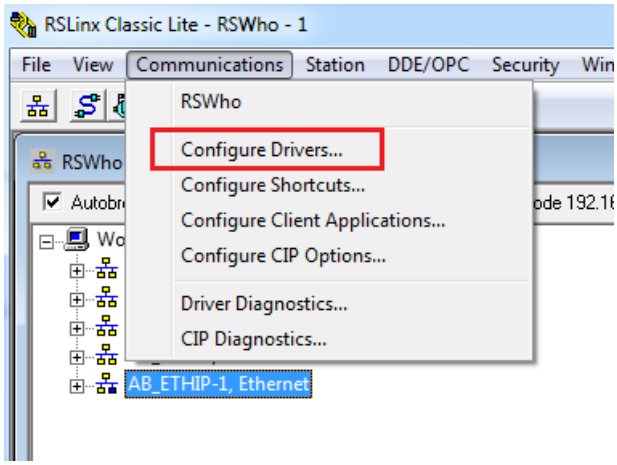


Figure 4.2 – RSLinx Driver Configuration

In the driver selection window, add the ***EtherNet/IP Driver***.

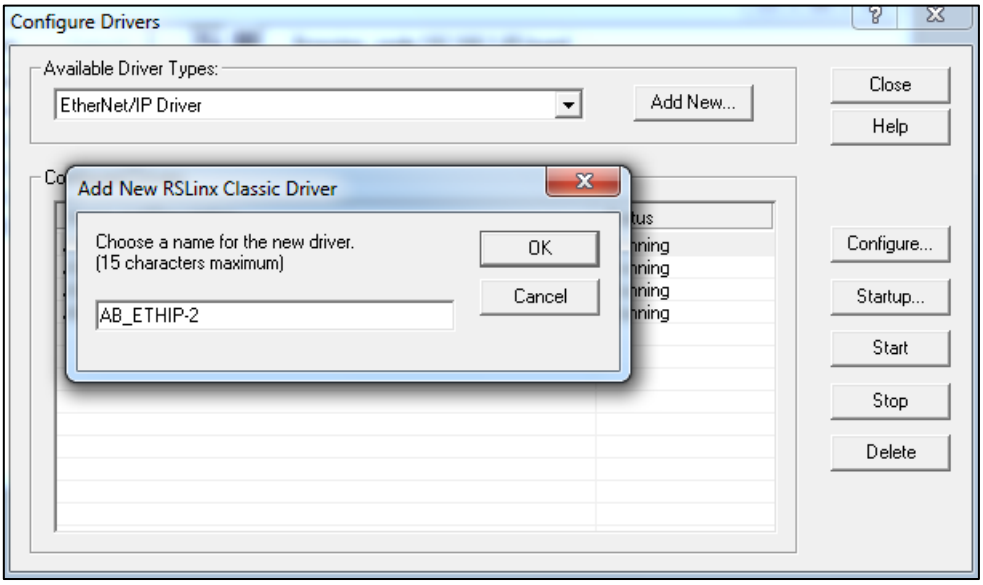


Figure 4.3 – RSLinx EtherNet/IP Driver

If the ControlNet Router is powered and connected on the network, it will appear in the RSLinx's RSWho browse window. Note that the device icon will be a yellow question mark, as

the EDS file has not yet been registered.

Right-click on the ControlNet Router in the RSWho browser and select the "***Upload EDS file from device***" option.

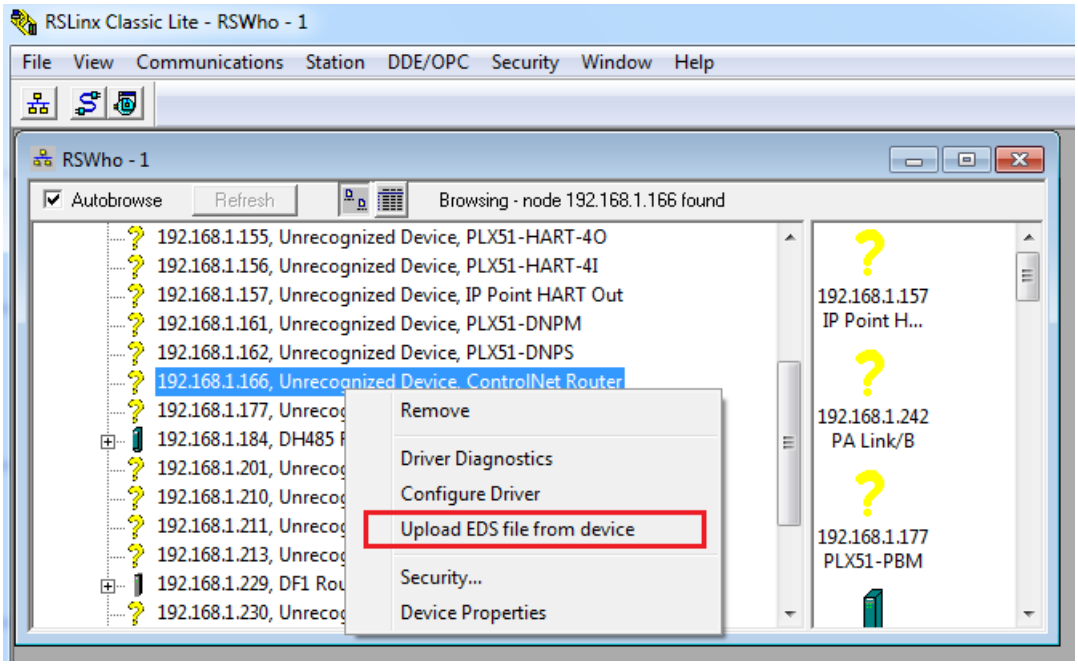


Figure 4.4 – RSLinx Upload EDS File from Device

The Rockwell Automation EDS Wizard will then launch. Follow the multiple steps by pressing the ***Next*** button and complete the EDS file registration.



Figure 4.5 – EDS Wizard

NOTE: Depending on the RSLinx revision, RSLinx may need to be **shut-down and re-started**. If RSLinx is running as a service, make sure the service is shut

down completely before re-starting.

After restarting RSLinx, the ControlNet Router will correctly appear in the RSWho browser. It will also show the underlying ControlNet Port, which, when expanded, will show the underlying ControlNet network nodes (and further down Logix Controller).

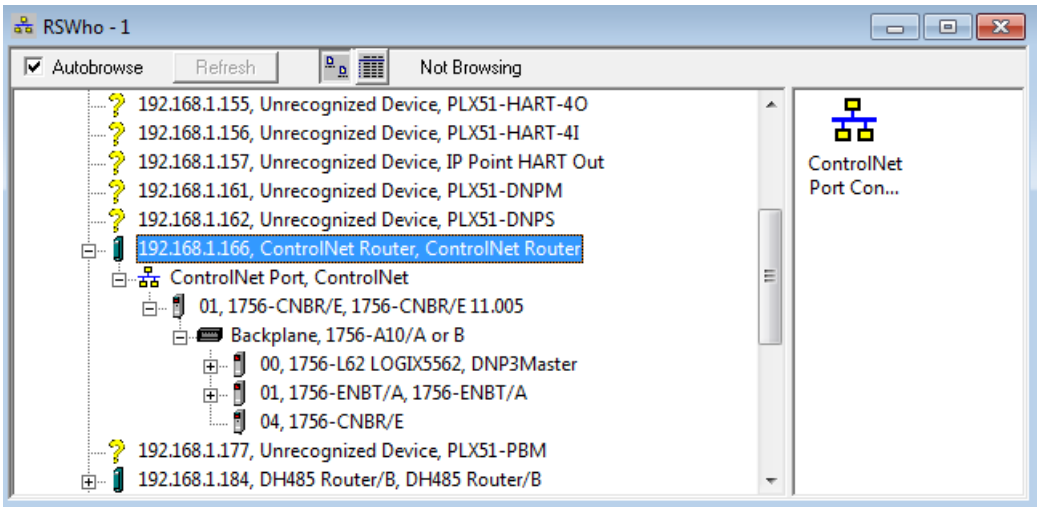


Figure 4.6 – Browsing the Logix Controller via ControlNet Router

4.1.2. Manually Installing the EDS File Option

To install the ControlNet Router's EDS file manually, it must first be downloaded from the Aparian website: https://www.aparian.com/products/controlnetrouter#downloads

The EDS file and associated icon are zipped in a single file. Save the file to local hard drive and unzip the file.

Using the Windows start button launch the Rockwell Software's ***EDS Hardware Installation Tool***.

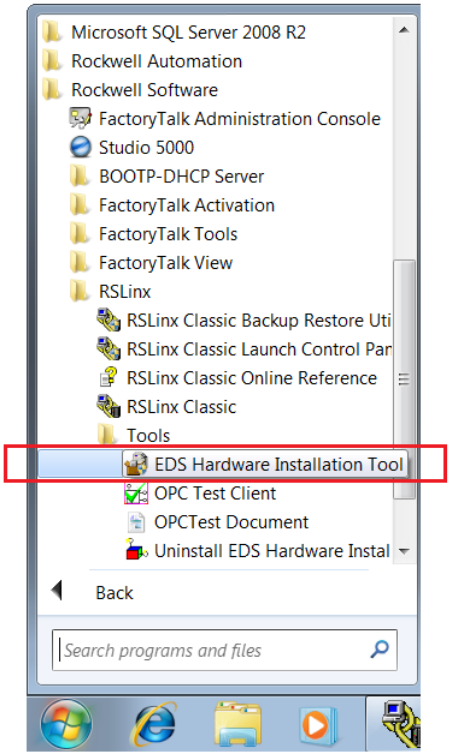


Figure 4.7 – Launch EDS Hardware Installation Tool

In the ***Hardware Installation Tool***, select the ***Add*** option.

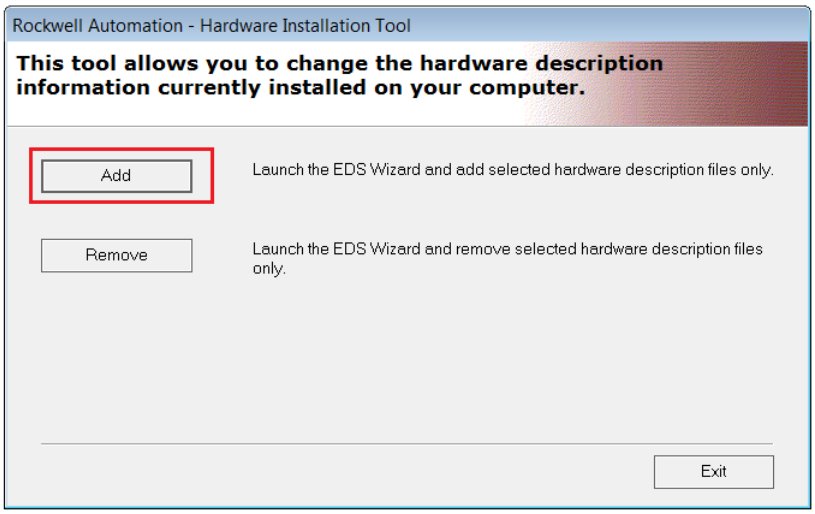


Figure 4.8 – Hardware Installation Tool - ADD

Use the ***Browse*** button to navigate to the folder where the EDS file was unzipped and select the EDS file.

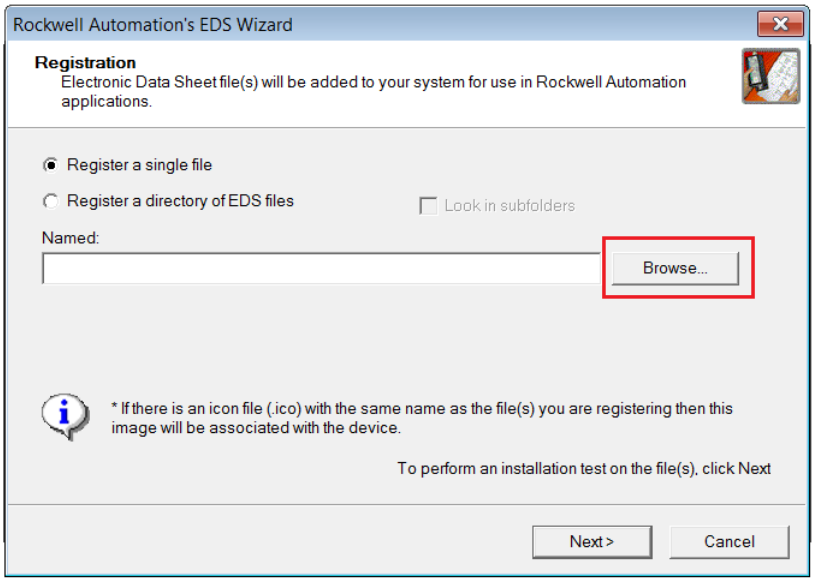


Figure 4.9 – EDS Registration

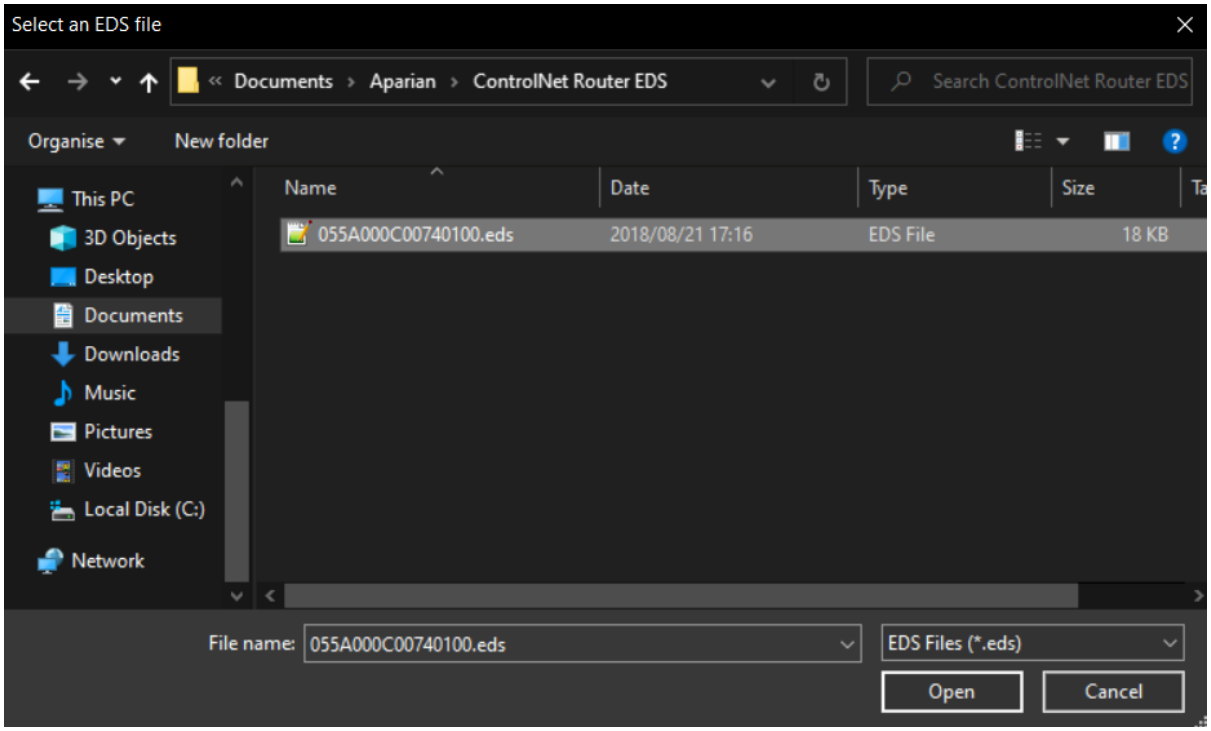


Figure 4.10 – EDS File Select

Follow the prompts and select the Next button to complete the EDS file registration.

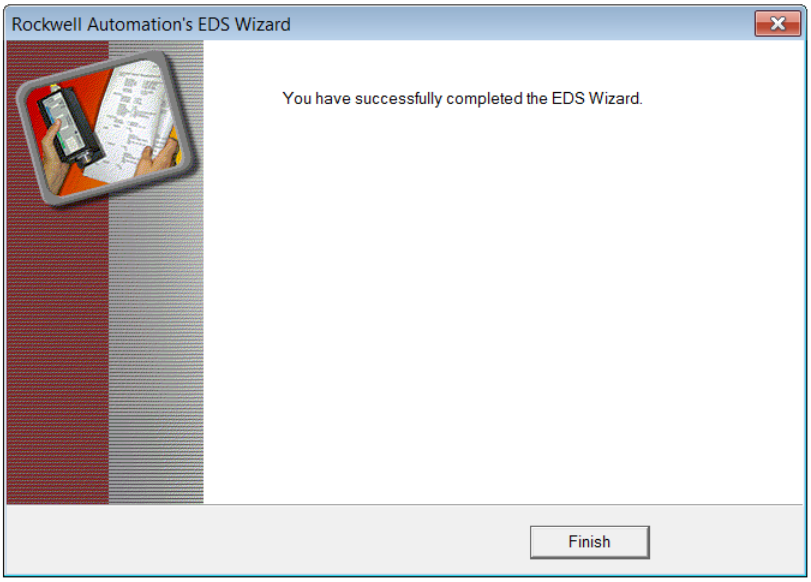


Figure 4.11 – EDS Registration Complete

NOTE: Depending on the RSLinx revision, RSLinx may need to be **shut-down and re-started**. If RSLinx is running as a service, make sure the service is shut down completely before re-starting.

4.2. RSLOGIX 5000

Programming using RSLogix 5000 can then continue as normal. Inside RSLogix5000 under the ***Communications*** menu, select the ***Who Active*** item.

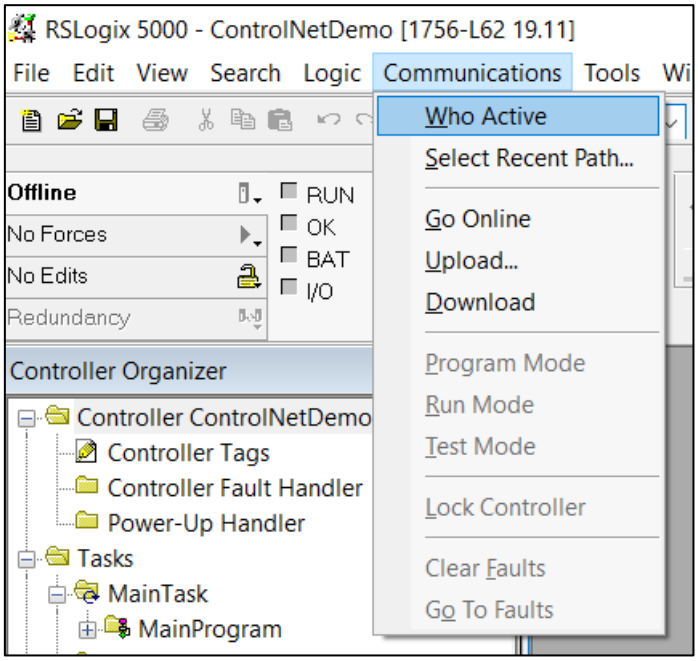


Figure 4.12 – System Communication

In the Communications window, browse to the Logix controller under the ControlNet Router, and select Online, Upload or Download as required.

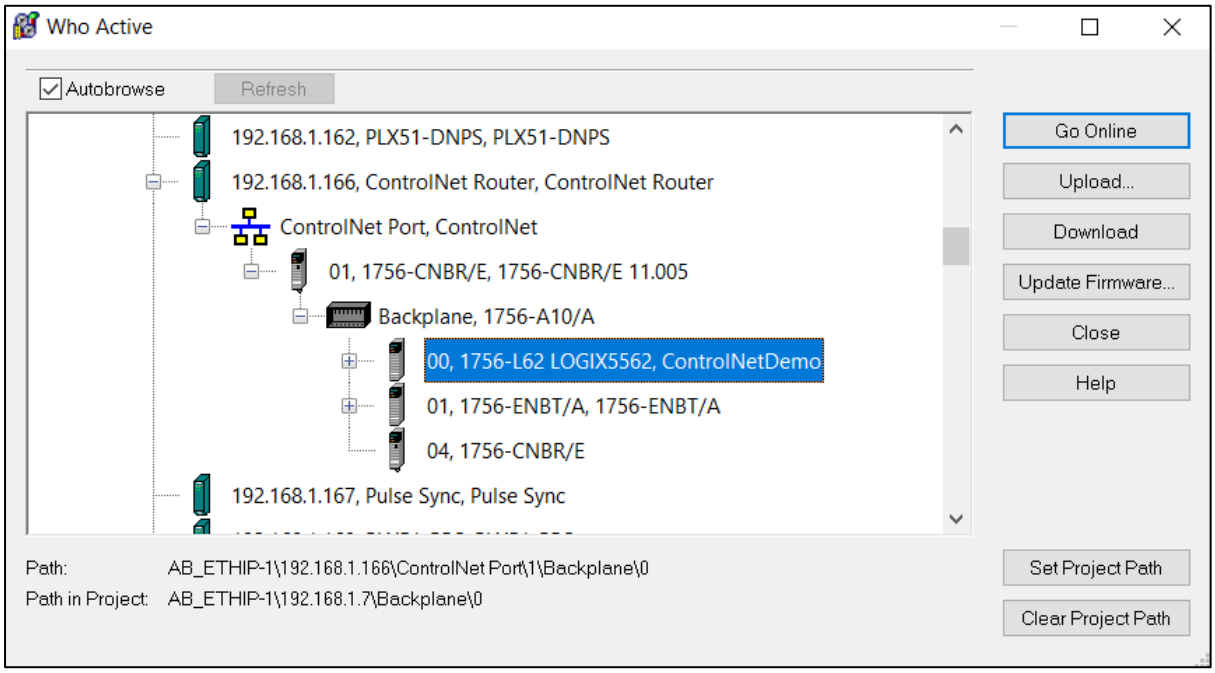


Figure 4.13 – Select Device

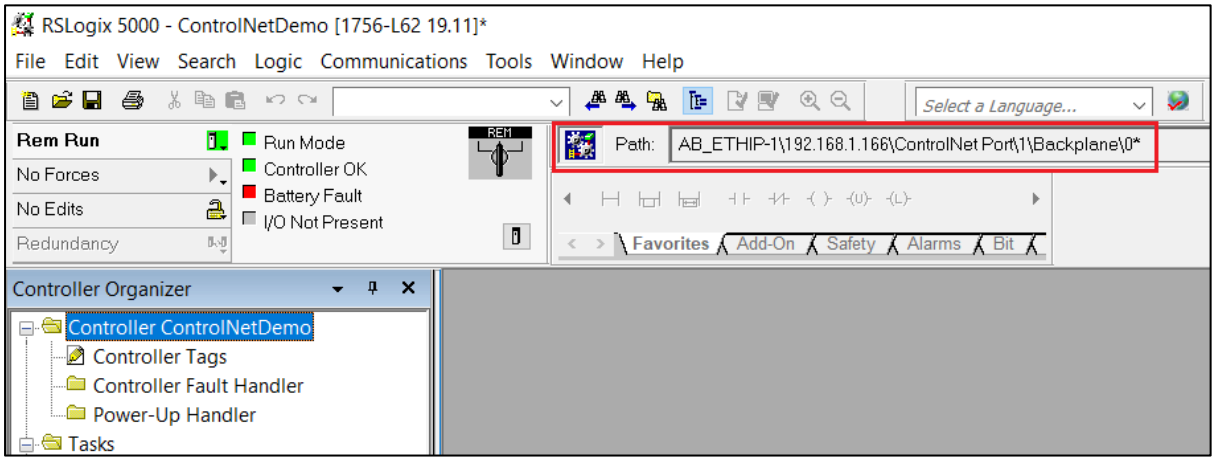


Figure 4.14 – Online with Controller

5. DEVICE FIRMWARE UPDATE

The ControlNet Router module supports in-field firmware upgrading. The latest firmware for the module can be downloaded from the Aparian website ***www.aparian.com***. The firmware is digitally signed, so only the correct firmware can be used.

To firmware upgrade the module, follow the steps below:

- From the tools menu in Slate, select the *DeviceFlash* utility.

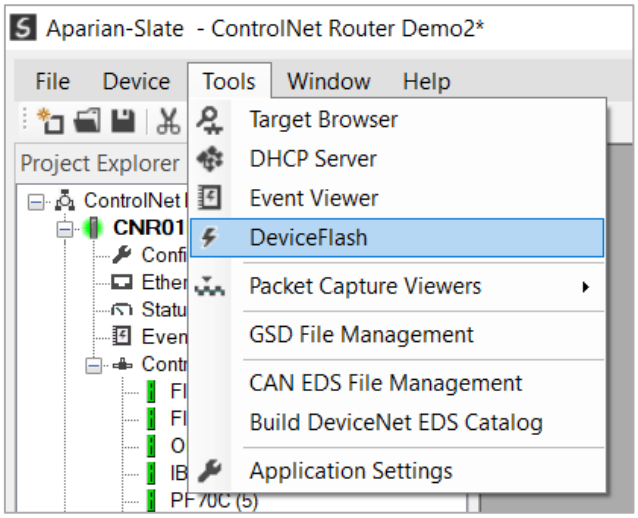


Figure 5.1 – Select DeviceFlash utility from Slate

- When the utility opens, the user will be prompted to select the binary file to be used to firmware upgrade the module.

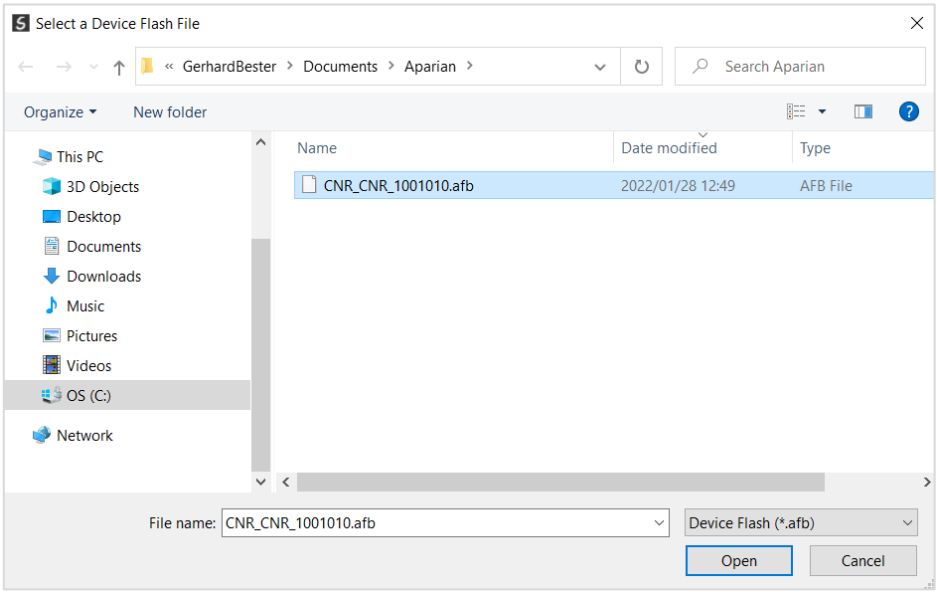


Figure 5.2 – Select the binary file

- After selecting the file, the user will be prompted to select the device to firmware upgrade on the local network.

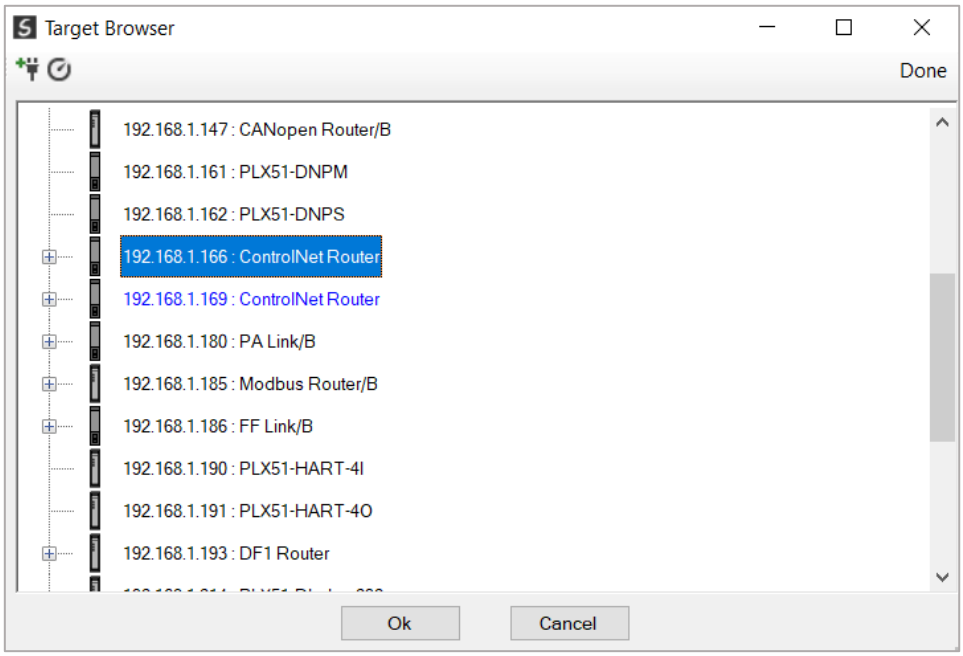


Figure 5.3 – Select the device to be updated

- After the device selection the user will be prompted if the device flash must start. The firmware update will take less than 2 minutes to complete.

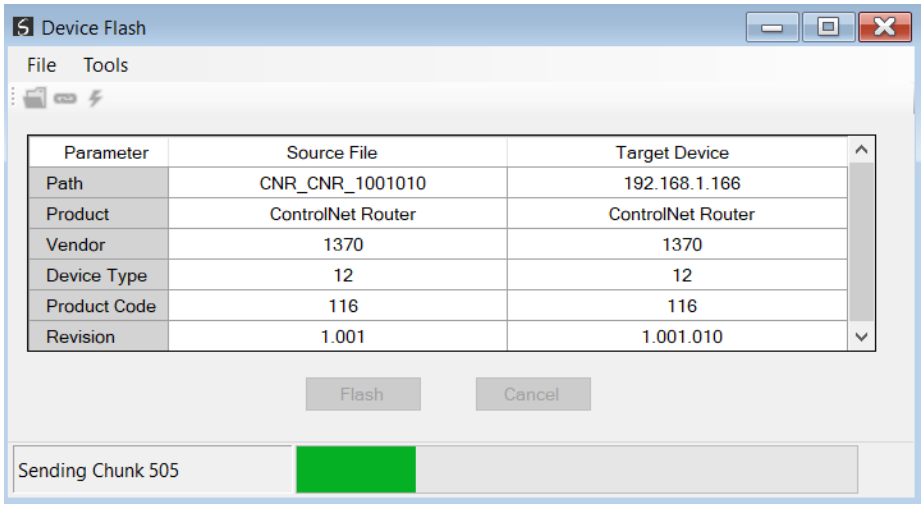


Figure 5.4 – Firmware update busy

- Once the firmware update has successfully completed, the Target Device textboxes will display green.

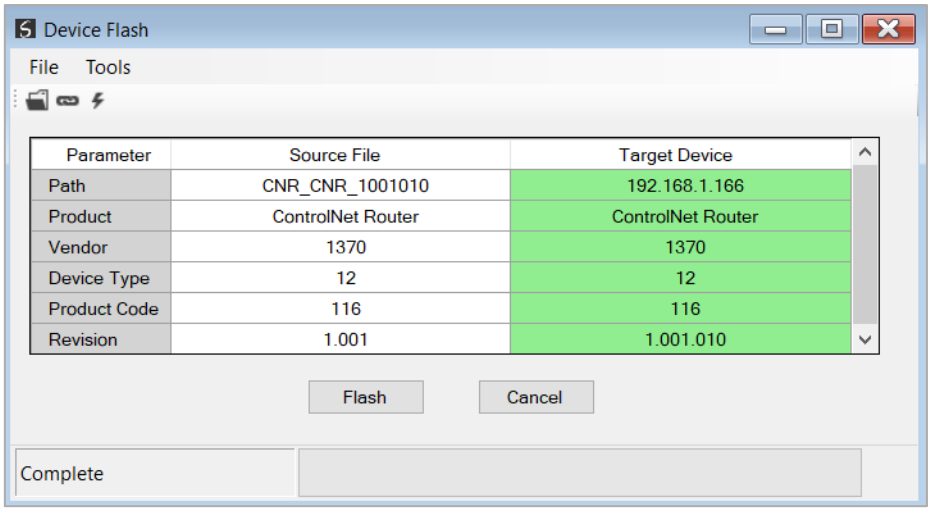


Figure 5.5 – Firmware update successfully completed.

NOTE: If for any reason the firmware update failed (e.g. power down during the update), then the module will revert back to the bootloader. The user can then simply reflash the module again to update it to the latest application firmware.

6. OPERATION

6.1. CONTROLNET TARGET

A Logix controller can own the ControlNet Router over ControlNet using a scheduled ControlNet connection when the ControlNet Router is operating as a ControlNet target. This will allow the ControlNet Router to exchange data with the Logix controller using the input and output assembly of the scheduled ControlNet connection.

6.1.1. SCHEDULED

When the module uses Scheduled ControlNet the Logix controller will establish a cyclic scheduled communication connection with the ControlNet Router. An input and output assembly is exchanged at a fix interval (RPI). The UDTs provided will convert the input and output arrays into tag-based assemblies. Refer to the additional information section in this document for the input and output UDTs.

6.1.1.1. INPUT ASSEMBLY (CONTROLNET)

The following parameters are used in the input assembly of the module when using the UDT ***AparianCNRInput***.

Parameter	Datatype	Description
GeneralStatus	SINT	Bit 0 – Configuration Valid Bit 1 – Routing Inhibited
CNetNodeNumber	SINT	The Node Number of the ControlNet Router on the network.
CNetLedState	SINT	The current status of the ControlNet LEDs. ControlNet A LED – Bit 0 to 3 ControlNet B LED – Bit 4 to 7 See the appendix for an explanation of the LED States.
CNetNetMode	SINT	The current Net Mode of the ControlNet network.
DeviceTemperature	REAL	The internal temperature of the ControlNet Router module.
MappedData	SINT[]	The data that can mapped from PLC5 file N7 and F8 when operating as a PCCC Client or by the Internal Mapping. Can be up to 400 bytes.

Table 6.1 – ControlNet Target input assembly parameters

6.1.1.2. OUTPUT ASSEMBLY (CONTROLNET)

The following parameters are used in the output assembly of the module.

Parameter	Datatype	Description
General Control	DINT	Bit 0 - Inhibit Routing
MappedData	SINT[]	The data that is mapped to PLC5 file N9 and F10 when operating as a PCCC Client or by the Internal Mapping. Can be up to 400 bytes.

Table 6.2 - ControlNet Target output assembly parameters

6.2. CONTROLNET ORIGINATOR

The ControlNet Router module can operate as a ControlNet originator. In this mode the module can exchange data from the Ethernet network with ControlNet devices using either the input and output assemblies of the Scheduled ControlNet connection to the device or using an explicit (unscheduled UCMM) ControlNet message to read or write data.

6.2.1. SCHEDULED CONTROLNET CONNECTIONS

In the example below, the ControlNet Router is owned by a Logix controller over EtherNet/IP while the ControlNet Router is owning some ControlNet IO. The data from the Logix controller is exchanged with the ControlNet IO.

Once the ControlNet Scheduled connections are setup and established then any Internal Map copying to a ControlNet IO device will be written to the output assembly of selected ControlNet device (Originator to Target). In the Internal Map the user will specify which device instance is the target and at what offset the data must be written in the output assembly.

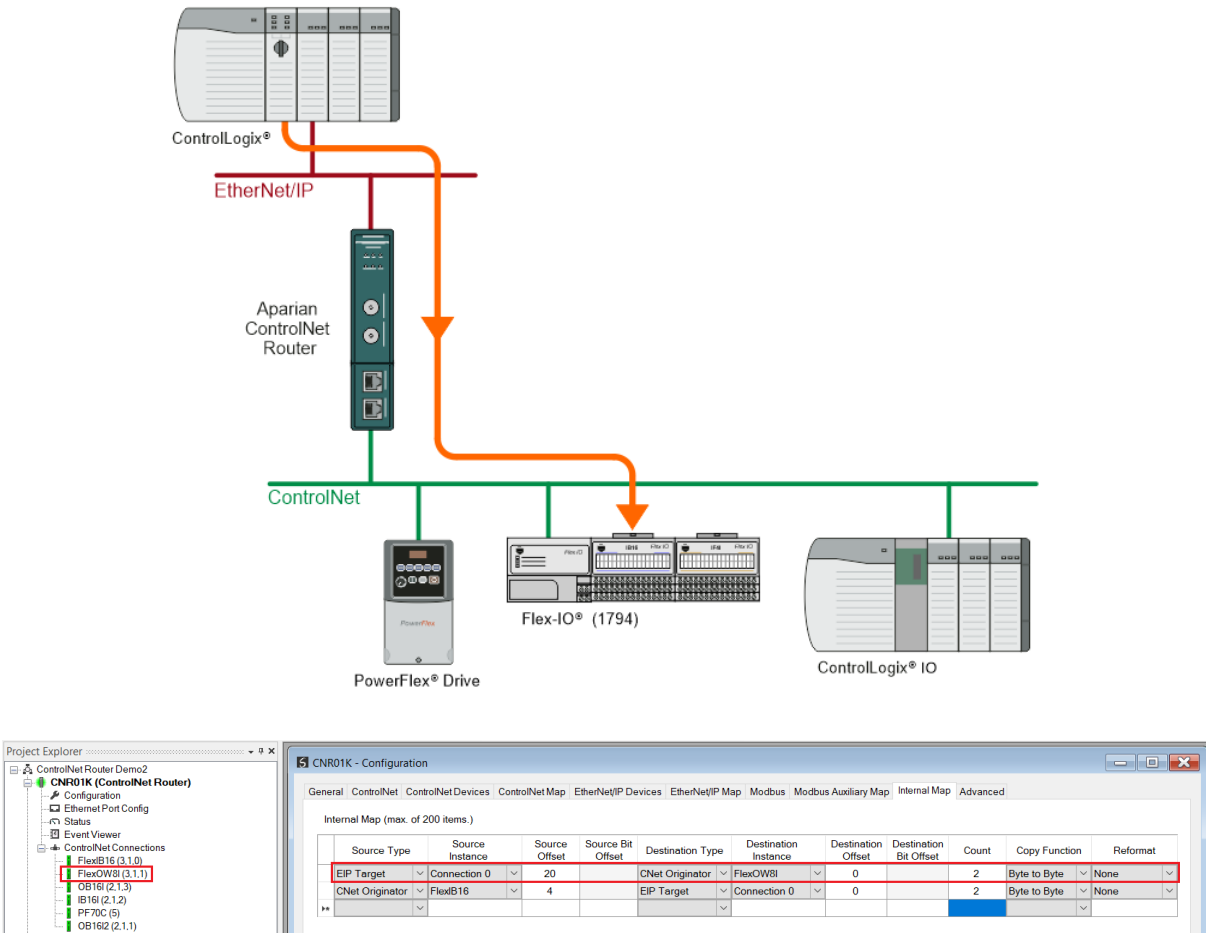


Figure 6.1 – Internal Mapping from EtherNet/IP to ControlNet Originator

Any Internal Map copying from a ControlNet IO device will be read from the input assembly of selected ControlNet device (Target to Originator). In the Internal Map the user will specify which device instance is the target and at what offset the data must be read from the input assembly.

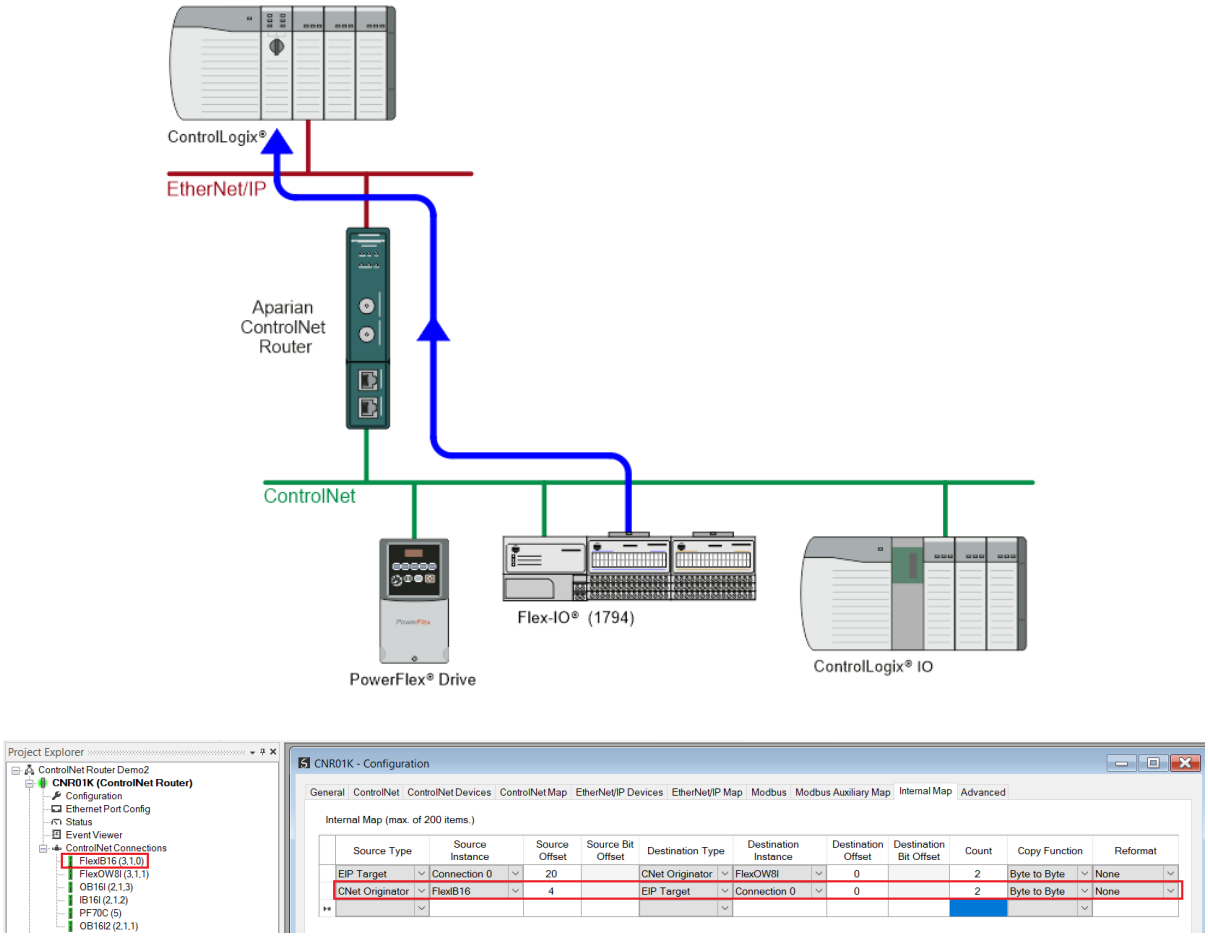


Figure 6.2 – Internal Mapping from ControlNet Originator to EtherNet/IP

6.2.2. EXPLICIT MESSAGING

When using the ControlNet Explicit Messaging, the user can configure up to 10 ControlNet devices which will be used for the Explicit Messaging. This configuration is located in the *ControlNet Devices* tab. Following this, the ControlNet Map of explicit messages needs to be configured. The Explicit Messaging uses the internal data space (IDS) which is size where data can be stored for exchanges between the explicit ControlNet devices and the Ethernet network.

The Input and Output IDS Offset is where the Explicit ControlNet device data will be read from or written to. The data in the IDS can then, in turn, be copied to or from, the Ethernet network using the Internal Map in the configuration.

In the example below, the ControlNet Router is receiving data from a Logix controller on EtherNet/IP, copying 4 bytes from the Class 1 EtherNet/IP Output assembly to the Internal Data Space at address 5030.

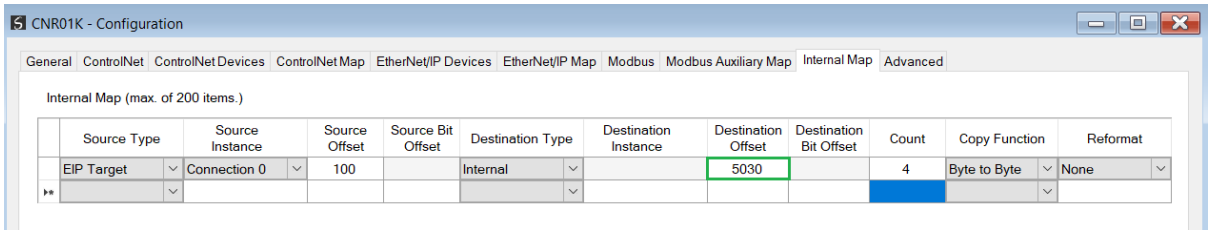


Figure 6.3 – Internal Mapping from EtherNet/IP to IDS

The ControlNet Router then uses the data from the EtherNet/IP Output assembly (which was copied to IDS offset 5030) to execute a Set Single Attribute to a target ControlNet device using explicit messaging.

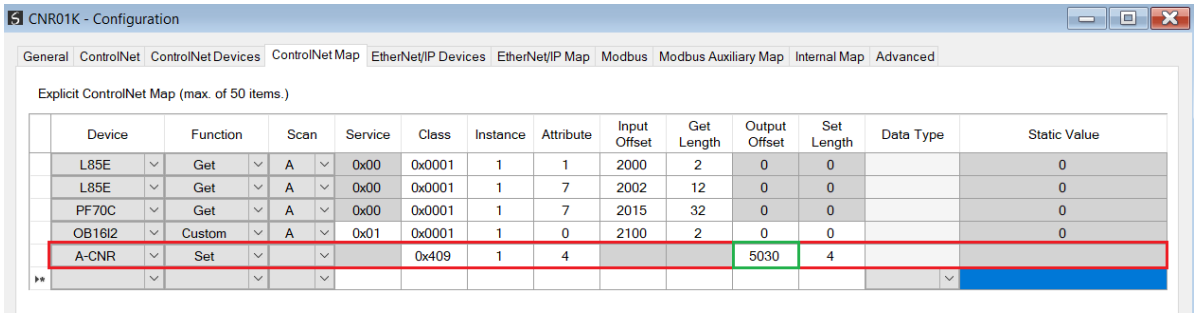


Figure 6.4 – Explicit Messaging – Set Function from IDS to ControlNet device

In the next example below, the ControlNet Router is receiving data from a ControlNet device using a Get Explicit Unscheduled ControlNet message and saving the data at IDS offset 6070.

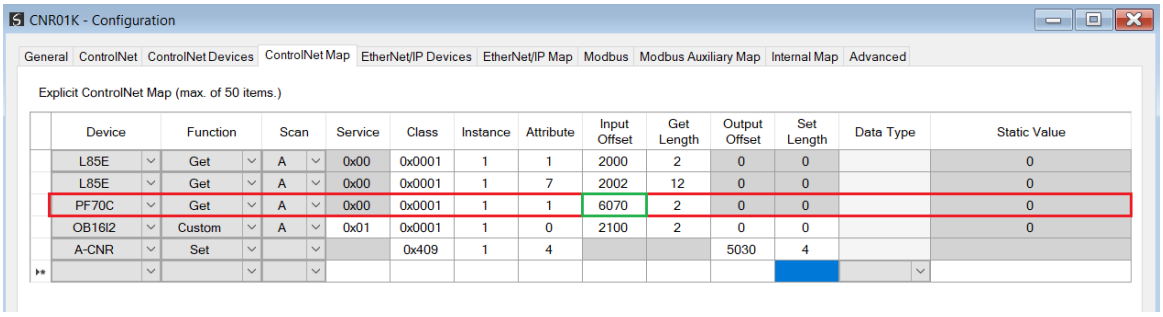


Figure 6.5 – Explicit Messaging – Get Function from ControlNet device to IDS

The ControlNet Router will then map the received data at IDS offset 6070 to the EtherNet/IP Input Assembly being sent back to the Logix controller.

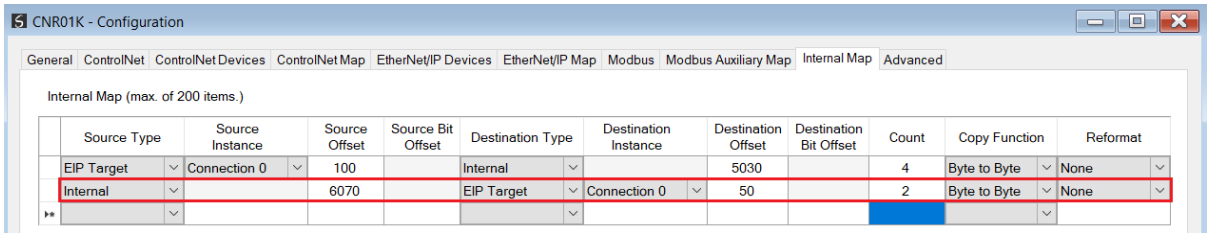


Figure 6.6 – Explicit Messaging – Internal Mapping from IDS to EtherNet/IP

6.3. CONTROLNET BRIDGE

When operating in Bridge Mode, the A-CNR will allow ControlNet IO to be connected to the Logix Controller via the A-CNR Add-on Profile (AOP). The ControlNet IO will be scheduled dynamically as the connections are established.

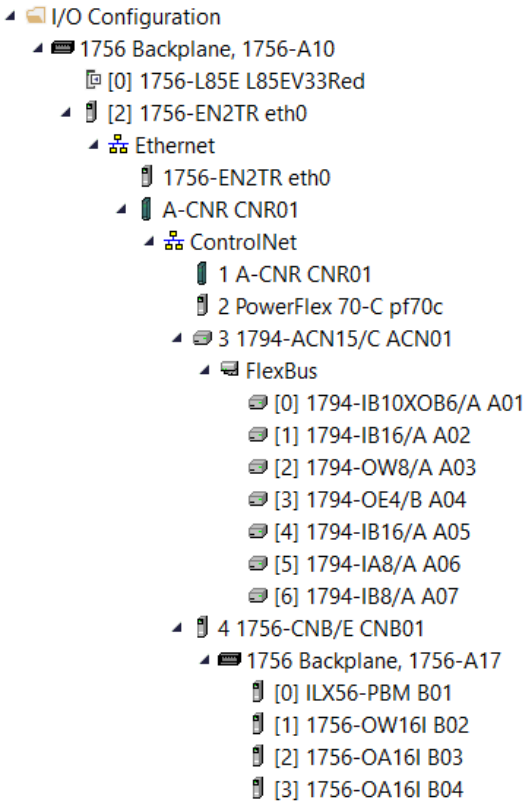


Figure 6.7 – ControlNet Bridge Mode – ControlNet IO

The A-CNR AOP will also allow the user to parameterize ControlNet IO (e.g., PowerFlex drive) vin Studio 5000 via the AOP of the respective ControlNet IO.

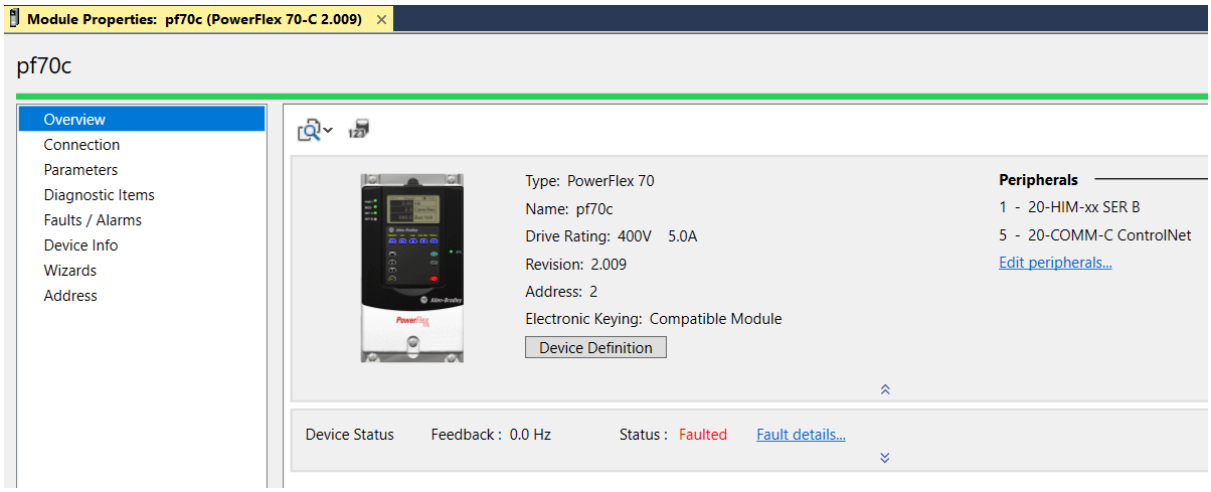


Figure 6.8 – ControlNet Bridge Mode – ControlNet IO Parameterization Overview

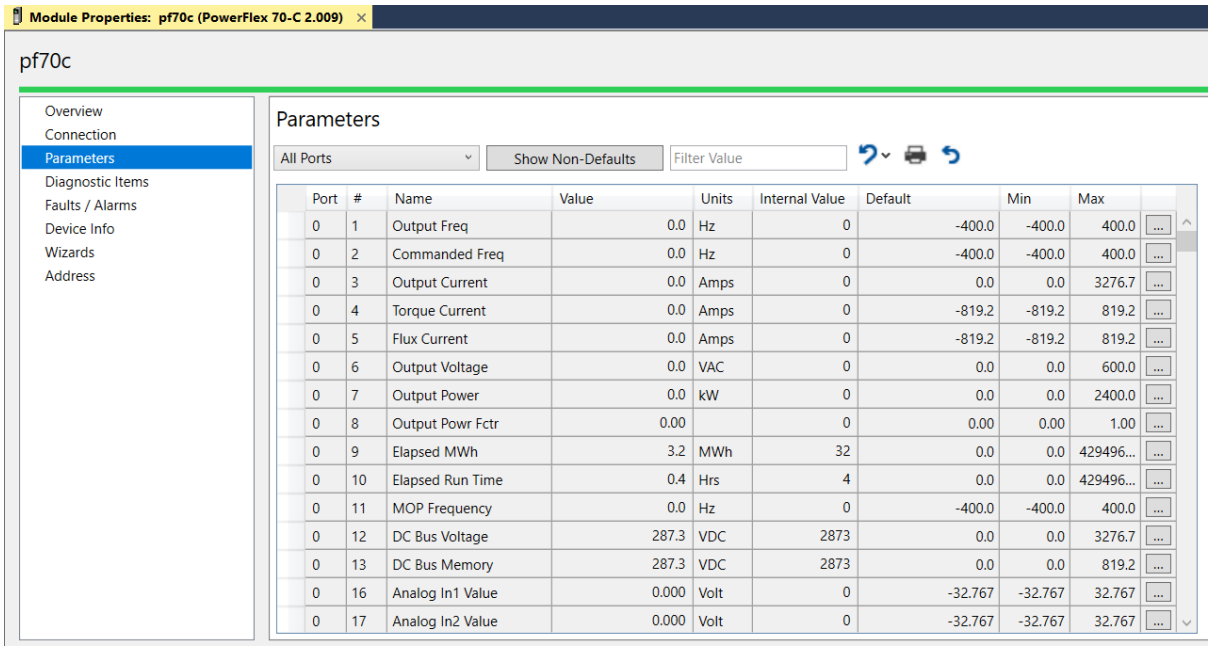


Figure 6.9 – ControlNet Bridge Mode – ControlNet IO Parameterization Parameters

6.3.1. PLC5 IO Data Exchange

When operating in bridge mode, the A-CNR can be used to integrate PLC5 IO into Logix over ControlNet.

NOTE: Adding and communicating with PLC5 IO from the Logix environment via the A-CNR, is done in the exact same way as when using a traditional 1756-CNB ControlNet bridge.

The PLC5 IO Rack can be added to the ControlNet IO tree in Logix (under the A-CNR). This can be done either manually or by module discovery when online in Studio5000 (as shown below).

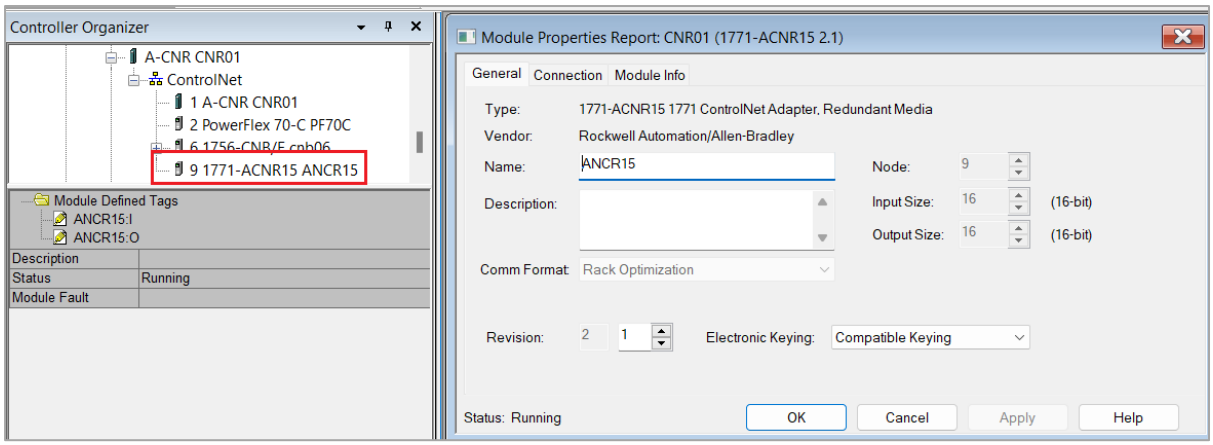


Figure 6.10 – PLC5 ControlNet Bridge – 1771-ACNR15

Once the 1771-ACNR15 is connected to the Logix controller (via the A-CNR), the data for Digital Input/Output modules will appear in the Logix tags for the 1771-ACNR15.

Name	Value	Style	Data Type
ANCR15:I	{...}		AB:1771_ACN15_40Bytes:I:0
ANCR15:I.RackStatus	2#0000_0000_0000_0000_0000_0000_0000_0000	Binary	DINT
ANCR15:I.InputGroupsValid	2#0000_0000_0000_0000_0000_0000_0100_0000	Binary	DINT
ANCR15:I.Data	{...}	Binary	INT[16]
ANCR15:I.Data[0]	2#0000_0000_0000_0000	Binary	INT
ANCR15:I.Data[1]	2#0000_0000_0000_0000	Binary	INT
ANCR15:I.Data[2]	2#0000_0000_0000_0000	Binary	INT
ANCR15:I.Data[3]	2#0000_0000_0000_0000	Binary	INT
ANCR15:I.Data[4]	2#0000_0000_0000_0000	Binary	INT
ANCR15:I.Data[5]	2#0000_0000_0000_0000	Binary	INT
ANCR15:I.Data[6]	2#0000_0000_0000_0010	Binary	INT
ANCR15:I.Data[7]	2#0000_0000_0000_0000	Binary	INT
ANCR15:I.Data[8]	2#0000_0000_0000_0000	Binary	INT
ANCR15:I.Data[9]	2#0000_0000_0000_0000	Binary	INT
ANCR15:I.Data[10]	2#0000_0000_0000_0000	Binary	INT
ANCR15:I.Data[11]	2#0000_0000_0000_0000	Binary	INT
ANCR15:I.Data[12]	2#0000_0000_0000_0000	Binary	INT
ANCR15:I.Data[13]	2#0000_0000_0000_0000	Binary	INT
ANCR15:I.Data[14]	2#0000_0000_0000_0000	Binary	INT
ANCR15:I.Data[15]	2#0000_0000_0000_0000	Binary	INT
ANCR15:O	{...}		AB:1771_ACN15_32Bytes:O:0
ANCR15:O.Data	{...}	Binary	INT[16]
ANCR15:O.Data[0]	2#0000_0000_0000_0000	Binary	INT
ANCR15:O.Data[1]	2#0000_0000_0000_0000	Binary	INT
ANCR15:O.Data[2]	2#0000_0000_0000_0000	Binary	INT
ANCR15:O.Data[3]	2#0000_0000_0000_0000	Binary	INT

Figure 6.11 –1771-ACNR15 – Input / Output assembly for status and Digital Input/Outputs

A Block Transfer Read/Write will be used in Logix to exchange data with 1771 Analog Cards (e.g., 1771-IFE). A MSG instruction will be used in Logix with a Message Type *Block Transfer Read* or *Block Transfer Write*.

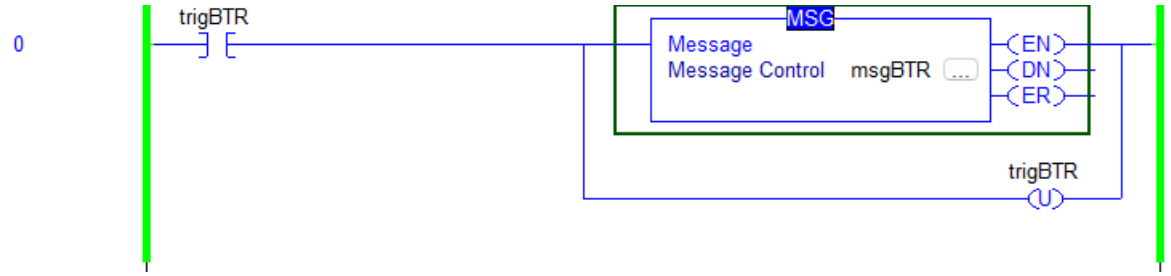


Figure 6.12 –1771 Analog Cards – MSG instruction

NOTE: Before the Block Transfer Read/Write can be successfully executed, the 1771-ACNR15 ControlNet bridge will need to have a Scheduled ControlNet connection with the A-CNR module (i.e., it will need to be connected in the Logix IO tree to the Logix controller under the A-CNR).

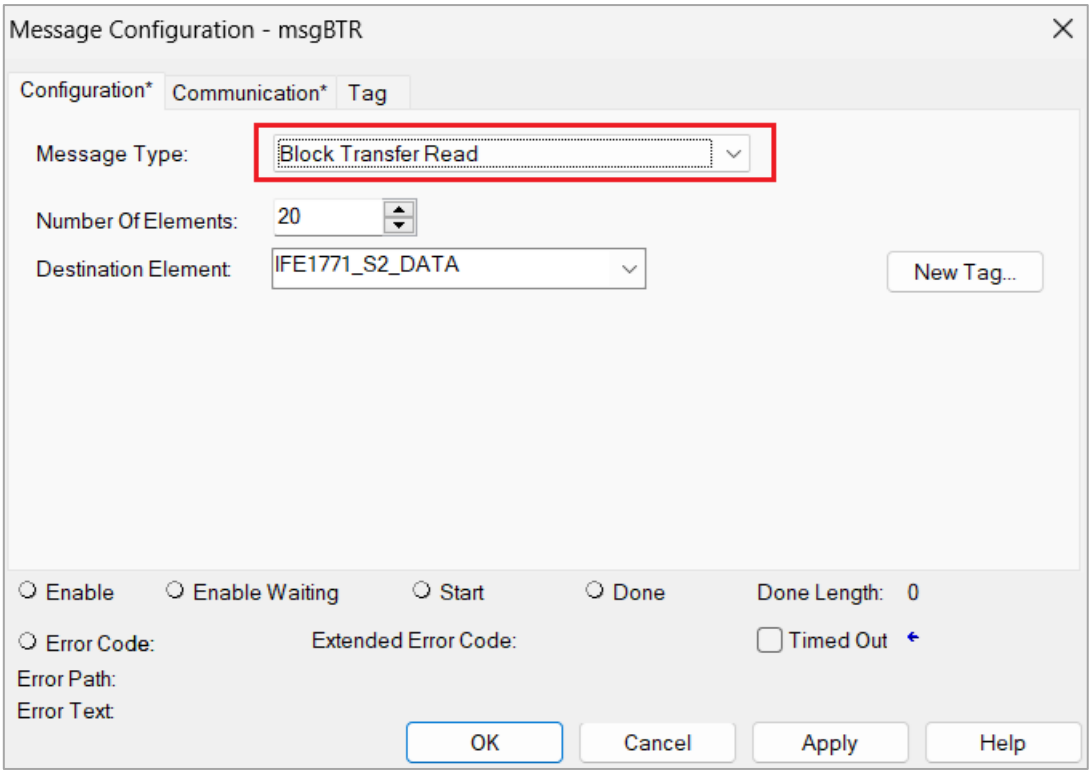


Figure 6.13 –1771 Analog Cards – Message Type

The path to the 1771-ACNR15 can either be entered manually or by browsing for the device under the A-CNR in the Logix IO tree.

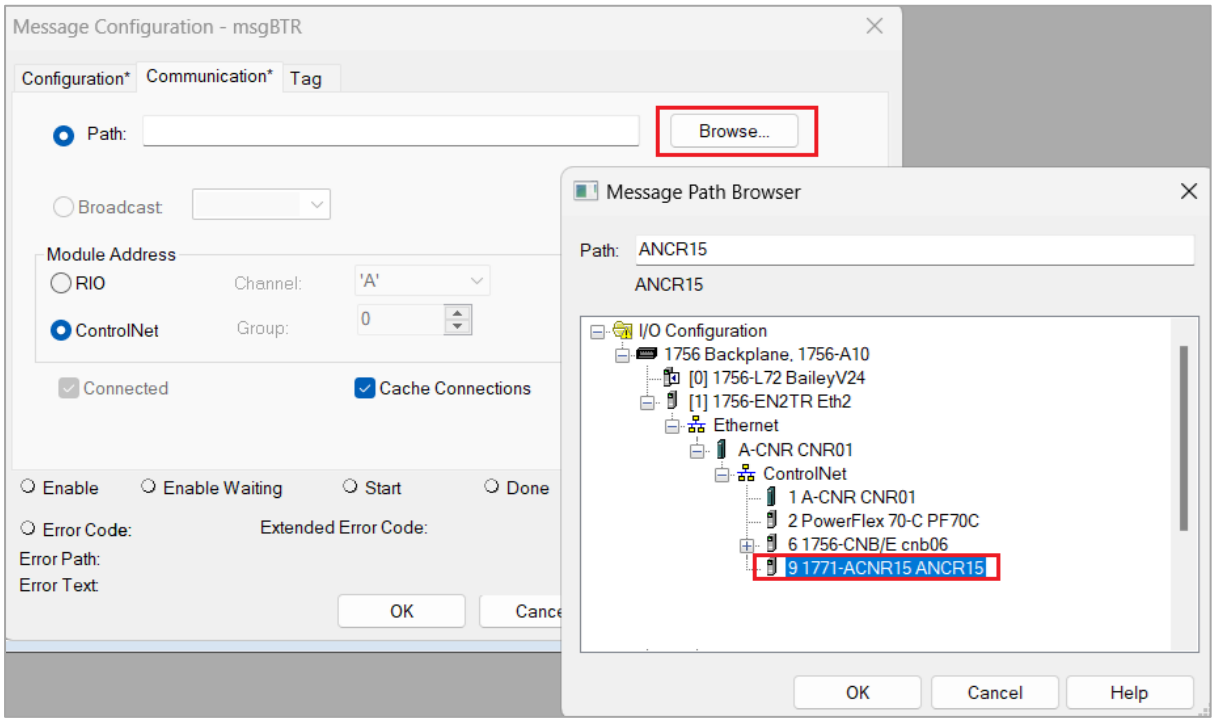


Figure 6.14 –1771 Analog Cards – Path to 1771-ACNR15

Once the Path has been setup to the 1771-ACNR15 ControlNet bridge, the Slot will need to be selected. The Slot will be the target 1771 Analog Card for the message exchange (note that the first slot starts at 0).

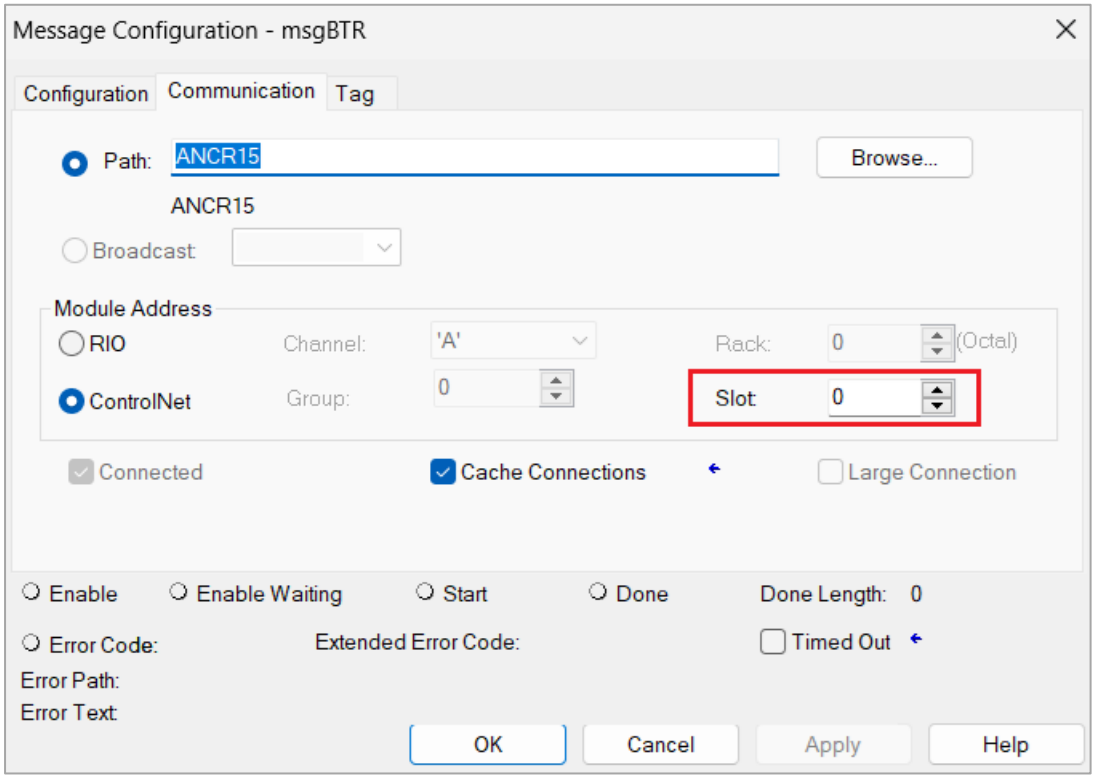


Figure 6.15 –1771 Analog Cards – Slot of the Analog Card

NOTE: Only one Block Transfer Read/Write per 1771 Analog Card can be executed at a time. Multiple simultaneous MSG instructions to the same 1771 Analog Card will result in some of the MSG instructions failing.

6.4. ETHERNET/IP TARGET

A controller (e.g. Logix controller) can own the ControlNet Router over EtherNet/IP using a Class 1 EtherNet/IP connection when the ControlNet Router is operating as an EtherNet/IP target. This will allow the ControlNet Router to exchange data with the controller using the input and output assembly of the Class 1 EtherNet/IP connection.

6.4.1. CLASS 1 CONNECTION

When the module uses EtherNet/IP (as a Target) the Logix controller will establish a cyclic Class 1 EtherNet/IP communication connection with the ControlNet Router. An input and output assembly is exchanged at a fix interval (RPI). The UDTs provided will convert the input and output arrays into tag-based assemblies. Refer to the additional information section in this document for the input and output UDTs.

6.4.1.1. INPUT ASSEMBLY (ETHERNET/IP)

The following parameters are used in the input assembly of the module when using the UDT ***AparianCNREInput***.

Parameter	Datatype	Description
GeneralStatus	SINT	Bit 0 – Configuration Valid Bit 1 – Routing Inhibited
CNetNodeNumber	SINT	The Node Number of the ControlNet Router on the network.
CNetLedState	SINT	The current status of the ControlNet LEDs. ControlNet A LED – Bit 0 to 3 ControlNet B LED – Bit 4 to 7 See the appendix for an explanation of the LED States.
CNetNetMode	SINT	The current Net Mode of the ControlNet network.
DeviceTemperature	REAL	The internal temperature of the ControlNet Router module.
MappedData	SINT[]	The data that can mapped from the Internal Mapping. Can be up to 492 bytes.

Table 6.3 – EtherNet/IP Target input assembly parameters

6.4.1.2. OUTPUT ASSEMBLY (ETHERNET/IP)

The following parameters are used in the output assembly of the module.

Parameter	Datatype	Description
General Control	DINT	Bit 0 - Inhibit Routing
MappedData	SINT[]	The data that is mapped to by the Internal Mapping. Can be up to 492 bytes.

Table 6.4 - EtherNet/IP Target output assembly parameters

6.5. ETHERNET/IP ORIGINATOR

The ControlNet Router module can operate as an EtherNet/IP originator. In this mode the module can exchange data from the ControlNet network with EtherNet/IP devices using either the input and output assemblies of the Class 1 EtherNet/IP connection to the device or using explicit (Class 3 or UCMM) EtherNet/IP messages.

6.5.1. ETHERNET/IP CLASS 1 CONNECTIONS

In the example below, the ControlNet Router is owned by a Logix controller over ControlNet while the ControlNet Router is owning some EtherNet/IP IO. The data from the Logix controller is exchanged with those of the EtherNet/IP IO.

Once the EtherNet/IP Class 1 connections are setup and established then any Internal Map copying to an EtherNet/IP device will be written to the output assembly of the selected EtherNet/IP device (Originator to Target). In the Internal Map the user will specify which device instance is the target and at what offset the data must be written in the output assembly.

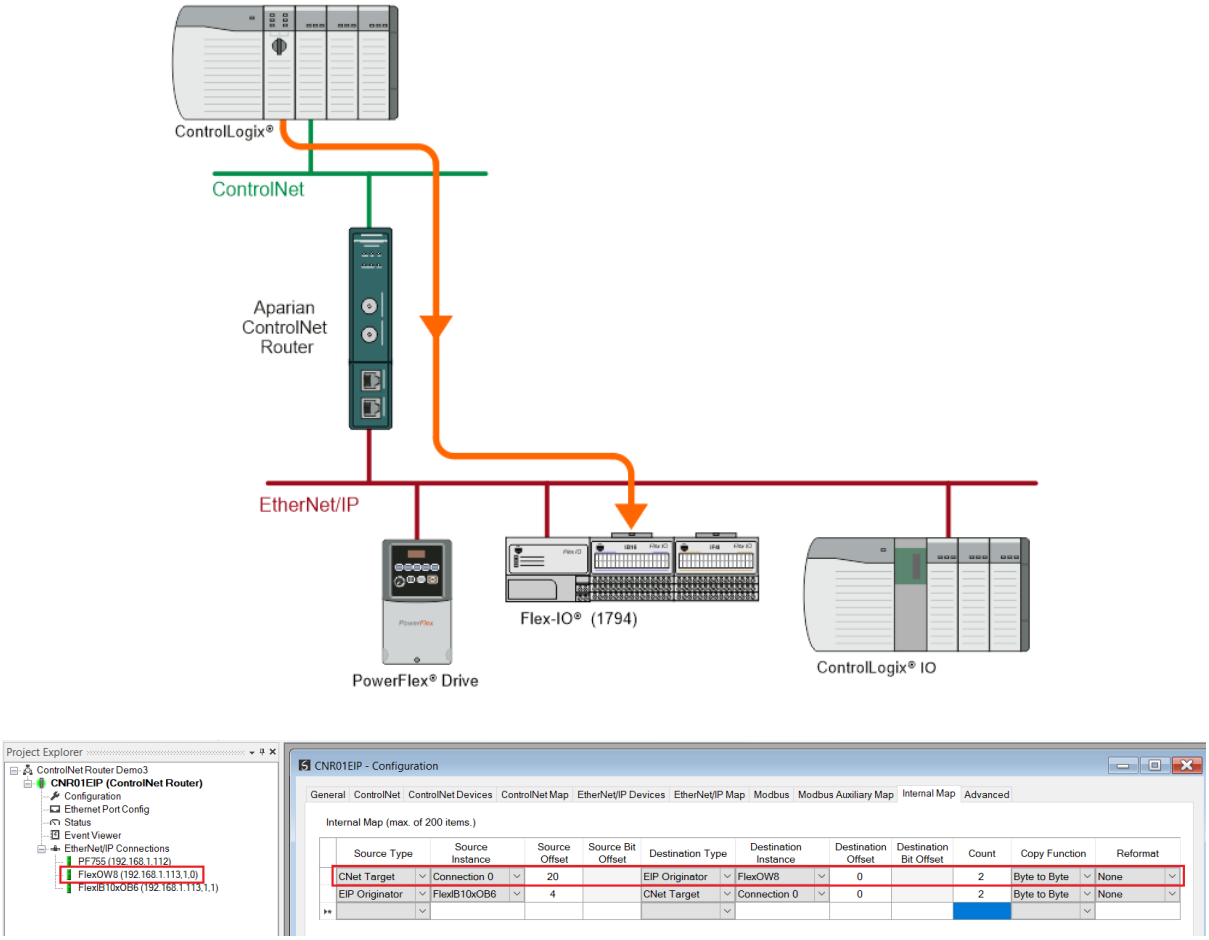


Figure 6.16 – Internal Mapping from ControlNet to EtherNet/IP Originator

Any Internal Map copying from an EtherNet/IP device will be read from the input assembly of selected EtherNet/IP device (Target to Originator). In the Internal Map the user will specify which device instance is the target and at what offset the data must be read from the input assembly.

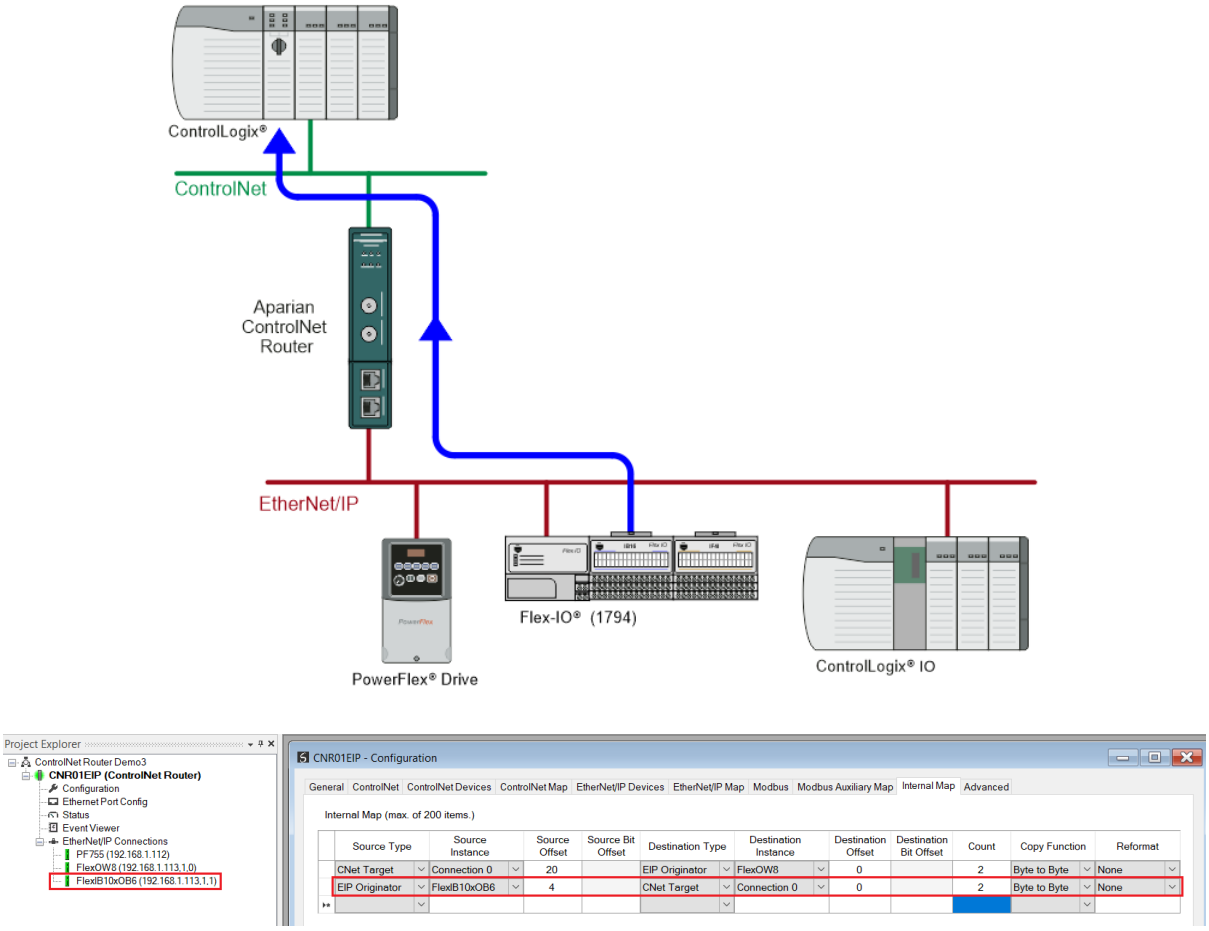


Figure 6.17 – Internal Mapping from EtherNet/IP Originator to ControlNet

6.5.2. EXPLICIT MESSAGING

When using the EtherNet/IP Explicit Messaging, the user can configure up to 10 EtherNet/IP devices which will be used for the Explicit Messaging. This configuration is located in the *EtherNet/IP Devices* tab. Following this, the EtherNet/IP Map of explicit messages needs to be configured. The Explicit Messaging uses the internal data space (IDS) where data can be stored for exchanges between the explicit EtherNet/IP devices and the ControlNet network.

The Input and Output IDS Offset is where the Explicit EtherNet/IP device data will be read from, or written to. The data in the IDS can then, in turn, be copied to or from the ControlNet network using the Internal Map in the configuration.

In the below example, the ControlNet Router is receiving data from a Logix controller on ControlNet, copying 4 bytes from the Scheduled ControlNet Output assembly to the Internal Data Space at address 4080.

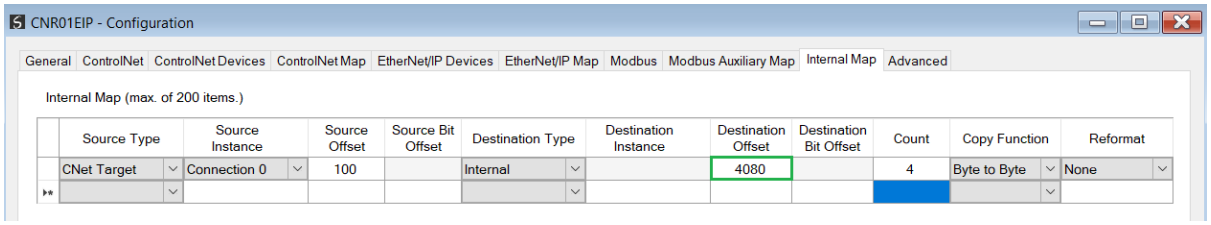


Figure 6.18 – Internal Mapping from ControlNet to IDS

The ControlNet Router then uses the data from the ControlNet Output assembly (which was copied to IDS offset 4080) to execute a Set Single Attribute to a target EtherNet/IP device using explicit messaging.

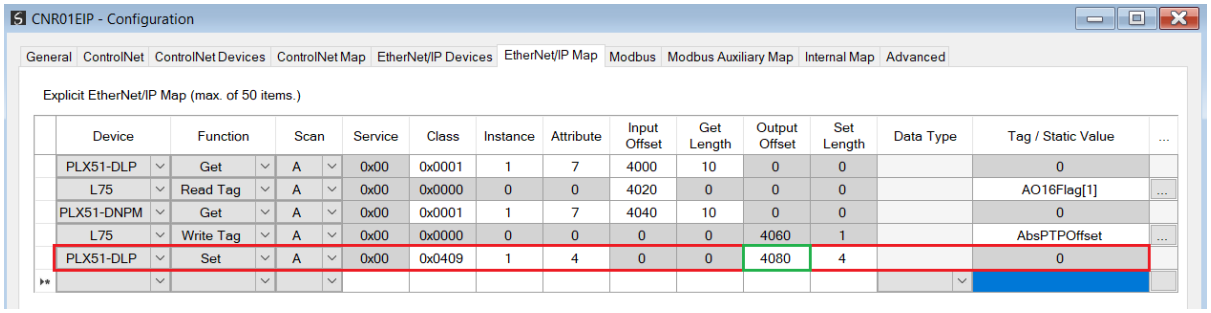


Figure 6.19 – Explicit Messaging – Set Function from IDS to EtherNet/IP device

In the next example below, the ControlNet Router is receiving data from an EtherNet/IP device using a Get Explicit EtherNet/IP message and saving the data at IDS offset 4040.

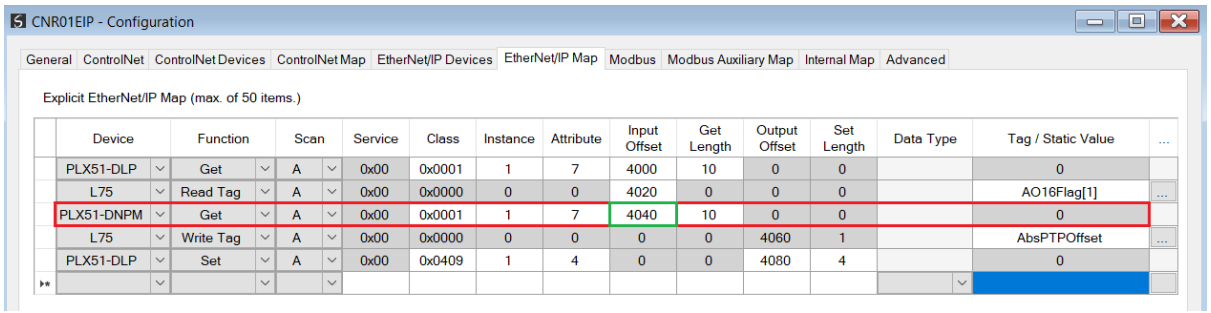


Figure 6.20 – Explicit Messaging – Get Function from EtherNet/IP device to IDS

The ControlNet Router will then map the received data at IDS offset 4040 to the ControlNet Input Assembly being sent back to the Logix controller.

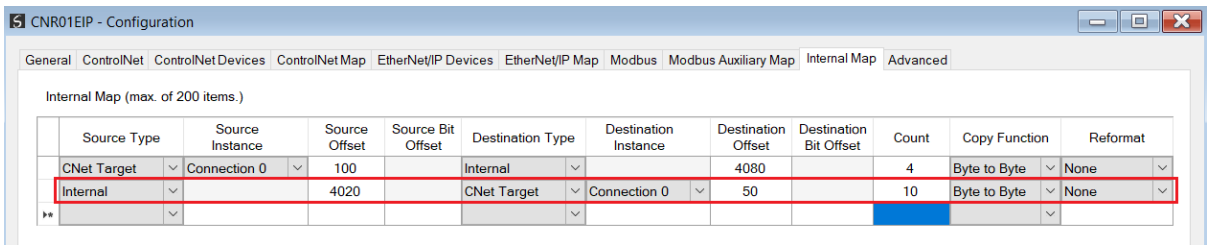


Figure 6.21 – Explicit Messaging – Internal Mapping from IDS to ControlNet

6.6. MODBUS TCP MASTER

When the ControlNet Router has the Ethernet Mode set to Modbus TCP Master, then the ControlNet data can be mapped to and from configurable internal Modbus Registers using the Internal Map.

The internal Modbus Registers are then asynchronously exchanged with up to 20 Modbus devices as configured in the Modbus Auxiliary Map. In this mapping the user can exchange (read or write) data between the internal Modbus Registers and a remote Modbus device.

In the example below the ControlNet Router with the Ethernet Mode set to Modbus TCP Master will read multiple Modbus Holding Registers from a Modbus Slave device and then map the received data to a ControlNet Input assembly where the module ControlNet Mode is set to Target.

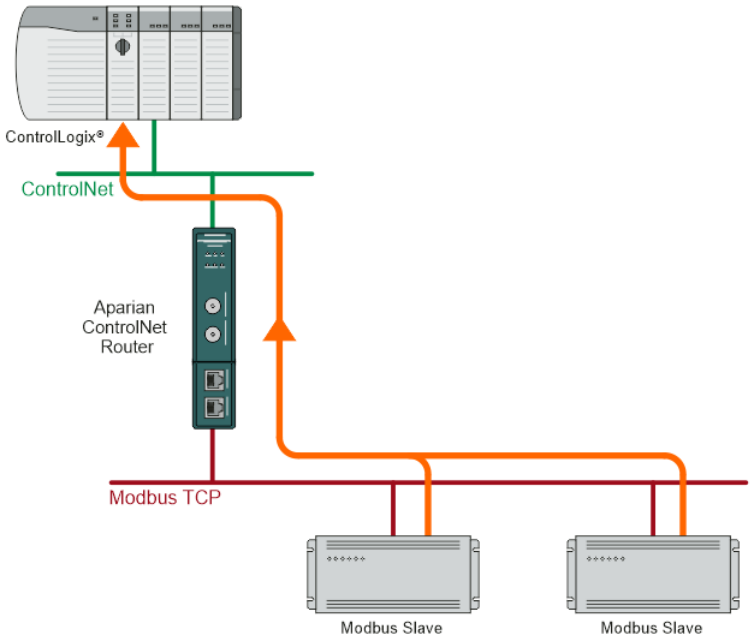


Figure 6.22 – Modbus Master to ControlNet operation

For this example the user will configure the Modbus Auxiliary Map to read data from a Modbus Slave device. The ControlNet Router will request data from Modbus Holding Register 4000 (from the external Modbus Slave) and write it to the module's internal Modbus Holding Register 3000.

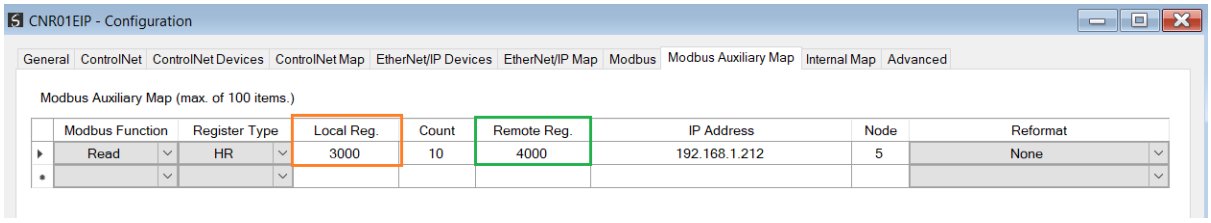


Figure 6.23 – Modbus Master Aux Mapping

Next the data saved in the internal Modbus Register at MB Holding Register 3000 is mapped to the ControlNet Scheduled Target connection input assembly with offset 50.

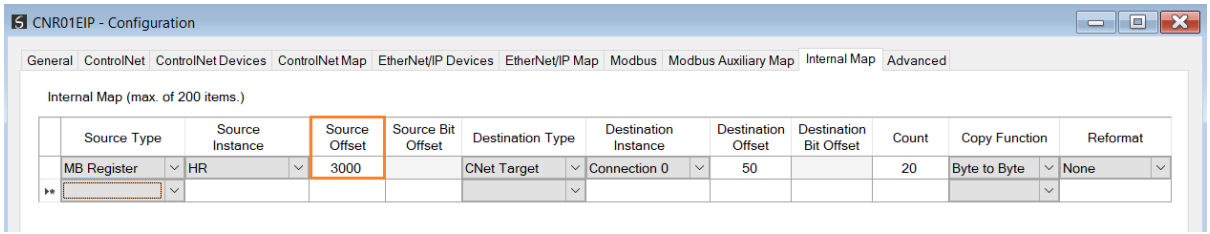


Figure 6.24 – Internal Mapping from Modbus Register to ControlNet

NOTE: The user will need to ensure that when writing to the ControlNet Router Modbus Holding Registers that the registers holding data from the device are not inadvertently overwritten.

6.7. MODBUS TCP SLAVE

When the ControlNet Router has the Ethernet Mode set to Modbus TCP Slave, then the ControlNet data can be mapped to and from configurable internal Modbus Registers and offsets using the Internal Map.

The internal Modbus Registers can then be asynchronously exchanged with a remote Modbus TCP Master. The remote Modbus Master can read or write to the configured Modbus addresses to access the ControlNet data that has been mapped to the Modbus Registers.

In the example below the ControlNet Router, with the Ethernet Mode set to Modbus TCP Slave, will have multiple Modbus Holding Registers written from a Modbus TCP Master and then map the received data to ControlNet IO that is owned with a scheduled ControlNet connection (when ControlNet mode is Originator).

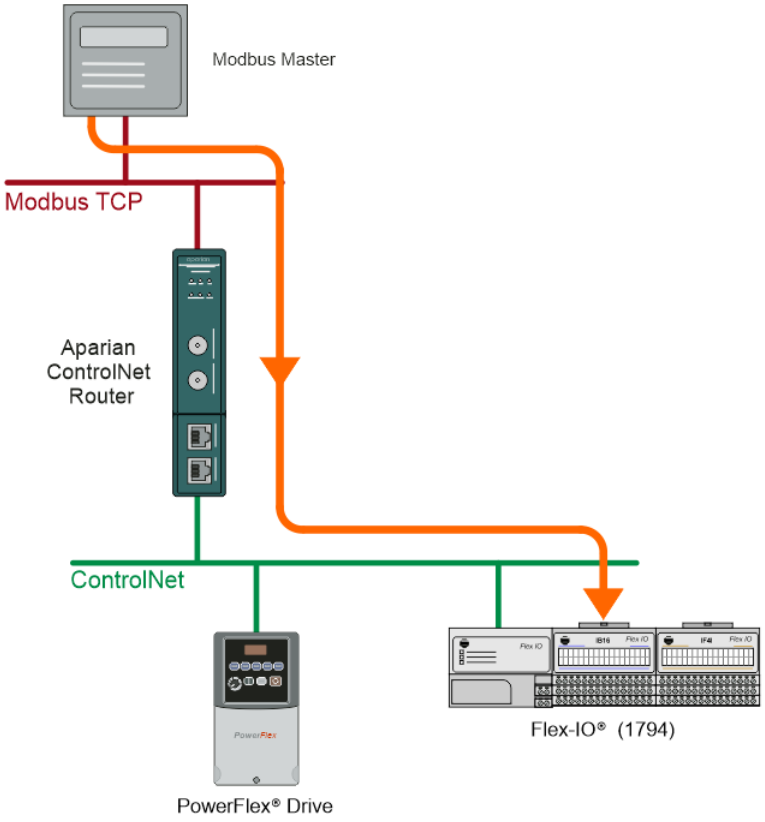


Figure 6.25 – Modbus TCP Master to ControlNet IO operation

For this example the remote Modbus TCP Master will write data to Modbus Holding Register 4000 in the ControlNet Router. The ControlNet Router will map the received Modbus data (at Holding Register 4000) to the output assembly of ControlNet IO (FlexOW8) being owned (using Scheduled ControlNet) by the ControlNet Router.

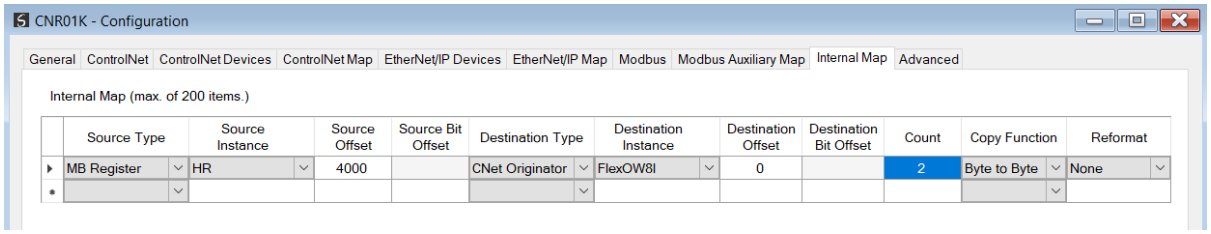


Figure 6.26 – Internal Mapping from Modbus Register to ControlNet IO



NOTE: The user will need to ensure that when writing to the Control Router Modbus Holding Registers that the registers holding data from the device are not inadvertently overwritten.

6.8. FTVIEW / PANELVIEW INTERFACING

The ControlNet Router can exchange data with a PanelView using a Scheduled ControlNet connection. The data being exchanged can then be accessed from a PanelView HMI by using

PLC5 emulation.

This ControlNet Router is scheduled using RSNetWorx for ControlNet (see section *ControlNet Configuration – RSNetWorx* for a detailed explanation on how to schedule the ControlNet Router over a ControlNet network).

The data in the *MappedData* tags (see the Logix Operation section) in the Logix input and output assembly of the ControlNet Router is used by the ControlNet Router to provide PLC5-Type Files that can be accessed from the PanelView by using the PLC5 driver. The sizes configured in Slate for the input and output assemblies will be used to provide the correct size PLC5-Type Files which the PanelView can access (see section *FTView Configuration*).

PanelView / FTView Operation	Integer	Float
Read Data	N9	F10
Write Data	N7	F8
Read Diagnostics	N11	-

Table 6.5 – PLC5 Emulation Data File Summary

6.8.1. PANELVIEW READING DATA FROM LOGIX

The PanelView will read N9 when accessing integer values from Logix and F10 when accessing floating/real values. The data being written from the Logix Controller to the *MappedData* tag of the Output Assembly UDT will be copied to both file N9 and F10, but in N9 the data is formatted for integers while in F10 the data is formatted for real values. Below is a diagram of the PanelView reading data from the Logix Controller over Scheduled ControlNet using the PLC5 emulation.

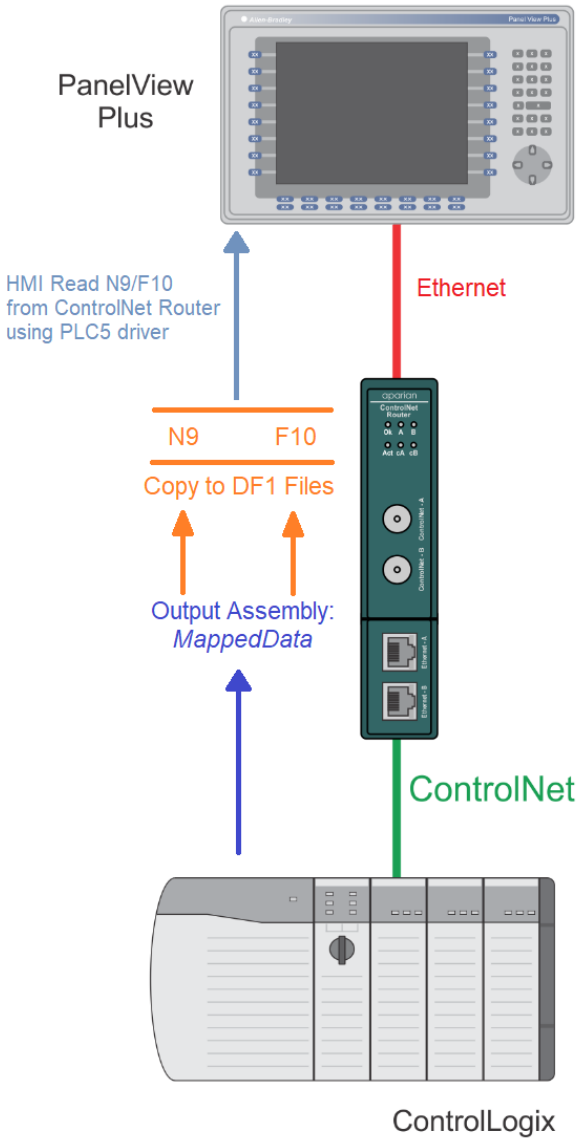


Figure 6.27 – PanelView reading Logix data over Scheduled ControlNet

6.8.2. PANELVIEW WRITING DATA TO LOGIX

The PanelView will write to N7 when sending integer values to Logix, and F8 when sending floating/real values. The data being read by the Logix Controller from the *MappedData* tag of the Input Assembly UDT will be copied from both file N7 and F8 respectively, but in N7 the data is formatted for integers, while in F8 the data is formatted for real values. Below is a diagram of the PanelView writing data to the Logix Controller over Scheduled ControlNet using the PLC5 emulation.

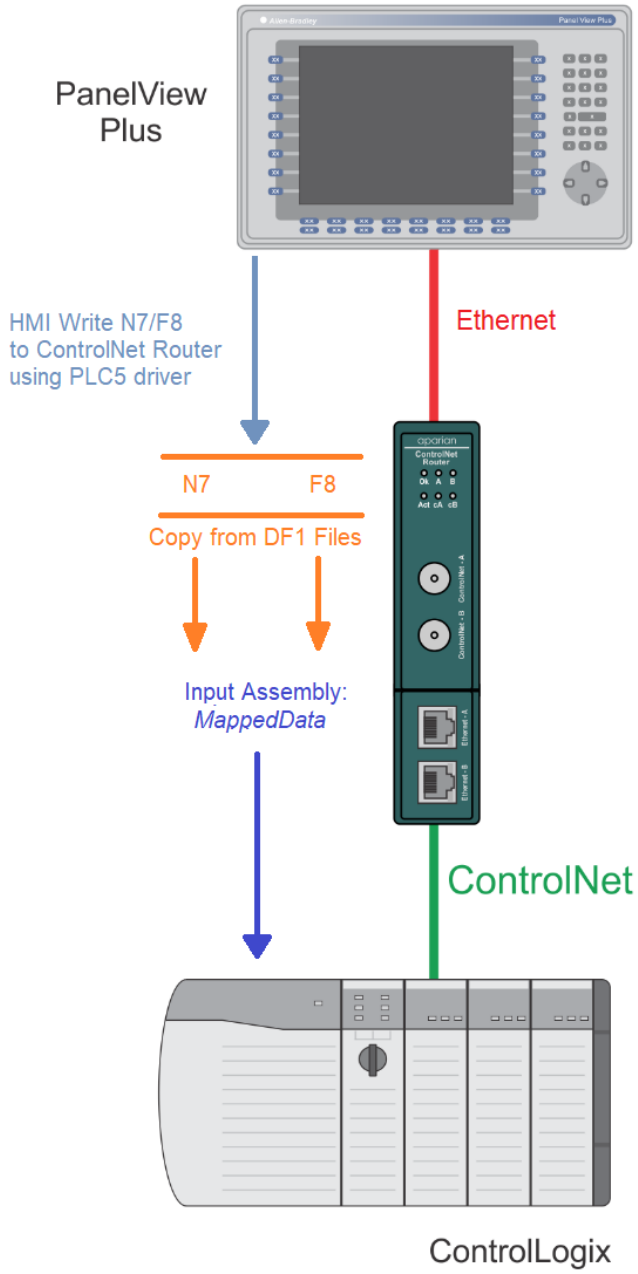


Figure 6.28 – PanelView writing Logix data over Scheduled ControlNet

6.8.3. PanelView Reading Diagnostic Data from ControlNet Router

The PanelView will read N11 when accessing diagnostics information from the ControlNet Router. See the *Diagnostic Data* section for the format of the diagnostic data. The data will be read similar to the above section where the PanelView will read the Logix data using the PLC5 driver emulation.

6.9. INTERNAL DATA SPACE (IDS)

The ControlNet Router Status and Control can be mapped to a specific Internal Data Space register offset. The Internal Status Offset Start is the address in the IDS where the module status will be written to. The Internal Control Offset Start is the address in the IDS where the module control will be read from.

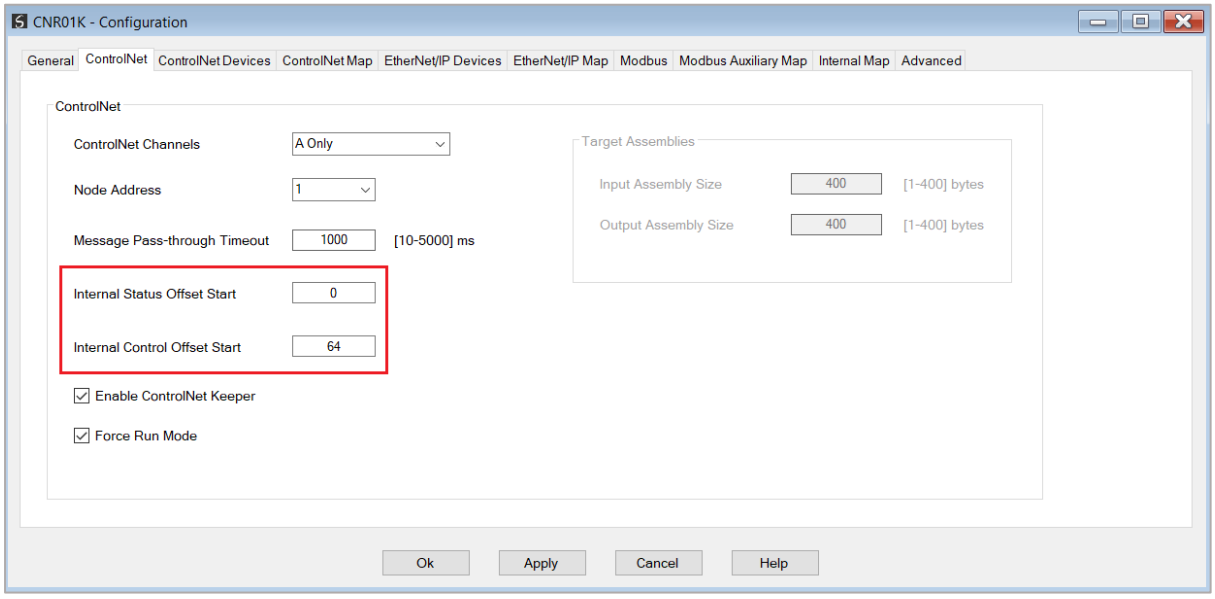


Figure 6.29 – Internal Status and Control Offset

6.9.1. INTERNAL STATUS

The Internal Status structure is shown below.

Field	Byte Offset	Parameter Size	Data Type	Description
Uptime	0	4	DINT	The number of seconds since the module was powered up.
Date Year	4	2	INT	The module UTC Year.
Date Month	6	1	SINT	The module UTC Month.
Date Day	7	1	SINT	The module UTC Day.
Time Hour	8	1	SINT	The module UTC Hour.
Time Minute	9	1	SINT	The module UTC Minute.
Time Second	10	1	SINT	The module UTC Second.
Device Temperature	11	4	REAL	Current temperature of the device in °C.
Hardware MAC address	15	6	SINT[6]	The MAC address of the module.

Ethernet Port1 Status	21	1	SINT	Status of Ethernet Port 1.
Link Up	*21.0*	-	*BIT*	*0 – The Link is down* *1 – The Link is up*
Ethernet Port2 Status	22	1	SINT	Status of Ethernet Port 2.
Link Up	*22.0*	-	*BIT*	*0 – The Link is down* *1 – The Link is up*
DLR Status	23	1	SINT	DLR Status when DLR is enabled
Enabled	*23.0*	-	*BIT*	*0 – Disabled* *1 – Enabled*
Topology	*23.1*	-	*BIT*	*0 – Linear* *1 – Ring Topology*
Status	*23.2*	-	*BIT*	*0 – Normal* *1 – Fault*
NTP Status	24	1	SINT	NTP Status when NTP is enabled
Enabled	*24.0*	-	*BIT*	*0 – Disabled* *1 – Enabled*
Connection to Server	*24.1*	-	*BIT*	*0 – Not Locked to a NTP Server* *1 – Locked to a NTP Server*
Firmware Major Rev	25	1	SINT	Application Firmware Major Rev
Firmware Minor Rev	26	1	SINT	Application Firmware Minor Rev
Firmware Micro Rev	27	1	SINT	Application Firmware Micro Rev
Module Status	28	2	INT	General Module status
Config Valid	*28.0*	-	*BIT*	*The configuration in the module is valid*
Routing Inhibited	*28.1*	-	*BIT*	*Routing has been inhibited*
EIP Originator Comms Ok	*28.2*	-	*BIT*	*EtherNet/IP communication is ok when the module is configured as a EtherNet/IP Originator.*
Modbus Communication Ok	*28.3*	-	*BIT*	*Modbus communication is ok when the module is configured as a Modbus Master or Modbus Slave.*
CNet Originator Comms Ok	*28.4*	-	*BIT*	*ControlNet communication is ok when the module is configured as a ControlNet Originator.*
EtherNet/IP Owned	*28.5*	-	*BIT*	*Indication that the module is owned by a EtherNet/IP originator when the module is operating as a EtherNet/IP target.*
ControlNet Owned	*28.6*	-	*BIT*	*Indication that the module is owned by a ControlNet originator when the module is*

				operating as a ControlNet target.
ControlNet Keeper Active	*28.7*	*-*	*BIT*	*The module ControlNet Keeper functionality has been enabled and is active.*
ControlNet Keeper CRC Mismatch	*29.0*	*-*	*BIT*	*When the module has the Keeper functionality enabled and another Keeper is active with a signature that does not match the local module.*

Table 6.6 – IDS Status Regiser Format

6.9.2. INTERNAL CONTROL

The Internal Control structure is shown below.

Field	Byte Offset	Parameter Size	Data Type	Description
Connection Originator Run Mode	0	1	SINT	The mode of the connection when the module is operating as a connection originator (EtherNet/IP or ControlNet). 0 – Program Mode 1 – Run Mode

Table 6.7 – IDS Control Regiser Format

7. DIAGNOSTICS

7.1. LEDS

The module provides three LEDs for diagnostics purposes as shown in the front view figure below. A description of each LED is given in the table below.

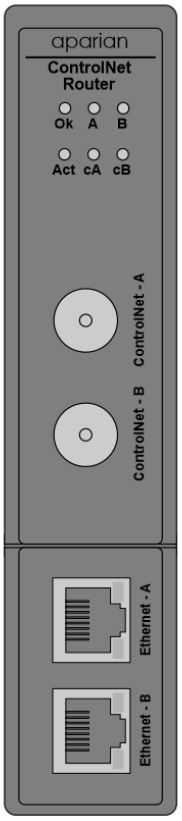


Figure 7.1 – ControlNet Router front view

LED	Description
Ok	The module LED will provide information regarding the system-level operation of the module. If the LED is **red**, then the module is not operating correctly. For example, if the module application firmware has been corrupted or there is a hardware fault the module will have a red Module LED. If the LED is briefly **flashing red**, and then returning to either flashing green or solid green, then there is a duplicate IP address on the Ethernet network similar to that of the local module. If the LED is **green (flashing)**, then the module has booted and is running correctly **without** any application configuration loaded.

	If the LED is **green (solid)**, then the module has booted and is running correctly **with** application configuration loaded.
A / B	The Ethernet LED will light up when an Ethernet link has been detected (by plugging in a connected Ethernet cable). The LED will flash every time traffic was detected. This module has two Ethernet ports A and B. Each LEDs represents each specific port.
Act	The Act LED shows the routing activity for the module. Every time there is a message being exchanged with either the ControlNet or Ethernet network, the LED will flash green. If there was any mapping or exchange issue, the LED will flash red.
cA / cB	The status of the ControlNet link. See the appendix for details on the ControlNet LED status.

Table 7.1 - Module LED operation

7.2. MODULE STATUS MONITORING IN SLATE

The ControlNet Router provides various statistics which can assist with module operation, maintenance, and fault finding. The statistics can be accessed in full by Slate or using the web server in the module.

To view the module's status in the Aparian-Slate environment, the module must be online. If the module is not already Online (following a recent configuration download), then right-click on the module and select the ***Go Online*** option.

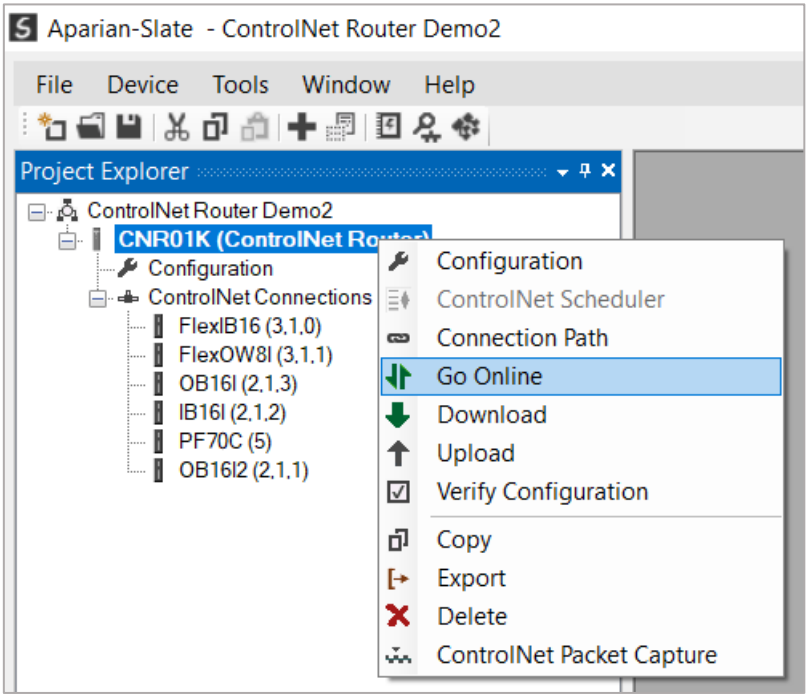


Figure 7.2 - Selecting to Go Online

The Online mode is indicated by the green circle behind the module in the Project Explorer

tree.

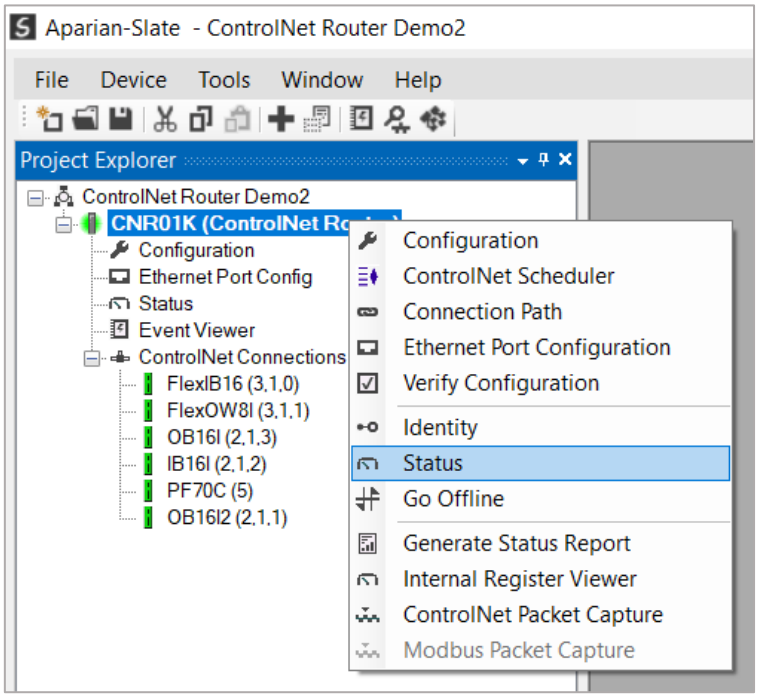


Figure 7.3 - Selecting online Status

The Status monitoring window can be opened by either double-clicking on the ***Status*** item in the Project Explorer tree, or by right-clicking on the module and selecting ***Status***. The status window contains multiple tabs to display the current status of the module.

7.2.1. GENERAL

The General tab displays the general status for the local ControlNet Router module.

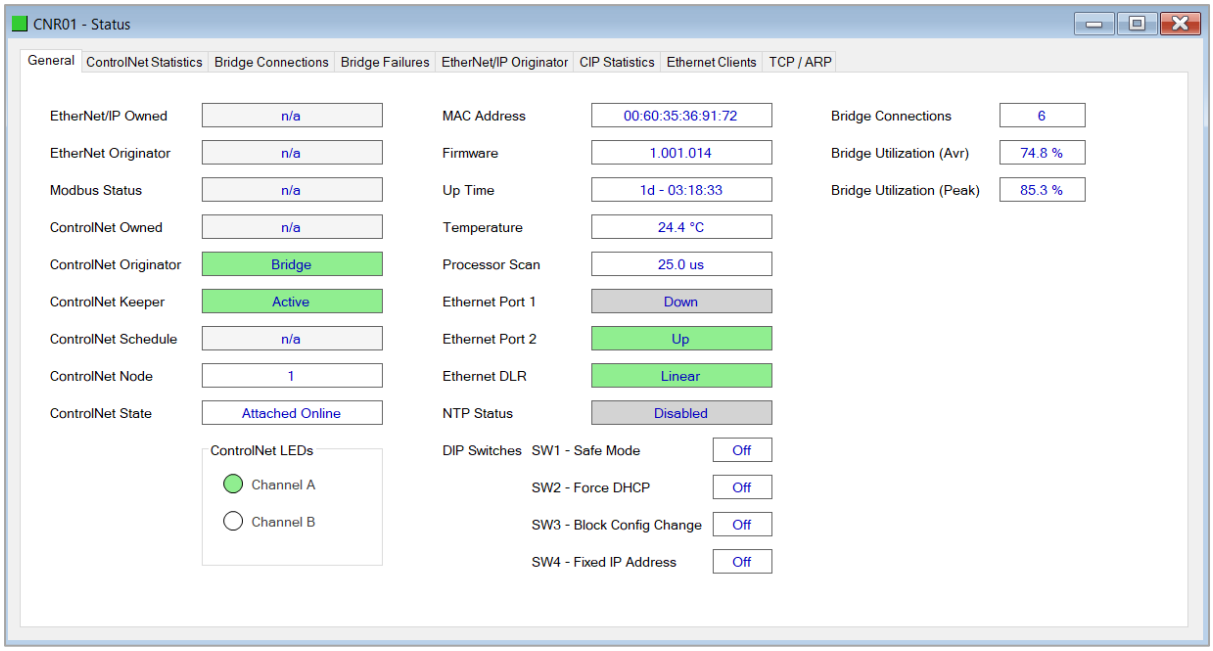


Figure 7.4 - Status monitoring – General

The General tab displays the following general parameters:

Parameter	Description
EtherNet/IP Owned	When the module is configured as an EtherNet/IP Target, this will indicate if the module is owned by an EtherNet/IP connection originator.
EtherNet/IP Originator	When the module is configured as an EtherNet/IP Originator, this will show if all the Class 1 and Explicit Message connections to EtherNet/IP target devices are established and returning valid data.
Modbus Status	When the module is operating as a **Modbus TCP Slave**, this parameter will indicate that the module has received a valid Modbus request within the Modbus inactivity time. When the module is operating as a **Modbus TCP Master**, this parameter will indicate that all the mapping items in the Modbus Auxiliary Map are executing correctly.
ControlNet Owned	When the module is configured as a ControlNet Target, this will indicate if the module is owned by a ControlNet connection originator.
ControlNet Originator	When the module is configured as a ControlNet Originator, this will show if all the Scheduled and Unscheduled Explicit Message connections to ControlNet target devices are established and returning valid data.
ControlNet Keeper	When the module has the *Enabled ControlNet Keeper* set in the ControlNet parameters, this parameter will show if the module has valid ControlNet Keeper configuration data saved and if it is active.
ControlNet Schedule	When the module is a ControlNet Originator and has ControlNet IO in the *ControlNet Connections* tree in Slate, this parameter will indicate if a valid ControlNet scheduled has been downloaded to the module.
ControlNet Node	The current ControlNet node number.
ControlNet State	This is the Net Mode of the ControlNet network. See the appendix for more details.
ControlNet LEDs	The current status of the ControlNet LEDs. See the appendix for more details.
MAC Address	Displays the module's unique Ethernet MAC address.
Firmware	The version of the module's firmware.
Up Time	Indicates the elapsed time since the module was powered-up.
Temperature	The internal temperature of the module.
Processor Scan	The amount of time (microseconds) taken by the module's processor in the last scan.
Ethernet Port 1/2	This is the status of each Ethernet port. **Down**

	The Ethernet connector has **not been** successfully connected to an Ethernet network. **Up** The Ethernet connector has successfully connected to an Ethernet network. **Mirror Enabled** The Ethernet port is mirroring the traffic on the other Ethernet port.
Ethernet DLR (Device Level Ring)	The status of the Ethernet DLR. **Disabled** Device Level Ring functionality has been disabled. **Linear** The DLR functionality has been enabled and the Ethernet network architecture is linear. **Ring – Fault** The DLR functionality has been enabled and the Ethernet network architecture is ring, but there is a fault with the network. **Ring – Ok** The DLR functionality has been enabled and the Ethernet network architecture is ring and is operating as expected.
NTP Status	The status of the local NTP Client. **Disabled** The NTP time synchronization has been disabled. **Locked** NTP time synchronization has been enabled and the module has locked onto the target time server. **Not Locked** NTP time synchronization has been enabled and the module has not locked onto the target time server.
DIP Switch Position	The status of the DIP switches when the module booted up. Note that this status will not change if the DIP switches are altered when the module is running.
Bridge Connections	When the module is operating as a **ControlNet Bridge**, this will be the number of connections the Logix Controller has established with the ControlNet IO via the A-CNR.
Bridge Utilization (Avr)	When the module is operating as a **ControlNet Bridge**, this will be the **average** ControlNet Utilization.

Bridge Utilization (Peak)	When the module is operating as a **ControlNet Bridge**, this will be the **peak** ControlNet Utilization.

Table 7.2 - Parameters displayed in the Status Monitoring – General Tab

7.2.2. CONTROLNET STATISTICS

The ControlNet Statistics tab displays the statistics associated with the ControlNet communication network.

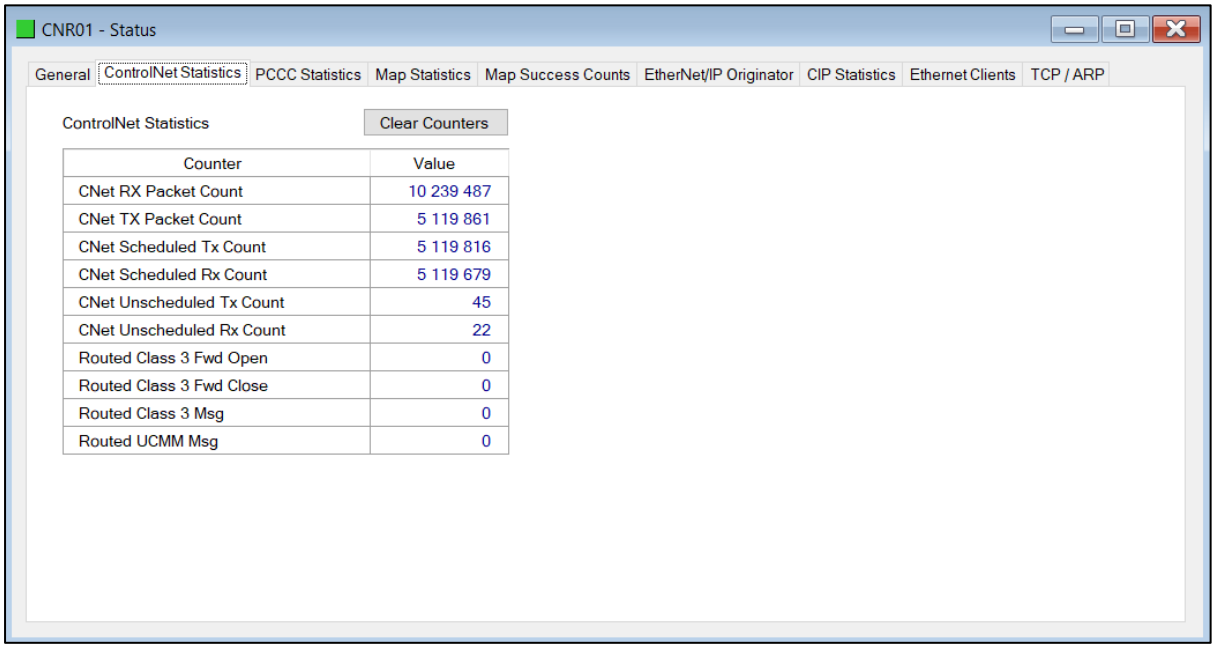


Figure 7.5 - Status monitoring – ControlNet Statistics

Statistic	Description
CNet Rx Packet Count	Number of ControlNet packets received.
CNet Tx Packet Count	Number of ControlNet packets sent.
CNet Scheduled Tx Count	Number of Scheduled ControlNet packets received.
CNet Scheduled Rx Count	Number of Scheduled ControlNet packets sent.
CNet Unscheduled Tx Count	Number of Unscheduled ControlNet packets received.
CNet Unscheduled Rx Count	Number of Unscheduled ControlNet packets sent.
Routed Class 3 Forward Open	Number of Class 3 Forward Opens received for routed messaging.
Routed Class 3 Forward Close	Number of Class 3 Forward Closes received for routed messaging.
Routed Class 3 Message	Number of Routed Class 3 Messages received.
Routed UCMM Message	Number of Routed UCMM Messages received.

Table 7.3 – ControlNet statistics

7.2.3. BRIDGE CONNECTIONS

The Bridge Connections tab displays the connections established to ControlNet IO from the Logix Controller via the A-CNR.

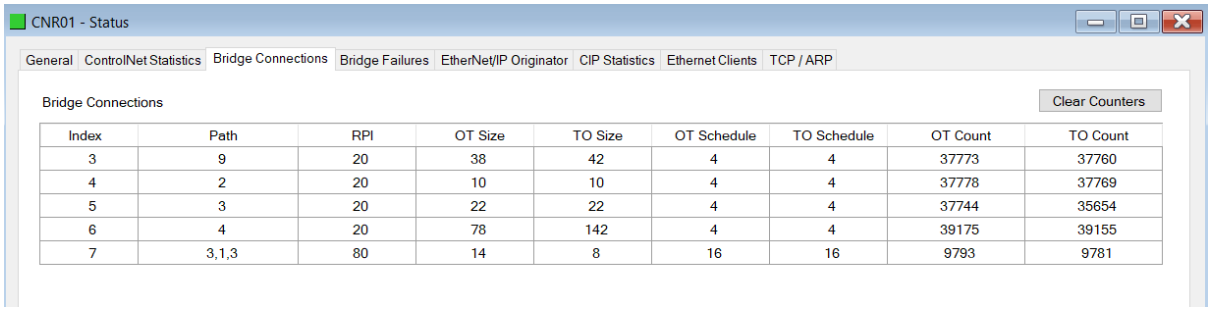


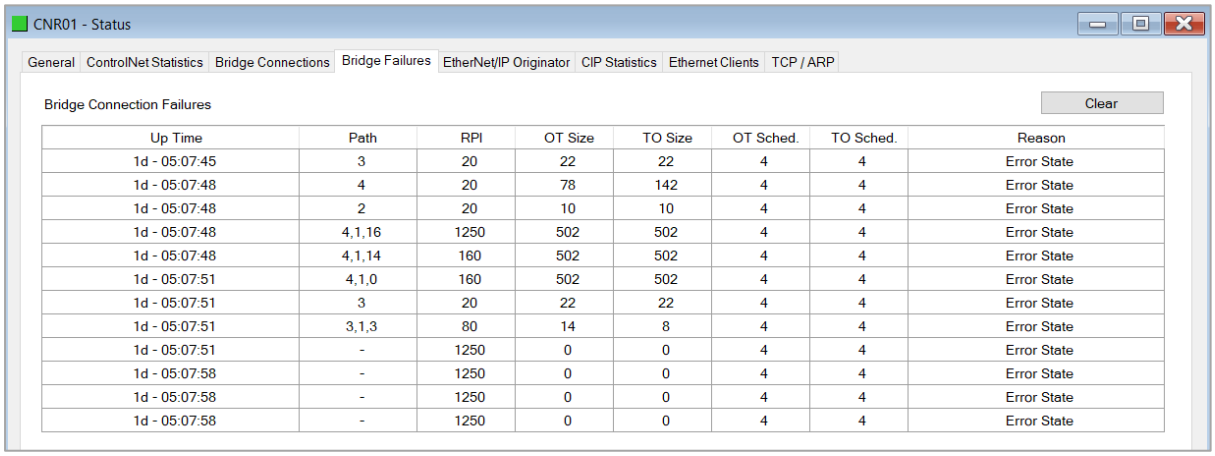
Figure 7.6 - Status monitoring – Bridge Connections

Statistic	Description
Index	The connection index as it was established.
Path	The connection path to the device. If the connection is directly to the device, only the target ControlNet node will be in the path.
RPI	The Requested Packet Interval (RPI) in milliseconds of the connection.
OT Size	The Originator to Target connection size.
TO Size	The Target to Originator connection size.
OT Schedule	The Originator to Target schedule.
TO Schedule	The Target to Originator schedule.
OT Count	The number of scheduled packets sent from the Originator to the Target.
TO Count	The number of scheduled packets sent from the Target to the Originator.

Table 7.4 – Bridge Connections

7.2.4. BRIDGE FAILURES

The Bridge Failure tab displays the last 24 connections failures that have occurred between the Logix Controller and the ControlNet IO.



Up Time	Path	RPI	OT Size	TO Size	OT Sched.	TO Sched.	Reason
1d - 05:07:45	3	20	22	22	4	4	Error State
1d - 05:07:48	4	20	78	142	4	4	Error State
1d - 05:07:48	2	20	10	10	4	4	Error State
1d - 05:07:48	4,1,16	1250	502	502	4	4	Error State
1d - 05:07:48	4,1,14	160	502	502	4	4	Error State
1d - 05:07:51	4,1,0	160	502	502	4	4	Error State
1d - 05:07:51	3	20	22	22	4	4	Error State
1d - 05:07:51	3,1,3	80	14	8	4	4	Error State
1d - 05:07:51	-	1250	0	0	4	4	Error State
1d - 05:07:58	-	1250	0	0	4	4	Error State
1d - 05:07:58	-	1250	0	0	4	4	Error State
1d - 05:07:58	-	1250	0	0	4	4	Error State

Figure 7.7 - Status monitoring – Bridge Failures

Statistic	Description
Uptime	The time since the module was powered on, when the event occurred.
Path	The connection path to the device. If the connection is directly to the device, only the target ControlNet node will be in the path.
RPI	The Requested Packet Interval (RPI) in milliseconds of the connection.
OT Size	The Originator to Target connection size.
TO Size	The Target to Originator connection size.
OT Schedule	The Originator to Target schedule.
TO Schedule	The Target to Originator schedule.
Reason	The reason for the connection failure.

Table 7.5 – Bridge Failures

7.2.5. CONTROLNET EXPLICIT

The ControlNet Explicit Statistics tab displays the statistics associated with ControlNet Device unscheduled explicit mapping.

NOTE: This tab is only applicable when the module has the ControlNet mode set to Originator.

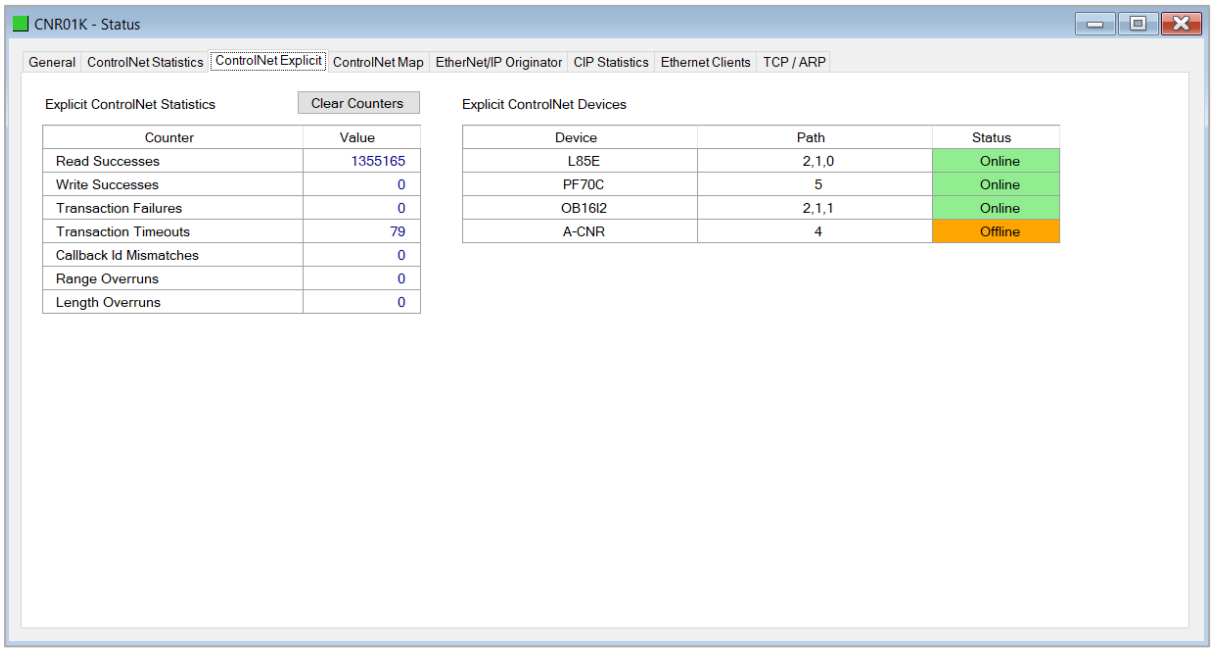


Figure 7.8 - Status monitoring – ControlNet Explicit Statistics

Statistic	Description
Read Successes	The number of successful reads from the target ControlNet device.
Write Successes	The number of successful write to the target ControlNet device.
Transaction Failures	The number of failed reads/writes to the target ControlNet device (e.g. error response).
Transaction Timeouts	The number of times the target ControlNet device failed to respond.
Callback Id Mismatches	The ControlNet Unscheduled UCMM response does not match the request.
Range Overruns	The number of times the returned data length + the input assembly location is greater than the *MappedData* tag area.
Length Overruns	The number of times the returned data is greater than the configured get length.

Table 7.6 – ControlNet Explicit Statistics

7.2.6. CONTROLNET MAP

The ControlNet Map tab displays the success counts for each ControlNet device mapped item.

NOTE: This tab is only relevant when the module has the ControlNet mode set to Originator.

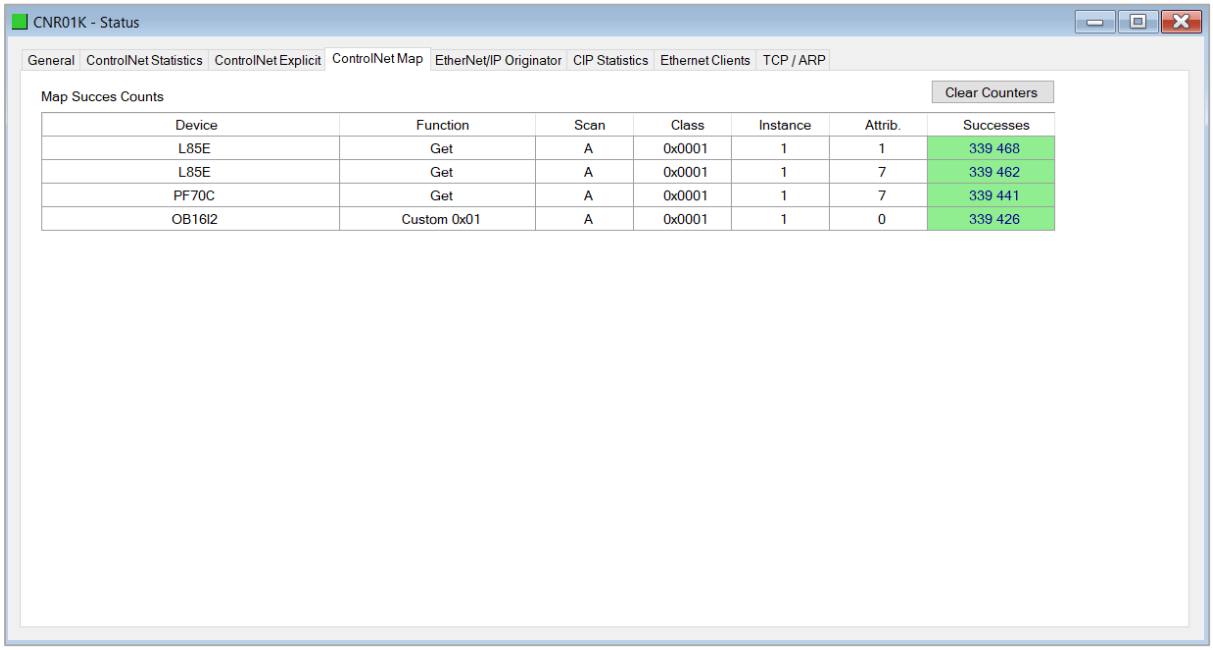


Figure 7.9 - Status monitoring – ControlNet Map

Each time a mapped item is executed successfully its associated count will increase. The count cell will momentarily be highlighted green following a successful transaction.

7.2.7. PCCC STATISTICS

The PCCC tab displays the Ethernet PCCC statistics.

NOTE: This tab is only applicable when the module has the Ethernet mode set to PCCC Client.

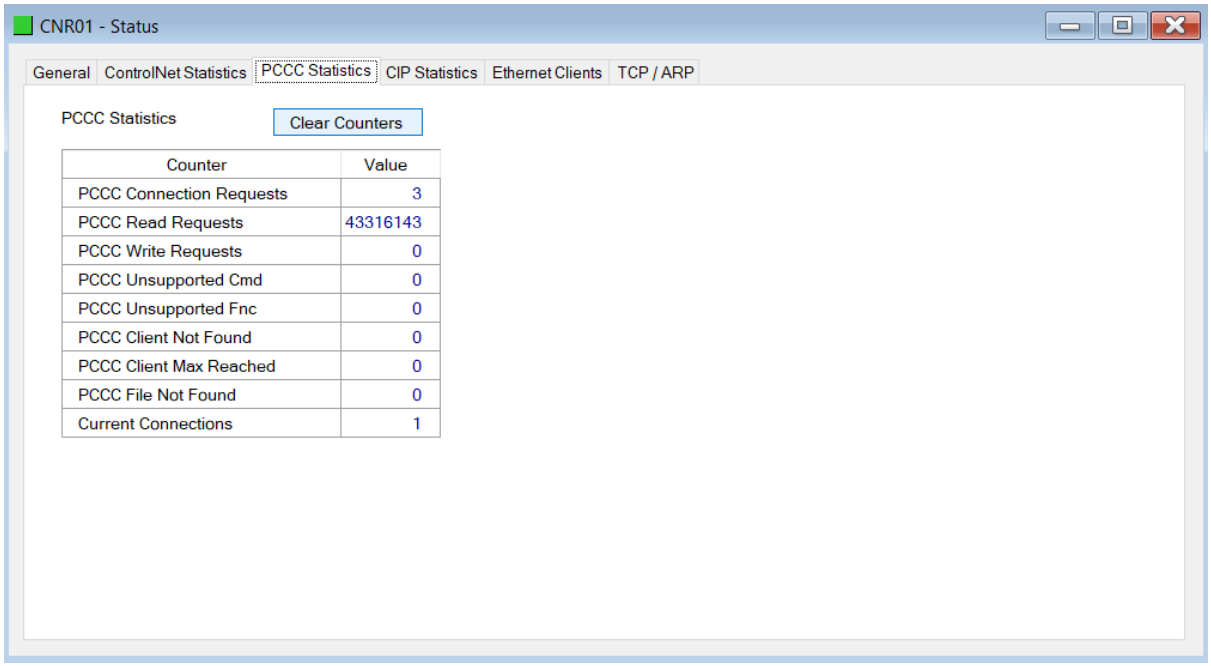


Figure 7.10 - Status monitoring – PCCC Statistics

Statistic	Description
PCCC Connection Requests	The number of PCCC connection establishment requests received.
PCCC Read Requests	The number of Read requests received.
PCCC Write Requests	The number of Write requests received.
PCCC Unsupported Cmd	The number of requests rejected due to an unsupported command.
PCCC Unsupported Fnc	The number of requests rejected due to an unsupported function code.
PCCC Client Not Found	The number of requests rejected due to no matching connection.
PCCC Client Max Reached	The number of connection request rejections due to maximum connection count reached.
PCCC File Not Found	The number of requests rejected due to an unsupported PLC file number.
Current Connections	The current number of active connections.

Table 7.7 – PCCC statistics

7.2.8. ETHERNET/IP EXPLICIT

The EtherNet/IP Explicit Statistics tab displays the statistics associated with EtherNet/IP Device explicit mapping.

NOTE: This tab is only applicable when the module has the Ethernet mode set to EtherNet/IP Originator.

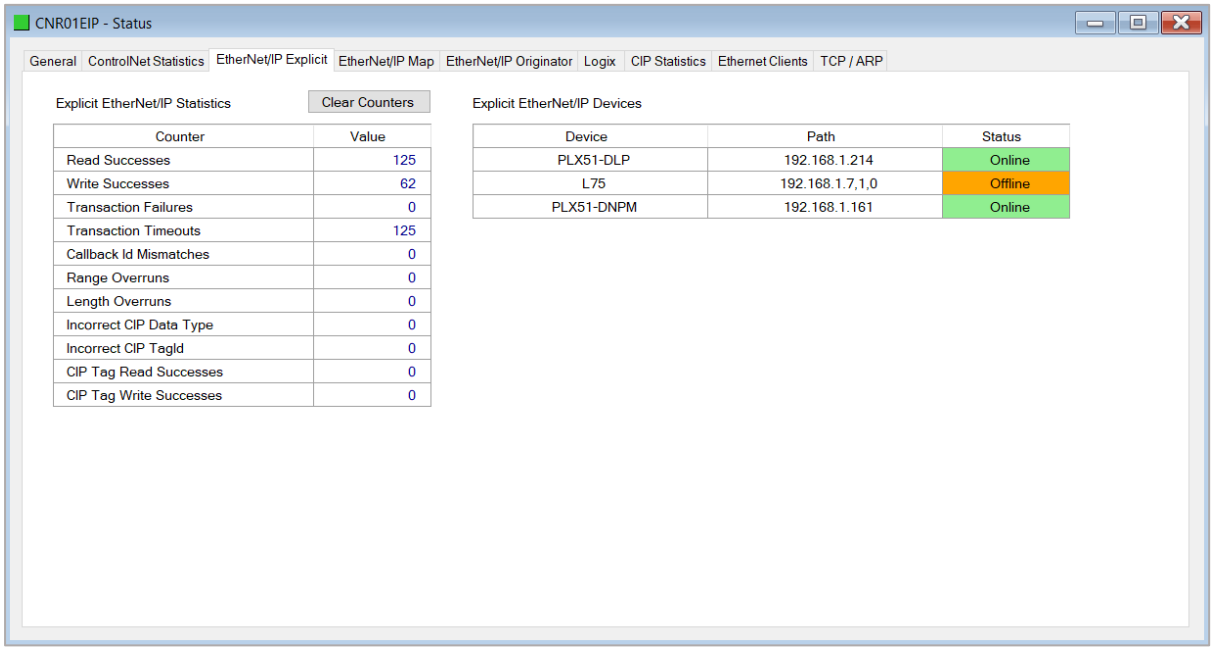


Figure 7.11 - Status monitoring – EtherNet/IP Explicit

Statistic	Description
Read Successes	The number of successful reads from the target EtherNet/IP device.
Write Successes	The number of successful write to the target EtherNet/IP device.
Transaction Failures	The number of failed reads/writes to the target EtherNet/IP device (e.g. error response).
Transaction Timeouts	The number of times the target EtherNet/IP device failed to respond.
Callback Id Mismatches	The EtherNet/IP UCMM or Class 3 response does not match the request.
Range Overruns	The number of times the returned data length + the input assembly location is greater than the *MappedData* tag area.
Length Overruns	The number of times the returned data is greater than the configured get length.
Incorrect CIP Data Type	When the Explicit Message Function is a Tag Read/Write, this statistic will increase when the incorrect CIP data type was returned in the response.
Incorrect CIP Tag Id	When the Explicit Message Function is a Tag Read/Write, this statistic will increase when the incorrect CIP UDT tag ID was returned in the response.

CIP Tag Read Successes	When the Explicit Message Function is a Tag Read, this statistic will increase when there was a successful Logix Tag Read.
CIP Tag Write Successes	When the Explicit Message Function is a Tag Write, this statistic will increase when there was a successful Logix Tag Write.

Table 7.8 – EtherNet/IP Explicit Statistics

7.2.9. EtherNet/IP Map

The EtherNet/IP Map tab displays the success counts for each EtherNet/IP device mapped item.

NOTE: This tab is only applicable when the module has the Ethernet mode set to EtherNet/IP Originator.

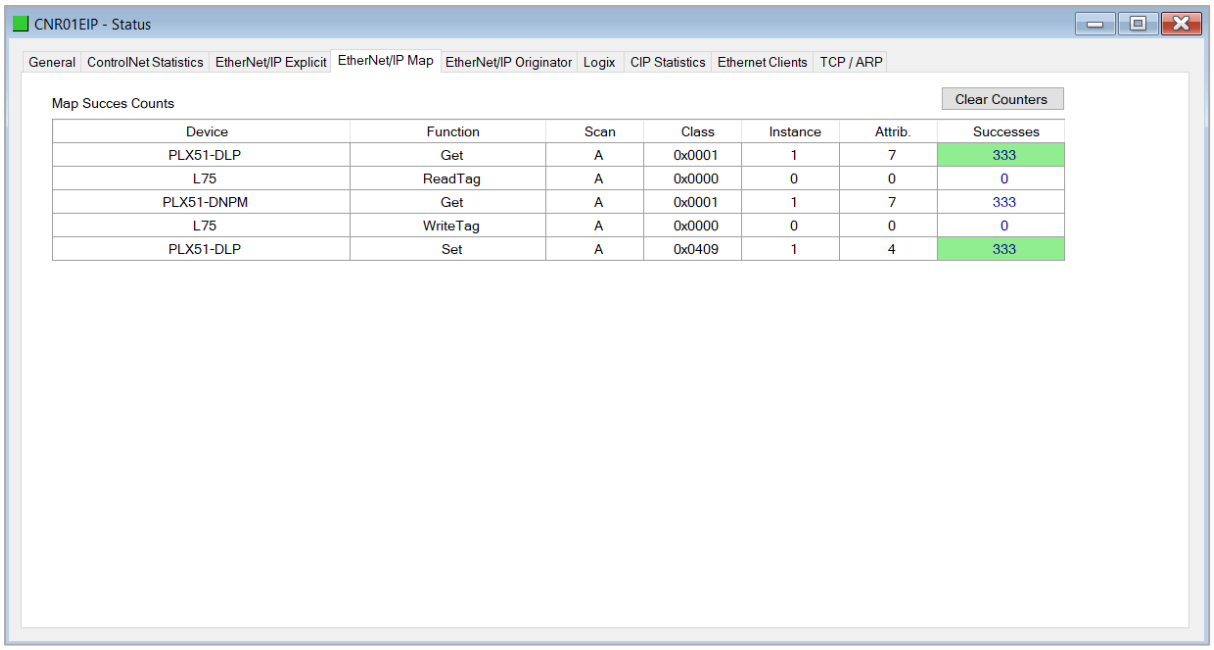


Figure 7.12 - Status monitoring – EtherNet/IP Map

Each time a mapped item is executed successfully its associated count will increase. The count cell will momentarily be highlighted green following a successful transaction.

7.2.10. ETHERNET/IP ORIGINATOR

The EtherNet/IP Originator tab displays the EtherNet/IP Class 1 connection status and statistics for each configured EtherNet/IP device.

NOTE: This tab is only applicable when the module has the Ethernet mode set to EtherNet/IP Originator.

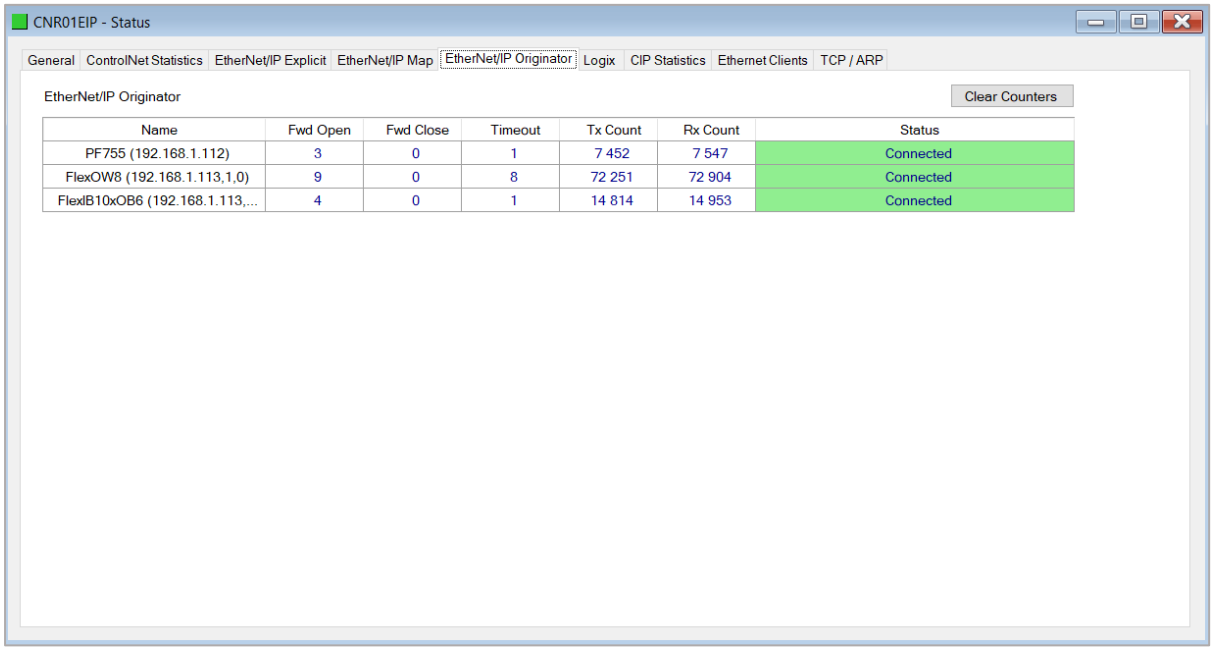


Figure 7.13 - Status monitoring – EtherNet/IP Originator

Statistic	Description
Status	The current connection status of the module. **Connected** The device is connected and exchanging data using Class 1 cyclic communication. **Offline** The device it offline and not connected **Various response faults** If the connection parameters entered are not correct, then generally the target device will reply with the specific reason for the connection reject, for example: Ownership Conflict Connection In Use Or Duplicate Forward Open
Class 1 Originator Statistics	
Forward Open Count	The number of Class 1 Forward Open (connection establishment) messages sent to this device.
Forward Close Count	The number of Class 1 Forward Close (connection termination) messages sent or received from this device.

Connection Timeouts	The number of this connection was closed due to timeouts.
Tx Count	Number of Class 1 messages sent to the specific target device.
Rx Count	Number of Class 1 messages received from the specific target device.

Table 7.9 – EtherNet/IP Class 1 status and statistics

7.2.11. LOGIX

The Logix tab displays the Logix statistics for the explicit EtherNet/IP Tag Read/Write message instructions.

NOTE: This tab is only relevant when the module has the Ethernet mode set to EtherNet/IP Originator and Logix Tag Read/Write functions are being used in the EtherNet/IP Explicit Message Map.

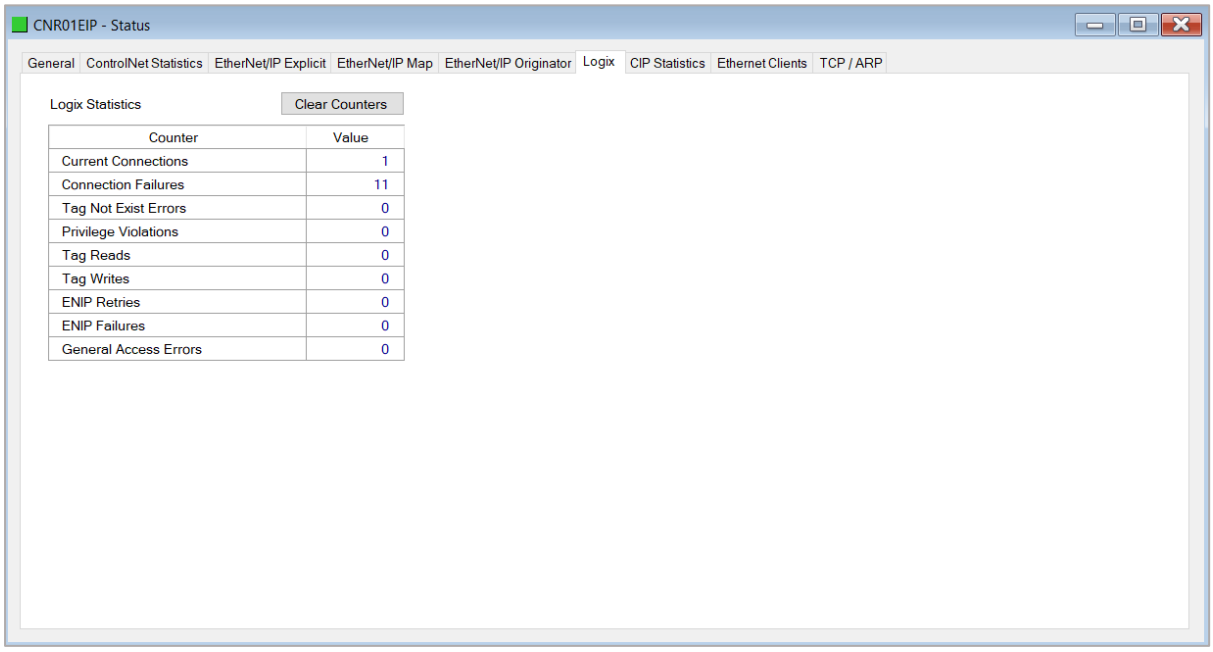


Figure 7.14 - Status monitoring – Logix Statistics

Parameter	Description
Current Connections	The number of current open class 3 connections.
Connection Failures	The number of failed attempts at establishing a class 3 connection with a Logix controller.
Tag Not Exist Errors	The number of tag read and tag write transactions that failed due to the destination tag not existing.
Privilege Violations	The number of tag read and tag write transactions that failed due to a privilege violation error. Note: This may be caused by the ***External Access*** property of the

	Logix tag being set to either ***None*** or ***Read Only***.
Tag Reads	The number of tag read transactions executed by the ControlNet Router module.
Tag Writes	The number of tag write transactions executed by the ControlNet Router module.
ENIP Retries	This count increases when no response was received from the Logix Controller by the time the ENIP timeout is reached.
ENIP Failures	This count increases when the ENIP Retry Limit is reached and no response has been received from the Logix Controller.
Tag Access General Error	This count increases when a tag cannot be accessed for any other reason not reported above.

Table 7.10 – Logix Statistics Tab

7.2.12. MODBUS

The Modbus tab displays the Modbus statistics for the Modbus Read and Write Message Exchanges when the module is a Modbus TCP Slave or Modbus TCP Master.

NOTE: The Modbus statistics tab is only displayed if the module has the primary interface set to Modbus TCP Master or Modbus TCP Slave.

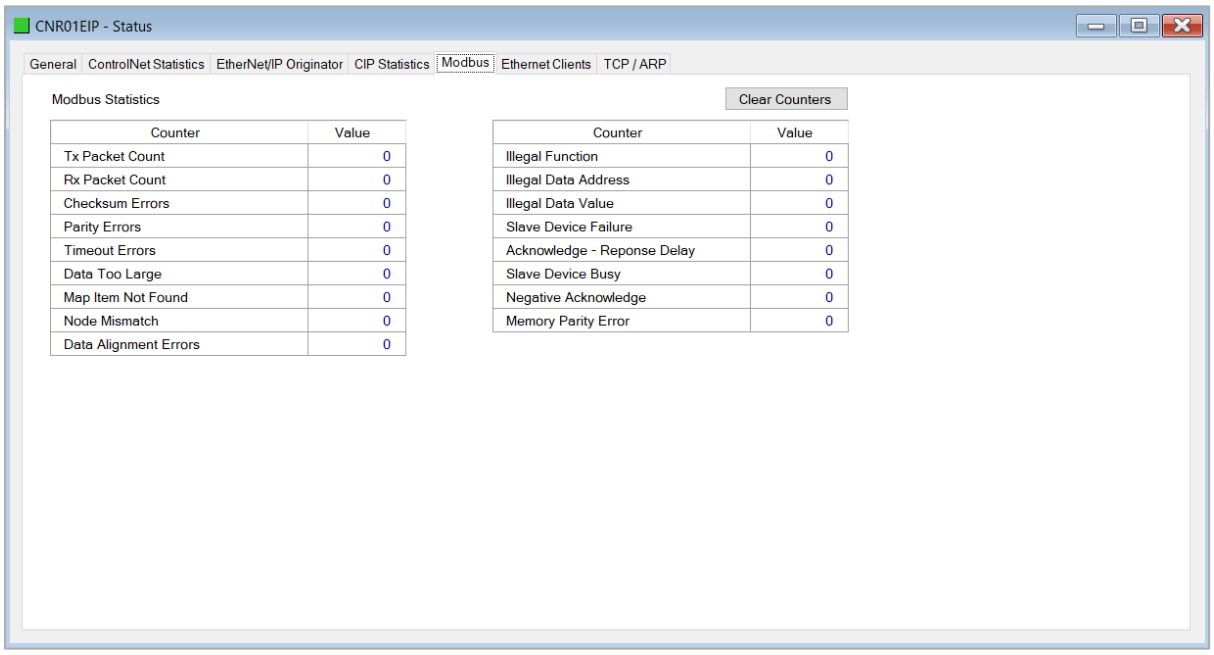


Figure 7.15. - Status monitoring – Modbus Statistics

The Modbus tab displays the following parameters:

Statistic	Description
Tx Packet Count	The number of Modbus packets sent by the module.
Rx Packet Count	The number of Modbus packets received by the module.
Checksum errors	The number of corrupted Modbus packets received by the module.
Parity errors	The number of bytes with parity errors received by the module.
Timeout Errors	The number of message response timeouts the module has encountered.
Data Too Large	The number of Modbus requests or responses where the data was too large to process.
Map Item Not Found	The number of Modbus requests did not match any mapped items.
Node Mismatch	The received Modbus request did not match the module's Modbus node address.
Data Alignment Errors	The Modbus request and associated mapped item is not byte aligned with the destination Logix tag.
Illegal Function	The number of times the Modbus device responded with an Illegal Function exception.
Illegal Data Address	The number of times the Modbus device responded with an Illegal Data Address exception.
Illegal Data Value	The number of times the Modbus device responded with an Illegal Data Value exception.
Slave Device Failure	The number of times the Modbus device responded with a Device Failure exception.
Acknowledge –Response Delay	The number of times the Modbus device responded with an Acknowledge exception.
Slave Device Busy	The number of times the Modbus device responded with a Slave Busy exception.
Negative Acknowledge	The number of times the Modbus device responded with a Negative Acknowledge exception.
Memory Parity Error	The number of times the Modbus device responded with a Memory Parity exception.

Table 7.11 - Modbus Statistics Tab

7.2.13. CIP STATISTICS

The CIP tab displays the Ethernet CIP statistics.

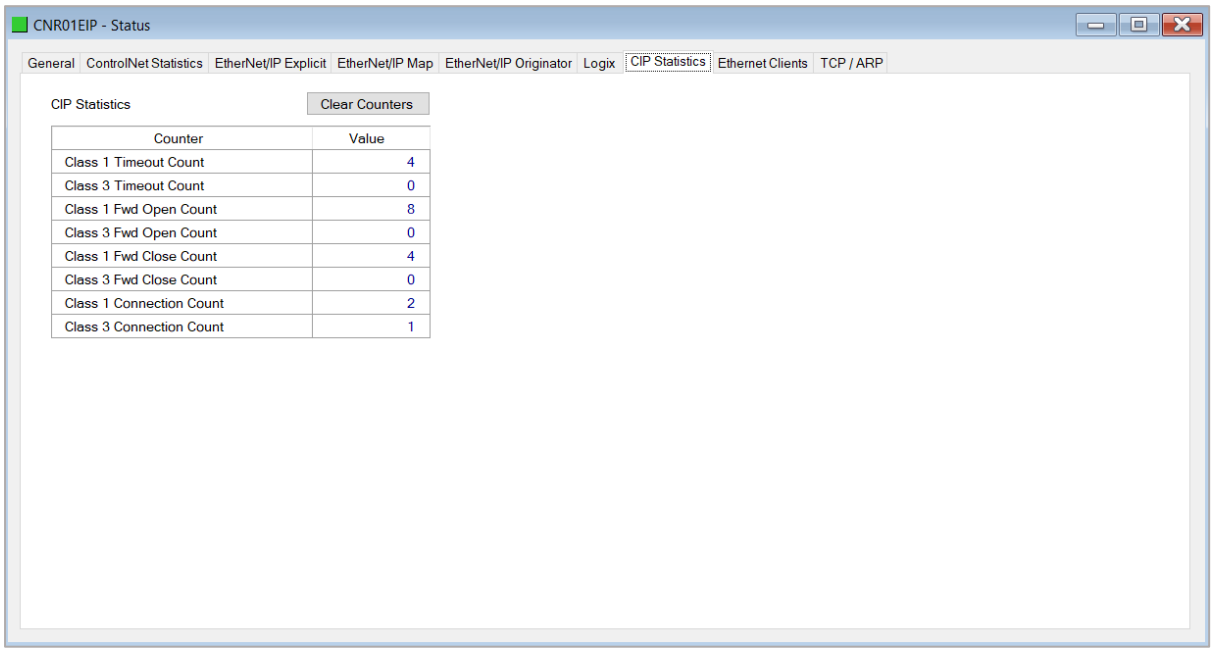


Figure 7.16 - Status monitoring – CIP Statistics

Statistic	Description
Class 1 Timeout Count	The number of Class 1 connections closed due to Timeouts.
Class 3 Timeout Count	The number of Class 3 connections closed due to Timeouts.
Class 1 Forward Open Count	The number of Class 1 Forward Open (connection establishment) messages sent.
Class 3 Forward Open Count	The number of Class 3 Forward Open (connection establishment) messages sent.
Class 1 Forward Close Count	The number of Class 1 Forward Close (connection termination) messages sent.
Class 3 Forward Close Count	The number of Class 3 Forward Close (connection termination) messages sent.
Class 1 Connection Count	The current number of active Class 1 connections.
Class 3 Connection Count	The current number of active Class 3 connections.

Table 7.12 – Mapped Item statistics

7.2.14. ETHERNET CLIENTS

The Ethernet Clients tab displays details of the Ethernet and EtherNet/IP clients connected to the ControlNet Router.

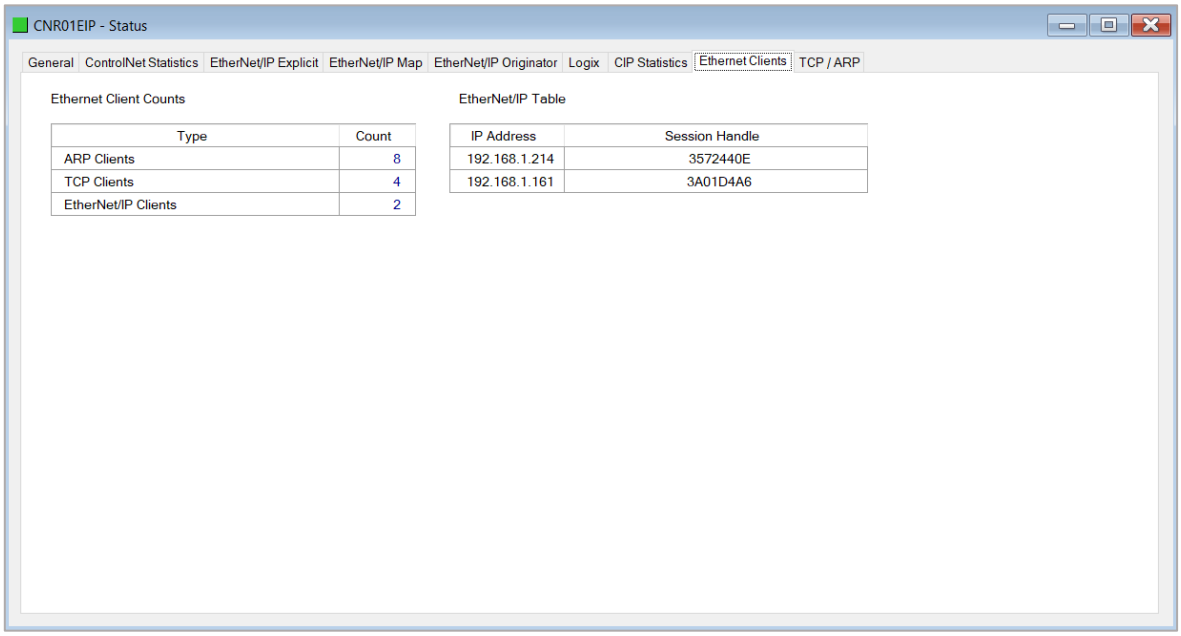


Figure 7.17 –Status monitoring – Ethernet Client Statistics

7.2.15. TCP/ARP

The TCP/ARP tab displays details of the internal Ethernet ARP and TCP lists of the ControlNet Router.

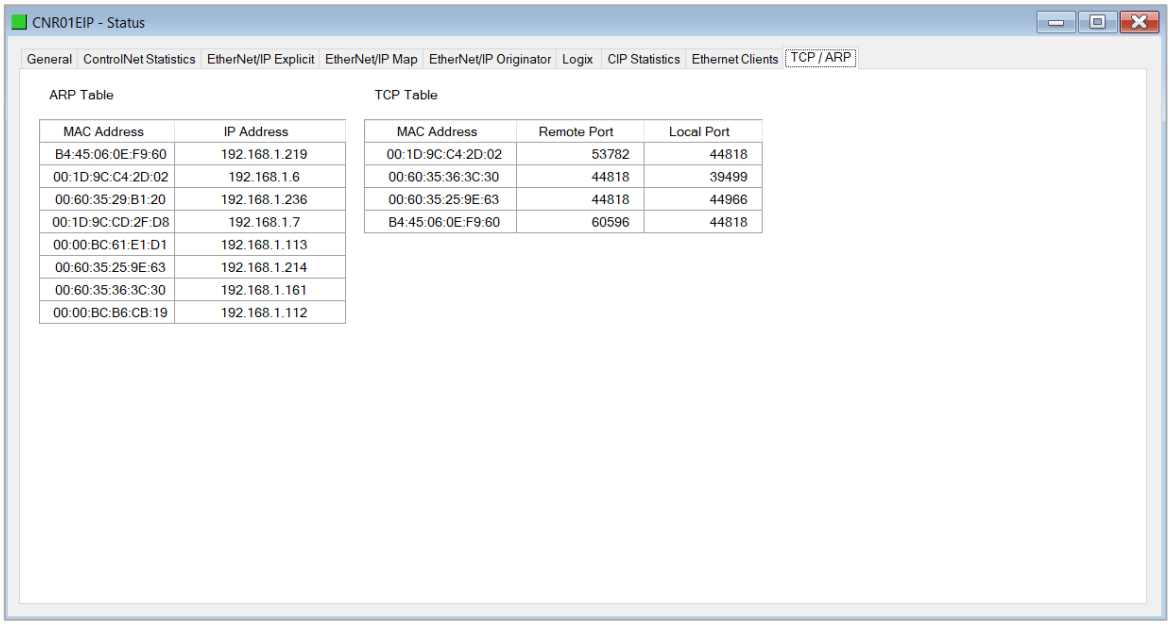


Figure 7.18 – Status monitoring – Ethernet TCP / ARP Statistics

7.3. TARGET DEVICE STATUS MONITORING IN SLATE

The ControlNet Router can also provide individual statistics and status for each for each of the EtherNet/IP Class 1 or ControlNet Scheduled IO devices when the Ethernet Mode is *EtherNet/IP Originator* or the ControlNet Mode is *ControlNet Originator.*

7.3.1. ETHERNET/IP

When online with the module in Slate, right-click on the desired EtherNet/IP device under the *EtherNet/IP Connections* tree in Slate and select *Status*.

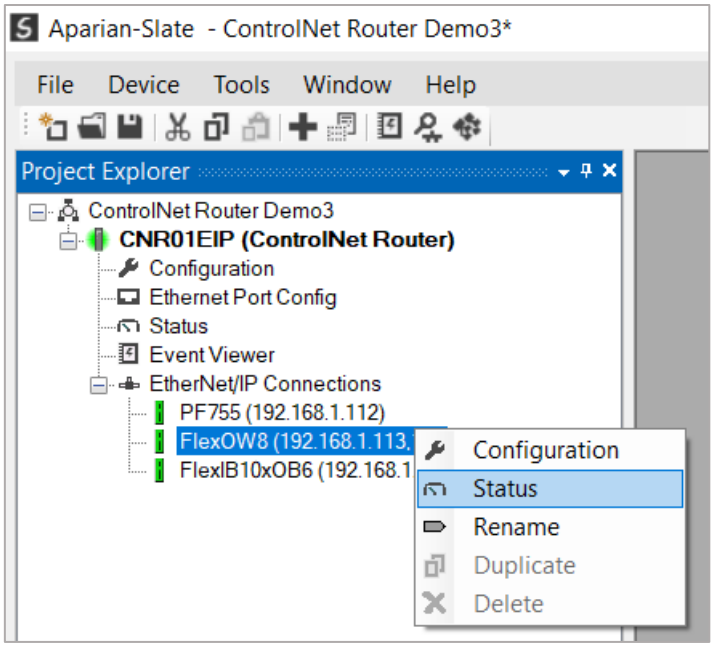


Figure 7.19 – EtherNet/IP Device Status – Status selection

7.3.1.1. GENERAL

The General Status for the EtherNet/IP device shows the connection statistics and parameters associated with the EtherNet/IP Class 1 connection.

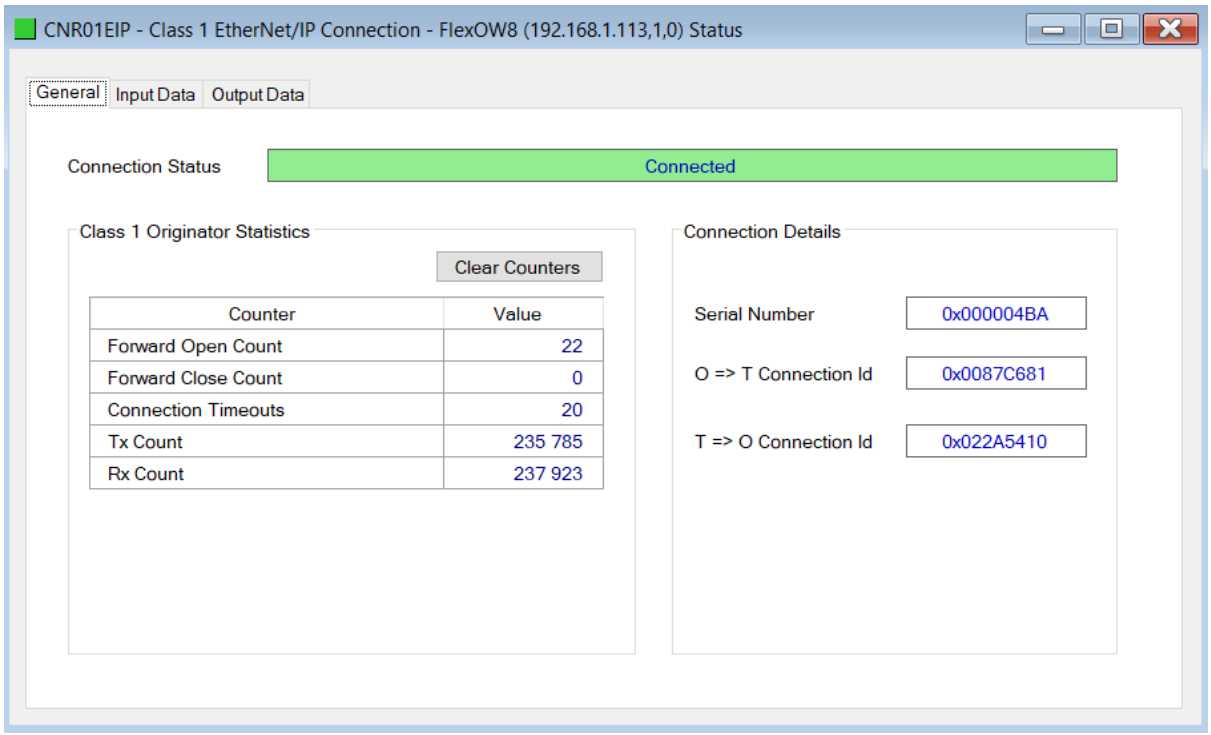


Figure 7.20 – EtherNet/IP Device Status – General Status

Statistic	Description
Connection Status	The current connection status of the module. **Connected** The device is connected and exchanging data using Class 1 cyclic communication. **Offline** The device it offline and not connected **Various response faults** If the connection parameters entered are not correct, then generally the target device will reply with the specific reason for the connection reject, for example: Connection Status: Invalid Originator To Target Size
Class 1 Originator Statistics	
Forward Open Count	The number of Class 1 Forward Open (connection establishment) messages sent to this device.
Forward Close Count	The number of Class 1 Forward Close (connection termination) messages sent or received from this device.
Connection Timeouts	The number of this connection was closed due to timeouts.

Tx Count	Number of Class 1 messages sent to the specific target device.
Rx Count	Number of Class 1 messages received from the specific target device.
Connection Details	
Serial Number	The active connection's serial number.
O -> T Connection Id	The active connection Originator to Target Connection Id.
T -> O Connection Id	The active connection Target to Originator Connection Id.

Table 7.13 – EtherNet/IP Class 1 Device status and statistics

7.3.1.2. INPUT DATA

The Input Data for the EtherNet/IP device shows the Input Assembly associated with the EtherNet/IP Class 1 connection.

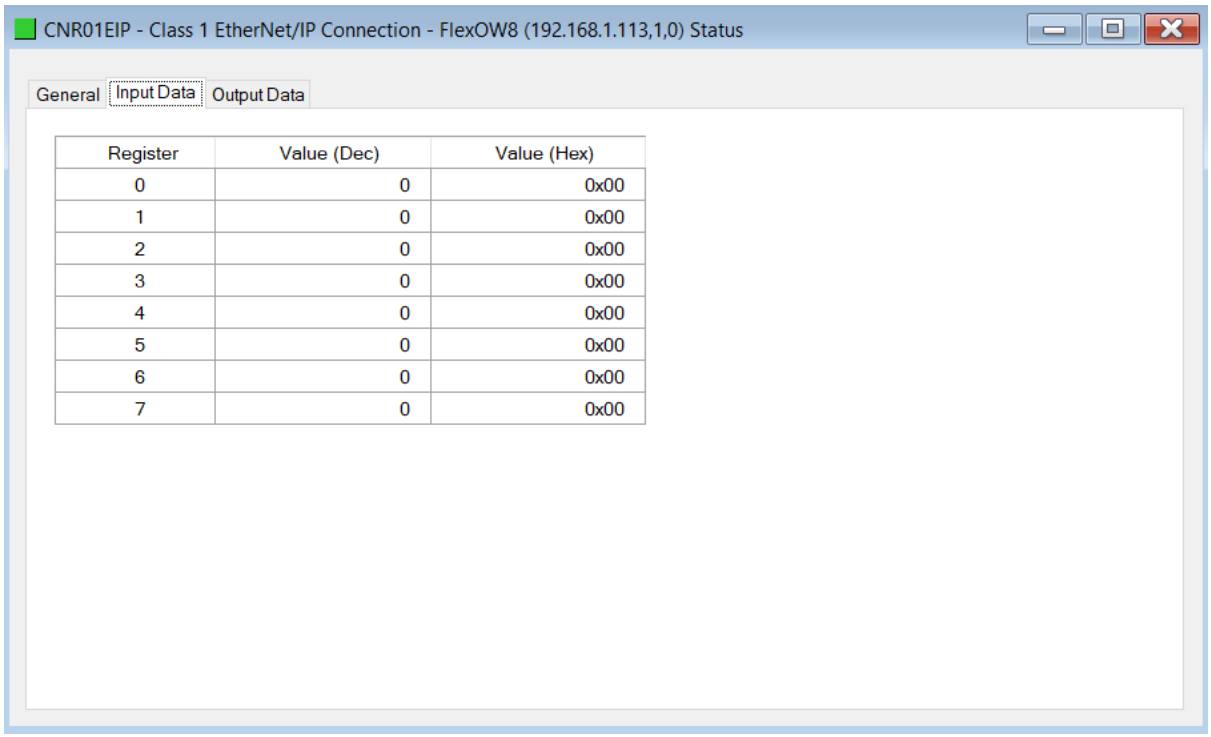


Figure 7.21 – EtherNet/IP Device Status – Input Data

7.3.1.3. OUTPUT DATA

The Output Data for the EtherNet/IP device shows the Output Assembly associated with the EtherNet/IP Class 1 connection.

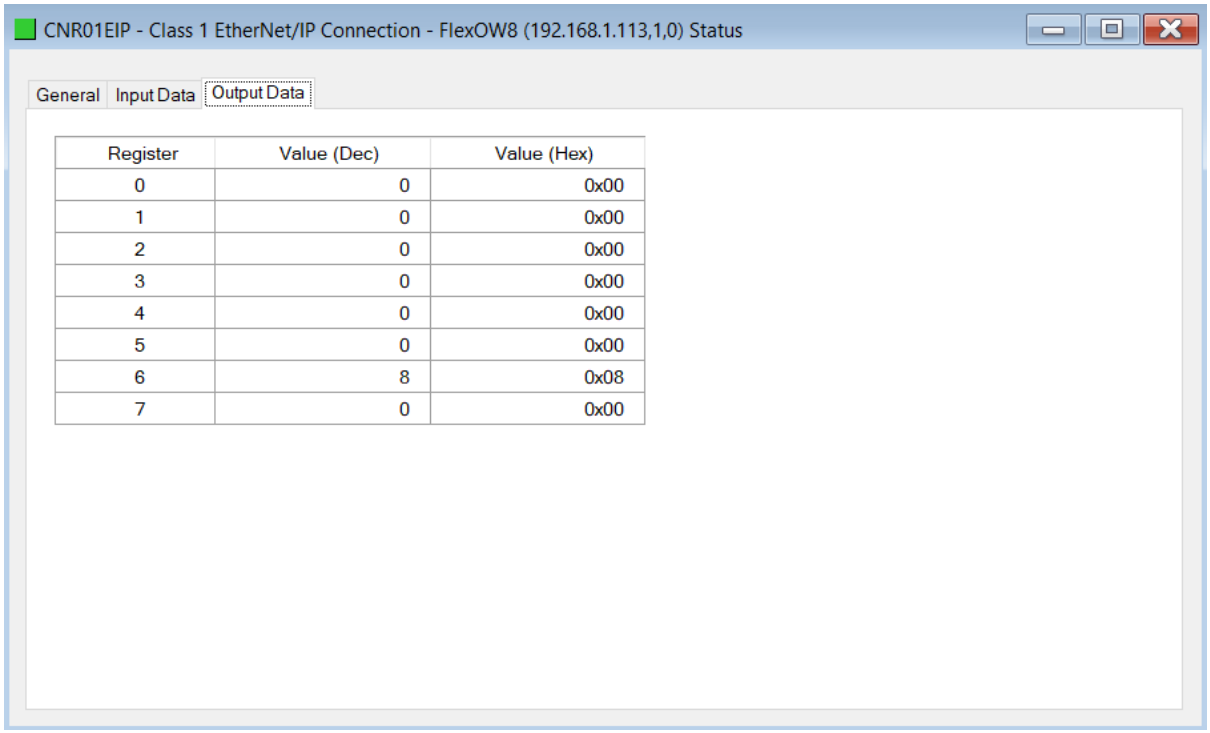


Figure 7.22 – EtherNet/IP Device Status – Output Data

7.3.2. CONTROLNET

When online with the module in Slate, right-click on the desired ControlNet device under the *ControlNet Connections* tree in Slate and select *Status*.

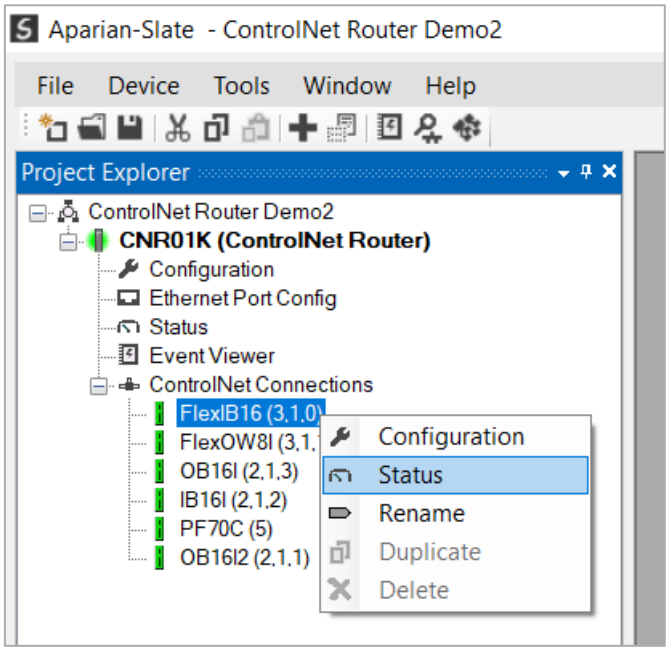


Figure 7.23 – ControlNet Device Status – Status selection

7.3.2.1. GENERAL

The General Status for the ControlNet device shows the connection statistics and parameters associated with the ControlNet Scheduled connection.

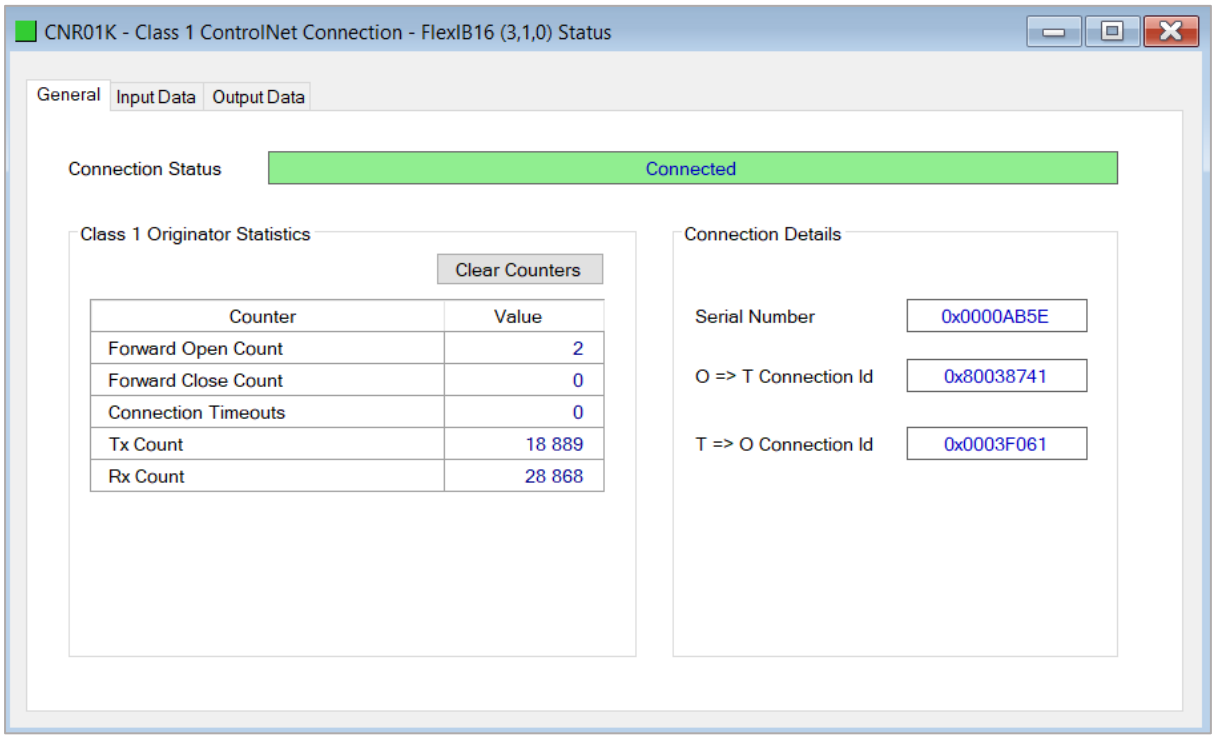


Figure 7.24 – ControlNet Device Status – General Status

Statistic	Description
Connection Status	The current connection status of the module. **Connected** The device is connected and exchanging data using Scheduled communication. **Offline** The device it offline and not connected **Various response faults** If the connection parameters entered are not correct, then generally the target device will reply with the specific reason for the connection reject, for example: Connection Status: Invalid Originator To Target Size
Class 1 Originator Statistics	
Forward Open Count	The number of Scheduled Forward Open (connection establishment) messages sent to this device.
Forward Close Count	The number of Scheduled Forward Close (connection termination) messages sent or received from this device.
Connection Timeouts	The number of this connection was closed due to timeouts.
Tx Count	Number of Scheduled messages sent to the specific target device.

Rx Count	Number of Scheduled messages received from the specific target device.
Connection Details	
Serial Number	The active connection's serial number.
O -> T Connection Id	The active connection Originator to Target Connection Id.
T -> O Connection Id	The active connection Target to Originator Connection Id.

Table 7.14 – ControlNet Scheduled Device status and statistics

7.3.2.2. INPUT DATA

The Input Data for the ControlNet device shows the Input Assembly associated with the ControlNet Scheduled connection.

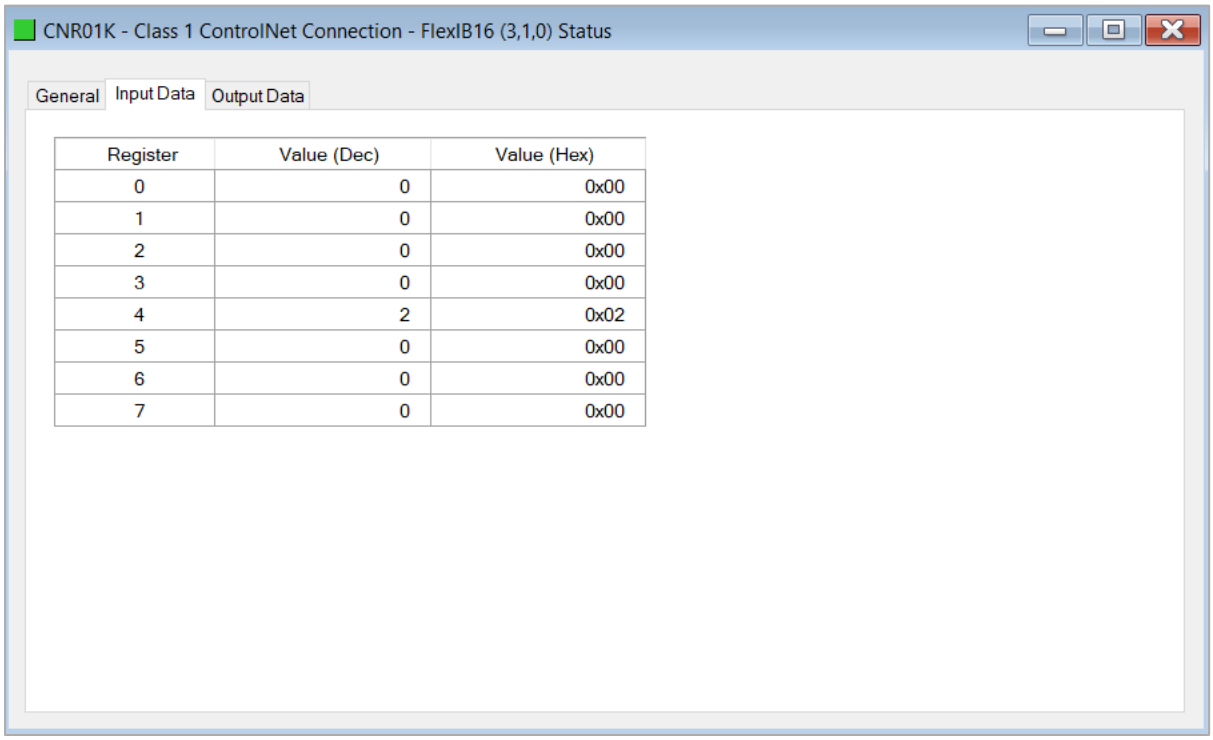


Figure 7.25 – ControlNet Device Status – Input Data

7.3.2.3. OUTPUT DATA

The Output Data for the ControlNet device shows the Output Assembly associated with the ControlNet Scheduled connection.

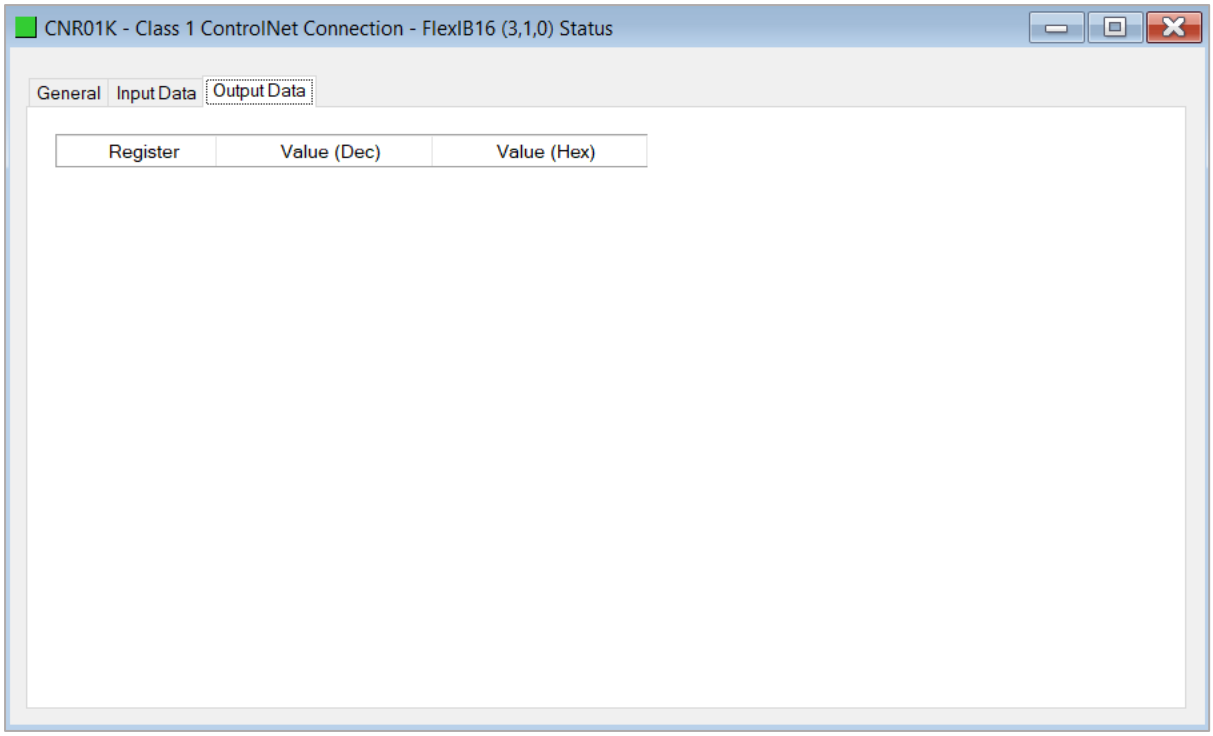


Figure 7.26 – ControlNet Device Status – Output Data

7.4. MODULE EVENT LOG

The ControlNet Router module logs various diagnostic records to an internal event log. These logs are stored in non-volatile memory and can be displayed using Slate or via the web interface. To view them in Slate, select the ***Event Viewer*** option in the Project Explorer tree.

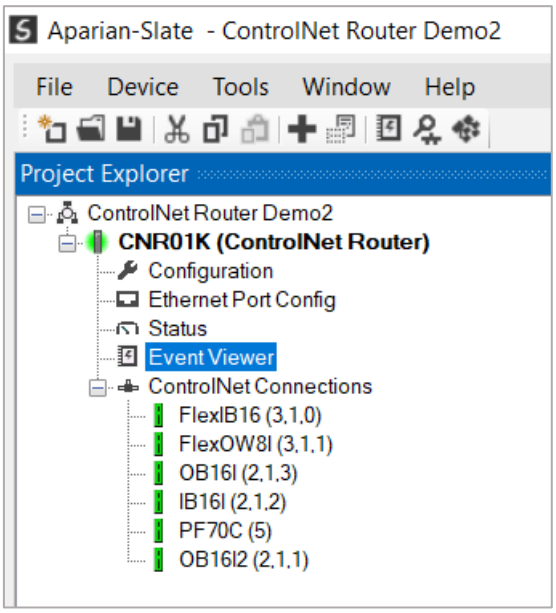
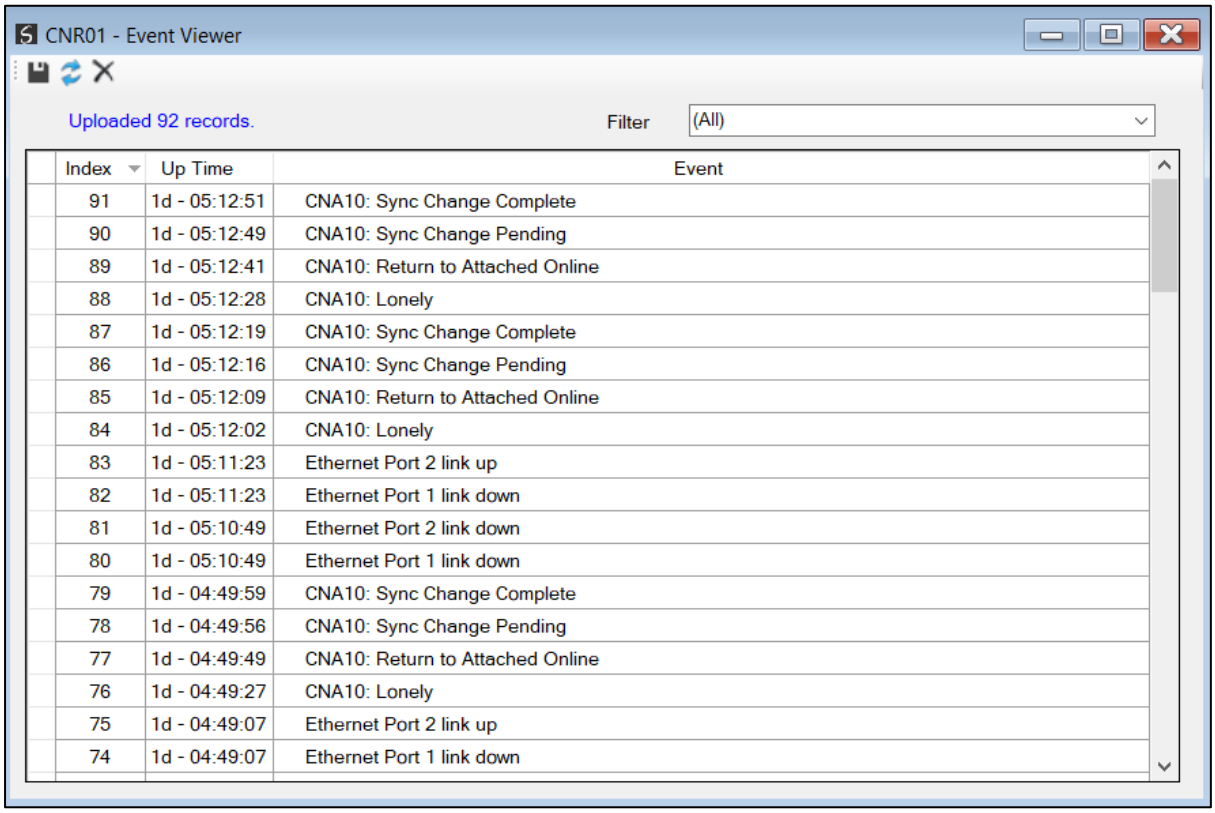


Figure 7.27 - Selecting the module Event Log

The Event Log window will open and automatically read all the events from the module. The log entries are sorted so as to have the latest record at the top. Custom sorting is achieved by double-clicking on the column headings.



Index	Up Time	Event
91	1d - 05:12:51	CNA10: Sync Change Complete
90	1d - 05:12:49	CNA10: Sync Change Pending
89	1d - 05:12:41	CNA10: Return to Attached Online
88	1d - 05:12:28	CNA10: Lonely
87	1d - 05:12:19	CNA10: Sync Change Complete
86	1d - 05:12:16	CNA10: Sync Change Pending
85	1d - 05:12:09	CNA10: Return to Attached Online
84	1d - 05:12:02	CNA10: Lonely
83	1d - 05:11:23	Ethernet Port 2 link up
82	1d - 05:11:23	Ethernet Port 1 link down
81	1d - 05:10:49	Ethernet Port 2 link down
80	1d - 05:10:49	Ethernet Port 1 link down
79	1d - 04:49:59	CNA10: Sync Change Complete
78	1d - 04:49:56	CNA10: Sync Change Pending
77	1d - 04:49:49	CNA10: Return to Attached Online
76	1d - 04:49:27	CNA10: Lonely
75	1d - 04:49:07	Ethernet Port 2 link up
74	1d - 04:49:07	Ethernet Port 1 link down

Figure 7.28 - Module Event Log

The log can also be stored to a file for future analysis, by selecting the Save button in the tool menu. To view previously saved files, use the Event Log Viewer option under the Tools menu.

7.5. WEB SERVER

The ControlNet Router provides a web server allowing a user without Slate to view various diagnostics of the module. This includes Ethernet parameters, system event log, advanced diagnostics, and application diagnostics (ControlNet diagnostics).

NOTE: The web server is view **only** and therefor no parameters or configuration can be altered from the web interface.

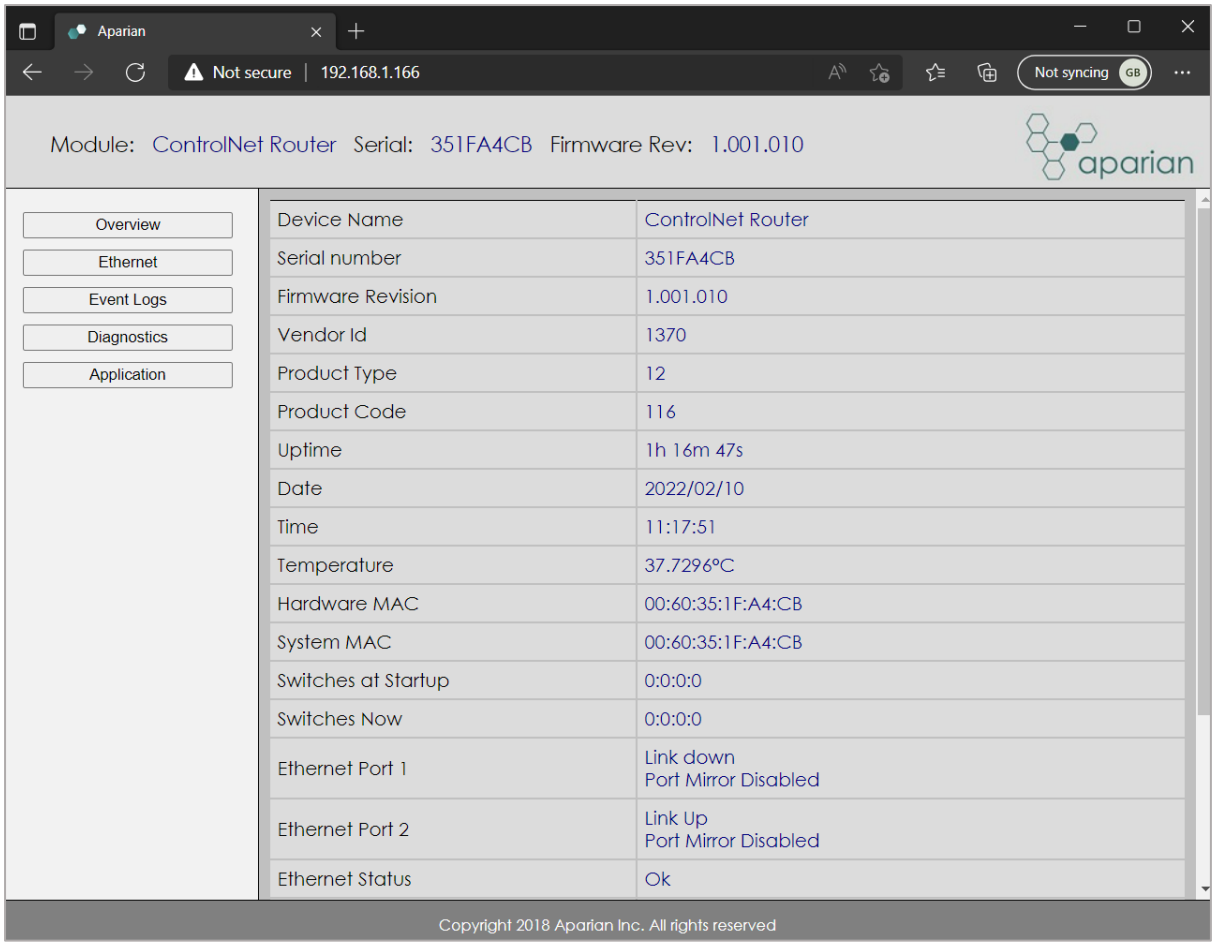


Figure 7.29 – Web interface

7.6. CONTROLNET PACKET CAPTURE

The module provides the capability to capture the ControlNet traffic for analysis. The will allow the user and a remote support team to resolve any possible issues on site. To invoke the capture of the module, right-click on the ControlNet Router and select the ***ControlNet Packet Capture*** option.

There are two capture modes as described below:

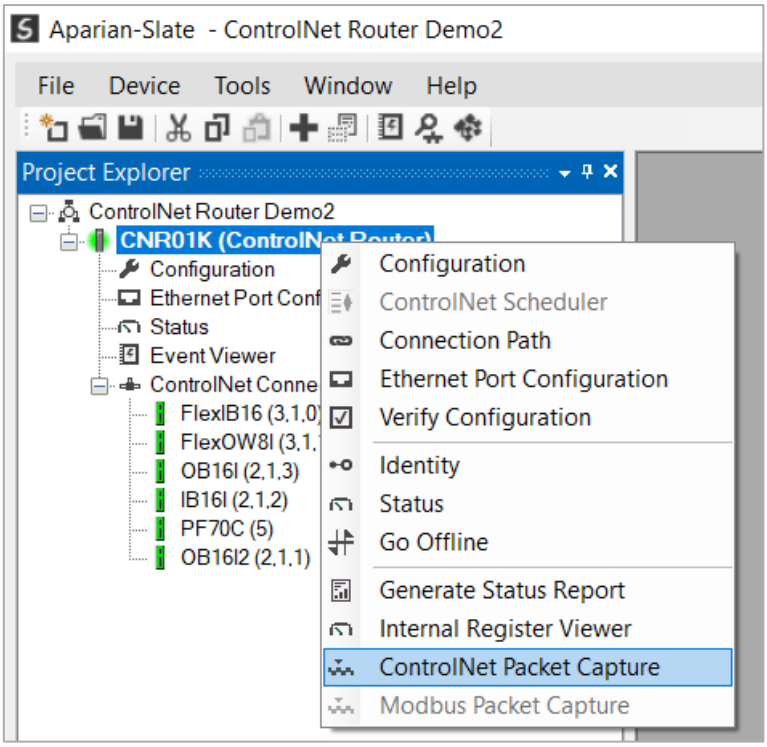


Figure 7.30 - Selecting ControlNet Packet Capture

7.6.1. Operational Capture

The operational capture is the ControlNet packet capture when the module is in normal routing operation mode. This is set when the *Main Mode* parameter in the General Configuration is set to *Operational.*

This capture mode will allow the module to capture all data to and from the ControlNet Router module. The ControlNet Packet Capture window will open and automatically start capturing all ControlNet packets.

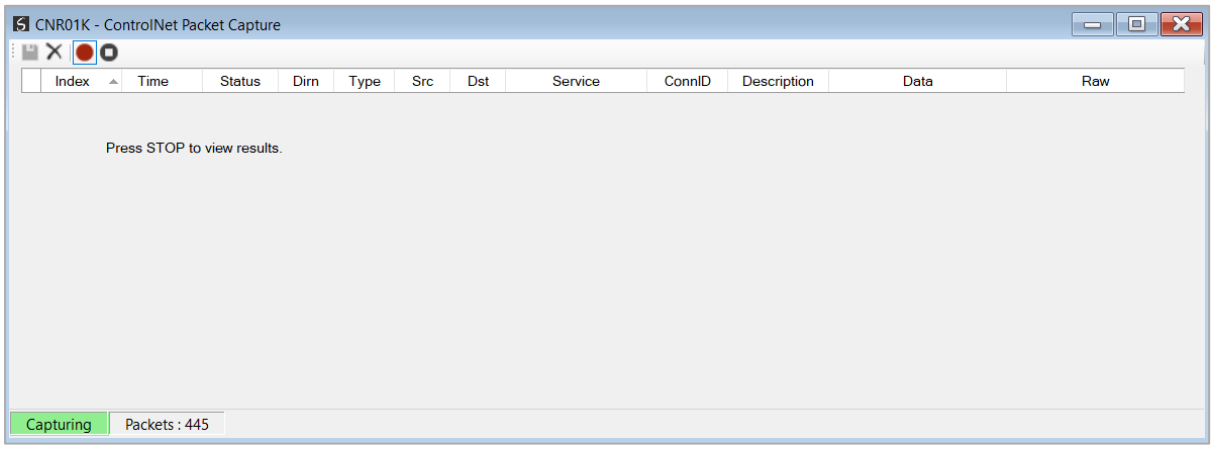
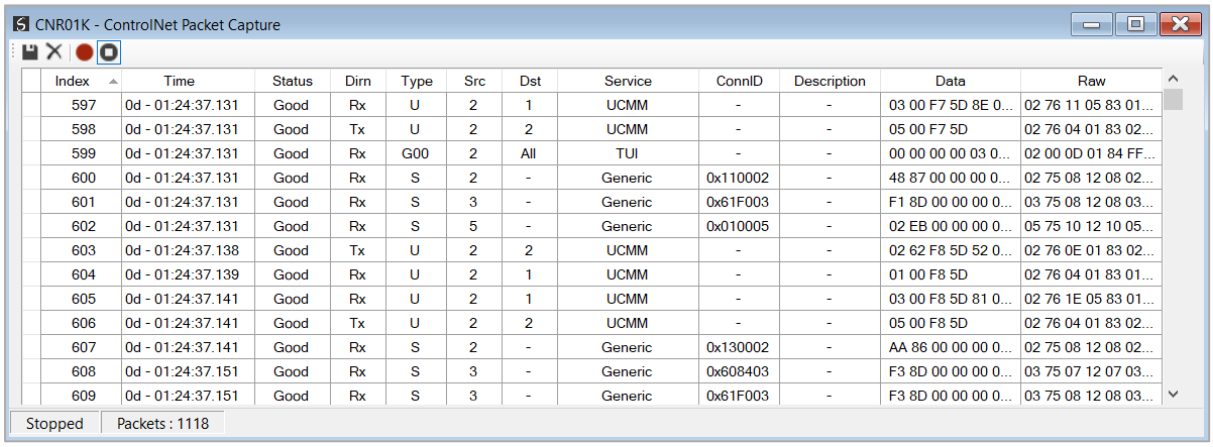


Figure 7.31 – ControlNet packet capture

To display the captured ControlNet packets, the capture process must first be stopped, by

pressing the Stop button.



Index	Time	Status	Dirn	Type	Src	Dst	Service	ConnID	Description	Data	Raw
597	0d - 01:24:37.131	Good	Rx	U	2	1	UCMM	-	-	03 00 F7 5D 8E 0...	02 76 11 05 83 01...
598	0d - 01:24:37.131	Good	Tx	U	2	2	UCMM	-	-	05 00 F7 5D	02 76 04 01 83 02...
599	0d - 01:24:37.131	Good	Rx	G00	2	All	TUI	-	-	00 00 00 00 03 0...	02 00 0D 01 84 FF...
600	0d - 01:24:37.131	Good	Rx	S	2	-	Generic	0x110002	-	48 87 00 00 00 0...	02 75 08 12 08 02...
601	0d - 01:24:37.131	Good	Rx	S	3	-	Generic	0x61F003	-	F1 8D 00 00 00 0...	03 75 08 12 08 03...
602	0d - 01:24:37.131	Good	Rx	S	5	-	Generic	0x010005	-	02 EB 00 00 00 0...	05 75 10 12 10 05...
603	0d - 01:24:37.138	Good	Tx	U	2	2	UCMM	-	-	02 62 F8 5D 52 0...	02 76 0E 01 83 02...
604	0d - 01:24:37.139	Good	Rx	U	2	1	UCMM	-	-	01 00 F8 5D	02 76 04 01 83 01...
605	0d - 01:24:37.141	Good	Rx	U	2	1	UCMM	-	-	03 00 F8 5D 81 0...	02 76 1E 05 83 01...
606	0d - 01:24:37.141	Good	Tx	U	2	2	UCMM	-	-	05 00 F8 5D	02 76 04 01 83 02...
607	0d - 01:24:37.141	Good	Rx	S	2	-	Generic	0x130002	-	AA 86 00 00 00 0...	02 75 08 12 08 02...
608	0d - 01:24:37.151	Good	Rx	S	3	-	Generic	0x608403	-	F3 8D 00 00 00 0...	03 75 07 12 07 03...
609	0d - 01:24:37.151	Good	Rx	S	3	-	Generic	0x61F003	-	F3 8D 00 00 00 0...	03 75 08 12 08 03...

Figure 7.32 – ControlNet Packet Capture complete

The captured ControlNet packets are tabulated as follows:

Statistic	Description
Index	The packet index, incremented for each packet sent or received.
Time	The elapsed time since the module powered up.
Status	The status of the packet. Received packets are checked for valid ControlNet constructs and valid checksums.
Dirn	The direction of the packet, either transmitted (Tx) or received (Rx).
Type	The ControlNet MAC Frame Market Type G – General S – Scheduled U – Unscheduled M - Moderator
Src	The source ControlNet address.
Dst	The destination ControlNet address.
Service	The ControlNet LPacket Service Description
ConnID	The ControlNet Connection ID, where applicable.
Description	Description of the packet that was received.
Data	The application layer packet data.
Raw Data	The raw packet data.

Table 7.15 – ControlNet Packet Capture fields

The packet capture can be saved to a file for further analysis, by selecting the ***Save*** button on the toolbar. Previously saved ControlNet Packet Capture files can be viewed by selecting the

ControlNet Packet Capture Viewer option in the tools menu.

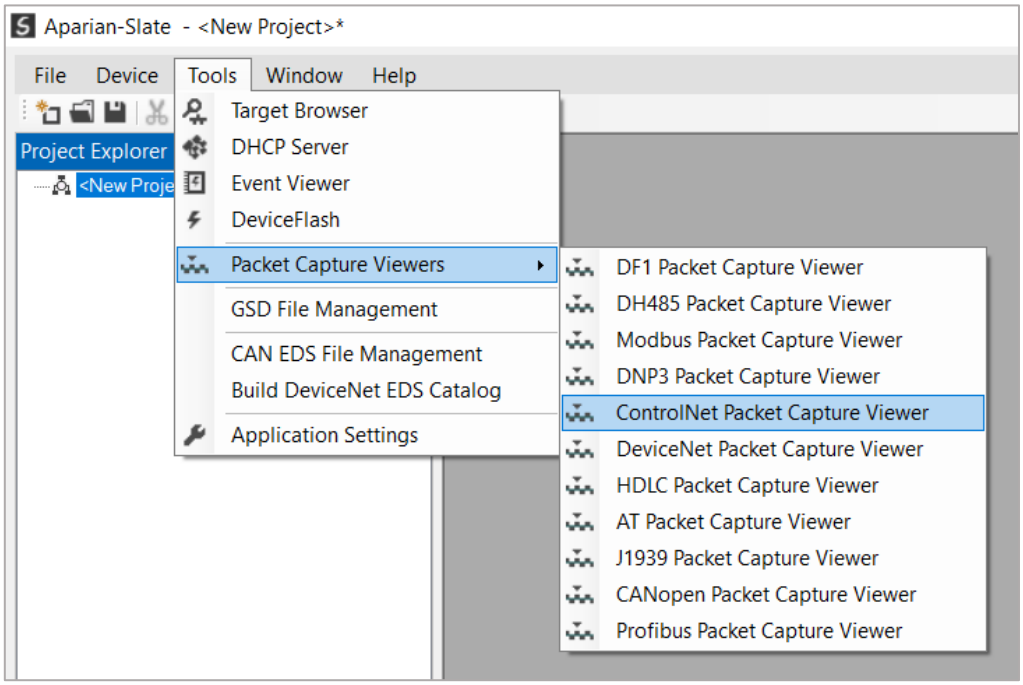


Figure 7.33 - Selecting the ControlNet Packet Capture Viewer

7.6.2. Promiscuous Capture

The Promiscuous Capture is the ControlNet packet capture when the module is not actively part of the ControlNet network, but rather listening to **all** packet being sent on the network. This is set when the ***Main Mode*** parameter in the General Configuration is set to ***Promiscuous Capture***.

When starting the capture, the user will be prompted with the ControlNet Pre-Capture Filter to help filter out the non-required packets (e.g., filter out the Moderator Frame).

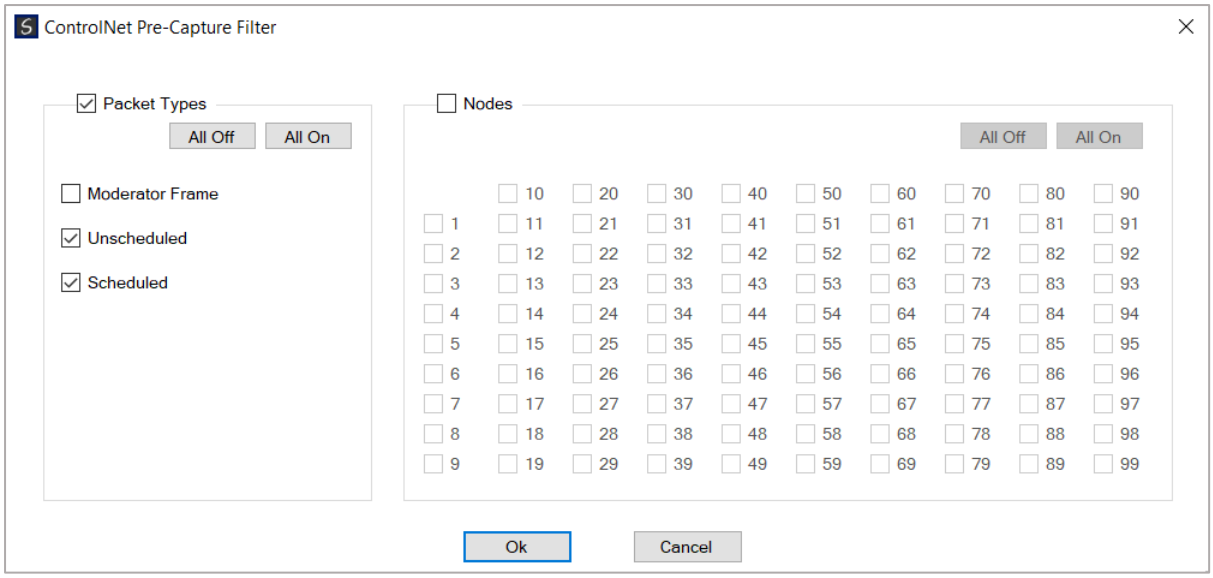


Figure 7.34 – ControlNet Pre-Capture Filter

The ControlNet Packet Capture window will open and automatically start capturing all ControlNet packets.

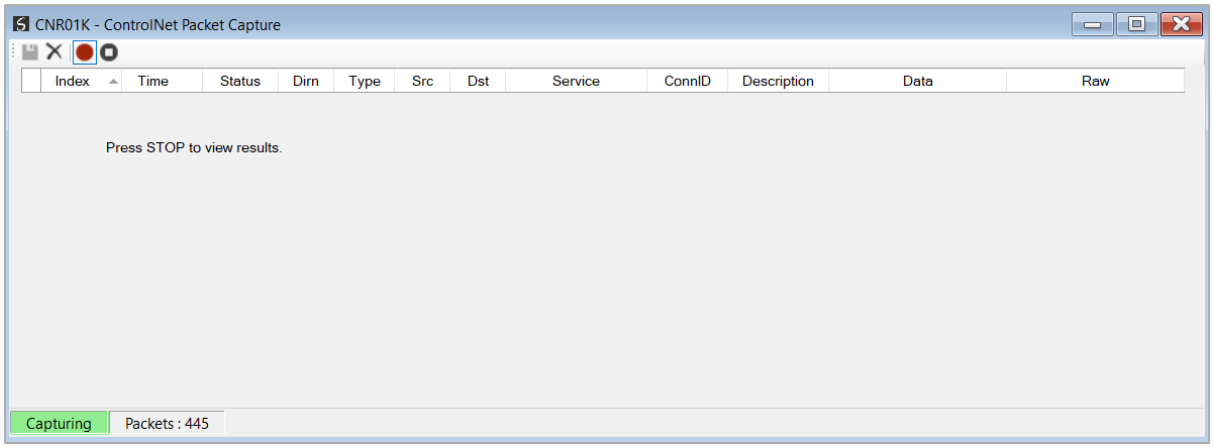


Figure 7.35 – ControlNet packet capture – Promiscuous Mode

To display the captured ControlNet packets, the capture process must first be stopped, by pressing the Stop button.

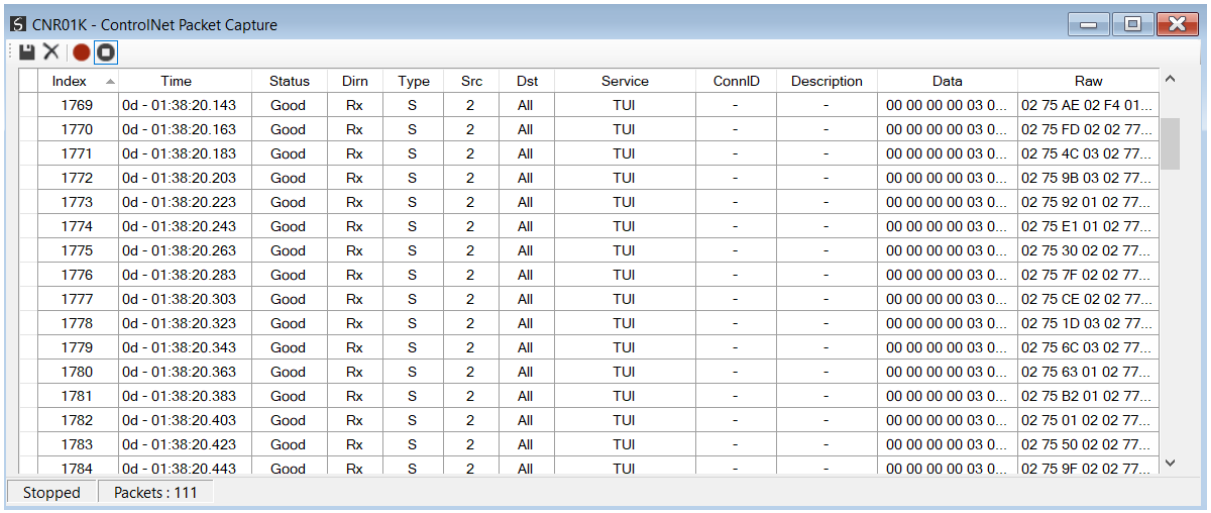


Figure 7.36 – ControlNet Packet Capture complete – Promiscuous Mode

The captured ControlNet packets are tabulated as follows:

Statistic	**Description**
Index	The packet index, incremented for each packet sent or received.
Time	The elapsed time since the module powered up.
Status	The status of the packet. Received packets are checked for valid ControlNet constructs and valid checksums.
Dirn	The direction of the packet, either transmitted (Tx) or received (Rx).
Type	The ControlNet MAC Frame Market Type G – General S – Scheduled U – Unscheduled M - Moderator
Src	The source ControlNet address.
Dst	The destination ControlNet address.
Service	The ControlNet LPacket Service Description
ConnID	The ControlNet Connection ID, where applicable.
Description	Description of the packet that was received.
Data	The application layer packet data.
Raw Data	The raw packet data.

Table 7.16 – ControlNet Packet Capture fields

7.7. MODBUS PACKET CAPTURE

The module provides the capability to capture the Modbus traffic for analysis. The will allow the user and a remote support team to resolve any possible issues on site. To invoke the capture of the module, double-click on the Modbus Packet Capture item in the Project Explorer tree.

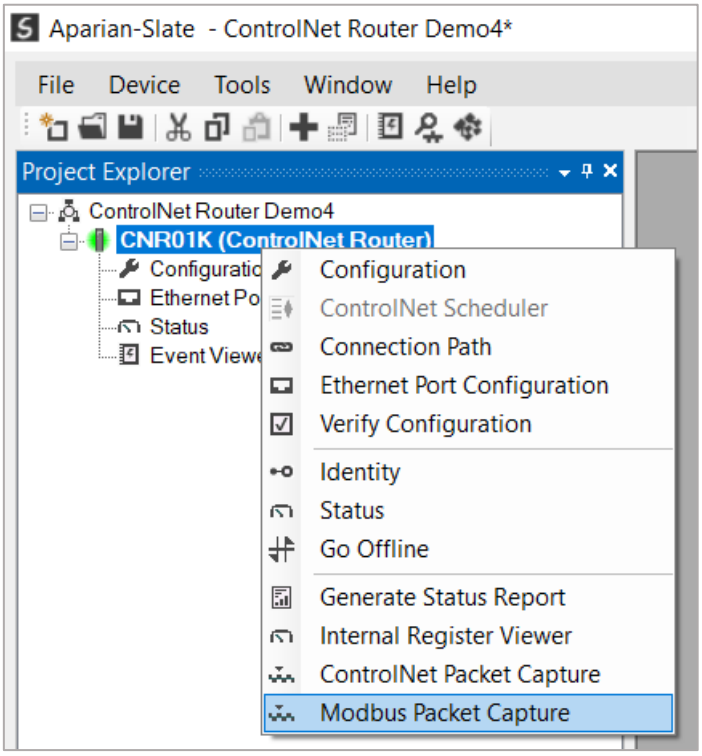


Figure 7.37 - Selecting Modbus Packet Capture

The Modbus Packet Capture window will open and automatically start capturing all Modbus packets.

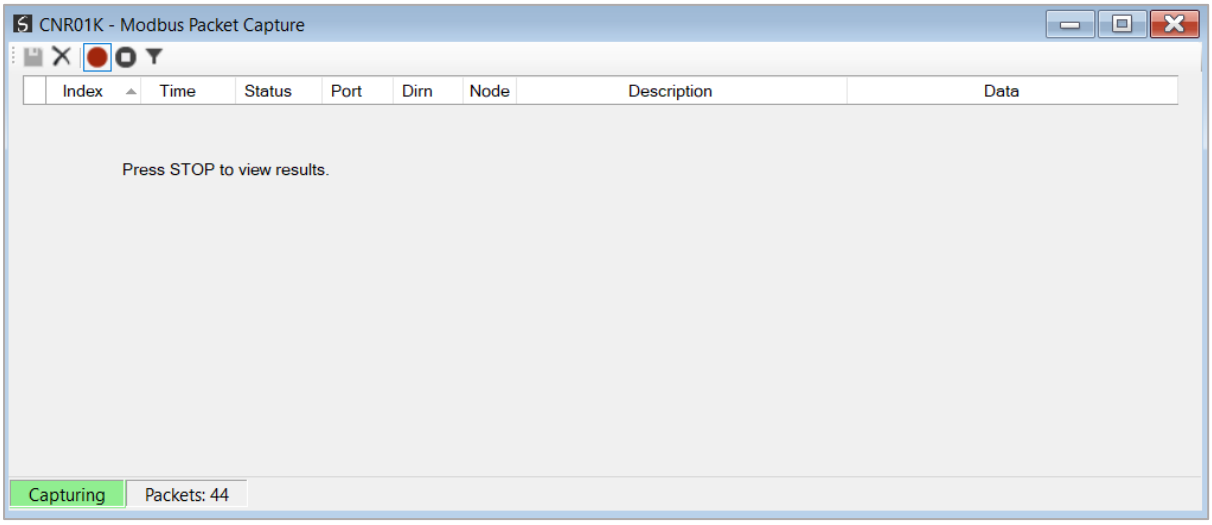


Figure 7.38 – Modbus packet capture

To display the captured Modbus packets, the capture process must first be stopped, by pressing the Stop button.

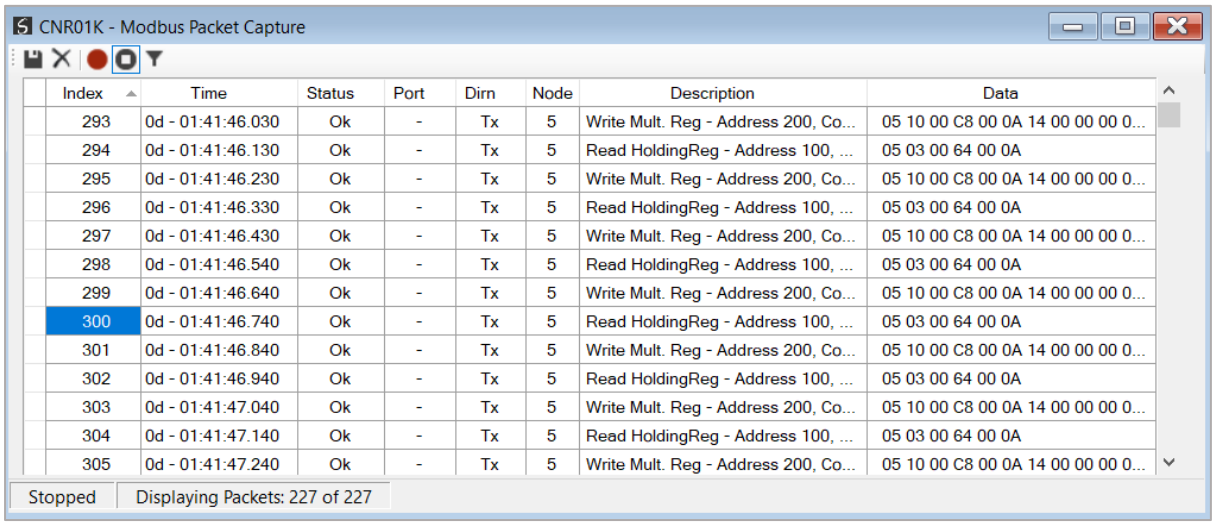


Figure 7.39 – Modbus Packet Capture complete

The captured Modbus packets are tabulated as follows:

Statistic	Description
Index	The packet index, incremented for each packet sent or received.
Time	The elapsed time since the module powered up.
Status	The status of the packet. Received packets are checked for valid Modbus constructs and valid checksums.
Port	Port on where the data was sent or received (TCP, RTU232, RTU485)
Dirn	The direction of the packet, either transmitted (Tx) or received (Rx).
Node	The Source Node address for the packet
Description	Description of the packet that was received.
Data	The raw packet data.

Table 7.17 – Modbus Packet Capture fields

The packet capture can be saved to a file for further analysis, by selecting the ***Save*** button on the toolbar. Previously saved Modbus Packet Capture files can be viewed by selecting the ***Modbus Packet Capture Viewer*** option in the tools menu.

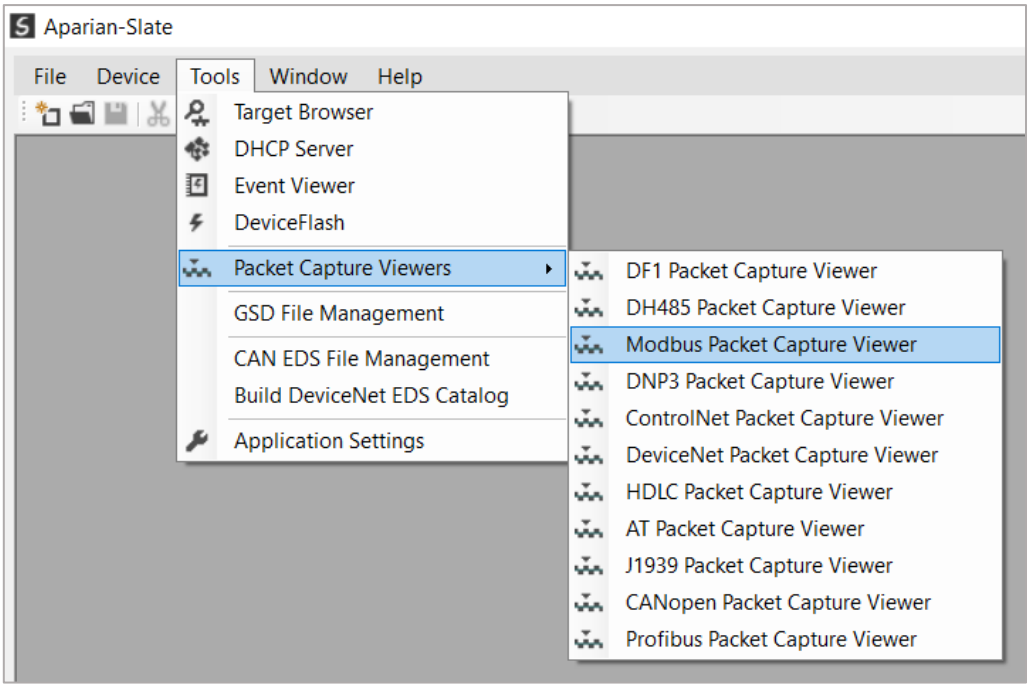


Figure 7.40 - Selecting the Modbus Packet Capture Viewer

7.8. MODULE STATUS REPORT

For assisting with support Slate can generate a status report for the module which is a word document that can be emailed to support. To generate this report the user can right-click on the module (when online in Slate) and select *Generate Status Report*.

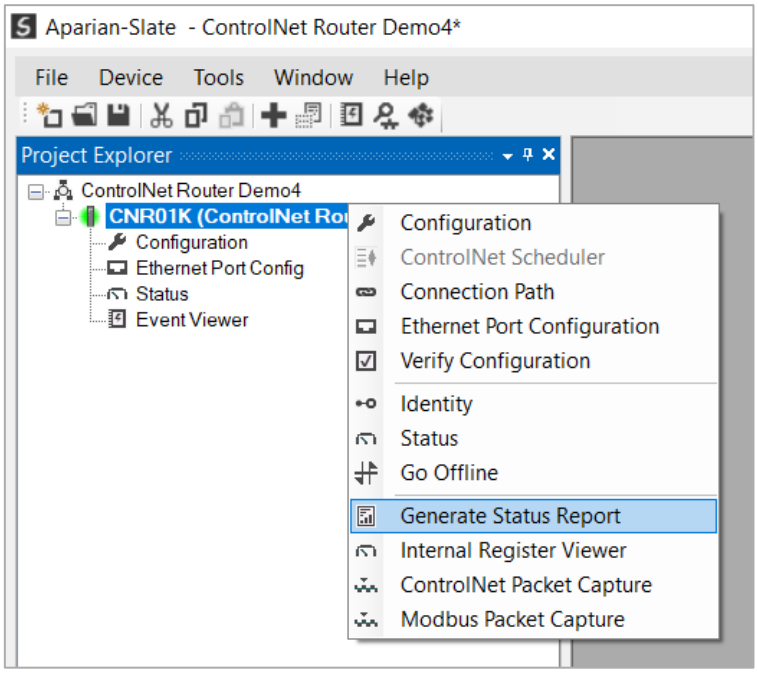


Figure 7.41 – Module Status Report

7.9. INTERNAL REGISTER VIEWER

The module provides an Internal Register Viewer that allows the user to view the data registers in the Internal Data Space (IDS) and Modbus Registers.

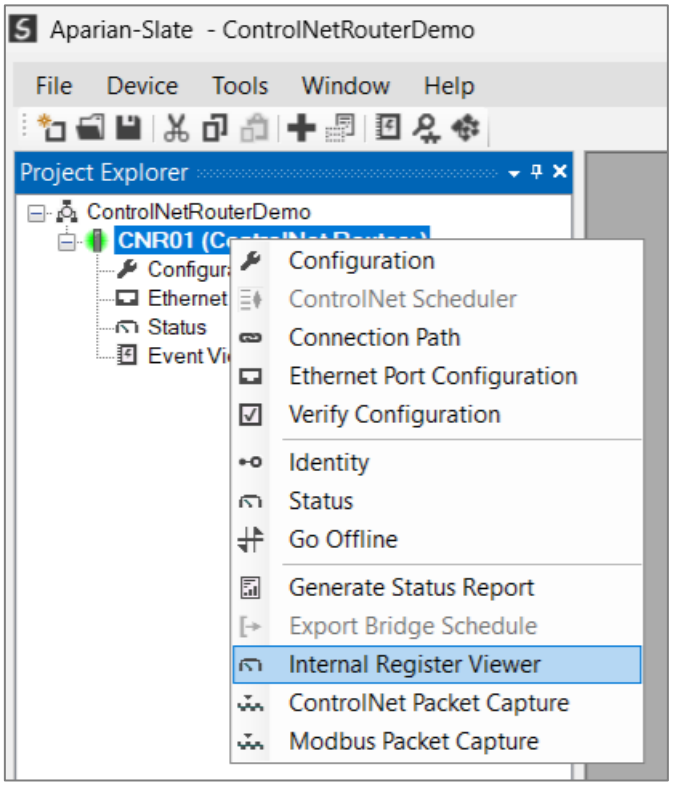


Figure 7.42 – Internal Register Viewer - Select

Once opened, the register type can be selected. Note that the Modbus Registers will only be available if the Primary Interface configured is for Modbus Server or Modbus Client.

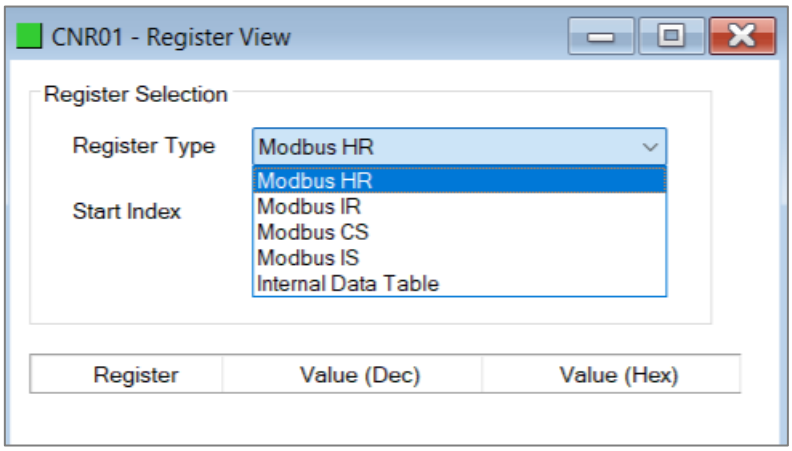


Figure 7.43 – Internal Register Viewer – Register Type

The Start and Stop Index can then be selected to determine how much data to display and refresh.

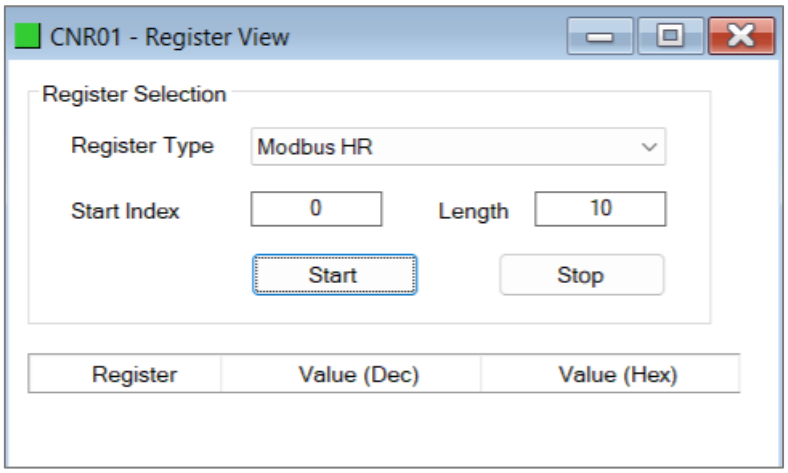


Figure 7.44 – Internal Register Viewer – Indexes

The data will appear and start refreshing by pressing the *Start* button.

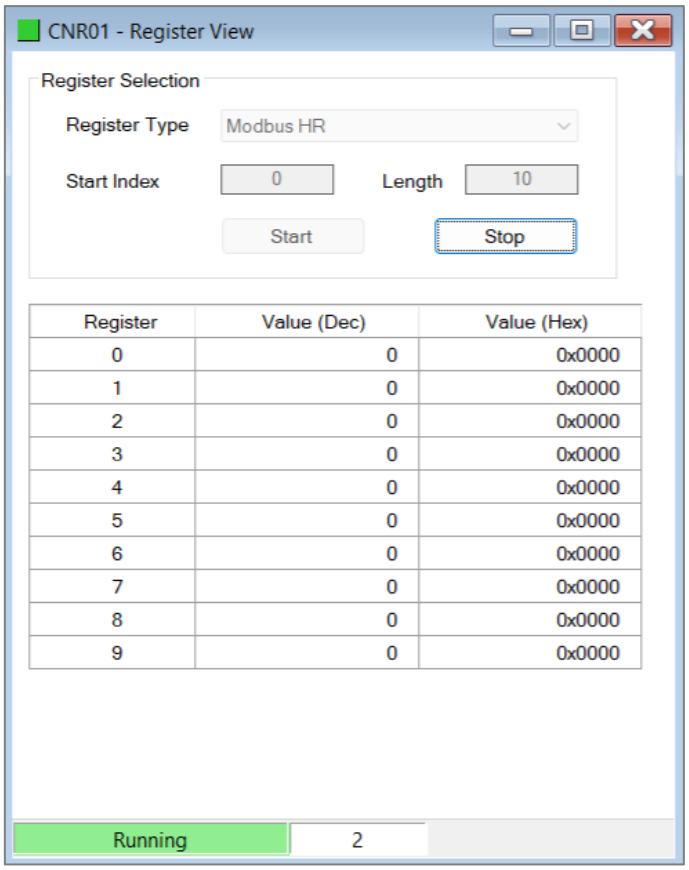


Figure 7.45 – Internal Register Viewer – Running

8. TECHNICAL SPECIFICATIONS

8.1. DIMENSIONS

Below are the enclosure dimensions as well as the required DIN rail dimensions. All dimensions are in millimetres.

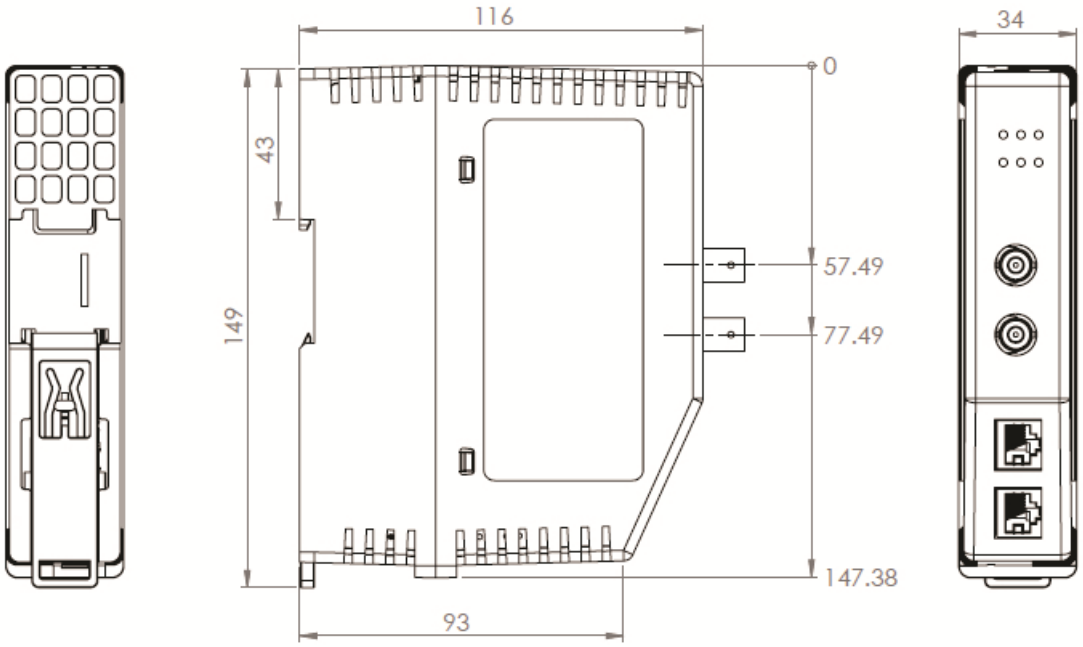


Figure 8.1 – ControlNet Router enclosure dimensions

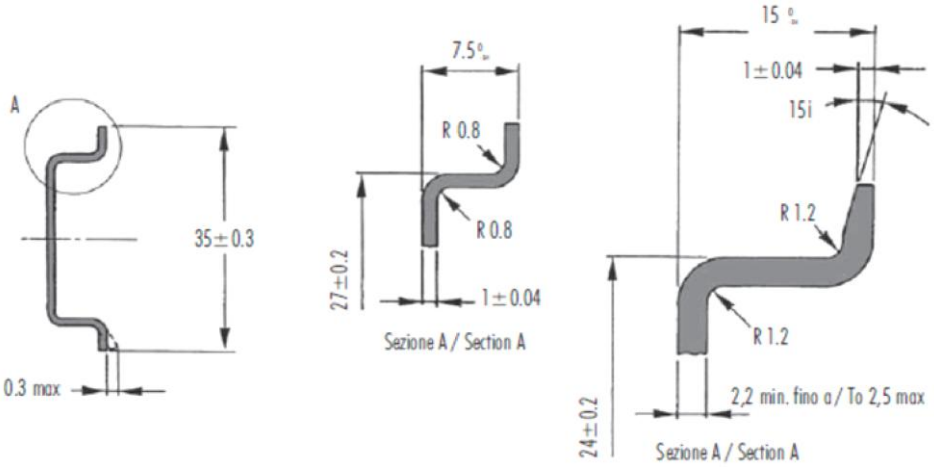


Figure 8.2 - Required DIN Rail dimensions

8.2. ELECTRICAL

Specification	Rating
Power requirements	Input: 10 – 32V DC, (85 mA @ 24 VDC)
Power consumption	2W (max)
Connector	3-way terminal
Conductors	24 – 18 AWG
Enclosure rating	IP20, NEMA/UL Open Type
Temperature	-20 – 70 °C
Earth connection	Yes, terminal based
Emissions	IEC61000-6-4
ESD Immunity	EN 61000-4-2
Radiated RF Immunity	IEC 61000-4-3
EFT/B Immunity	EFT: IEC 61000-4-4
Surge Immunity	Surge: IEC 61000-4-5
Conducted RF Immunity	IEC 61000-4-6

Table 8.1 - Electrical specification

8.3. ETHERNET

Specification	Rating
Connector	RJ45
Conductors	CAT5 STP/UTP
ARP connections	Max 100
TCP connections	Max 100
CIP connections	Max 50
Communication rate	10/100Mbps
Duplex mode	Full/Half
Auto-MDIX support	Yes
Embedded switch	Yes, 2 x Ethernet ports
Device Level Ring (DLR)	Supported
Network Time Protocol (NTP)	Supported

Table 8.2 - Ethernet specification

8.4. CONTROLNET

Specification	Rating
Connectors	2 x BNC connectors (ControlNet A and B).
Redundant ControlNet	Supported
Conductors	Quad shield RG-6 coaxial cable
Routing Supported (RSLogix programming)	Yes

Table 8.3 – ControlNet specification

8.5. CONTROLNET TARGET

Specification	Rating
Scheduled Connection Size	Max Input Size – 408 bytes (400 bytes mapped data) Max Output Size – 404 bytes (400 bytes mapped data)
Unscheduled Routed Client Max	40
Scheduled Connection Count	1

Table 8.4 – ControlNet Target specification

8.6. CONTROLNET ORIGINATOR

Specification	Rating
Scheduled Connection Max Count (Hardware)	10
Scheduled Connection Max Data (Hardware)	**Input Connection Data** 500 bytes per connection **Output Connection Data** Total of 277 bytes for 1 connection. Total of 269 bytes for 2 connections. Total of 261 bytes for 3 connections. Total of 253 bytes for 4 connections. Total of 245 bytes for 5 connections. Total of 237 bytes for 6 connections. Total of 229 bytes for 7 connections. Total of 221 bytes for 8 connections. Total of 213 bytes for 9 connections. Total of 205 bytes for 10 connections.

Scheduled Connection Max Count (Software)	30
Scheduled Connection Max Data (Software)	**Input Connection Data** 500 bytes per connection **Output Connection Data** 500 bytes per connection
Explicit Unscheduled Connection Max	10
Multiple ControlNet Router Connection Originators	Supported

Table 8.5 – ControlNet Originator specification

8.7. CONTROLNET BRIDGE

Specification	Rating
Scheduled Connection Max Count *connections can be Rack Optimized	30
Scheduled Connection Max Data	**Input Connection Data** 500 bytes per connection **Output Connection Data** 500 bytes per connection

Table 8.6 – ControlNet Originator specification

8.8. PCCC

Specification	Rating
Max PCCC Connections	10
Max PCCC Payload	1000 bytes

Table 8.7 – PCCC specification

8.9. ETHERNET/IP TARGET

Specification	Rating
Class 1 Connection Size	Max Input Size – 500 bytes (492 bytes mapped data) Max Output Size – 496 bytes (492 bytes mapped data)
Class 1 Connection Count	1
Class 3 Messaging Supported	Yes
UCMM Messaging Supported	Yes

Table 8.8 – EtherNet/IP Target specification

8.10. ETHERNET/IP ORIGINATOR

Specification	Rating
Class 1 Cyclic Connections Supported	Yes
Class 3 / UCMM Connections Supported	Yes
Class 1 Connection Count	16
Class 3 / UCMM Target Device Count	10
Class 3 / UCMM Mapping Count	50
Direct-To-Tag Logix Support	Yes

Table 8.9 – EtherNet/IP Originator specification

8.11. MODBUS TCP MASTER

Specification	Rating
Modes Supported	Modbus TCP
Max Modbus Slave device	20
Max Modbus Mapping	100
Mapping Ranges	Holding Register 0 – 65535 Input Register 0 – 65535 Input Status 0 – 65535 Coil Status 0 – 65535
Base Offset	Modbus (Base 0) PLC (Base 1)
Configurable Modbus TCP Port	Yes
Data Re-formatting Supported	BB AA BB AA DD CC CC DD AA BB DD CC BB AA

Table 8.10 – Modbus Master specification

8.12. MODBUS TCP SLAVE

Specification	Rating

Modes Supported	Modbus TCP
Mapping Ranges	Holding Register 0 – 65535 Input Register 0 – 65535 Input Status 0 – 65535 Coil Status 0 – 65535
Base Offset	Modbus (Base 0) PLC (Base 1)
Configurable Modbus TCP Port	Yes

Table 8.11 – Modbus Slave specification

8.13. CERTIFICATIONS

Certification	Mark
CE Mark	CE
RoHS2 Compliant	RoHS2
ODVA Conformance	EtherNet/IP™ * F/W 1.001
UL Mark File: E494895	c UL us LISTED CLASS 1, DIV 2, GROUPS A, B, C, D
ATEX	Ex II 3 G Ex ec IIC T5 -20°C ≤ Ta ≤ 70 °C

UKCA	

Table 8.12 – Certifications

9. APPENDIX

Value	LED State	Channel A		Channel B	
		Flash Phase 0	Flash Phase 1	Flash Phase 0	Flash Phase 1
000	Off	Off	Off	Off	Off
001	Green	Green	Green	Green	Green
010	Flashing Green	Green	Off	Green	Off
011	Flashing Red	Red	Off	Red	Off
100	Flashing Red/Green	Red	Green	Red	Green
101	Railroad Red	Red	Off	Off	Red
110	Railroad Red/Green	Red	Green	Green	Red
111	Red	Red	Red	Red	Red

Table 9.1 – ControlNet LED States

Mode (Bits 3..0)	Description
0x0	Invalid
0x1	Power-Up
0x2	Check for Cable
0x3	Waiting to Rogue
0x4	Check for Moderator Listen-Only
0x5	I'm Alive
0x6	Attached On-Line
0x7	Forced Listen-Only
0x8	Duplicate Listen-Only

Table 9.2 – ControlNet Net Mode

10. INDEX

A

Adanced, 101, 102
assembly instance, 35, 78

C

CANopen Router, 7, 170
CANopen Router, 170
CANopen Router general configuration, 44, 63, 82, 84, 85, 87, 88, 90, 105, 106
Connection path, 116
Contact Us, 18
ControlNet, 7, 10, 12, 17, 19, 20, 29, 30, 31, 32, 64, 116, 154, 160, 180, 181, 193, 195, 196
ControlNet general configuration, 30, 32, 64
ControlNet parameters, 30
ControlNet Router, 7, 10, 12, 17, 20, 29, 30, 36, 78, 116, 130, 139, 154, 180, 181, 193

D

DHCP, 20, 24, 25, 26
dimensions, 193
DIN rail, 20, 21, 193
DIP, 20
DIP Switch, 158

E

Ethernet Bridge, 35, 77
Ethernet connector, 22
Ethernet/IP, 53, 102
EtherNet/IP, 7, 27, 53, 102
EtherNet/IP Devices configuration, 52, 101
EtherNet/IP Map configuration, 53, 54, 56, 57, 102, 103, 115, 116

F

firmware upgrade, 30

I

input assembly, 130, 140
Input assembly, 160
input voltage, 22

L

LED, 153, 154

M

MODBUS, 171

O

output assembly, 130, 131, 139, 140, 146

P

partial import, 38, 81
PCCC, 195, 196, 197
Protocol, 24

R

requested packet interval (RPI), 36, 79
Rockwell Automation, 28
RSLinx, 28
RSLogix 5000, 35, 36, 37, 38, 39, 77, 78, 79, 80, 82

S

Safe Mode, 20
Slate, 10, 17, 24, 26, 28, 30, 118, 154, 174, 180, 181
statistics, 167, 169, 170, 172, 176, 179, 184, 187, 189
Statistics, 154
Support email, 18

T

Target Browser, 26, 27, 53, 102, 117

U

UDTs, 17
User Defined Types (UDTs), 37, 80

W

web server, 154, 181