Modbus Router/B

User Manual

A-MBR/B

Document No. D106-018
08/2025
Revision 1.15

CONTENTS

1. Preface 8
1.1. Introduction to the Modbus Router 8
1.2. Features 11
1.3. Architecture 13
1.4. Additional Information 19
1.5. Support 19
2. Installation 20
2.1. Module Layout 20
2.2. Module Mounting 21
2.3. Power 23
2.4. RS232/RS485 Port 23
2.5. RS485 Termination 24
2.6. Ethernet Ports 24
2.7. USB Port 25
3. Setup 26
3.1. Install Configuration Software 26
3.2. Network Parameters 26
3.3. Creating a New Project 31
3.4. General Parameters 33
3.5. Modbus Parameters 37
3.6. Ports Parameters 39
3.7. Scheduled Tag Mode 40
3.7.1. Logix Controller Type 41
3.7.1.1. EtherNet/IP 41
3.7.1.2. USB 45
3.7.2. MicroLogix/SLC Controller Type 49
3.8. Reactive Tag Mode 51
3.8.1.1. Logix Controller Type 52
3.8.1.1.1. EtherNet/IP 52
3.8.1.1.2. USB 56
3.8.1.2. MicroLogix/SLC Controller Type 60

3.8.1.3. Multi-Server 62
3.9. Unscheduled Mode 63
3.10. Enhanced Mode 64
3.10.1. Modbus Mode 64
3.10.1.1. Server 64
3.10.1.1.1. Multi-Server 64
3.10.1.1.2. Internal Data Space Mapping 65
3.10.1.2. Client 68
3.10.1.2.1. Modbus Auxiliary Map 68
3.10.1.2.2. Internal Data Space Mapping 70
3.10.2. EtherNet/IP Mode 72
3.10.2.1. Target 72
3.10.2.1.1. Studio 5000 Configuration (Version 21+) 73
3.10.2.1.2. RSLogix 5000 Configuration (Pre-Version 21) 77
3.10.2.1.3. Internal Data Space Mapping 81
3.10.2.2. Originator 83
3.10.2.2.1. EtherNet/IP Class 1 Device Connections 83
3.10.2.2.2. EtherNet/IP Explicit Message Device Connections 92
3.10.2.2.3. Internal Data Space Mapping 96
3.11. Bridge Mode 98
3.12. Concentrator Mode 100
3.13. Internal Data Space Map 101
3.13.1. Copy From 102
3.13.1.1. Internal 103
3.13.1.2. EIP Target 103
3.13.1.3. EIP Originator 104
3.13.1.4. MB Register 105
3.13.1.5. MB Aux 105
3.13.1.6. System 106
3.13.2. Copy To 108
3.13.2.1. Internal 108
3.13.2.2. EIP Target 109
3.13.2.3. EIP Originator 109

3.13.2.4. MB Register....110
3.14. Advanced....111
3.15. Module Download....111
4. Device Firmware Update....115
5. Security....118
5.1. Securing Process....119
5.1.1. Configuration....119
5.1.1.1. General Configuration....120
5.1.1.2. Password Rule Configuration....122
5.1.1.3. Custom Users Configuration....123
5.1.1.4. Services Configuration....124
5.1.1.5. IP Access Control List Configuration....125
5.1.1.6. MAC Access Control List Configuration....128
5.1.2. Download Security Configuration....129
5.1.3. Initialize User Authentication....130
5.2. Operation....133
5.2.1. Initial User Login....133
5.2.2. Role Specific Operations....135
5.2.2.1. Security Status....135
5.2.3. Change Other Password....140
5.2.4. Clear All Security....141
5.3. Event Log....141
5.4. Reset Security Configuration....142
6. Operation....143
6.1. Scheduled Tag Mode....143
6.2. Reactive Tag Mode....145
6.3. Unscheduled Mode....147
6.3.1. Example....151
6.4. Enhanced Mode....151
6.4.1. EIP Target – Modbus Server....152
6.4.2. EIP Originator – Modbus Server....153
6.4.3. EIP Target – Modbus Client....154
6.4.4. EIP Originator – Modbus Client....156

6.5. Bridge Mode....157
6.6. Concentrator Mode....159
6.7. Internal Map Data Formats....160
6.7.1. System Status....160
6.7.2. MB Aux Item Status....162
6.7.3. EtherNet/IP IO Device Status....162
7. Logix Redundancy....164
7.1. Setup....164
7.2. Operation....166
8. Logix USB Connectivity....170
9. Diagnostics....172
9.1. LEDs....172
9.2. Module Status Monitoring in Slate....173
9.2.1. General....176
9.2.2. Modbus Statistics....179
9.2.3. Logix Statistics....180
9.2.4. SLC/MicroLogix Statistics....181
9.2.5. Map Items....182
9.2.6. Aux. Items....182
9.2.7. CIP Statistics....183
9.2.8. EtherNet/IP Explicit....183
9.2.9. EtherNet/IP Map....185
9.2.10. EtherNet/IP Originator....185
9.2.11. Ethernet Clients....186
9.2.12. TCP / ARP....186
9.2.13. USB Statistics....187
9.3. Target Device Status Monitoring In Slate....188
9.3.1. EtherNet/IP....188
9.3.1.1. General....189
9.3.1.2. Input Data....190
9.3.1.3. Output Data....191
9.4. Modbus Diagnostics....191
9.5. Modbus Packet Capture....192

9.6. Module Event Log 194
9.7. Web Server 196
9.8. Module Status Report 196
10. Technical Specifications 198
10.1. Dimensions 198
10.2. Electrical 199
10.3. Environmental 199
10.4. Ethernet 200
10.5. Serial Port (RS232) 200
10.6. Serial Port (RS485) 201
10.7. USB Port 201
10.8. Modbus – Reactive / Scheduled Mode 201
10.9. Modbus Client – Enhanced Mode 202
10.10. Modbus Server – Enhanced Mode 202
10.1. EtherNet/IP Target 203
10.2. EtherNet/IP Originator 203
10.3. Certifications 203
Index 205

Revision History

Revision	Date	Comment
1.0	7 July 2019	Initial document
1.1	13 August 2019	Minor corrections to hyperlinks
1.2	12 September 2019	Corrected RS485 connections
1.3	13 October 2019	Added additional USB routing functionality
1.4	15 October 2019	Added ODVA conformance mark
1.5	4 February 2020	Corrected RS485 polarity labels
1.6	9 September 2020	Added information regarding using the No USB EDS file to be able to add the module to Logix using the EDS AOP
1.7	25 November 2021	Minor corrections.
1.8	4 August 2022	Added information required for UL regarding open type device enclosures.
1.9	3 April 2023	Updated configuration to support Enhanced, Bridge, and Concentrator modes Added new multi-server configuration Added internal register and Modbus viewer Added EtherNet/IP originator mode Added Module Security
1.10	13 November 2023	Added UKCA Conformance Mark
1.11	27 November 2023	Updated RS485/RS232 Connector description Updated EtherNet/IP Class 1 Connection Import options
1.12	25 January 2024	Added Ethernet Switch Split-Dual IP Option
1.13	20 May 2025	Add duplicate IP address indication to LEDs in the Diagnostics section.
1.14	17 June 2025	Updated Concentrator Mode to support EtherNet/IP as well.
1.15	26 August 2025	Added option to disable MB Comms when EIP communication fails.

1. PREFACE

1.1. INTRODUCTION TO THE MODBUS ROUTER

This manual describes the installation, operation, and diagnostics of the Aparian Modbus Router Series B (MBR/B). The Modbus Router provides intelligent data routing between EtherNet/IP and Modbus (serial Modbus-RTU232, Modbus-RTU485, and Ethernet Modbus-TCP). The Modbus Router allows the user to integrate Modbus devices into a Rockwell Automation Logix platform (ControlLogix or CompactLogix) or PLC (SLC or MicroLogix) with minimal effort.

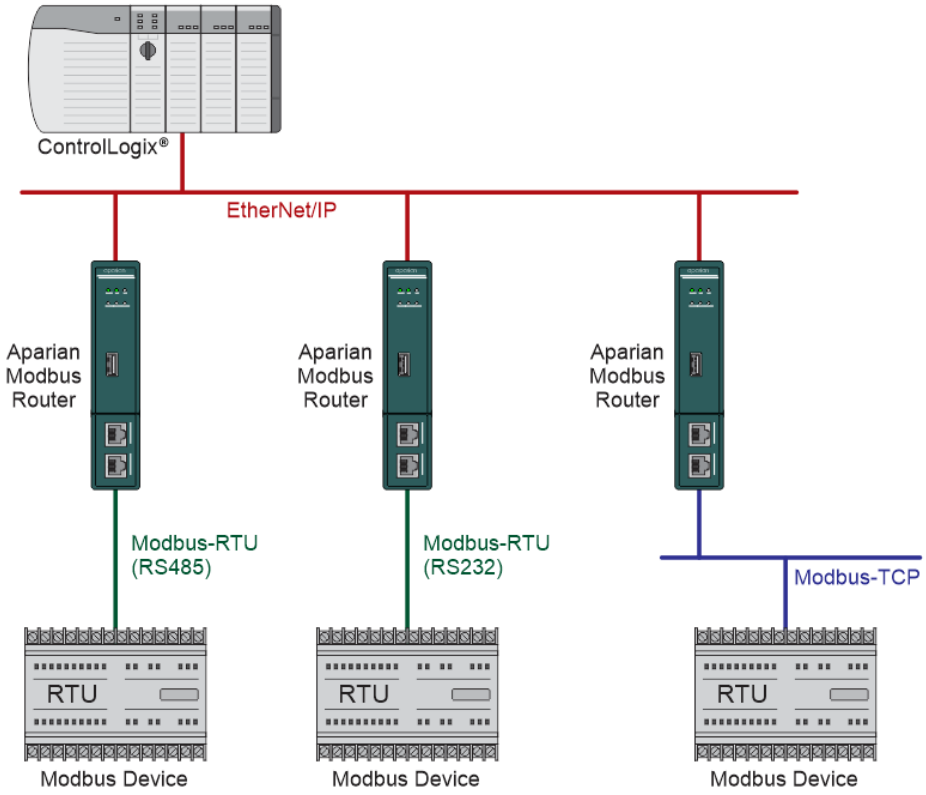


Figure 1.1 – Typical Setup.

3

The module can be configured in one of six router modes:

Scheduled Tag

The MBR/B operates as a **Modbus Client** and transfers data between a Modbus device and a number of Logix tags, using a preconfigured scheduled. No Logix or remote device configuration is required. The MBR/B can execute up to 100 Modbus message instructions (e.g., Modbus Holding Register Read) and exchange the Modbus data with Logix tags in up to three Logix Controllers. Each Modbus Message instruction is executed at one of four

configured rates. The Modbus data is exchanged using only one of the available ports (Ethernet TCP, RTU232, or RTU485).

Reactive Tag
The MBR/B operates as a **Modbus Server** and transfers data between Modbus Clients and a number of Logix tags, using a preconfigured tag map. No Logix or remote device configuration is required. The MBR/B can map up to 100 Modbus message instructions (e.g., Modbus Holding Register Read) and exchange the Modbus data with Logix tags in up to three Logix Controllers. The Modbus data is exchanged using only one of the available ports (Ethernet TCP, RTU232, or RTU485).

Unscheduled
The MBR/B operates as a **Modbus Client** and sends Modbus messages to Modbus devices when receiving a Logix Message Instruction. This provides the user with complete control (from Logix) of which Modbus instructions must be sent and at which frequency. The Modbus data is exchanged using only one of the available ports (Ethernet TCP, RTU232, or RTU485).

Enhanced
The user can decide if the MBR/B is operating as a Modbus Client or Server as well as if the MBR/B is operating as an EtherNet/IP Target or EtherNet/IP Originator.

In Enhanced Mode the MBR/B can be configured in any one of the four combinations:

Modbus	EtherNet/IP
Client	Target
Client	Originator
Server	Target
Server	Originator

Table 1.1 – Enhanced Mode Operating Combinations.

Modbus Mode

Modbus Server

The data from the configured EtherNet/IP devices (using either Class 1 Assemblies, Explicit messaging or Logix Direct-to-Tag reads/writes) will be written to, or read from, the module's internal Modbus Registers using the Internal Map. These registers can be accessed by a remote Modbus Client using either Modbus TCP, Modbus RTU232, Modbus RTU485, or any simultaneous combination of these.

Modbus Client

The data from the configured EtherNet/IP devices (using either Class 1 Assemblies, Explicit messaging or Logix Direct-to-Tag reads/writes) will be written to, or read

from, the module's internal Modbus Registers using the Internal Map. The Modbus Auxiliary Map can then be used to configure the Modbus data exchange between multiple remote Modbus Server devices and the module's internal Modbus registers. The Modbus communication can be via Modbus TCP, Modbus RTU232, Modbus RTU485, or any simultaneous combination of these.

EtherNet/IP Mode

EtherNet/IP Target

As an EtherNet/IP target, the module can exchange the Class 1 data with devices on the Modbus network:

- **EtherNet/IP Class 1 connection**

  Here a remote EtherNet/IP device (e.g. a Logix controller) establishes between 1 and 5 Class 1 connections to the module. Modbus data can be mapped into four separate input and output class 1 cyclic connections to the Logix controller (allowing up to 2KB input and 2KB output to be exchanged at the requested packet interval – RPI).

EtherNet/IP Originator

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

- **EtherNet/IP Explicit Messaging**

  This allows the MBR/B to exchange data with up to 10 EtherNet/IP devices. The module can use either Class 3 or Unconnected Messaging (UCMM) to Get and Set data in the remote EtherNet/IP devices.

- **Direct-To-Tag**

  This allows the MBR/B to exchange data with a Logix controller without the need to write any application code (e.g. ladder) in Studio 5000. The Modbus data is directly read from, or written to, Logix tags.

- **EtherNet/IP Class 1 connection**

  Modbus data can be mapped to a maximum of 10 EtherNet/IP devices using input and output class 1 cyclic connections. This will allow the MBR/B to "own" the EtherNet/IP target device and exchange Modbus data using the EtherNet/IP device's input and output assemblies.

Bridge

The MBR/B can operate as a transparent link for Modbus between different media (TCP, RTU232, and RTU485). This allows a Modbus client on Modbus TCP to communicate with Modbus devices on RS232 or RS485 and vice versa.

Concentrator
In Concentrator Mode, the MBR/B will collect data from multiple Modbus devices and provide the consolidated data to one or more Modbus Clients. The MBR/B will be a Modbus Server on one port, and Modbus Clients on the other ports. This will allow a Modbus SCADA to efficiently read the data from multiple Modbus devices, because the data from the various Modbus devices have been concentrated into the MBR/B.
NOTE: The module will still support an EtherNet/IP connection to a Logix controller to allow monitoring the Modbus data.

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

The module provides a range of statistics, a Modbus packet analyzer, and an Internal Modbus Register viewer to help fault find any problems.

The Modbus Router supports Modbus on three ports which can be configured from the Slate environment: Modbus-RTU232 (Serial), Modbus-RTU485 (Serial), and Modbus-TCP (Ethernet).

The Modbus Router uses isolated RS232 and isolated RS485 for Modbus serial communication providing better noise immunity. The RS232 and RS485 ports use a terminal block for convenient installation.

The MBR/B supports security services allowing the user to configure various levels of module security.

The module has two Ethernet ports and supports Device-Level-Ring (DLR) architectures.

A built-in webserver provides detailed diagnostics of system configuration and operation, including the display of Modbus operation and communication statistics, without the need for any additional software.

1.2. FEATURES

- MBR/B can operate as a Modbus Client or Modbus Server.
- Modbus Client mode can communicate with up to 100 Modbus Servers.
- Supports Modbus passthrough messaging.
- Modbus Server mode supports full Modbus data range.
- Modbus Server mode supports emulating multiple Modbus nodes.
- Configurable Modbus TCP Port numbers.

- Supports EtherNet/IP Target and Originator modes:
    - EtherNet/IP Target (Class 1 connection)
    - EtherNet/IP Originator (Class 1 connection with up to 10 EtherNet/IP devices and Explicit Messaging, including Direct-To-Tag Logix tag access, with up to 10 EtherNet/IP devices).
- Supports CIP USB interface for Logix controllers.
- EtherNet/IP Class 1 target connections support 2kbyte input and 2kbyte output data to Logix input and output assemblies.
- Supports Modbus media conversion allowing for a transparent link from one media to another (e.g., Modbus TCP to Modbus RTU232).
- Slate provides a Modbus packet capture utility for better diagnosis.
- Dual Ethernet ports which support Linear, Device-Level-Ring (DLR) or Split Dual-IP Modes.
- Network Time Protocol (NTP) supported for external time synchronization.
- Small form factor – DIN rail mounted.

The security services provide the following features:

- Configurable level of security
- Encrypted EtherNet/IP communication
- Fixed User Roles and Custom Users
- Configurable Password rules
- Login Expiry
- Failed Login Cool-off and Account Disable options
- Configurable Global Services and Custom Ports
- Optional IP Address Access Control List
- Optional MAC Access Control List
- Restrictions on Reset commands
- Restrictions on Device Flash options
- Security Audit trail in Event Log

1.3. ARCHITECTURE

The figures below provide an example of the typical network setup for connecting Modbus networks to either EtherNet/IP or PCCC networks.

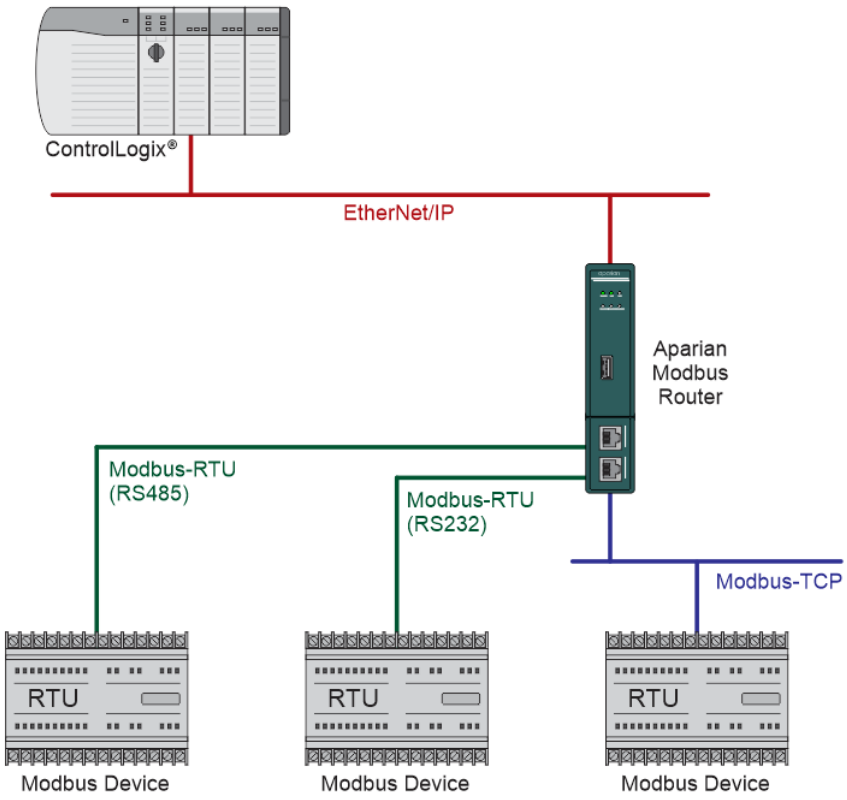


Figure 1.2 - Enhanced mode as a Modbus Client and EtherNet/IP Target.

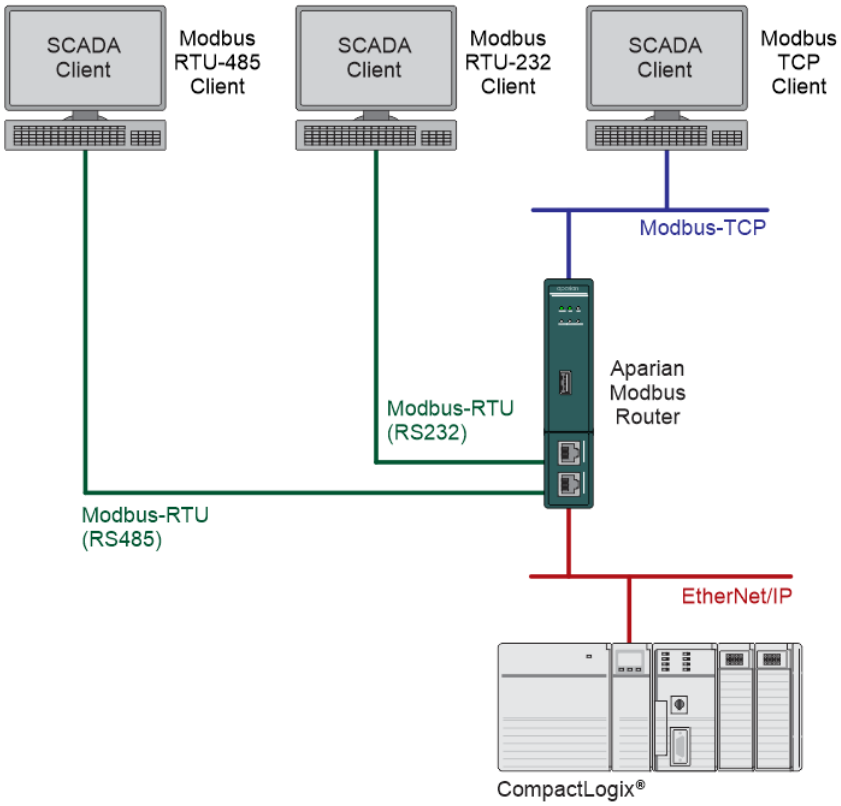


Figure 1.3 - Enhanced mode as a Modbus Server and EtherNet/IP Target.

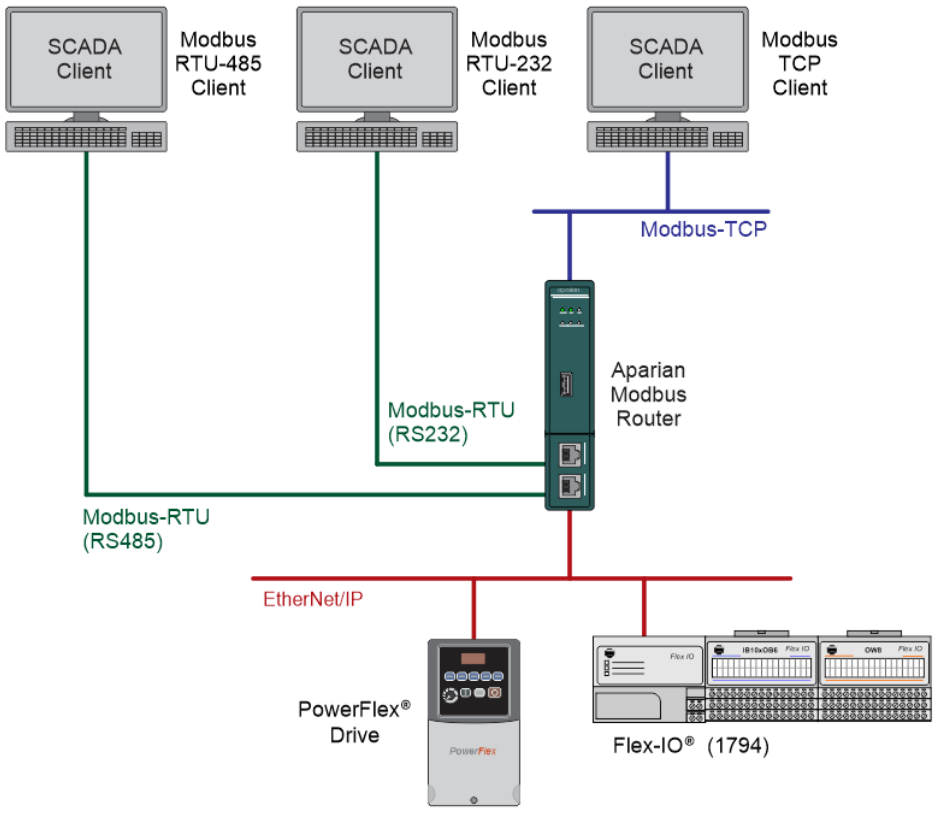


Figure 1.4 - Enhanced mode as a Modbus Server and EtherNet/IP Originator.

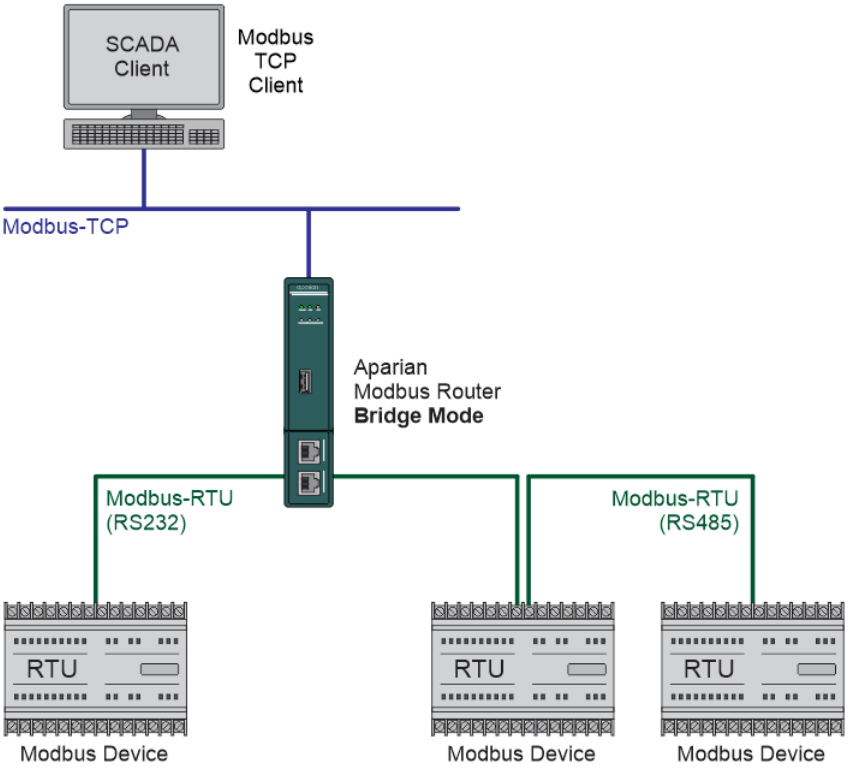


Figure 1.5 – Bridge Mode - from Modbus TCP to Modbus RTU232/RTU485.

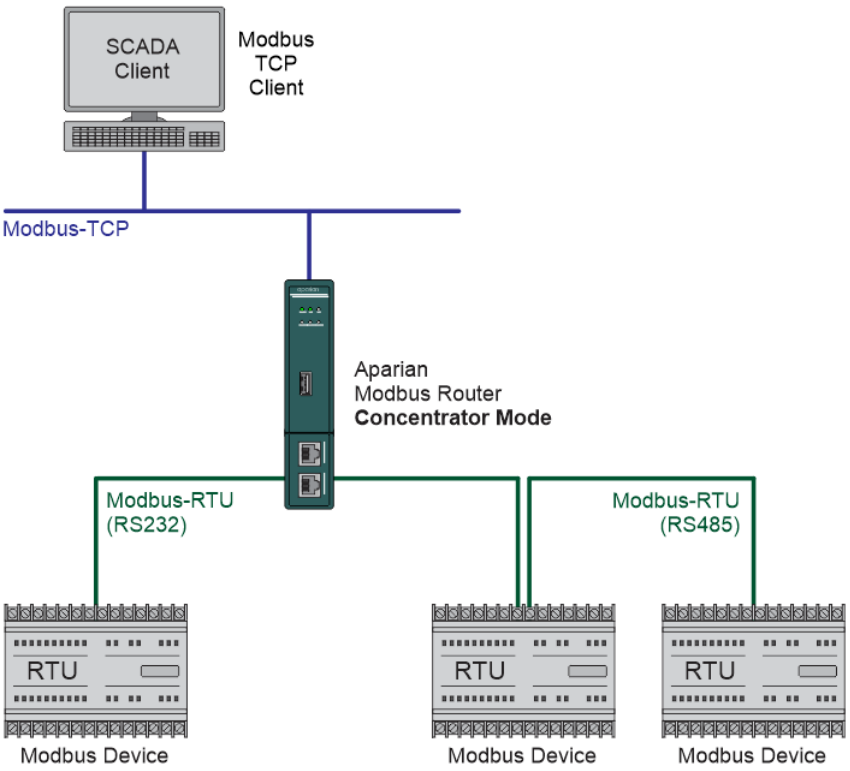


Figure 1.6 – Concentrator Mode – Modbus TCP SCADA interfacing with Modbus RTU devices.

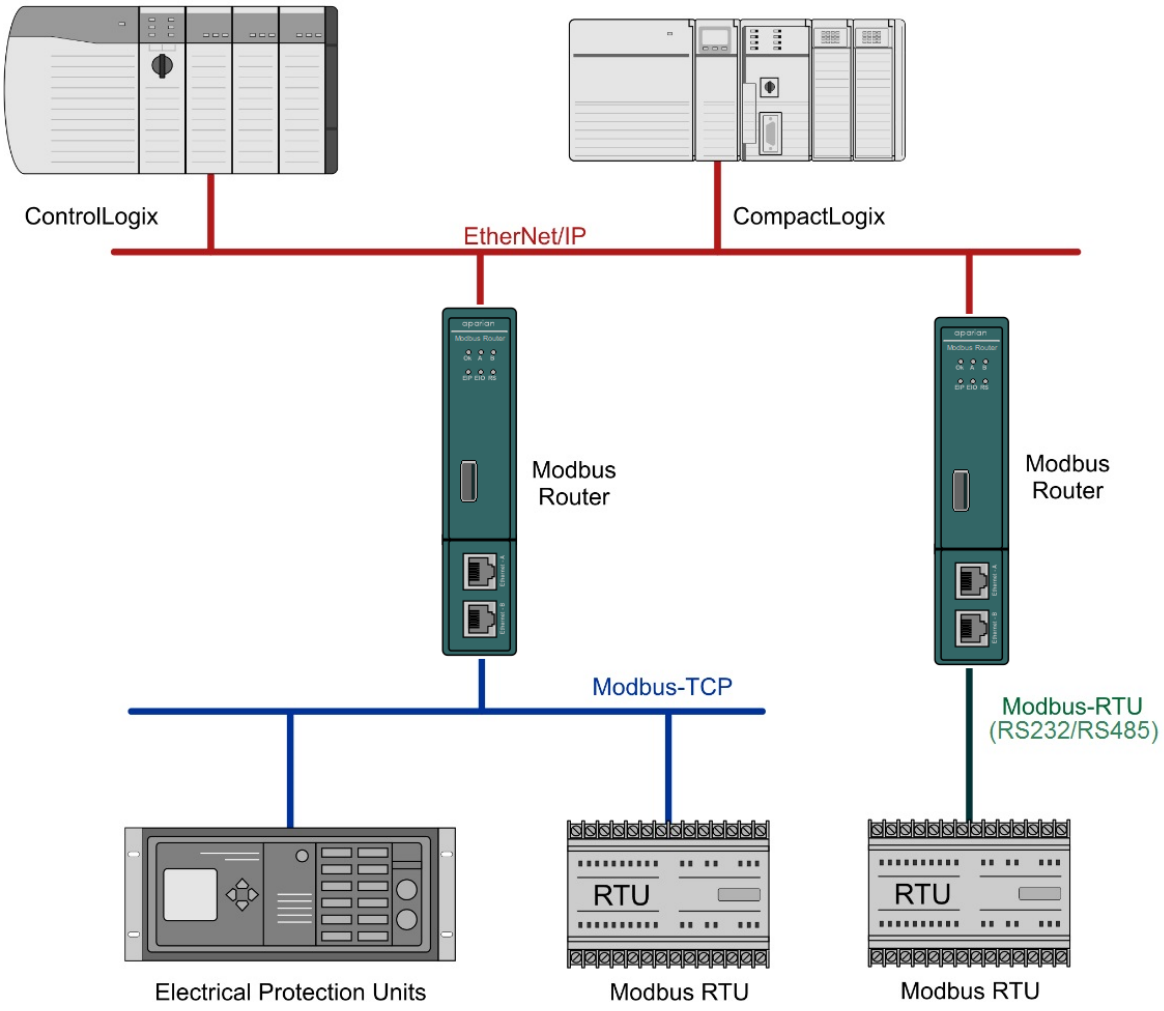


Figure 1.7 - Example of a typical network setup in scheduled/unscheduled mode.

Systems that rely on a central Logix Controller communicating to a number of remote Modbus devices, (e.g. Electrical Protection Units or Remote Terminal Units), may find the Modbus Router useful when operating in Scheduled Tag Mode as shown in the figure above. The module, acting as a Modbus client, will exchange data between the Modbus device and Logix platform at a configured interval without any need for additional coding or mapping.

The Modbus Router can also communicate with a Logix controller using the USB port on the front of the controller. This will allow the user to setup redundant Logix Controllers each with its own Modbus Router directly connected to the Logix Controller over USB. Only the Modbus Router connected to the Primary controller will be “active” with a configured Modbus TCP/IP address. When the primary and standby Logix Controllers swap the Modbus Routers will follow the primary controller and that specific Modbus Router will become active.

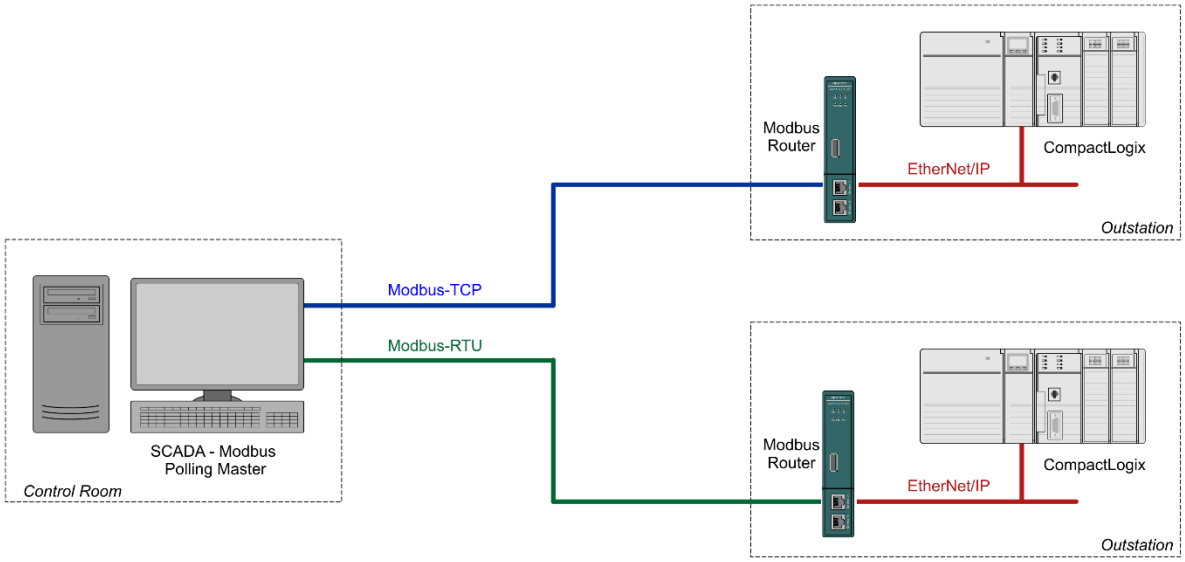


Figure 1.8 - Example of a typical network setup in reactive mode

By converting and redirecting serial Modbus messages from legacy devices to EtherNet/IP, the module provides an interface for data exchange to Allen-Bradley ControlLogix and CompactLogix platforms. This enables user to replace legacy devices and systems with minimal effort and downtime.

The Modbus Router allows a Logix platform to seamlessly integrate into a Modbus network with Reactive Tag Mode. The module will route Modbus messages directly to Logix tags with no need for additional ladder code.

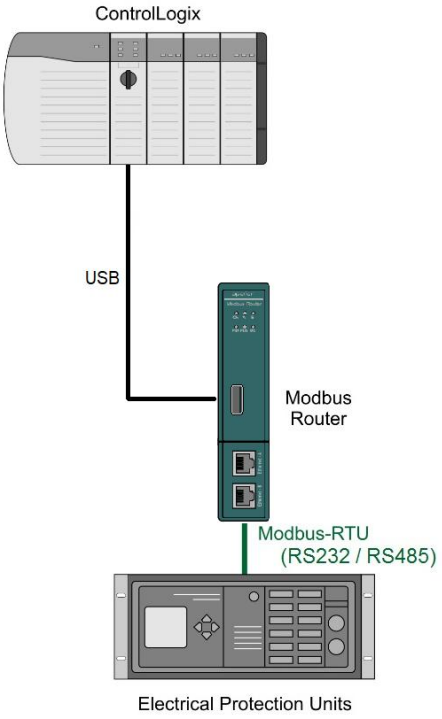


Figure 1.9 - Example of a typical network setup using the Logix Controller USB port.

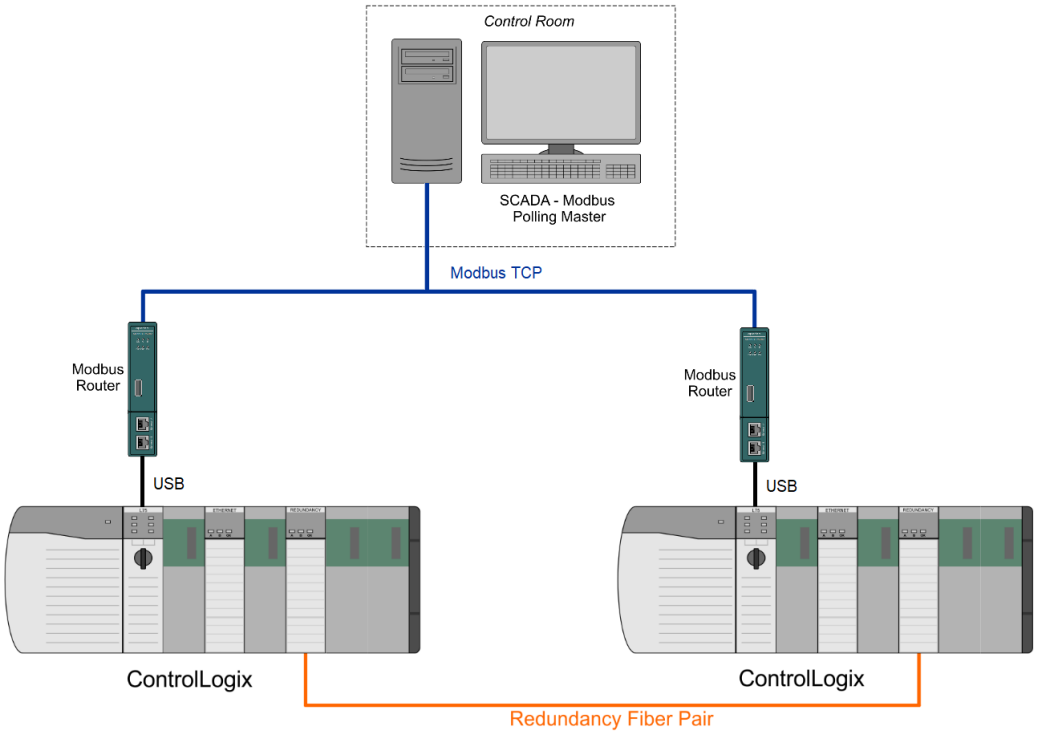


Figure 1.10 - Redundant Logix Controller communicating with Modbus Router over USB.

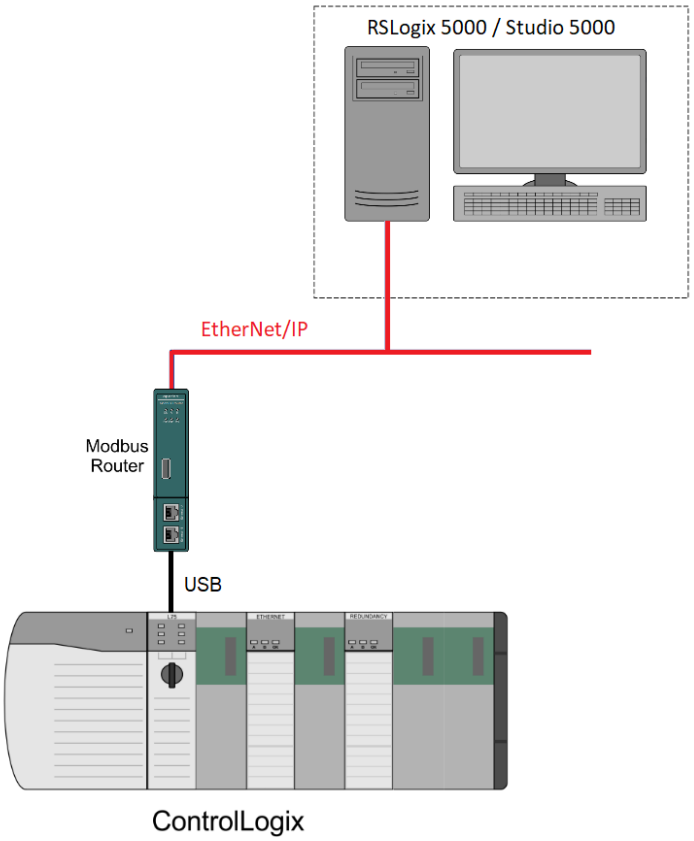


Figure 1.11 – Connect to Logix with Studio 5000 over USB.

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
Modbus Router User Manual Modbus Router Datasheet Example Code & UDTs	https://www.aparian.com/products/modbusrouterb
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
Modbus	http://www.modbus.org

Table 1.2 - 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	www.aparian.com/contact-us
Support email	support@aparian.com

Table 1.3 – Support Details.

2. INSTALLATION

2.1. MODULE LAYOUT

The module has two ports at the bottom of the enclosure as shown in the figure below. The ports are used for Ethernet, RS232 or RS485 serial, and power. The power port uses a three-way connector which is used for the DC power supply positive and negative (or ground) voltage as well as the earth connection.

The Ethernet cable must be wired according to industry standards which can be found in the additional information section of this document.

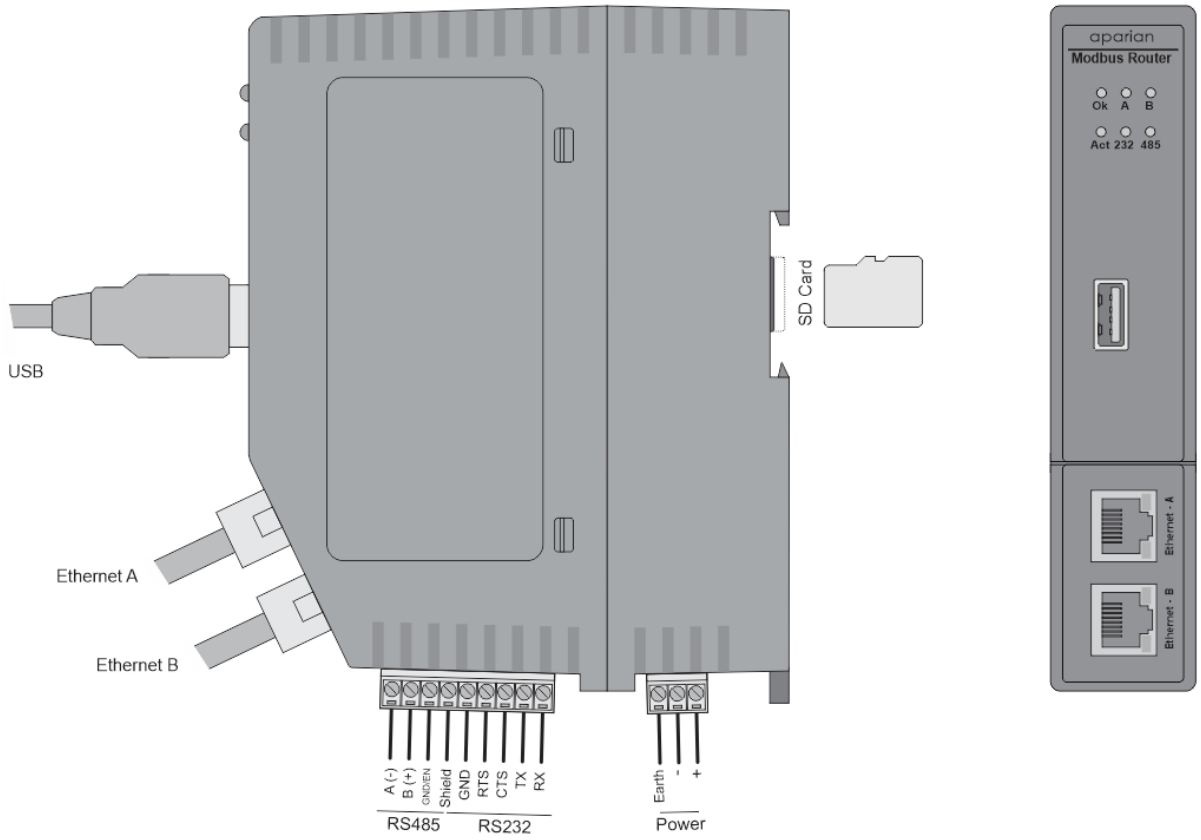


Figure 2.1 – Modbus Router side and front view.

The module provides six diagnostic LEDs as shown in the front view figure above. These LEDs are used to provide information regarding the module system operation, the Ethernet interface, and the auxiliary communication interface (RS232 or RS485).

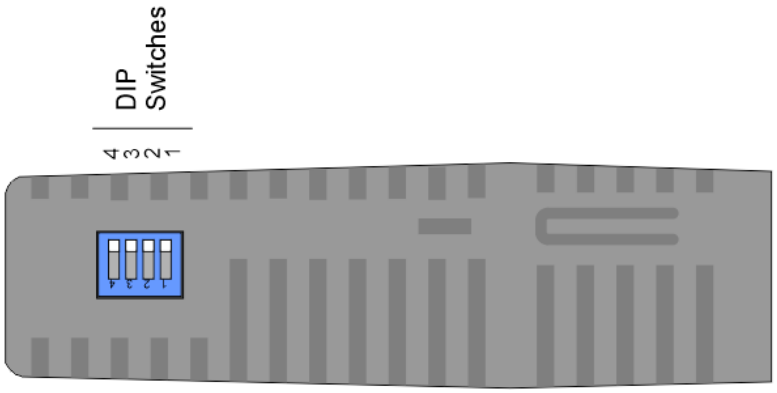


Figure 2.2 – Modbus Router top view.

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 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.

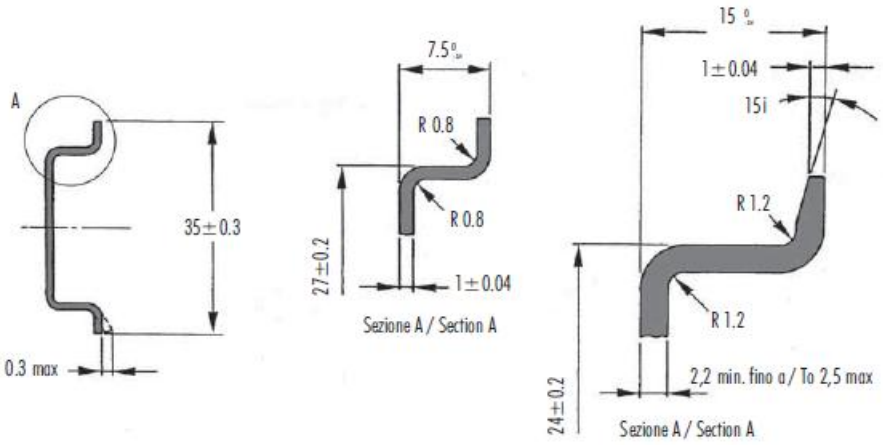


Figure 2.3 - DIN rail specification.

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.

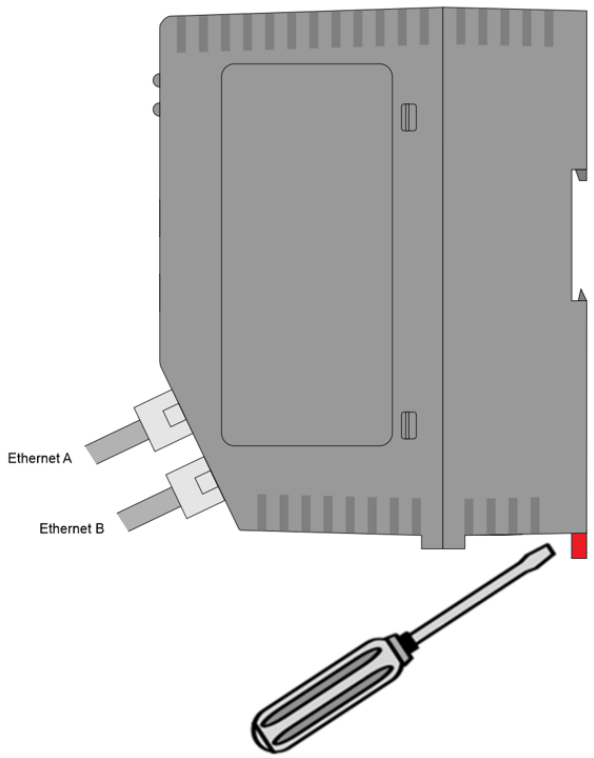


Figure 2.4 - DIN rail mounting.

2.3. POWER

A three-way power connector is used to connect Power+, Power– (GND), and earth. The module requires an input voltage of 10 – 32Vdc. **Refer** to the technical specifications section in this document.

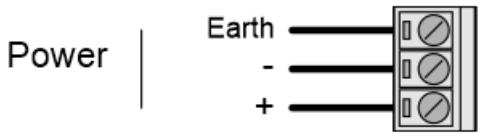


Figure 2.5 - Power connector.

2.4. RS232/RS485 PORT

The nine-way connector is used to connect the RS232 and RS485 conductors for serial communication. The shield terminal can be used for shielded cable in high noise environments.

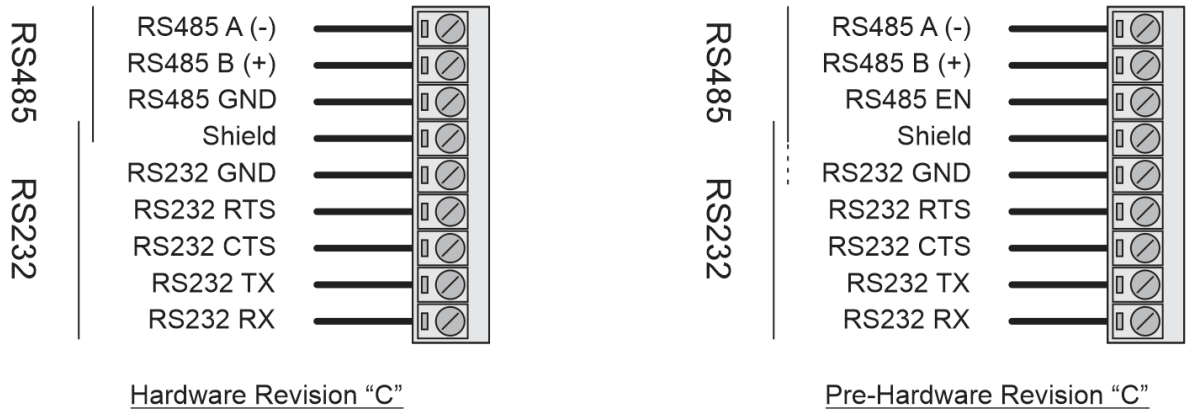


Figure 2.6 - RS232/RS485 connector.

The RS485 port provides the standard ***A(-)***, ***B(+)*** conductors.

For long RS485 networks, it is recommended to also connect the RS485 Ground reference. This is available as a dedicated **RS*485-GND*** pin on Hardware Revision "C" and later.
For Pre-Hardware Revision "C" the RS485 Ground wire must be connected to the **RS*232-GND*** pin, and the ***RS485 Ground Bridge*** option enabled in the configuration.

NOTE: The Hardware Revision is indicated on the product label by a single character to the right of the F/W REV.

The RS485 ***(EN***) transmit drive enable is provided to allow the module to be used with repeaters and radios that may require a transmit enable line. This pin is not available in Hardware Revision "C" and later.
Note that the ***EN*** pin is referenced to **RS*232-GND*** and requires the ***RS485 Ground Bridge*** option to be enabled in the configuration.

The RS232 port provides the standard communication pins (***RX***, ***TX***, and ***GND***) as well as hardware handshaking lines for legacy systems (***RTS*** – Request to Send, ***CTS*** – Clear to Send).

Both RS232 and RS485 share a common cable shield connection which should be connected to the shield of the outgoing cable (RS232 and/or RS485).

NOTE: The shield of the RS232/RS485 port is internally connected to the power connector earth. Thus, when using a shield, it is important to connect the Earth terminal on the power connector to a clean earth. Failing to do this can lower the signal quality of the RS232/RS485 communication.

NOTE: When using a shielded cable, it is important that only one end of the shield is connected to earth to avoid current loops. It is recommended to connect the shield to the Modbus Router module, and not to the other Modbus device.

2.5. RS485 TERMINATION

All RS485 networks need to be terminated at the extremities (start and end point) of the communication conductor. The termination for the RS485 network can be enabled/disabled via the module configuration. Enabling the termination will connect an internal 150 Ohm resistor across the positive (+) and negative (-) conductors of the RS485 network.

2.6. ETHERNET PORTS

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

The module has an embedded switch connecting the two Ethernet ports. The switch functionality is disabled when the Split (Dual-IP) mode is enabled.

2.7. USB PORT

The module supports USB2.0 on its USB port and will operate as a USB Host device. The user will require a USB Type-A connector on the Modbus Router side and generally a USB Type-B connector on the device side (i.e. Logix Controller).

Figure 2.7 – USB 2.0 Type-A to USB Type-B cable.

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.

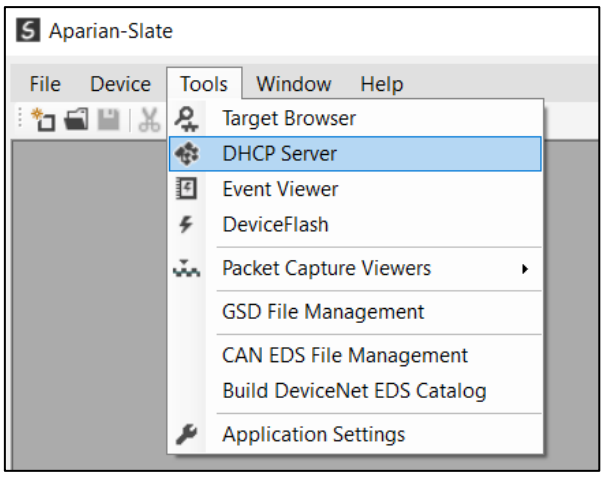


Figure 3.2 - Selecting DHCP Server.

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

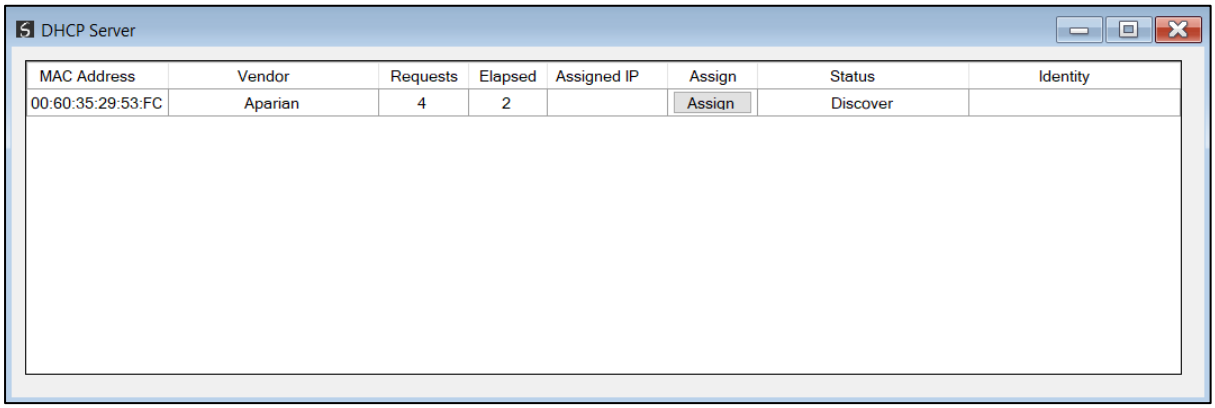


Figure 3.3 - DHCP Server.

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.

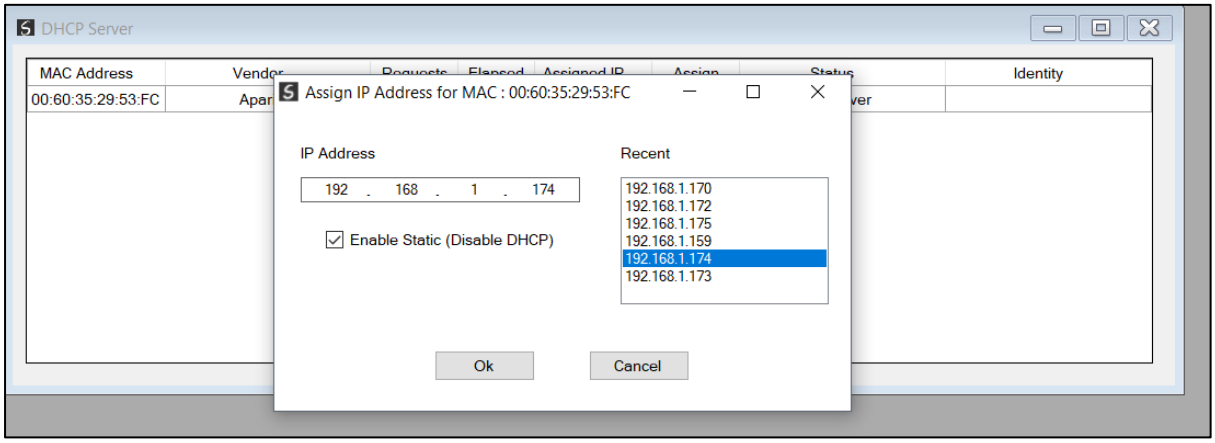


Figure 3.4 - Assigning IP Address.

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.

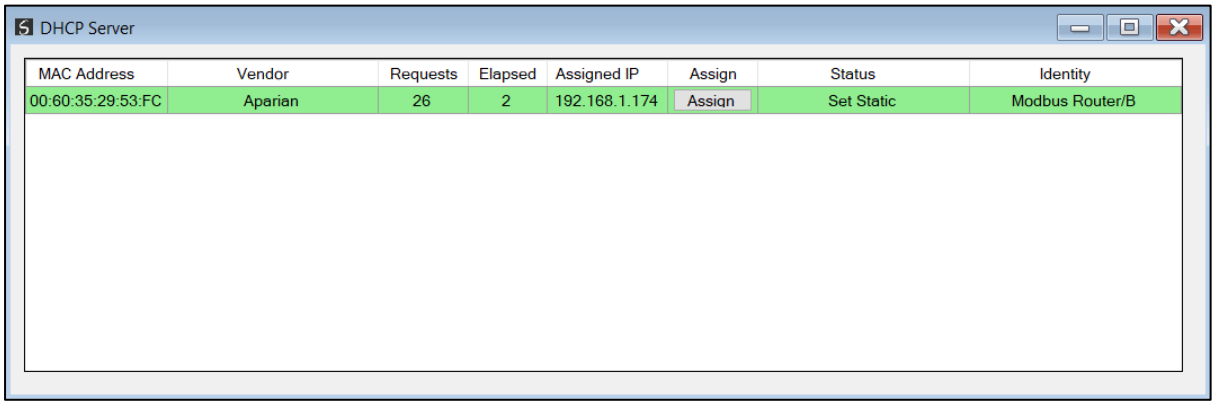


Figure 3.5 - Successful IP address assignment.

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.

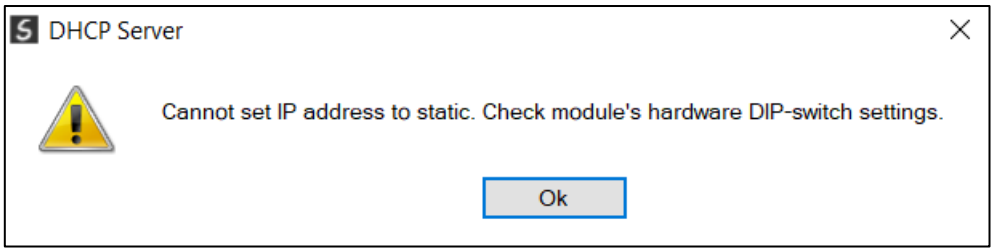


Figure 3.6 - Force DHCP warning.

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.

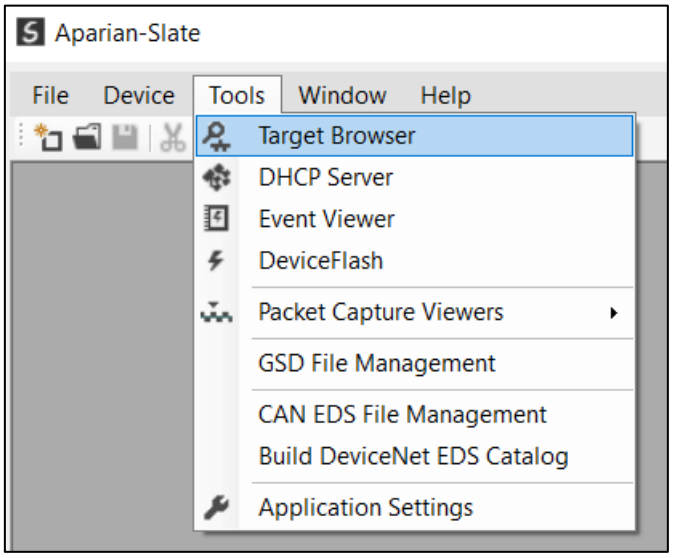


Figure 3.7 - Selecting the Target Browser.

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

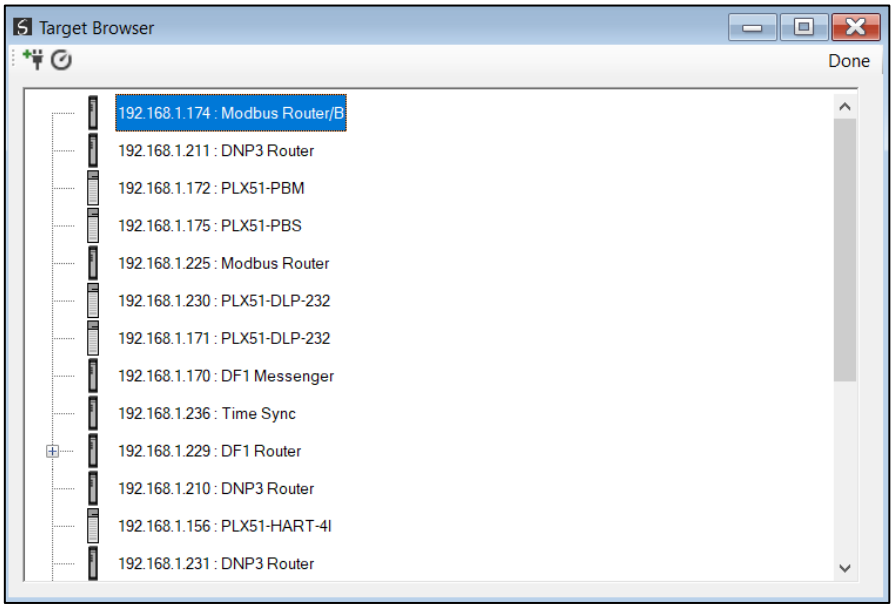


Figure 3.8 - Target Browser.

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

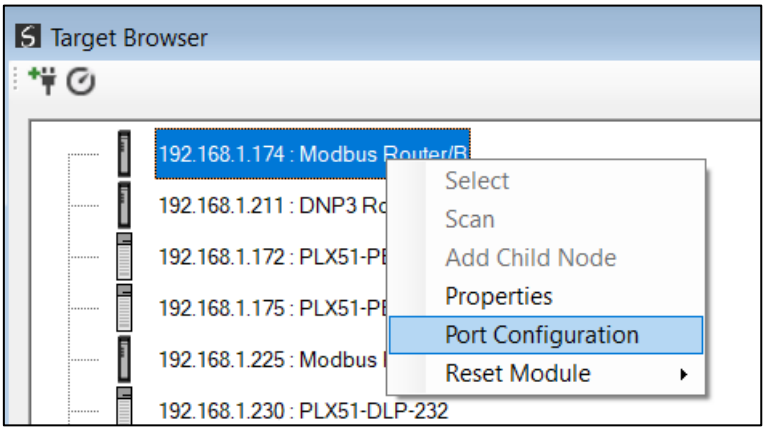


Figure 3.9 - Selecting Port Configuration.

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

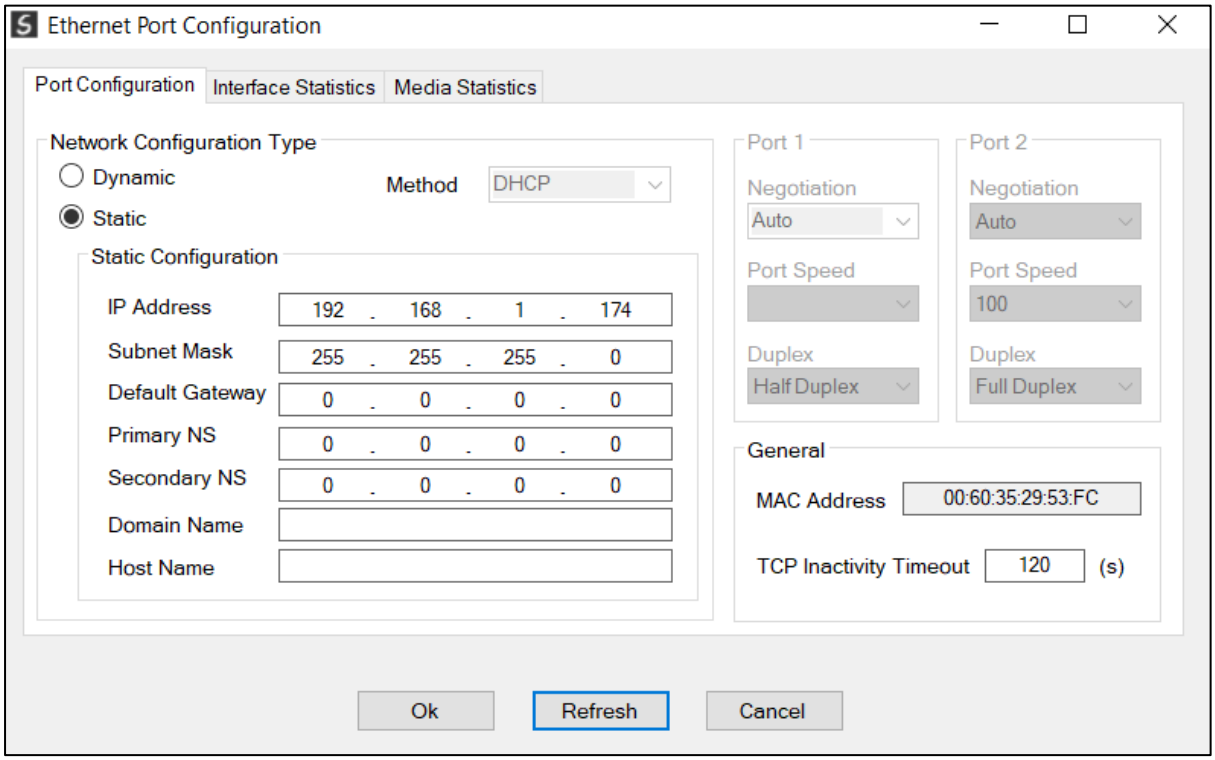


Figure 3.10 - Port Configuration.

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.

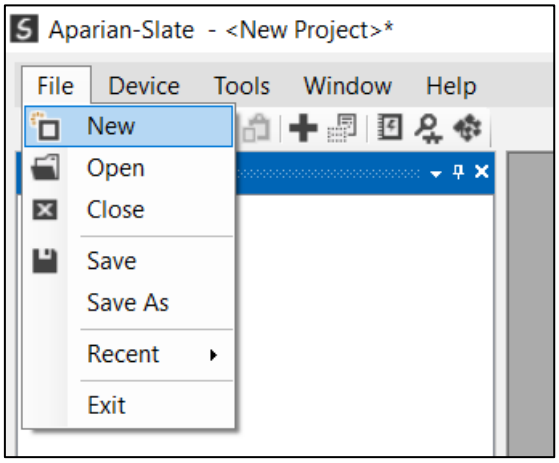


Figure 3.11 - Creating a new project.

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 under the Device menu.

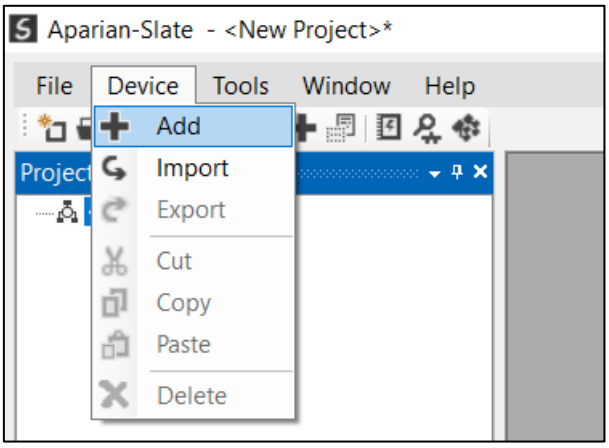


Figure 3.12 - Adding a new device.

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

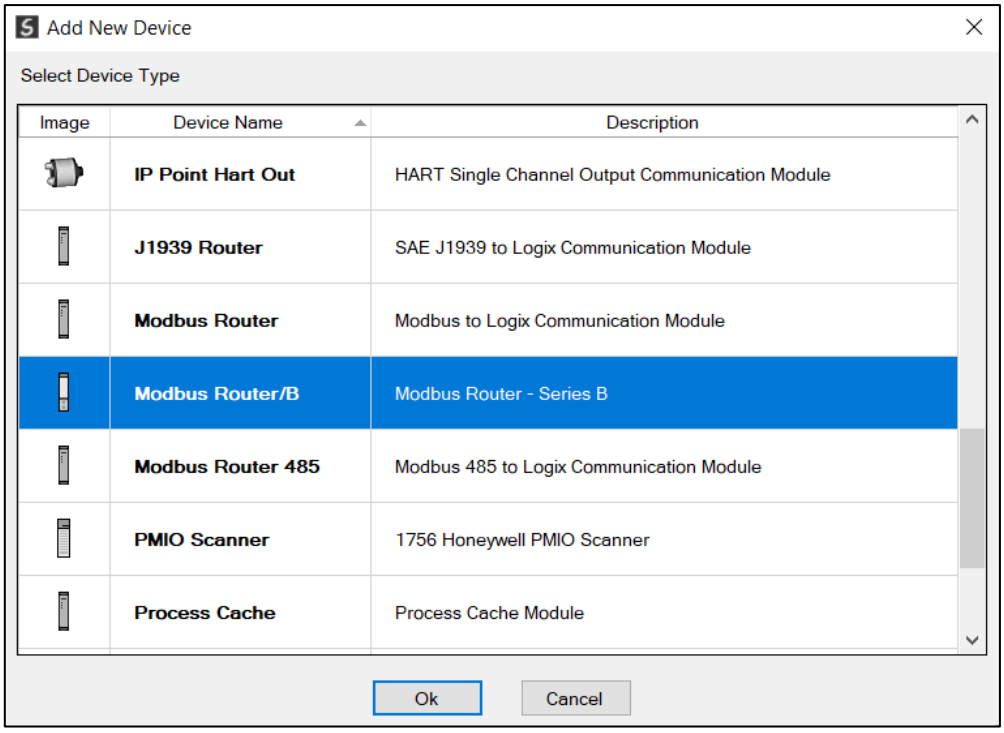


Figure 3.13 – Selecting a new Modbus Router.

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***.

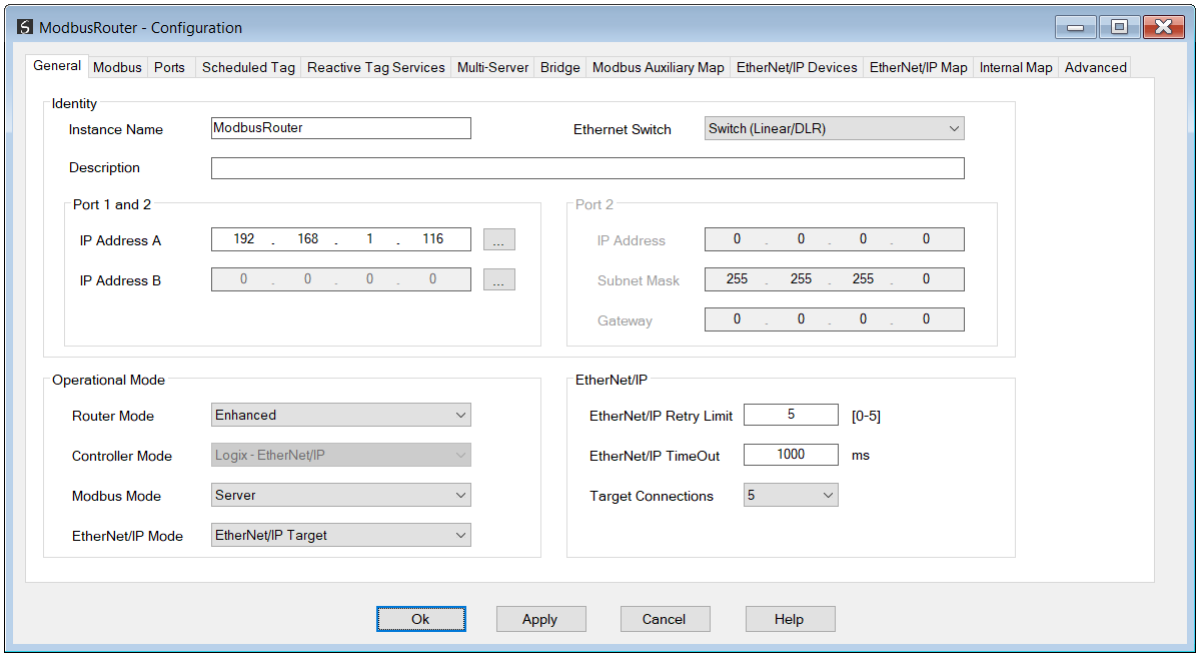


Figure 3.14 – Modbus Router configuration.

Refer to the additional information section in this document for Slate's installation and operation documentation.

3.4. GENERAL PARAMETERS

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

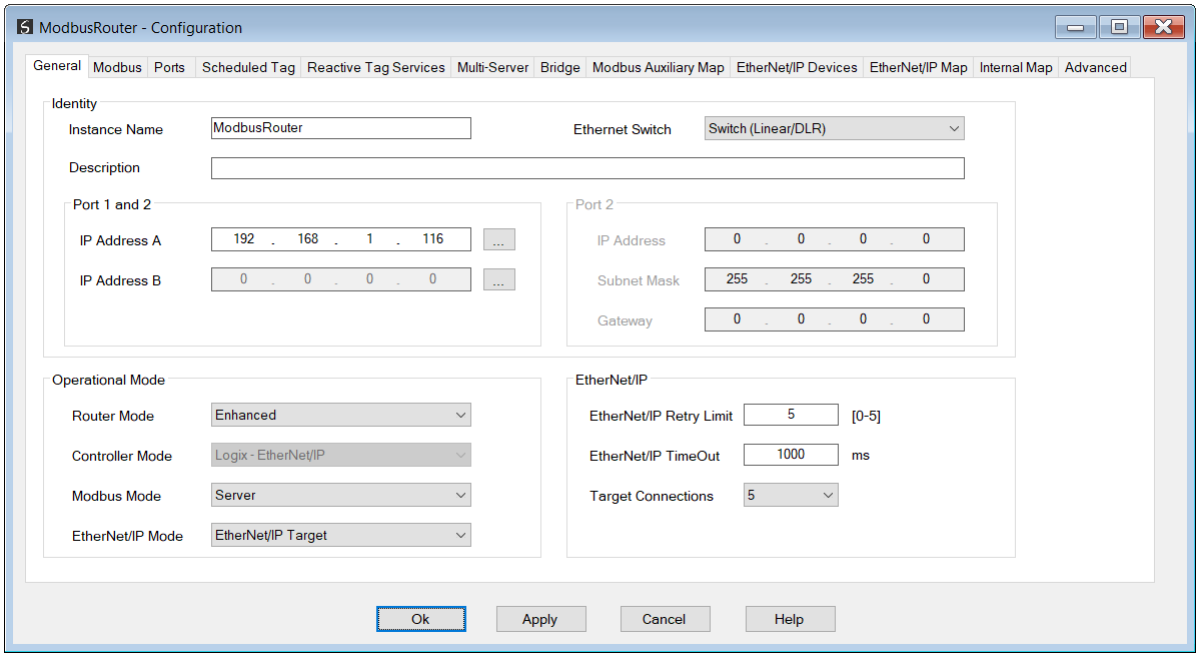


Figure 3.15 - General Configuration

The general configuration consists of the following parameters:

Parameter	Description
Instance Name	This parameter is a user defined name to identify between various Modbus Routers.
Description	This parameter is used to provide a more detailed description of the application for the module.
Ethernet Switch	The operational mode of the embedded Ethernet switch.

	Switch (Linear/DLR) This is the default mode where the embedded switch is enabled. The two Ethernet ports are identical in functionality and (foreign) traffic received on one port is transmitted out the other. Both ports use the IP address configured using DHCP or set the Target browser. **Split (Dual-IP)** In this mode the embedded switch is disabled. Traffic received on one port, is not transmitted to the other. Port 1: The IP address is the one configured using DHCP or via the Target browser. EtherNet/IP requests and connections will be processed. Modbus TCP requests will be processed. Port 2: The IP address is the one contained within the configuration below (Port 2). EtherNet/IP requests and connections will not be processed. Modbus TCP requests will be processed. **NOTE**: The Split (Dual-IP) functionality requires module firmware revision 2.001.012 or greater.
Port 1 IP Address A	When not operating in Logix USB Redundant mode, this is the IP address of the local Modbus Router. When in Logix USB Redundant mode this will be the IP address of Modbus Router A in the redundant setup.
Port 1 IP Address B	When not operating in Logix USB Redundant mode, this is not applicable. When in Logix USB Redundant mode this will be the IP address of Modbus Router B in the redundant setup.
Port 2 IP Address	**THIS IS RELEVANT ONLY IN SPLIT (DUAL-IP) ETHERNET SWITCH MODE** The IP address used on Port 2.
Port 2 Subnet Mask	**THIS IS RELEVANT ONLY IN SPLIT (DUAL-IP) ETHERNET SWITCH MODE** The Subnet Mask used on Port 2.
Port 2 Gateway	**THIS IS RELEVANT ONLY IN SPLIT (DUAL-IP) ETHERNET SWITCH MODE** The Default Gateway address used on Port 2. **NOTE**: If the IP address ranges for Port 1 and Port 2 overlap, then only one Port's Gateway address can be non-zero.
Router Mode	The mapping mode will determine how the Modbus messages are routed.

	Scheduled Tag mode In Scheduled Tag mode, the Modbus Router will initiate the exchange between the remote Modbus device and Logix. Either by reading data from a Modbus device and writing it into a preconfigured Logix tag or vice versa. Modbus communication in this mode is initiated by the Modbus Router. The *Controller Mode* selection will become available when this mode is selected. **Reactive Tag mode** In Reactive Tag mode the module will automatically route the Modbus message and function to the appropriate pre-configured Logix tag. Modbus communication in this mode is initiated by the remote Modbus device. The *Controller Mode* selection will become available when this mode is selected. **Unscheduled mode** In Unscheduled mode the Modbus Router routes Logix messages to the remote Modbus device and returns the result. There is little configuration required in the Modbus Router as the routing information is configured by Logix for each message transaction. Modbus communication in this mode is initiated by Logix. **Enhanced mode** In Enhanced mode, the user can decide if the MBR/B is operating as a Modbus Client or Server as well as if the MBR/B is operating as an EtherNet/IP Target or EtherNet/IP Originator. The *Modbus Mode* and *EtherNet/IP Mode* selection will become available when this mode is selected. **Bridge mode** In Bridge Mode, the MBR/B can operate as a transparent link for Modbus between different media (TCP, RTU232, and RTU485). This allows, for example, a Modbus client on Modbus TCP to communicate with multiple Modbus devices on RS232 or RS485 and vice versa. **Concentrator mode** In Concentrator Mode, the MBR/B will collect data from multiple Modbus devices and provide the consolidated data to one or more Modbus Client. The MBR/B will be a Modbus Server on one port, and a Modbus Client on the other ports. This will allow a Modbus SCADA to efficiently read the data from multiple Modbus devices, because the data from the various Modbus devices have been concentrated into the MBR/B.
Controller Mode	**THIS IS RELEVANT ONLY IN SCHEDULED AND REACTIVE MODES** This option will allow the user to communicate to either ControlLogix/CompactLogix or SLC/MicroLogix. When Logix has been selected the user can use either EtherNet/IP or USB. The Logix USB option also allows for Logix redundancy. The mapping section will provide more details regarding how to enter and use the target tag for each of the controller sets. **Logix – EtherNet/IP** ControlLogix or CompactLogix controllers using EtherNet/IP. **Logix – USB** ControlLogix or CompactLogix controllers using USB. **NOTE**: When used for Modbus Mapping communication the Modbus Router the USB cable must be plugged directly into the Logix Controller used. Plugging the USB cable into a routing device (e.g. EN2TR) will not work for Modbus Mapping. **Logix – USB Redundant** Redundant ControlLogix controllers using USB.

	NOTE: When used for Modbus Mapping communication the Modbus Router the USB cable must be plugged directly into the Logix Controller used. Plugging the USB cable into a routing device (e.g. EN2TR) will not work for Modbus Mapping. **MicroLogix/SLC Ethernet** MicroLogix or SLC controllers.
Modbus Mode	**THIS IS RELEVANT ONLY IN ENHANCED MODE** **Modbus Server** The data from the configured EtherNet/IP devices (using either Class 1 Assemblies, Explicit messaging, or Logix Direct-to-Tag reads/writes) will be written to, or read from, the module's internal Modbus Registers using the Internal Map. These registers can be accessed by a remote Modbus Client using either Modbus TCP, Modbus RTU232, Modbus RTU485, or any simultaneous combination of these. **Modbus Client** The data from the configured EtherNet/IP devices (using either Class 1 Assemblies, Explicit messaging, or Logix Direct-to-Tag reads/writes) will be written to, or read from, the module's internal Modbus Registers using the Internal Map. The Modbus Auxiliary Map can then be used to configure the Modbus data exchange between multiple remote Modbus Server devices and the module's internal Modbus registers. The Modbus communication can be via Modbus TCP, Modbus RTU232, Modbus RTU485, or any simultaneous combination of these.
EtherNet/IP Mode	**THIS IS RELEVANT ONLY IN ENHANCED MODE** **EtherNet/IP Target** As an EtherNet/IP target, the module can exchange the Class 1 data with devices on the Modbus network: • **EtherNet/IP Class 1 connection** Here a remote EtherNet/IP device (e.g. a Logix controller) establishes between 1 and 5 Class 1 connections to the module. Modbus data can be mapped into four separate input and output class 1 cyclic connections to the Logix controller (allowing up to 2KB input and 2KB output to be exchanged at the requested packet interval – RPI). **EtherNet/IP Originator** As an EtherNet/IP originator, the module can use one of two methods to read and write data to and from the Modbus network: • **EtherNet/IP Explicit Messaging** This allows the MBR/B to exchange data with up to 10 EtherNet/IP devices. The module can use either Class 3 or Unconnected Messaging (UCMM) to Get and Set data in the remote EtherNet/IP devices. • **Direct-To-Tag** This allows the MBR/B to exchange data with a Logix controller without the need to write any application code (e.g. ladder) in Studio 5000. The Modbus data is directly read from, or written to, Logix tags. **EtherNet/IP Class 1 connection**

	Modbus data can be mapped to a maximum of 10 EtherNet/IP devices using input and output class 1 cyclic connections. This will allow the MBR/B to “own” the EtherNet/IP target device and exchange Modbus data using the EtherNet/IP device’s input and output assemblies.
ENIP Retry Limit	The amount of EtherNet/IP retries the module will make once no response was received from the Logix Controller.
ENIP TimeOut	The time in milliseconds after which a retry is sent. Once the first retry is sent the next retry will be sent after the same amount of time. This will repeat until the ENIP Retry Limit is reached.
Target Connections	The number of EtherNet/IP Connections to be used in the exchange with Logix (1 to 5). The first connection is used for status information while connection 2 to 5 is used for Modbus data exchange (when operating in Enhanced mode). **NOTE**: This value must match that configured in the Logix IO tree

Table 3.1 - General configuration parameters

3.5. MODBUS PARAMETERS

The Modbus configuration is shown in the figure 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.* Once in the configuration window select the second tab at the top *Modbus*.

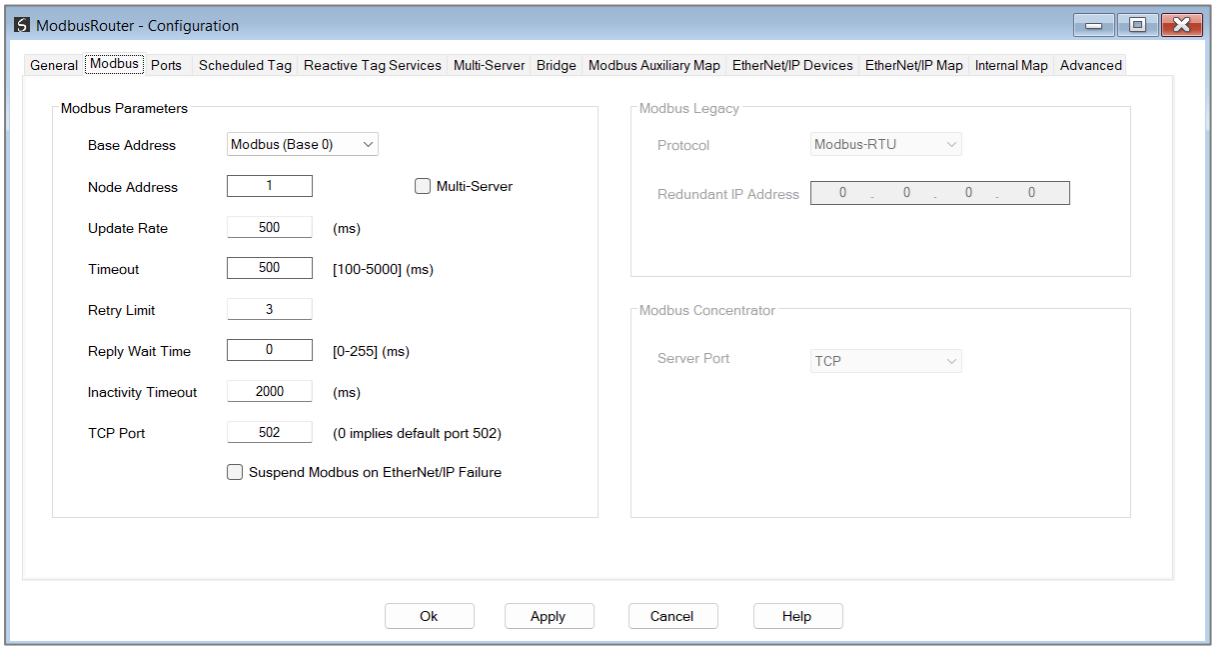


Figure 3.16 – Modbus Configuration

The Modbus configuration consists of the following parameters:

Parameter	Description
Modbus Parameters	
Base Address	The Modbus Router can be configured to operate in one of two Base Address modes: Modbus (Base 0) or PLC (Base 1). For example, when operating in Base 0 mode the first holding register address will be 40000 whilst in Base 1 mode the first holding register address will be 40001. Depending on the system being interfaced to the user will need to select the correct option. Failing to do this will result in the data being misaligned by one register.
Node Address	The Modbus node address of the Modbus Router.
Multi-Server	When this option is selected the user can select multiple nodes for the mapping items in the Reactive Tag Map or when the module is a Modbus Server in Enhanced mode. This will allow the Modbus Router to emulate multiple Modbus Node Addresses.
Update Rate	The period (in milliseconds) between Client requests (in the Modbus Auxiliary Map) to the target Modbus devices. **Only valid for Modbus Client when operating in Enhanced mode.**
Timeout	The time (in milliseconds) the module will wait for a valid Modbus response. **Only valid for Modbus Client when operating in Enhanced mode or Reactive Mode.**
Retry Limit	The number of successive Modbus request retries before the request is deemed to have failed. **Only valid for Modbus Client when operating in Enhanced mode.**
Reply Wait Time	The reply message wait is the minimum delay before the Modbus reply is transmitted to the Modbus device.
Inactivity Timeout	The amount of time during which no Modbus requests have been received before the MBR/B indicates that the connection to the Modbus Client is no longer active. **Only valid for Modbus Server when operating in Enhanced mode.**
TCP Port	The TCP port to be used for the Modbus-TCP communication. By default, the module will use the standard TCP port 502.
Suspend Modbus on EtherNet/IP Failure	When this option is set, the Modbus communication (when in Enhanced Mode) will be suspended when the EIP communication has failed (Target or Originator).
Modbus Legacy – Only Relevant for Scheduled, Reactive and Unscheduled Modes	
Protocol	The Modbus Router can interface to the Modbus device(s) on either serial (Modbus-RTU) or Ethernet (Modbus-TCP).
Redundant IP Address	When using Controller Mode *Logix – USB Redundant,* the user can configure an additional IP Address used **only** for Modbus TCP which will be "shared" by the partner Modbus Router. See the Logix Redundancy section for more information.
Modbus Concentrator – Only Relevant for Concentrator Mode	
Server Port	The Modbus port which will be communicating with the Modbus Client.

	Only valid for Concentrator Mode.

Table 3.2 – Modbus configuration parameters.

3.6. PORTS PARAMETERS

The Ports configuration is shown in the figure below. The Ports configuration window is opened by either double clicking on the module in the tree or right clicking the module and selecting *Configuration.* Once in the configuration window select the second tab at the top *Ports*.

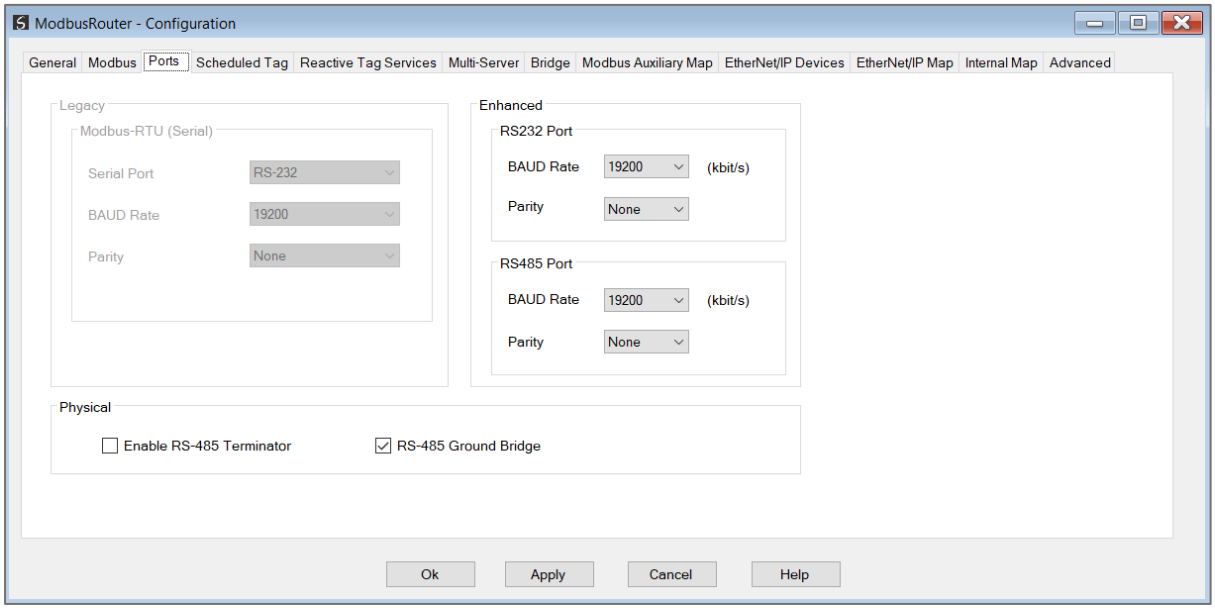


Figure 3.17 – Ports Configuration.

The Ports configuration consists of the following parameters:

Parameter	Description
Legacy – Only Relevant for Scheduled, Reactive and Unscheduled Modes	
Serial Port	The user can select either RS-232 or RS-485 for Modbus RTU.
BAUD Rate	The BAUD rate will configure at what speed the data is send across the RS232 or RS485 serial network. The module provides the following speeds: 1200, 2400, 4800, 9600, 19200, 38400, 57600, and 115200
Parity	The parity parameter will configure the parity of the module's RS232 or RS485 serial port. The module allows for Even, Odd, or None parity setting.
Enhanced – Only Relevant for Enhanced, Bridge, and Concentrator Modes	
RS232 Port – Baud Rate	The BAUD rate will configure at what speed the data is sent across the RS232 serial network. The module provides the following speeds:

	1200, 2400, 4800, 9600, 19200, 38400, 57600, and 115200
RS232 Port – Parity	The parity parameter will configure the parity of the module's RS232 serial port. The module allows for Even, Odd, or None parity setting.
RS485 Port – Baud Rate	The BAUD rate will configure at what speed the data is sent across the RS485 serial network. The module provides the following speeds: 1200, 2400, 4800, 9600, 19200, 38400, 57600, and 115200
RS485 Port – Parity	The parity parameter will configure the parity of the module's RS485 serial port. The module allows for Even, Odd, or None parity setting.
Physical	
Enable RS-485 Terminator	Enables the on-board 125Ω RS485 terminating resistor.
RS-485 Ground Bridge	Connects the isolated RS485 ground to the external Ground terminal. **NOTE:** When enabled, the isolated RS232 and RS485 grounds are connected.

Table 3.3 – Ports configuration parameters

NOTE: The Modbus Router supports 8 data bits and 1 stop bit.

3.7. SCHEDULED TAG MODE

The Scheduled Tag routing mode transfers data between Modbus devices and Logix controllers. Unlike the Reactive tag mode, the Modbus Router (when in the Scheduled Tag mode) initiates the messaging.

In this mode the Modbus Router transfers data between a Logix controller and Modbus devices without any configuration or programming required in either the Modbus device or the Logix controller. The data will be exchanged at a fixed interval which is configured in Slate.

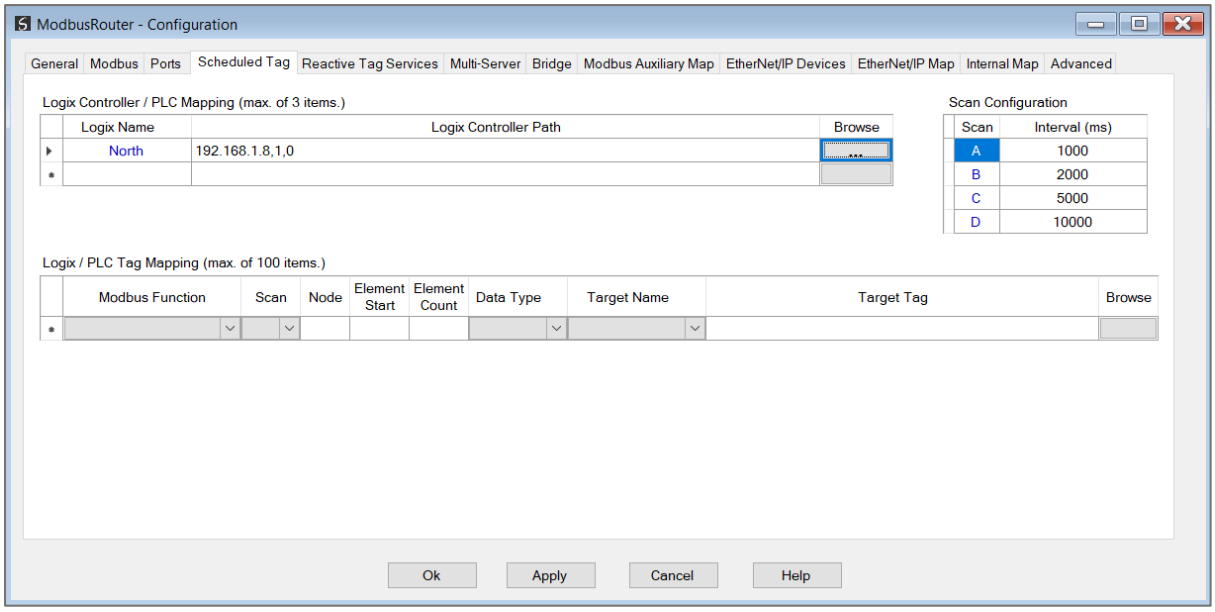


Figure 3.18 - Scheduled Tag configuration

The Schedule Tag mode is configured in three steps. First the user must create a Target Name (CIP path to the destination Logix controller) which will be used to link the Modbus function and register range to the destination Logix tag.

NOTE: When using USB to connect to the Logix Controller, the CIP path is **not** required for operation as the Modbus Router will be plugged directly into the Logix Controller.

3.7.1. Logix Controller Type

3.7.1.1. EtherNet/IP

The Logix controller paths 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.

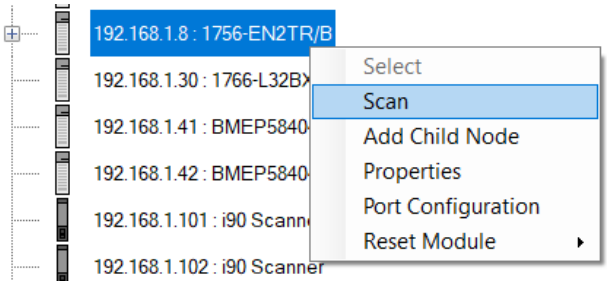


Figure 3.19 - Scanning node in the Target Browser.

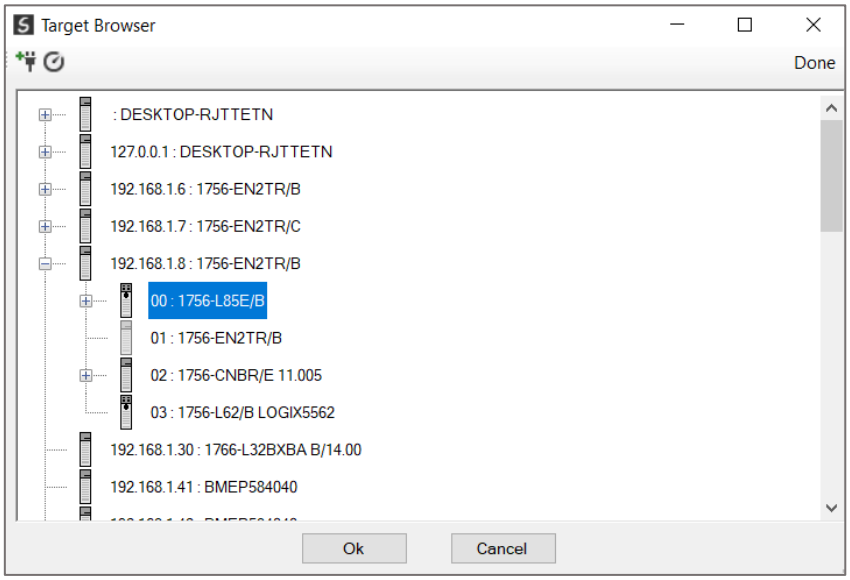


Figure 3.20 - Target Browser selection.

The required Logix controller can then be chosen by selecting it and clicking the Ok button, or by double-clicking on the controller module.

A maximum number of 3 controller mapping entries can be added.

The second part of the Scheduled Tag mode setup is to configure the scan intervals. The scan intervals allow different data items to be transferred at different rates. There are 4 scan classes, viz. A, B, C and D. The intervals for each can be adjusted by entering the scan time in milliseconds. The interval must be between 200 milliseconds and 60 seconds.

The third part of the Scheduled Tag mode setup is to configure the link between a Modbus function and register range combination to a Logix tag, and the associated action and scan required.

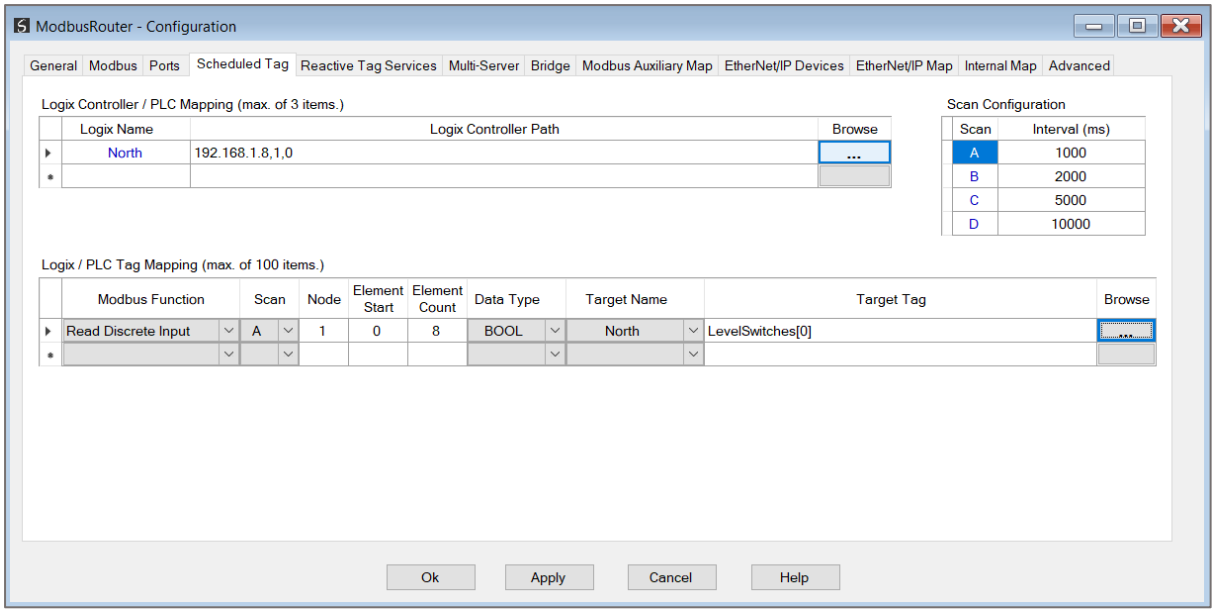


Figure 3.21 – Scheduled Tag Mapping.

The Modbus Function is used to determine the actual Modbus command to be sent to the device, as described in the table below:

Selected Function	Modbus Data Type	Single / Multiple	Modbus Function
Read Coil	Boolean	Both	01 - Read Coils
Read Discrete Input	Boolean	Both	02 - Read Discrete Inputs
Read Holding Register	Integer	Both	03 - Read Holding Registers
Read Input Register	Integer	Both	04 - Read Input Registers
Write Coil	Boolean	Single	05 - Write Single Coil
		Multiple	15 - Write Multiple Coils
Write Register	Integer	Single	06 - Write Single Register
		Multiple	16 - Write Multiple Registers

Table 3.4 - Modbus Function Mapping

The Scan field specifies at what rate the transaction will be executed. Select a scan class letter that matches the required interval. Care must be taken to select a realistic scan interval, taking into account the configured Baud rate and message size.

The IP address and Node is the remote device's IP address and node number.

NOTE: The IP address is only relevant (and visible) if the Modbus-TCP protocol is selected. The Node address is relevant for both Modbus protocols.

The range of data to be accessed from the remote device must also be specified. This is achieved by entering the *Register Start* and *Element Count* values.

Modbus read and write functions involve the transfer of either Booleans or integers (16 bit). A part or combination of integers may represent other types. For this reason, the Modbus Router allows the mapping of integer-based functions to the following Logix data types:

- SINT
- INT
- DINT
- REAL.

All discrete or Boolean based functions need to be mapped to a Logix BOOL array.

NOTE: When mapping a range of registers to a Logix array, it is important to ensure that the destination Logix array is sufficiently large to accommodate all the data.

NOTE: When mapping a range of registers to a Logix DINT or REAL array, the Register Start must be an even number, else the integer data will not be aligned with the 32-bit destination Logix tag.

NOTE: When writing to a DINT or REAL datatype the user cannot write a single Modbus element (16-bit). At least two elements will need to be written to either of the above 32-bit datatypes (DINT or REAL).

NOTE: When doing a single/multiple coil/discrete write to a Logix Bool array datatype the remaining bits of the 32-bit aligned Bool array will be overwritten.

One of the Target Names configured in the first step can be selected by means of the target Name combo box.

The Target Tag can be either entered manually or selected using the Tag Browser in Slate. The Tag Browser requires the controller to be available on the network.

To browse to the tag, click on the Browse button. The Tag Browser will open and scan all the tags inside that controller. If the controller has been recently scanned in this Slate session, then a cached version of the tags will be displayed. A rescan of the tags can be triggered by selecting the Refresh button in the Tag Browser's toolbar.

Only tags of a relevant type will be enabled, guiding the user to select a suitable tag.

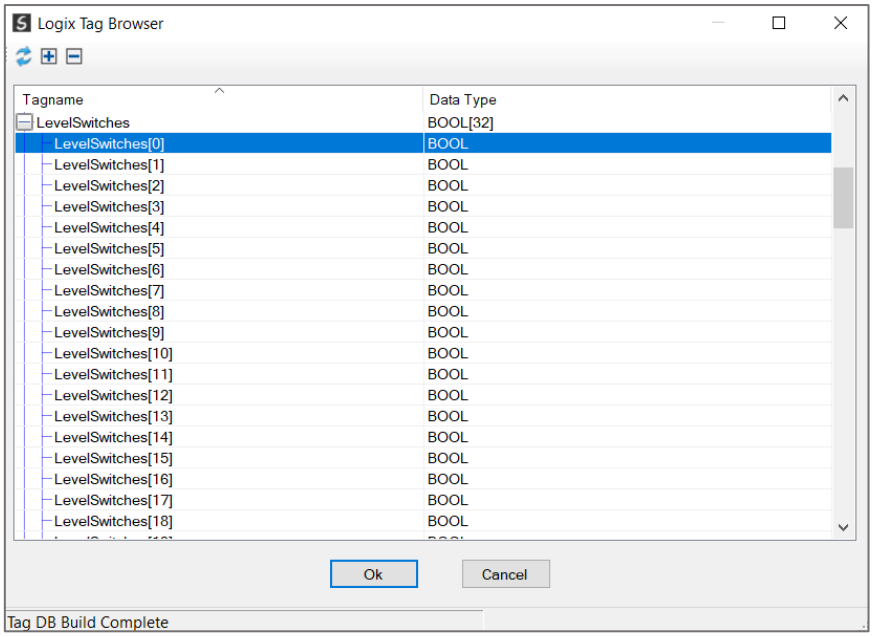


Figure 3.22 – Tag Browser tag selection.

3.7.1.2. USB

NOTE: The Modbus Router can communicate with only **one** Logix Controller.

NOTE: The Modbus Router **must** be directly connected to the Logix controller (via USB) with which it will exchange data.

NOTE: The user **does not** need to enter the Logix Controller path in the mapping, because the Modbus Router is directly connected to the Logix Controller (via USB). The Logix Controller Path will **only** be used for browsing the tags of the Logix Controller when connected on the same network.

NOTE: The user will need to ensure that if a Controller is being scanned for tags, that the CIP Path used is to the same Logix Controller directly connected to the Modbus Router (via USB)

The Logix controller path (used for scanning the Logix Controller tags) can either be entered manually, left blank (it the tags are manually being entered), 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.

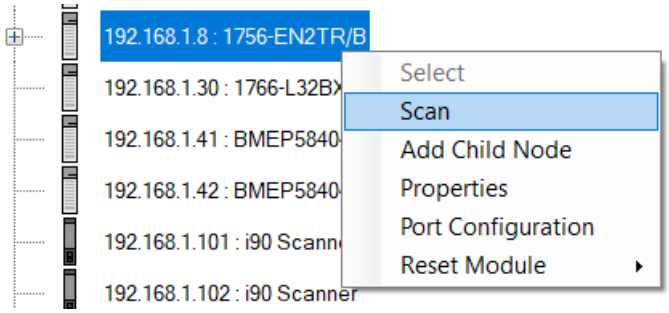


Figure 3.23 - Scanning node in the Target Browser.

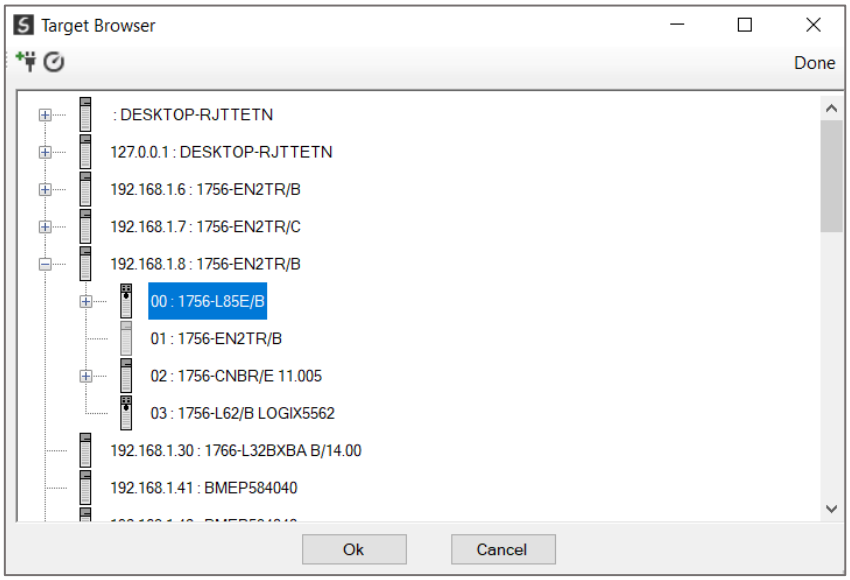


Figure 3.24 - Target Browser selection

The required Logix controller can then be chosen by selecting it and clicking the Ok button, or by double-clicking on the controller module.

The second part of the Scheduled Tag mode setup is to configure the scan intervals. The scan intervals allow different data items to be transferred at different rates. There are 4 scan classes, viz. A, B, C and D. The intervals for each can be adjusted by entering the scan time in milliseconds. The interval must be between 200 milliseconds and 60 seconds.

The third part of the Scheduled Tag mode setup is to configure the link between a Modbus function and register range combination to a Logix tag, and the associated action and scan required.

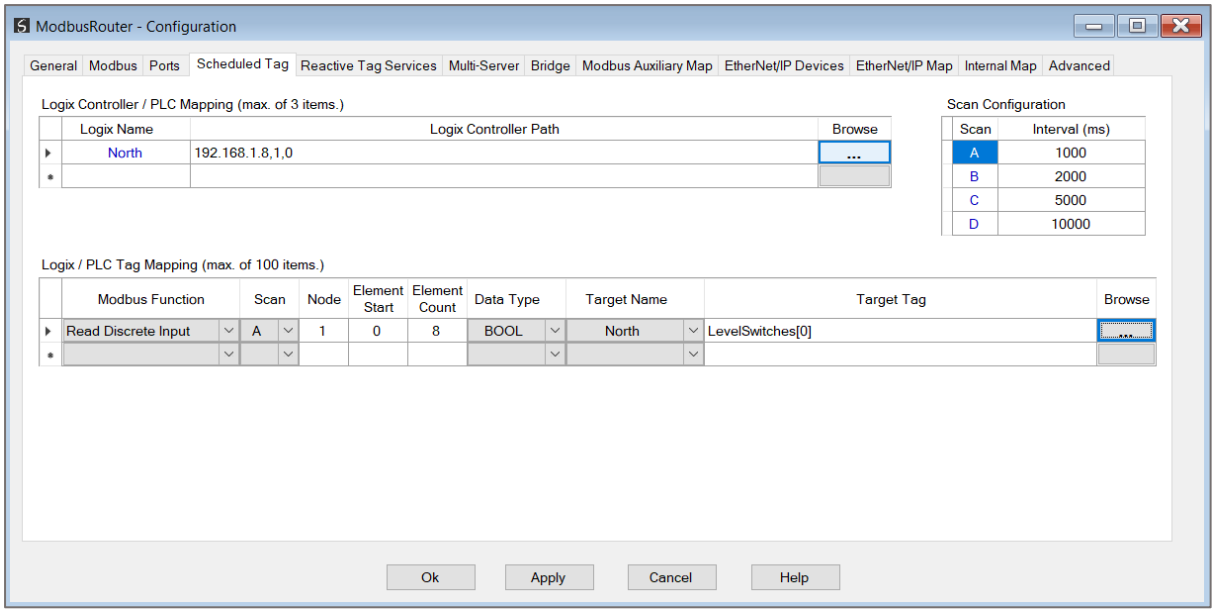


Figure 3.25 – Scheduled Tag Mapping.

The Modbus Function is used to determine the actual Modbus command to be sent to the device, as described in the table below:

Selected Function	Modbus Data Type	Single / Multiple	Modbus Function
Read Coil	Boolean	Both	01 - Read Coils
Read Discrete Input	Boolean	Both	02 - Read Discrete Inputs
Read Holding Register	Integer	Both	03 - Read Holding Registers
Read Input Register	Integer	Both	04 - Read Input Registers
Write Coil	Boolean	Single	05 - Write Single Coil
		Multiple	15 - Write Multiple Coils
Write Register	Integer	Single	06 - Write Single Register
		Multiple	16 - Write Multiple Registers

Table 3.5 - Modbus Function Mapping

The Scan field specifies at what rate the transaction will be executed. Select a scan class letter that matches the required interval. Care must be taken to select a realistic scan interval, taking into account the configured Baud rate and message size.

The IP address and Node is the remote device's IP address and node number.

NOTE: The IP address is only relevant (and visible) if the Modbus-TCP protocol is selected. The Node address is relevant for both Modbus protocols.

The range of data to be accessed from the remote device must also be specified. This is achieved by entering the *Register Start* and *Element Count* values.

Modbus read and write functions involve the transfer of either Booleans or integers (16 bit). A part or combination of integers may represent other types. For this reason, the Modbus Router allows the mapping of integer-based functions to the following Logix data types:

- SINT
- INT
- DINT
- REAL.

All discrete or Boolean based function need to be mapped to a Logix BOOL array.

NOTE: When mapping a range of registers to a Logix array, it is important to ensure that the destination Logix array is sufficiently large to accommodate all the data.

NOTE: When mapping a range of registers to a Logix DINT or REAL array, the Register Start must be an even number, else the integer data will not be aligned with the 32-bit destination Logix tag.

NOTE: When writing to a DINT or REAL datatype the user cannot write a single Modbus element (16-bit). At least two elements will need to be written to either of the above 32-bit datatypes (DINT or REAL).

NOTE: When doing a single/multiple coil/discrete write to a Logix Bool array datatype the remaining bits of the 32-bit aligned Bool array will be overwritten.

One of the Target Names configured in the first step can be selected by means of the target Name combo box.

NOTE: The target CIP Path is only used for browsing to the Logix Controller from Slate when using USB, because the Modbus Router is already directly connected to the Logix Controller (via USB).

The Target Tag can be either entered manually or selected using the Tag Browser in Slate. The Tag Browser requires the controller to be available on the network.

To browse to the tag, click on the Browse button. The Tag Browser will open and scan all the tags inside that controller. If the controller has been recently scanned in this Slate session, then a cached version of the tags will be displayed. A rescan of the tags can be triggered by selecting the Refresh button in the Tag Browser's toolbar.

Only tags of a relevant type will be enabled, guiding the user to select a suitable tag.

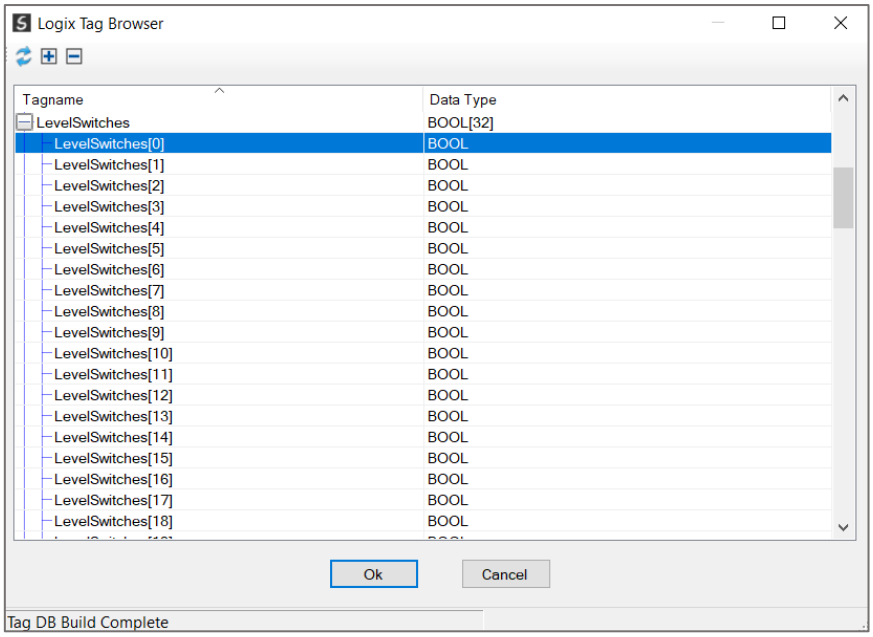


Figure 3.26 – Tag Browser tag selection.

3.7.2. MicroLogix/SLC Controller Type

The MicroLogix/SLC controller paths 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.

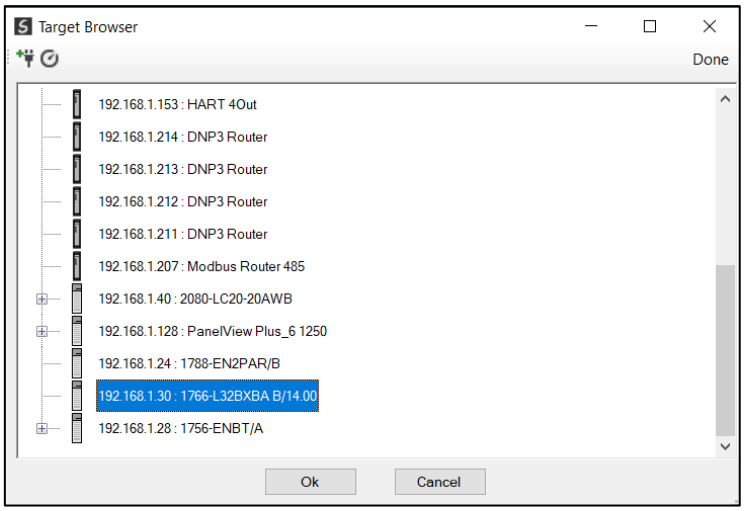


Figure 3.27 - Target Browser selection.

The required MicroLogix/SLC controller can then be chosen by selecting it and clicking the Ok button, or by double-clicking on the controller module. A maximum number of 3 controller mapping entries can be added.

The second part of the Scheduled Tag mode is to configure the link between a Modbus function and register range to a MicroLogix/SLC File number. This will allow the Modbus message initiator to effectively write to, or read from, a MicroLogix/SLC File using traditional Modbus functions.

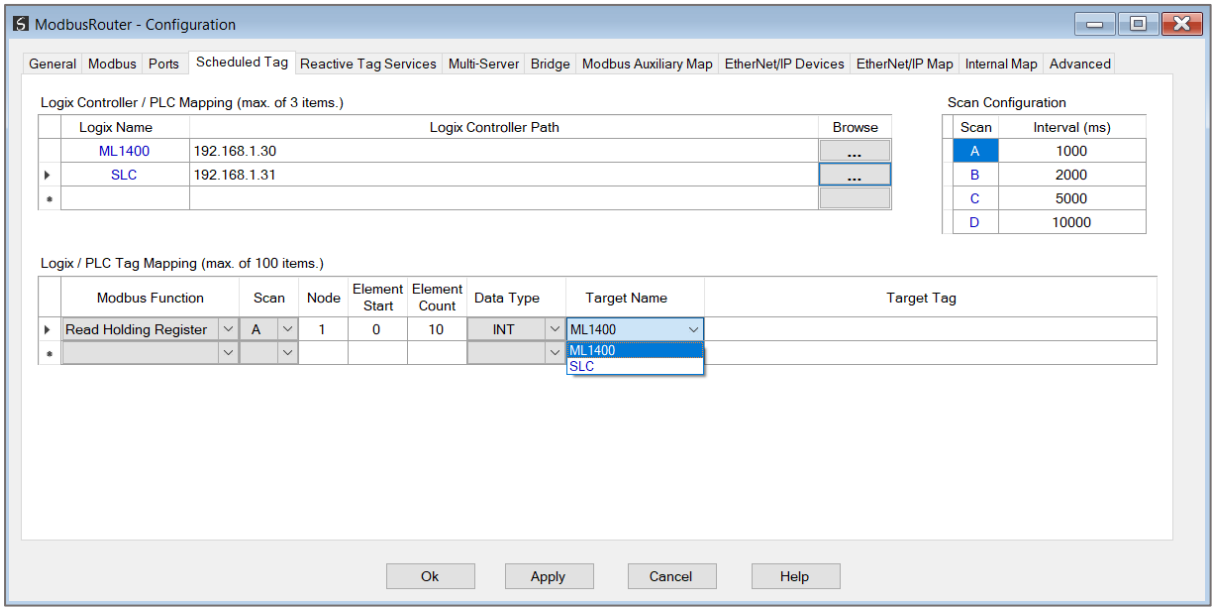


Figure 3.28 – Scheduled Tag Mapping (MicroLogix/SLC Controller Type).

Modbus read and write functions involve the transfer of either Booleans or integers (16 bit). A part or combination of integers may represent other types. For this reason, the Modbus Router allows the mapping of integer-based functions to the following data types when using the MicroLogix/SLC controller:

- BOOL
- INT
- REAL

All discrete or Boolean based function need to be mapped to a BOOL array (e.g. B3:0).

NOTE: When mapping a range of registers to a File array, it is important to ensure that the destination File array is sufficiently large to accommodate all the data.

NOTE: When mapping a range of registers to a Float File array, the Register Start must be an even number, else the integer data will not be aligned with the 32-bit destination Float File.

NOTE: When writing to a REAL datatype the user cannot write a single Modbus element (16-bit). At least two elements will need to be written a float datatype.

NOTE: When doing a single/multiple coil/discrete write to a MicroLogix/SLC Bool array datatype the remaining bits of the 32-bit aligned Bool array will be overwritten.

Next the range of Modbus data to be accessed must be specified. This is achieved by selecting a *Register Start* and *Element Count*.

If the Modbus *Element Count* is 1, it is possible to map a single integer register to a single integer tag. All other combinations will require a File array to be selected.

Below is an example of the target File selection. The File must be entered manually by typing in the needed File (e.g. N7:20).

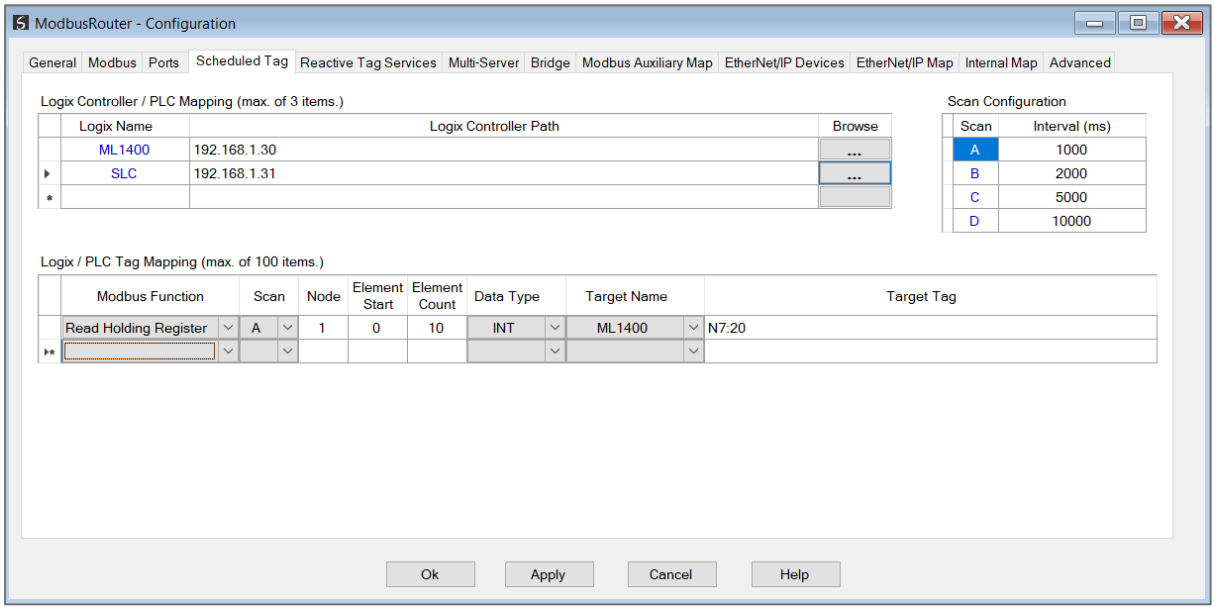


Figure 3.29 – File Selection for MicroLogix/SLC Controller Type.

The message routing executes in the exact same way as the Logix Controller type in section 3.7.1.

3.8. REACTIVE TAG MODE

The Reactive Tag routing mode allows mapping of virtual Modbus registers to Logix tags across multiple controllers. In this mode the Modbus Router will redirect a Modbus message to a Logix controller at a preconfigured path.

Thus, the routing of Modbus read and write register requests is managed by the Modbus Router and converted to direct Logix tag read and write functions.

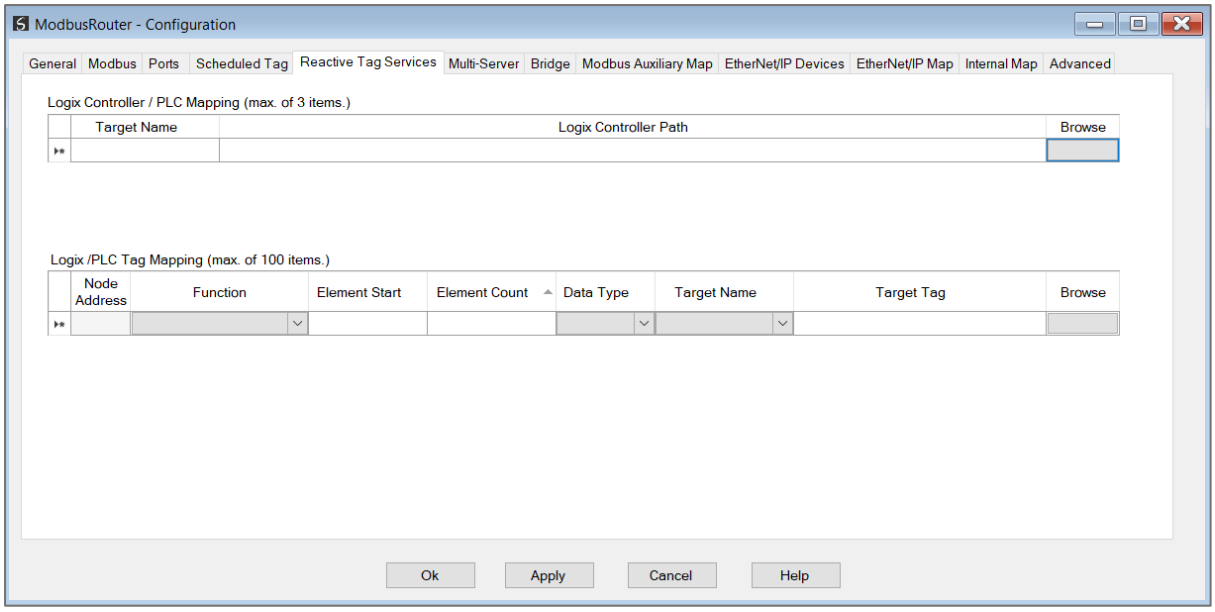


Figure 3.30 - Reactive Tag mode configuration.

The Reactive Tag mode is configured in two steps. First the user must create a Target Name (CIP path to the destination Logix controller) which will be used to link the Modbus function and register selection to the destination Logix tag.

3.8.1.1. LOGIX CONTROLLER TYPE

3.8.1.1.1. ETHERNET/IP

The Logix controller paths 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.

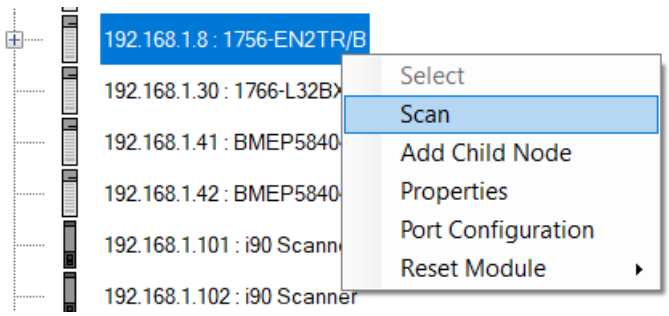


Figure 3.31 - Scanning node in the Target Browser.

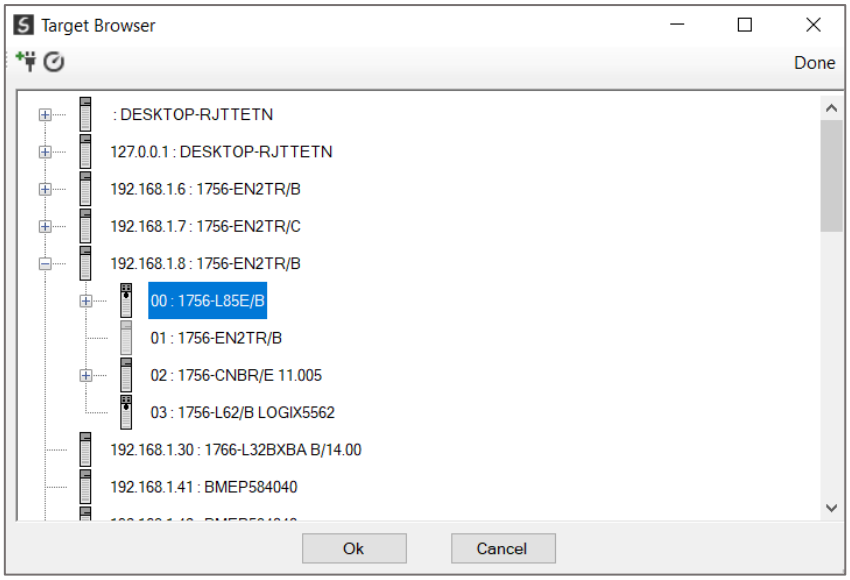


Figure 3.32 - Target Browser selection.

The required Logix controller can then be chosen by selecting it and clicking the Ok button, or by double-clicking on the controller module.

A maximum number of 3 controller mapping entries can be added.

The second part of the Reactive Tag mode is to configure the link between a Modbus function and register range to a Logix tag. This will allow the Modbus message initiator to effectively write to, or read from, a Logix tag using traditional Modbus functions.

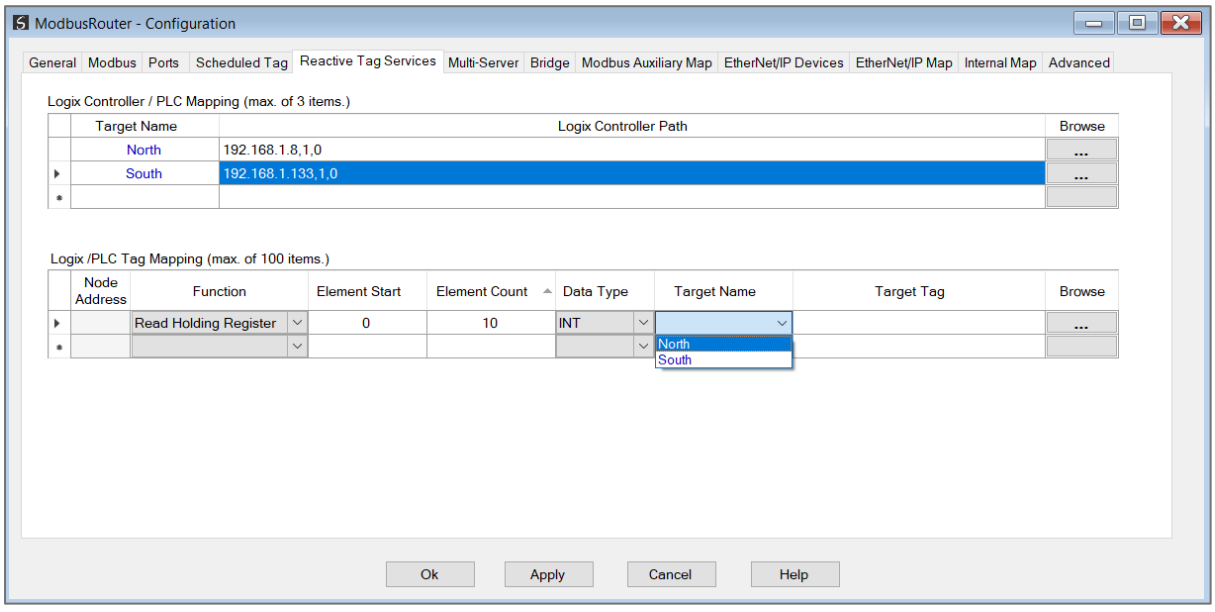


Figure 3.33 – Reactive Tag Mapping (Logix Controller Type)

Modbus read and write functions involve the transfer of either Booleans or integers (16 bit). A part or combination of integers may represent other types. For this reason, the Modbus Router allows the mapping of integer-based functions to the following Logix data types when using a ControlLogix or CompactLogix controller:

- SINT
- INT
- DINT
- REAL

All discrete or Boolean based function need to be mapped to a Logix BOOL array.

NOTE: When mapping a range of registers to a Logix array, it is important to ensure that the destination Logix array is sufficiently large to accommodate all the data.

NOTE: When mapping a range of registers to a Logix DINT or REAL array, the Register Start must be an even number, else the integer data will not be aligned with the 32-bit destination Logix tag.

NOTE: When writing to a DINT or REAL datatype the user cannot write a single Modbus element (16-bit). At least two elements will need to be written to either of the above 32-bit datatypes (DINT or REAL).

NOTE: When doing a single/multiple coil/discrete write to a Logix Bool array datatype the remaining bits of the 32-bit aligned Bool array will be overwritten.

Next the range of Modbus data to be accessed must be specified. This is achieved by selecting a *Register Start* and *Element Count*.

If the Modbus *Element Count* is 1, it is possible to map a single integer register to a single integer tag. All other combinations will require a Logix tag array to be selected.

Below is an example of the target tag selection. The Target Tag can be either entered manually or selected using the Tag Browser in Slate. The Tag Browser requires the controller to be available on the network.

To browse to the tag, click on the Browse button. The Tag Browser will open and scan all the tags inside that controller. If the controller has been recently scanned in this Slate session, then a cached version of the tags will be displayed. A rescan of the tags can be triggered by selecting the Refresh button in the Tag Browser's toolbar.

Only tags of a relevant type will be enabled, guiding the user to select a suitable tag.

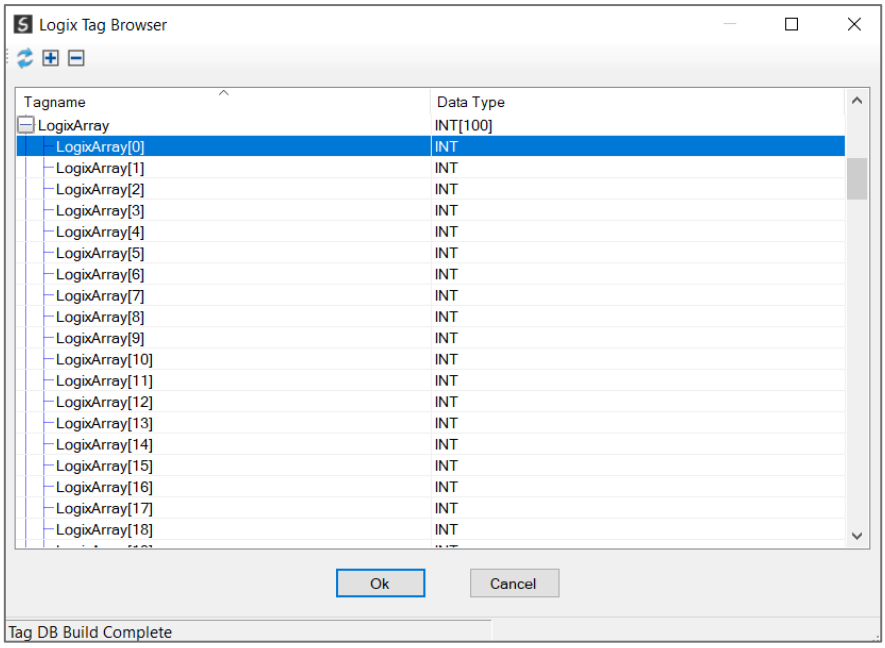


Figure 3.34 – Tag Browser tag selection.

The figure below is an example of how Modbus messages are routed to the Logix tags using the Reactive Tag Map mode.

NOTE: It is the user's responsibility to ensure that the Logix tag data type and size matches that of the Modbus message requests. Failing to do this can cause unexpected results and communication faults.

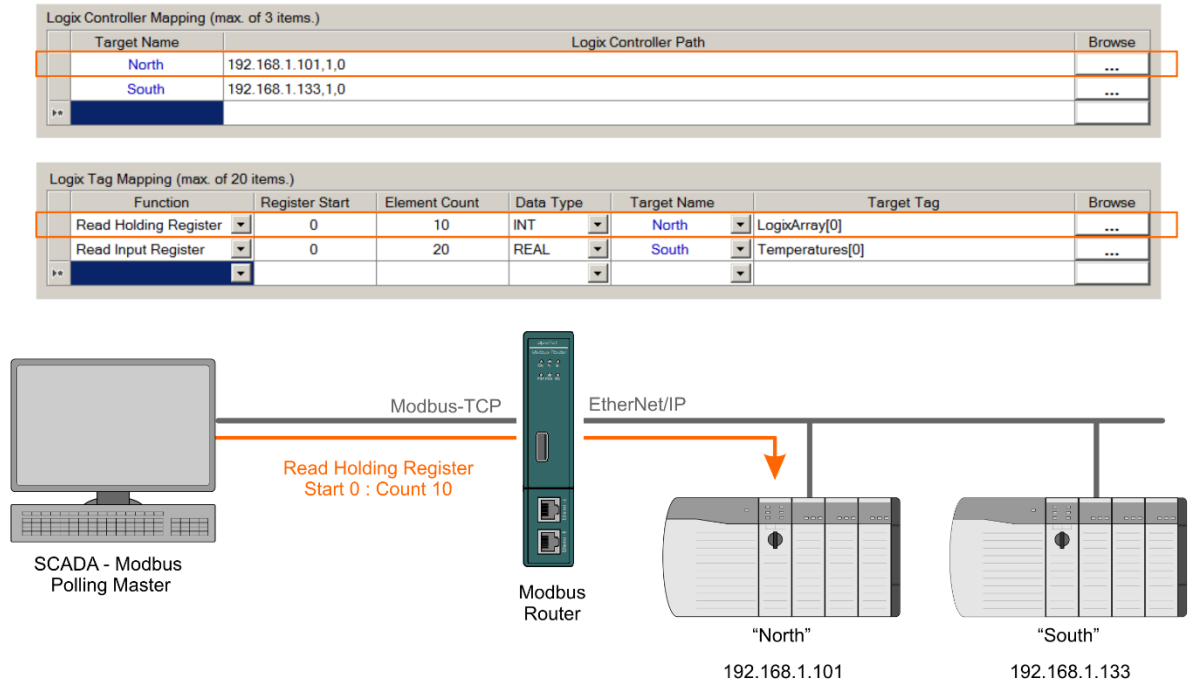


Figure 3.35 - Reactive Tag mode configuration in Slate (example route 1).

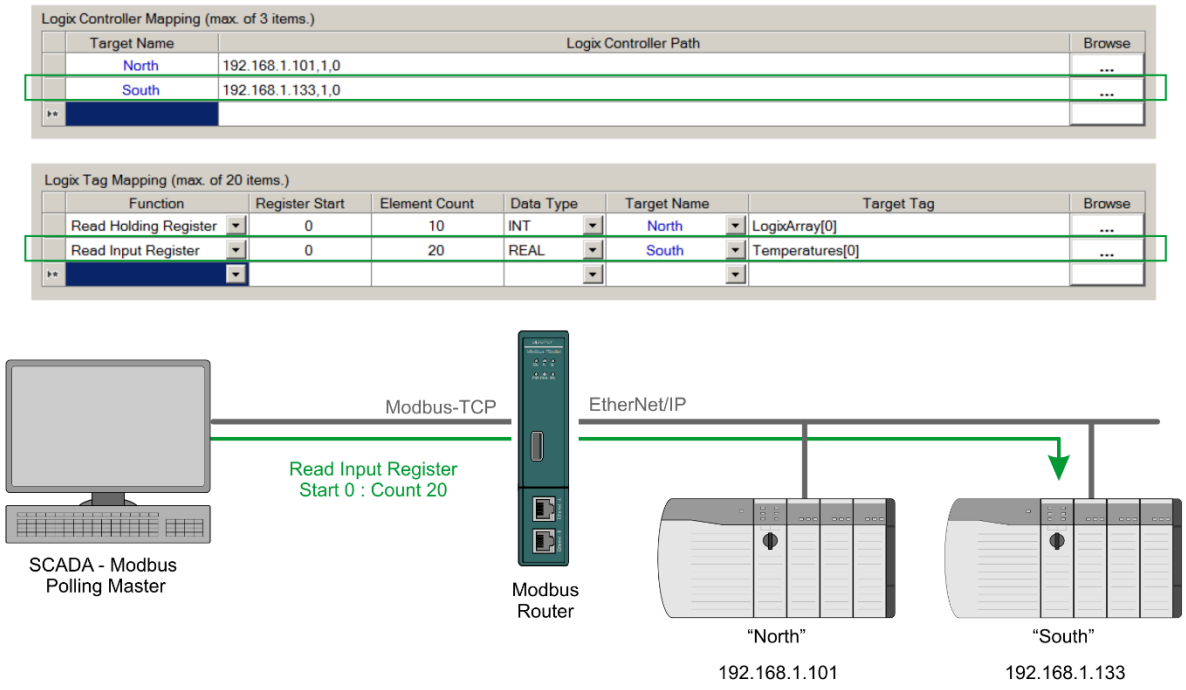


Figure 3.36 - Reactive Tag mode configuration in Slate (example route 2).

3.8.1.1.2. USB

NOTE: The Modbus Router can communicate only to a **single** Logix Controller.

NOTE: The Modbus Router **must** be directly connected to the Logix controller (via USB) with which it will exchange data.

NOTE: The user **does not** need to enter the Logix Controller path in the mapping, because the Modbus Router is directly connected to the Logix Controller (via USB). The Logix Controller Path will **only** be used for browsing the tags of the Logix Controller when connected on the same network.

NOTE: The user will need to ensure that if a Controller is being scanned for tags, that the CIP Path entered is to the same Logix Controller directly connected to the Modbus Router (via USB)

The Logix controller path (used for scanning the Logix Controller tags) can either be entered manually, left blank (it the tags are manually being entered), 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.

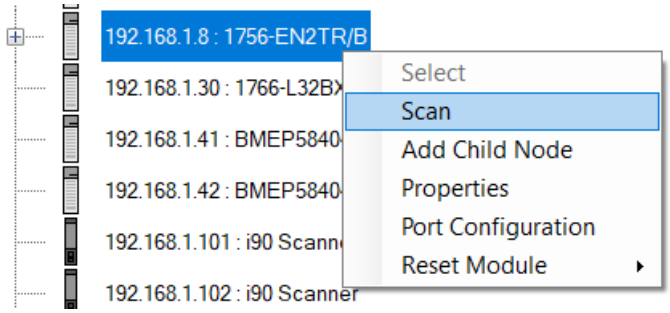


Figure 3.37 - Scanning node in the Target Browser.

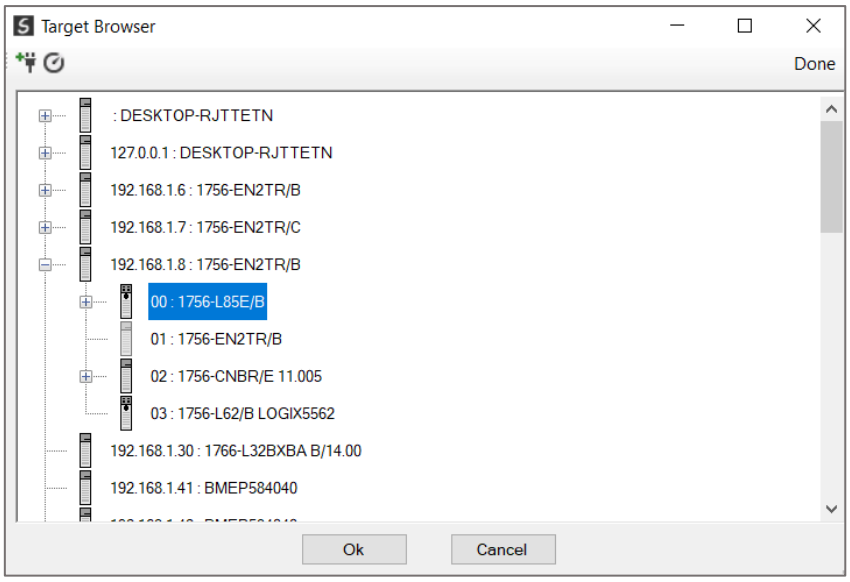


Figure 3.38 - Target Browser selection.

The required Logix controller can then be chosen by selecting it and clicking the Ok button, or by double-clicking on the controller module.

The second part of the Reactive Tag mode is to configure the link between a Modbus function and register range to a Logix tag. This will allow the Modbus message initiator to effectively write to, or read from, a Logix tag using traditional Modbus functions.

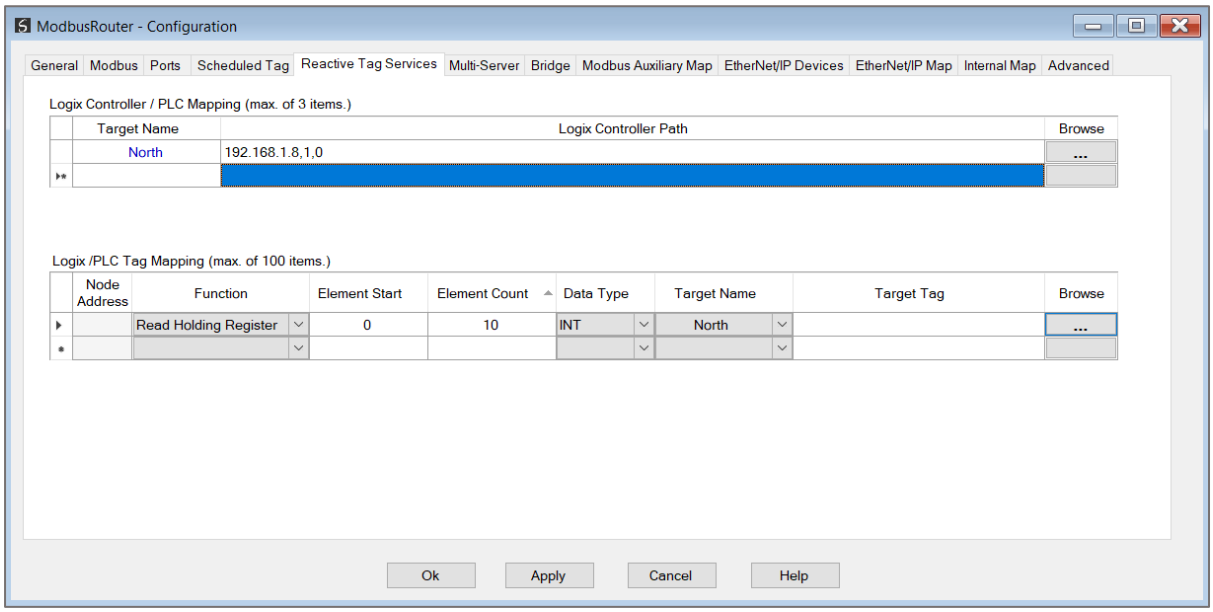


Figure 3.39 – Reactive Tag Mapping (Logix Controller Type).

Modbus read and write functions involve the transfer of either Booleans or integers (16 bit). A part or combination of integers may represent other types. For this reason, the Modbus Router allows the mapping of integer-based functions to the following Logix data types when using a ControlLogix or CompactLogix controller:

- SINT
- INT
- DINT
- REAL

All discrete or Boolean based function need to be mapped to a Logix BOOL array.

NOTE: When mapping a range of registers to a Logix array, it is important to ensure that the destination Logix array is sufficiently large to accommodate all the data.

NOTE: When mapping a range of registers to a Logix DINT or REAL array, the Register Start must be an even number, else the integer data will not be aligned with the 32-bit destination Logix tag.

NOTE: When writing to a DINT or REAL datatype the user cannot write a single Modbus element (16-bit). At least two elements will need to be written to either of the above 32-bit datatypes (DINT or REAL).

NOTE: When doing a single/multiple coil/discrete write to a Logix Bool array datatype the remaining bits of the 32-bit aligned Bool array will be overwritten.

Next the range of Modbus data to be accessed must be specified. This is achieved by selecting a *Register Start* and *Element Count*.

If the Modbus *Element Count* is 1, it is possible to map a single integer register to a single integer tag. All other combinations will require a Logix tag array to be selected.

Below is an example of the target tag selection. The Target Tag can be either entered manually or selected using the Tag Browser in Slate. The Tag Browser requires the controller to be available on the network.

To browse to the tag, click on the Browse button. The Tag Browser will open and scan all the tags inside that controller. If the controller has been recently scanned in this Slate session, then a cached version of the tags will be displayed. A rescan of the tags can be triggered by selecting the Refresh button in the Tag Browser's toolbar.

Only tags of a relevant type will be enabled, guiding the user to select a suitable tag.

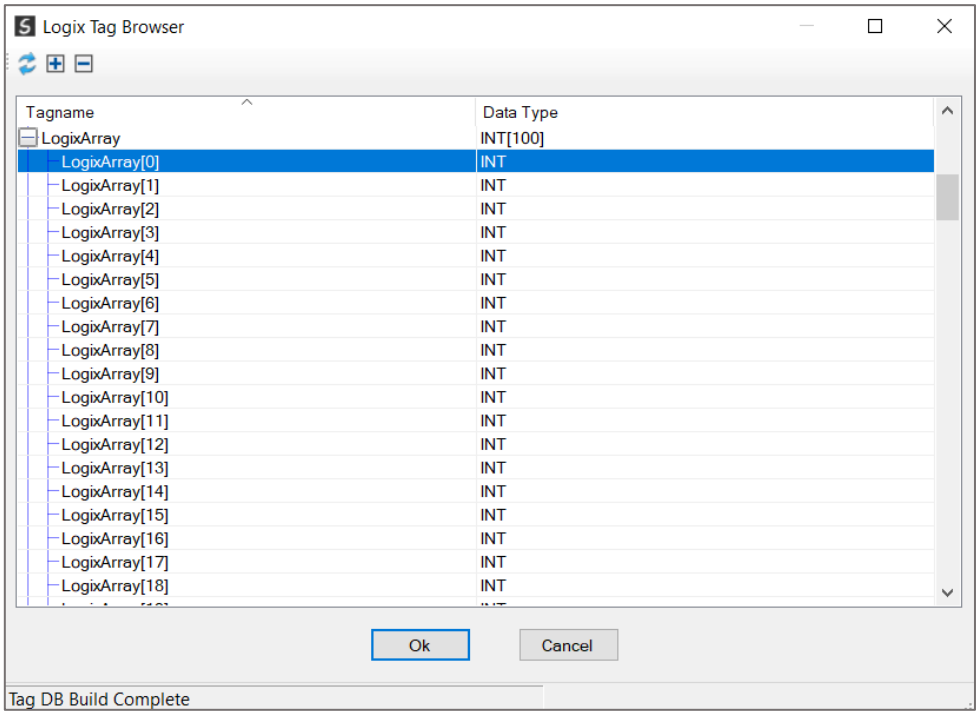


Figure 3.40 – Tag Browser tag selection.

The figure below is an example of how Modbus messages are routed to the Logix tags using the Reactive Tag Map mode.

NOTE: It is the user's responsibility to ensure that the Logix tag data type and size matches that of the Modbus message requests. Failing to do this can cause unexpected results and communication faults.

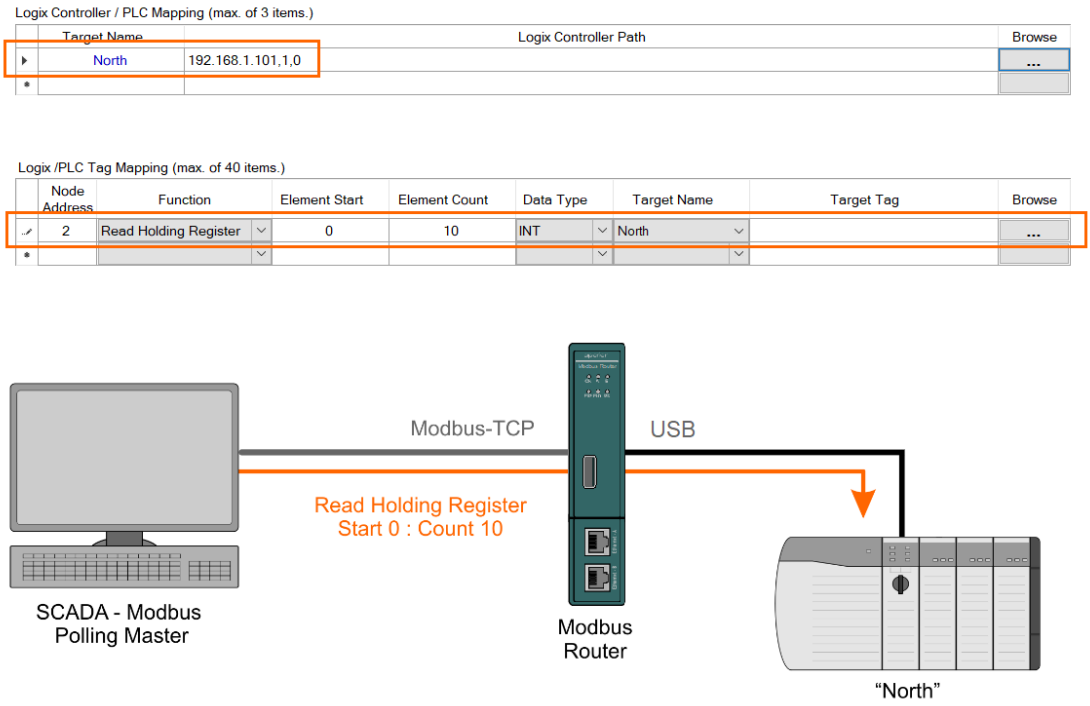


Figure 3.41 - Reactive Tag mode configuration in Slate (via USB).

3.8.1.2. MicroLogix/SLC Controller Type

The MicroLogix/SLC controller paths 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.

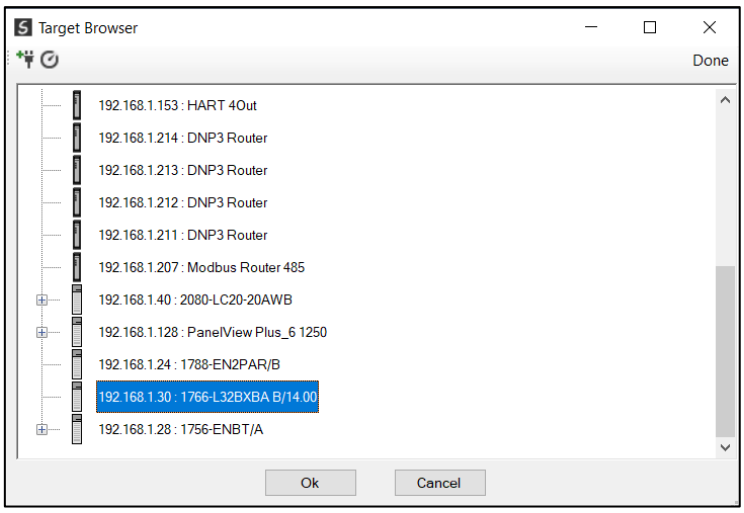


Figure 3.42 - Target Browser selection.

The required MicroLogix/SLC controller can then be chosen by selecting it and clicking the Ok button, or by double-clicking on the controller module. A maximum number of 3 controller mapping entries can be added.

The second part of the Reactive Tag mode is to configure the link between a Modbus function and register range to a MicroLogix/SLC File number. This will allow the Modbus message initiator to effectively write to, or read from, a MicroLogix/SLC File using traditional Modbus functions.

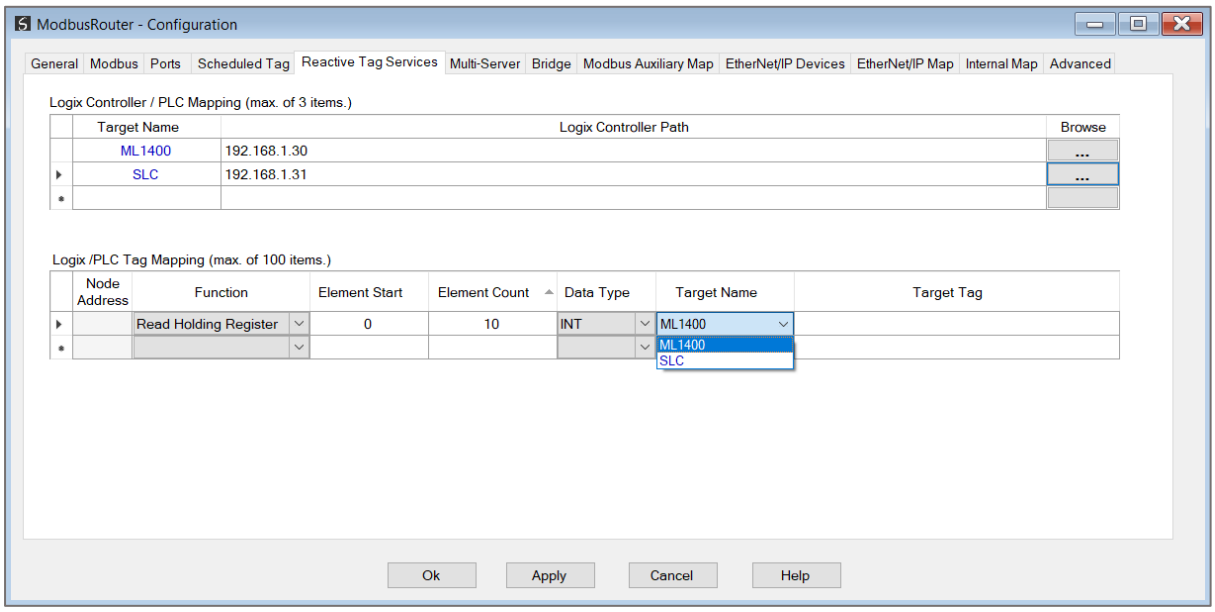


Figure 3.43 – Reactive Tag Mapping (MicroLogix/SLC Controller Type).

Modbus read and write functions involve the transfer of either Booleans or integers (16 bit). A part or combination of integers may represent other types. For this reason, the Modbus Router allows the mapping of integer-based functions to the following data types when using the MicroLogix/SLC controller:

- BOOL
- INT
- REAL

All discrete or Boolean based function need to be mapped to a BOOL array (e.g. B3:0).

NOTE: When mapping a range of registers to a File array, it is important to ensure that the destination File array is sufficiently large to accommodate all the data.

NOTE: When mapping a range of registers to a Float File array, the Register Start must be an even number, else the integer data will not be aligned with the 32-bit destination Float File.

NOTE: When writing to a REAL datatype the user cannot write a single Modbus element (16-bit). At least two elements will need to be written a float datatype.

NOTE: When doing a single/multiple coil/discrete write to a MicroLogix/SLC Bool array datatype the remaining bits of the 32-bit aligned Bool array will be overwritten.

Next the range of Modbus data to be accessed must be specified. This is achieved by selecting a *Register Start* and *Element Count*.

If the Modbus *Element Count* is 1, it is possible to map a single integer register to a single integer tag. All other combinations will require a File array to be selected.

Below is an example of the target File selection. The File must be entered manually by typing in the needed File (e.g. N7:20).

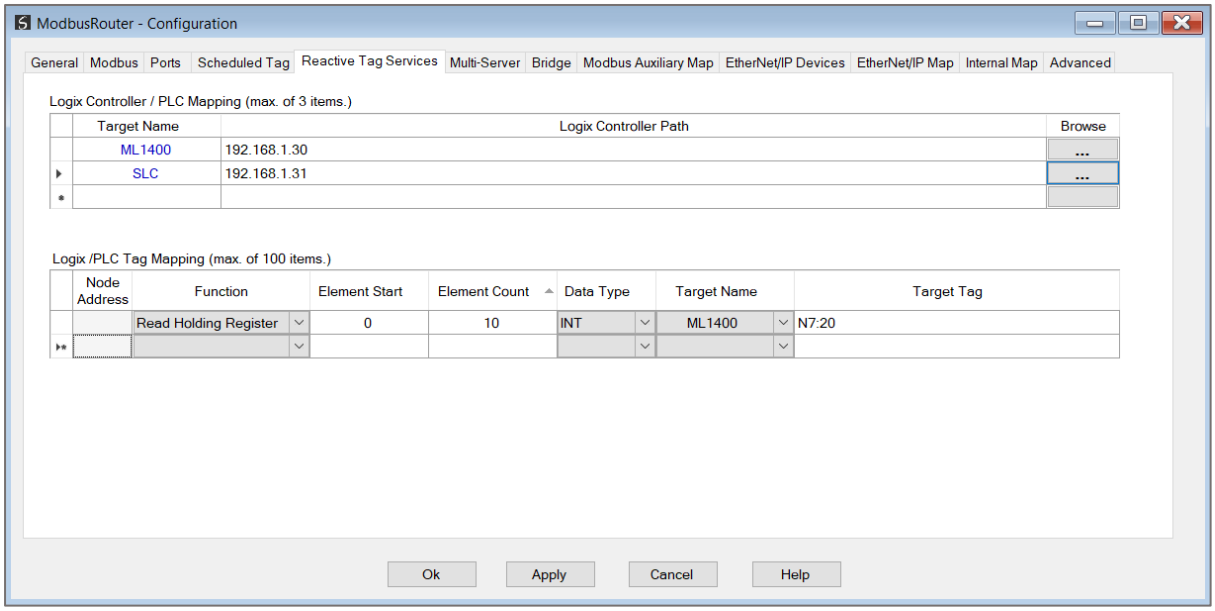


Figure 3.44 – File Selection for MicroLogix/SLC Controller Type.

The message routing executes in the exact same way as the Logix Controller type in section 3.8.1.1.

3.8.1.3. MULTI-SERVER

The MBR/B supports emulating multiple Modbus Server nodes when operating in Reactive Mode. When the *Multi-Server* option is selected in the *Modbus* tab (as shown below), then

the user will be able to enter a Modbus Node address for each item in the Reactive Tag mapped items.

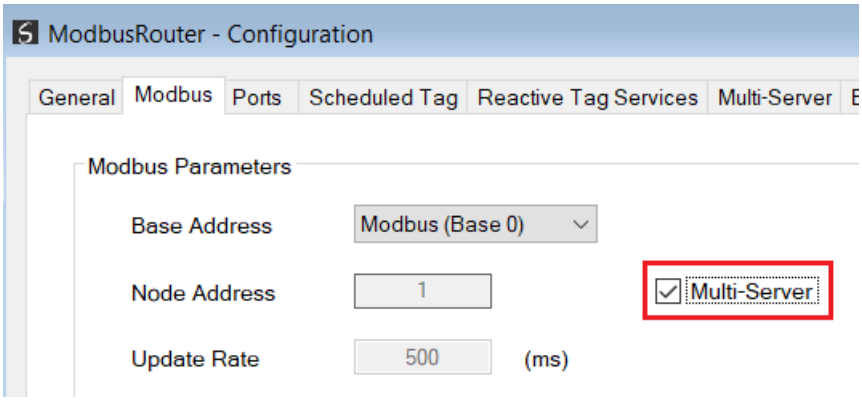


Figure 3.45 – Multi-Server option.

Once the Multi-Server option has been selected, then in the Reactive Tag Map, the Node Address for each mapped item will become available.

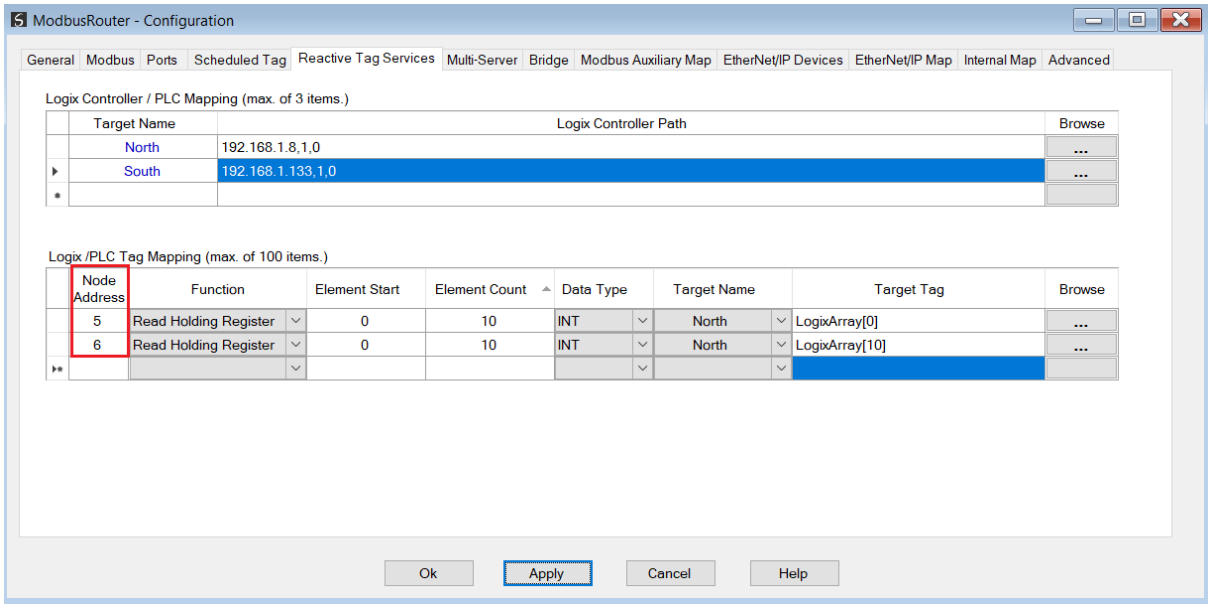


Figure 3.46 – Multi-Server Node address selection.

3.9. UNSCHEDULED MODE

There is no additional configuration required when using the Unscheduled Mode. The configuration required for the Modbus message is contained within the Message Block data, configured in Logix. **Refer** to the operation section for more information.

3.10. ENHANCED MODE

In Enhanced Mode the MBR/B can be configured in any of the four combinations:

Modbus	EtherNet/IP
Client	Target
Client	Originator
Server	Target
Server	Originator

Table 3.6 – Enhanced Mode Operating Combinations.

3.10.1. MODBUS MODE

3.10.1.1. SERVER

The MBR/B can operate as a Modbus Server for Modbus TCP, RTU232, and RTU485. One or more remote Modbus Clients can read and write to the full Modbus Register range in the MBR/B. The MBR/B can operate as a Modbus Server for Modbus TCP, Modbus RTU232, and Modbus RTU485 simultaneously. The user will need to configure the relevant Modbus Parameters as described in section 3.5 and section 3.6.

The Modbus Node Number will need to be configured in the *Modbus* parameters to allow a Modbus Client to access the MBR/B as a Modbus Server device. The MBR/B can also be configured for *Multi-Server,* allowing the module to emulate multiple Modbus node addresses.

3.10.1.1.1. MULTI-SERVER

The MBR/B supports emulating multiple Modbus Server nodes when operating as a Modbus Server in Enhanced Mode.

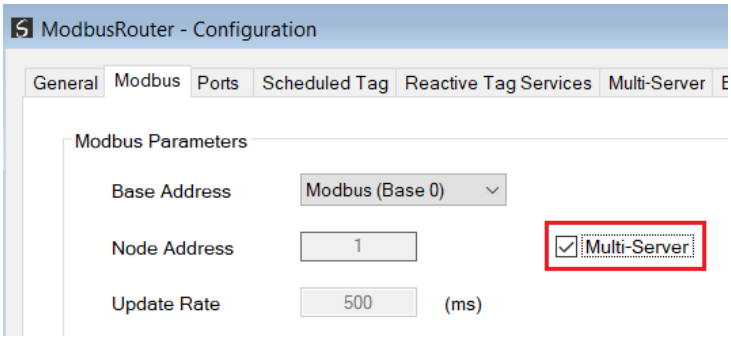


Figure 3.47 – Multi-Server selection.

Once the Multi-Server option has been selected, the *Multi-Server* tab will become available. The Multi-Server tab allows the MBR/B to emulate multiple Modbus Node address and also remap incoming requests to different internal Modbus Registers. This will allow the MBR/B to not only emulate the multiple Modbus Node address, but also provide unique data for each node even when the Modbus Client is reading the same Modbus address.

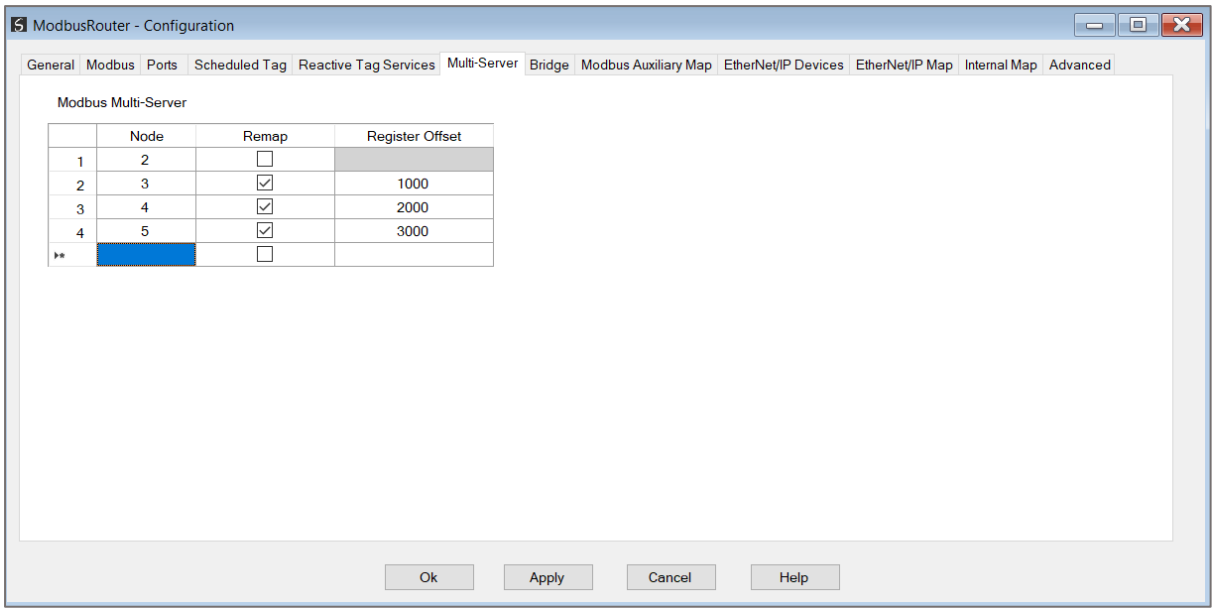


Figure 3.48 – Multi-Server Nodes and Remap.

In the above example, the MBR/B will respond to Modbus requests for node 2, 3, 4, and 5. When a Modbus request is received for node 2, the module will respond with the data in its internal Modbus registers at the requested Modbus Registers (e.g., if the request was received to read Modbus Holding Register 100, then the MBR/B will respond with the data in Modbus Holding Register 100 in its internal memory).

When a Modbus request is received for an emulated node that has the remap option configured, then the data being returned will be from the internal Modbus Register, but with the configured offset in the *Register Offset* parameter. For example, if a Modbus request is received for node 4 to read Holding Register 100, then the MBR/B will respond with the data in Modbus Holding Register 2100 (request offset of 100 + Remap offset of 2000) in its internal memory.

3.10.1.1.2. INTERNAL DATA SPACE MAPPING

When the module is operating as a Modbus Server, the data from the Modbus Registers (used to exchange data with the Modbus Client) can be mapped to the EtherNet/IP 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.

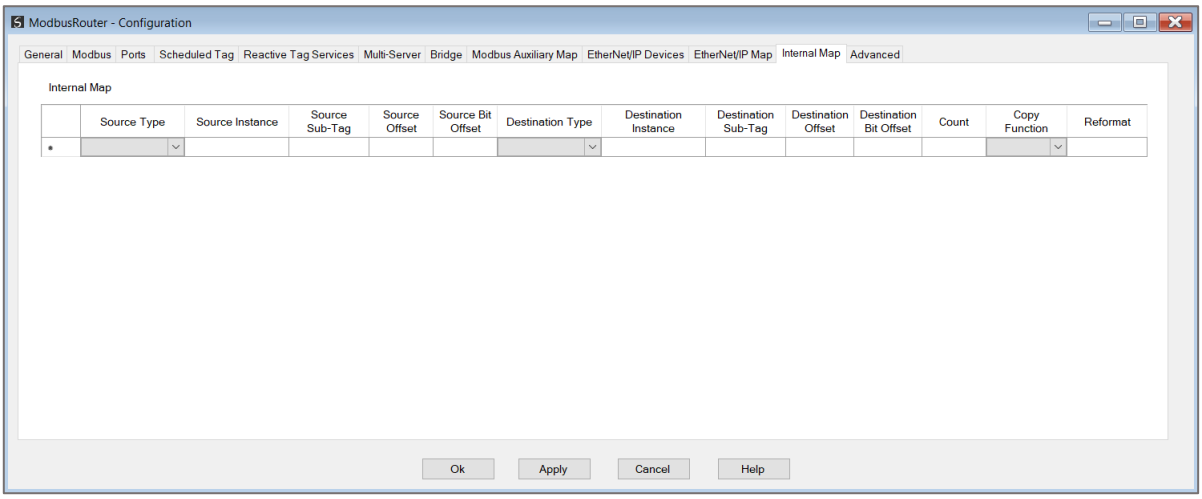


Figure 3.49 – Modbus Server – Internal Mapping

IDS COPY – MODBUS SOURCE

When copying Modbus Data or Status to the EtherNet/IP interface, the source type needs to be MB Register or MB Aux.

NOTE: The Modbus Aux source is only relevant when the MBR/B is operating in Enhanced mode as a Modbus Client.

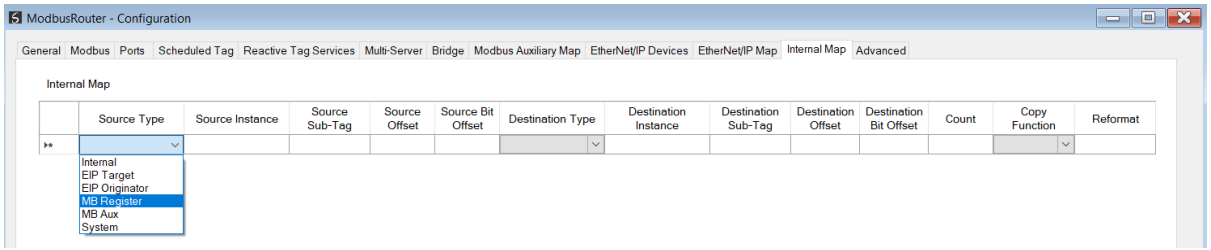


Figure 3.50 – IDS Copy - Modbus Register Source Type.

When using the MB Register source, the source instance will be the Modbus register type required.

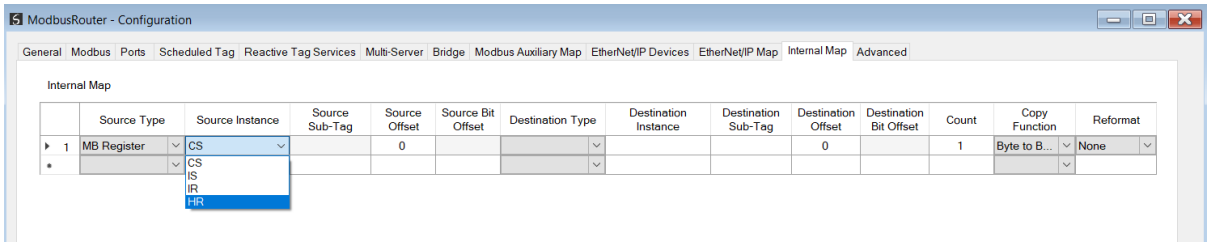


Figure 3.51 – IDS Copy - Modbus Register Source Instance.

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.

When using the MB Aux source, the source instance will be an item in the Modbus Auxiliary Map. For example, if there are two items in the Modbus Auxiliary Map, then there will be two Source Instances for the MB Aux Source Type.

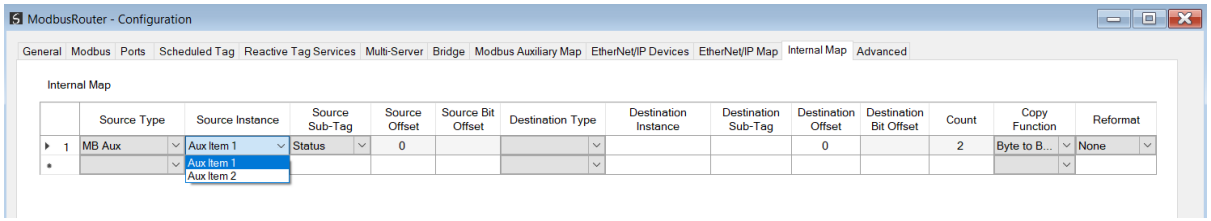


Figure 3.52 – IDS Copy - Modbus Register Source Instance.

The MB Aux format is as shown below:

Parameter	Data Type	Description
MB Aux Status	INT	Bit 0 – Last Transaction Ok

Table 3.7 – Modbus Aux Item Status.

IDS COPY – MODBUS DESTINATION

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

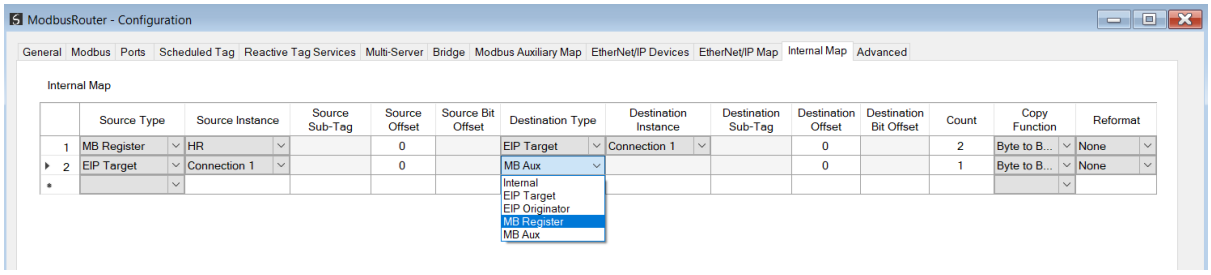


Figure 3.53 – IDS Copy - Modbus Register Destination Type.

The destination instance will be the Modbus register type required.

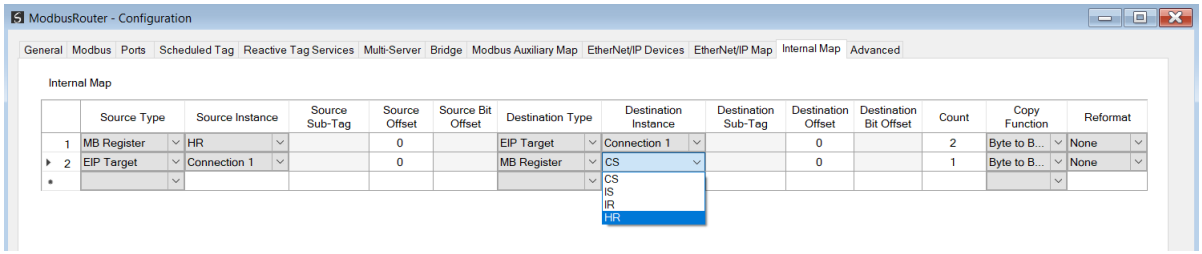


Figure 3.54 – IDS Copy - Modbus Register Destination Instance.

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.

3.10.1.2. CLIENT

The MBR/B can operate as a Modbus Client for Modbus TCP, RTU232, and RTU485. The user will need to configure the relevant Modbus Parameters as described in section 3.5 and section 3.6.

The user will also need to configure the Modbus Auxiliary Map which is used to execute the configured Modbus requests. This map will allow the user to configure various read and write functions to external Modbus Registers, to and from the internal Modbus registers.

3.10.1.2.1. 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 Client operating interface. Up to 100 mapping items (total) can be configured while communicating to up to 100 Modbus TCP Server 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***.

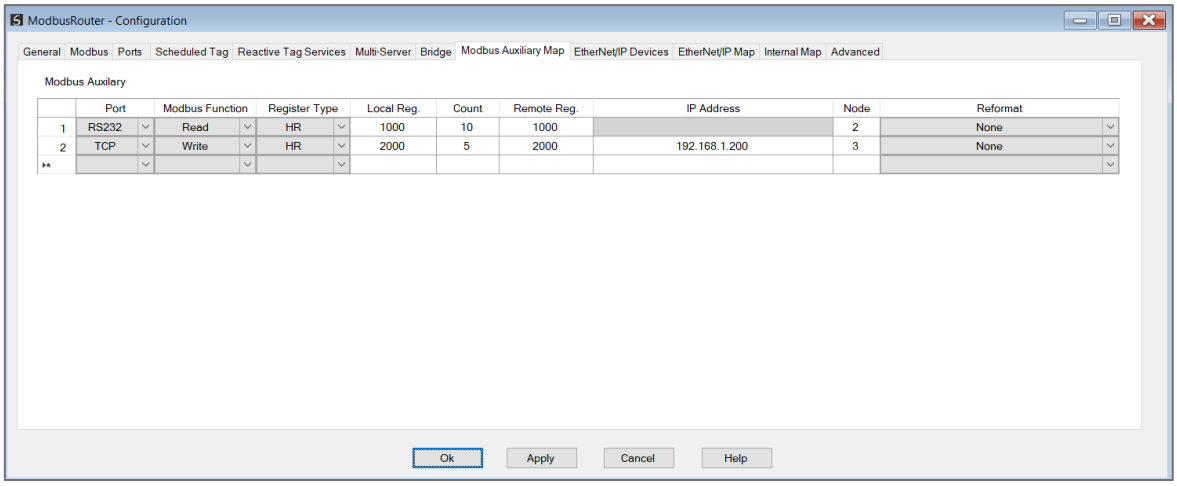


Figure 3.55 – Modbus Auxiliary Map Configuration.

The Modbus Auxiliary Map configuration consists of the following parameters:

Parameter	Description
Port	The external port to be used: **TCP** – Modbus TCP (Ethernet) **RS232** – Modbus RTU232 **RS485** – Modbus RTU485
Modbus Function	This is the Modbus function that is sent to the Modbus Server. **Read** – Read a Modbus Register (e.g. HR, IR, CS, or IS) from a Modbus Server. **Write** – Write a Modbus Register (e.g. HR or CS) to a Modbus Server.
Register Type	Modbus Register Type: **CS** – Coil Status **IS** – Input Status **IR** – Input Register **HR** – Holding Register
Local Reg.	The local (internal) MBR/B Modbus register address.
Count	The number of Modbus elements to read or write.
Remote Reg.	The remote Server Modbus address register.
IP Address	The IP address of the remote Modbus Server.
Node	The Modbus Node address of the remote Modbus Server.
Reformat	Used to specify how the data is formatted before writing to, or after reading from, the Modbus Server. **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 3.8 – Modbus Auxiliary Map parameters

3.10.1.2.2. INTERNAL DATA SPACE MAPPING

When the module is operating as a Modbus Client, the data from the Modbus Registers (used to exchange data with various Modbus Servers) can be mapped to the EtherNet/IP 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.

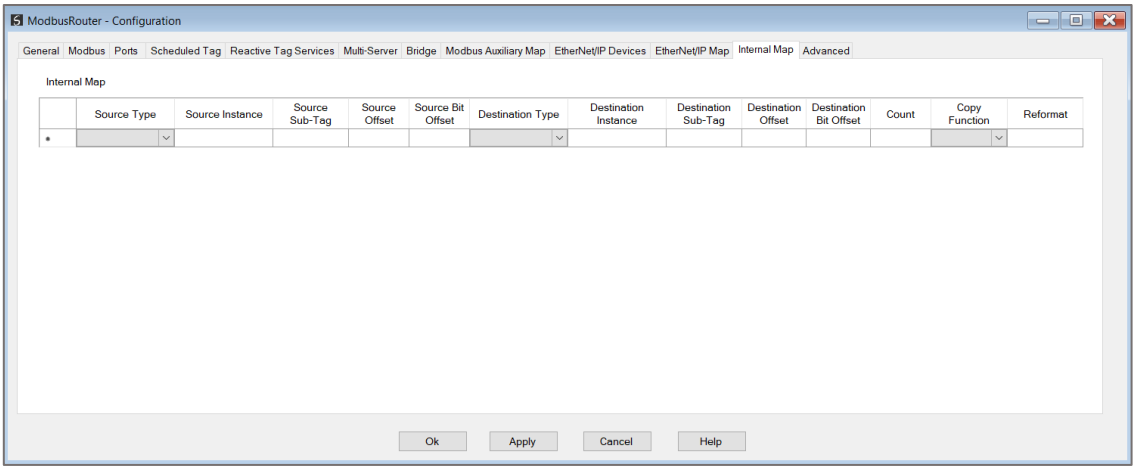


Figure 3.56 – Modbus Clients – Internal Mapping.

IDS COPY – MODBUS SOURCE

When copying Modbus Data or Status to the EtherNet/IP interface, the source type needs to be MB Register or MB Aux.

NOTE: The Modbus Aux source is only relevant when the MBR/B is operating in Enhanced mode as a Modbus Client.

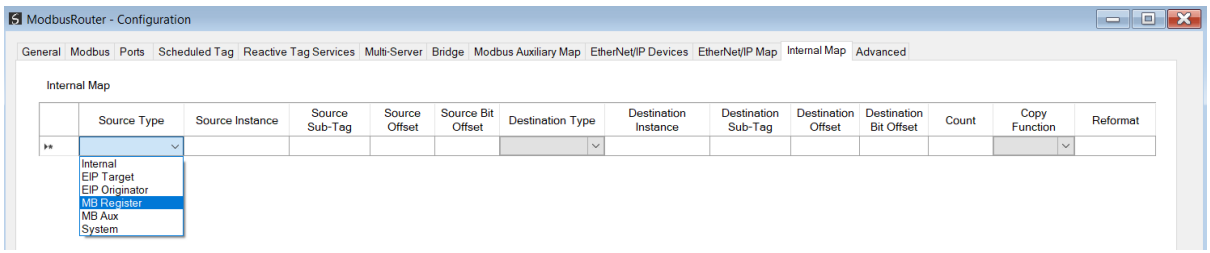


Figure 3.57 – IDS Copy - Modbus Register Source Type.

When using the MB Register source, the source instance will be the Modbus register type required.

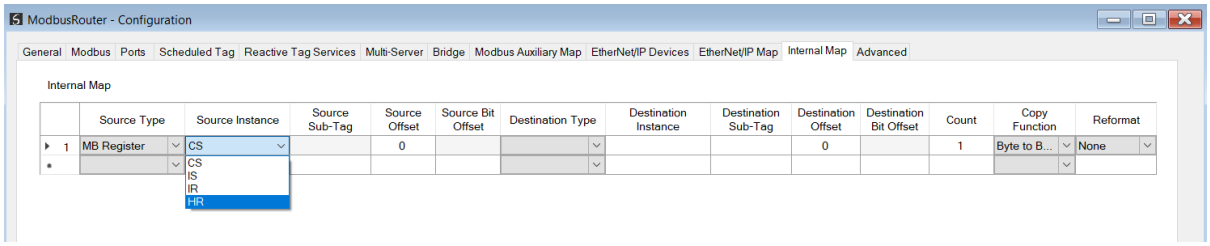


Figure 3.58 – IDS Copy - Modbus Register Source Instance.

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.

When using the MB Aux source, the source instance will be an item in the Modbus Auxiliary Map. For example, if there are two items in the Modbus Auxiliary Map, then there will be two Source Instances for the MB Aux Source Type.

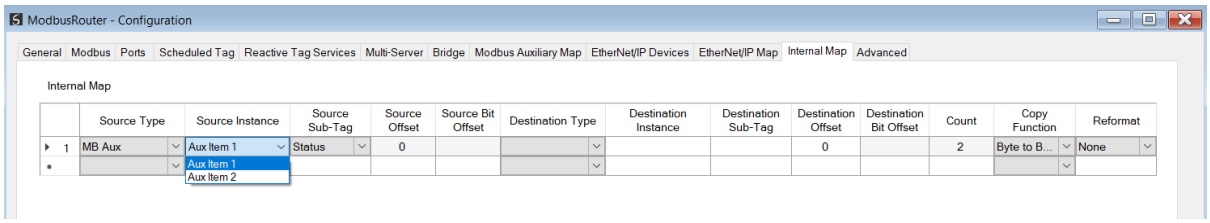


Figure 3.59 – IDS Copy - Modbus Register Source Instance.

The MB Aux format is as shown below:

Parameter	Data Type	Description
MB Aux Status	INT	Bit 0 – Last Transaction Ok

Table 3.9 – Modbus Aux Item Status.

IDS COPY – MODBUS DESTINATION

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

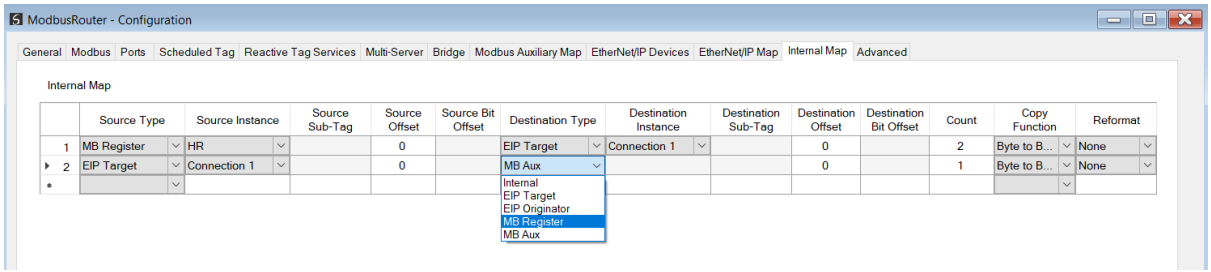


Figure 3.60 – IDS Copy - Modbus Register Destination Type.

The destination instance will be the Modbus register type required.

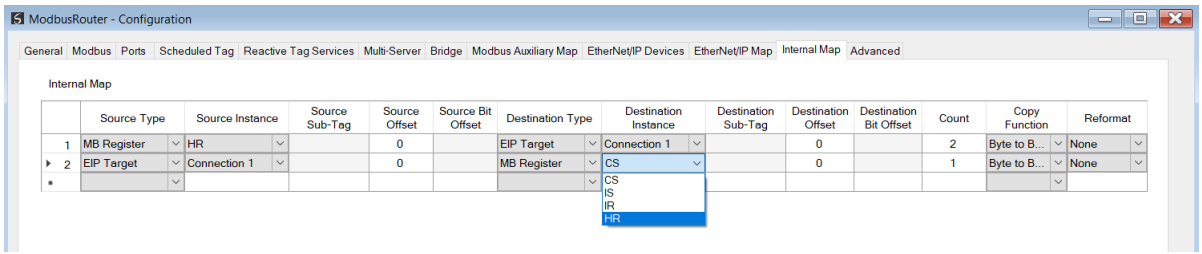


Figure 3.61 – IDS Copy - Modbus Register Destination Instance.

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.

3.10.2. EtherNet/IP Mode

3.10.2.1. Target

A controller (e.g. Logix controller) can own the MBR/B over EtherNet/IP using up to 5 Class 1 EtherNet/IP connections when the EtherNet/IP Mode is set to EtherNet/IP Target. This will allow the MBR/B to exchange data with the controller using the input and output assembly of the Class 1 EtherNet/IP connections. Data from Modbus devices can be mapped to the Logix controller over EtherNet/IP.

The user will need to add the MBR/B to the Logix IO tree under an EtherNet/IP bridge (e.g. 1756-EN2TR) or Ethernet Logix controller (e.g. 1756-L85E).

For Logix versions 20 and beyond, the modules can be added using the EDS Add-On-Profile (AOP), which is described in section 3.10.2.1.1.

For older versions (19 and below), the module must be added using a Generic Profile which is described in section 3.10.2.1.2.

NOTE: The MBR/B will need to be added to the Logix tree using an EDS AOP when operating in Enhanced Mode and mapping Modbus data to the Class 1 Input and Output Assemblies.

3.10.2.1.1. STUDIO 5000 CONFIGURATION (VERSION 21+)

ADD MODULE TO ETHERNET/IP I/O CONFIGURATION

The MBR/B can be easily integrated with Allen-Bradley Logix family of controllers. Integration with the Logix family in Studio5000 makes use of the EDS Add-On-Profile (AOP).

NOTE: Logix version 21 and newer supports EDS AOPs.

Before the module can be added to the tree the module's EDS file must be registered.

Using RSLinx, the EDS file can be uploaded from the device after which the EDS Hardware Installation tool will be invoked to complete the registration.

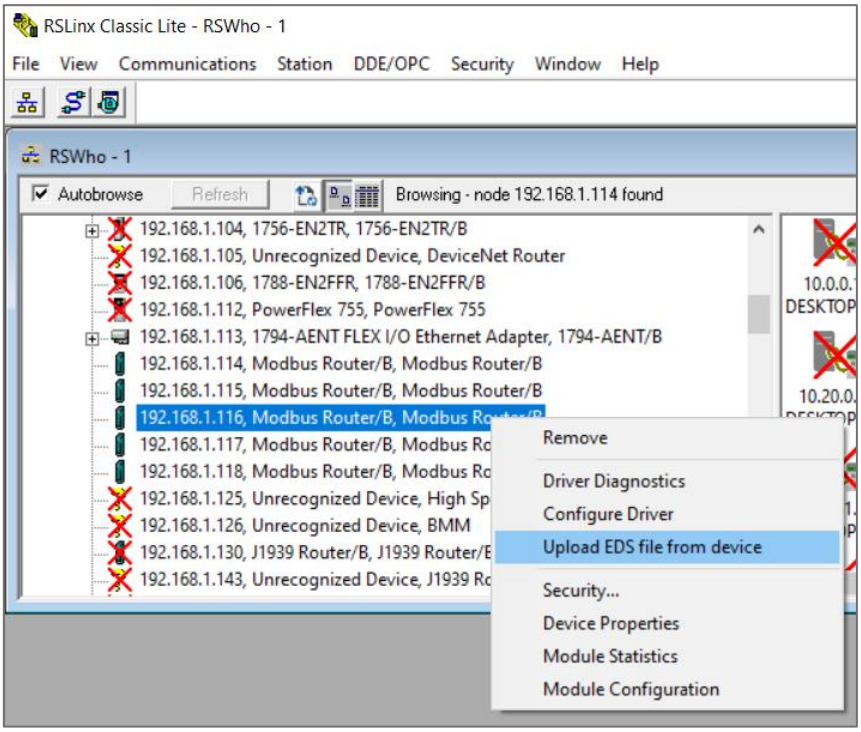


Figure 3.62 – EDS file upload from MBR/B.

Alternatively, the EDS file can be downloaded from the product web page at www.aparian.com and registered manually using the EDS Hardware Installation Tool shortcut under the Tools menu in Studio 5000.

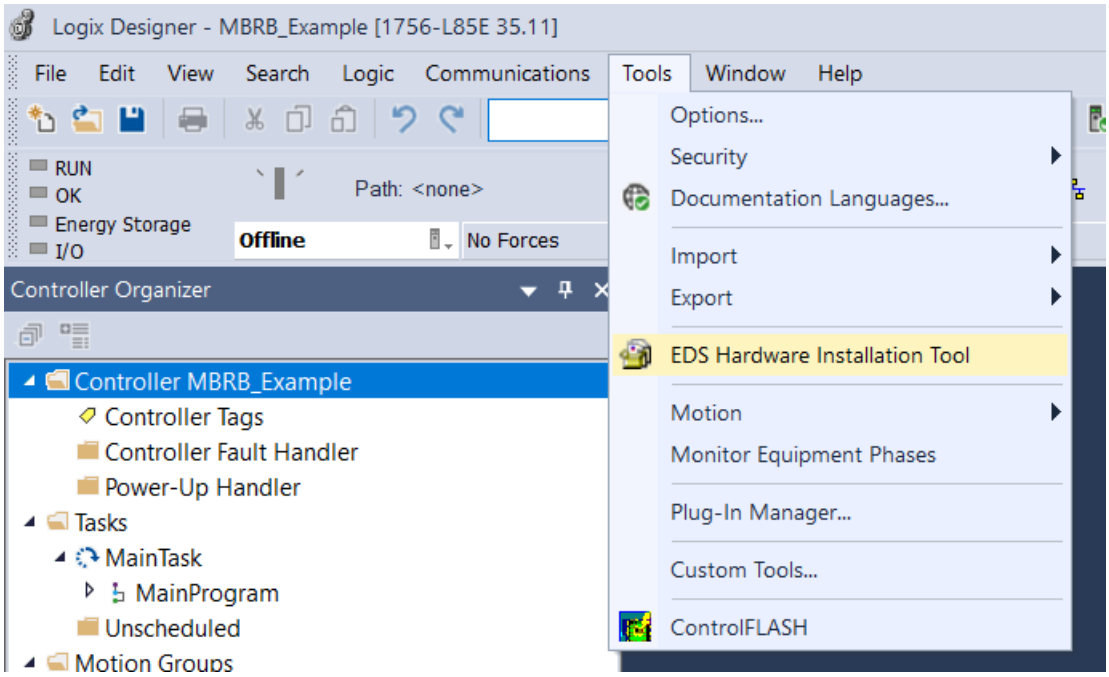


Figure 3.63 - EDS Hardware Installation Utility.

After the EDS file has been registered, the module can be added to the Logix IO tree in Studio 5000. Under a suitable Ethernet bridge module in the tree, select the Ethernet network, right-click and select the New Module option.

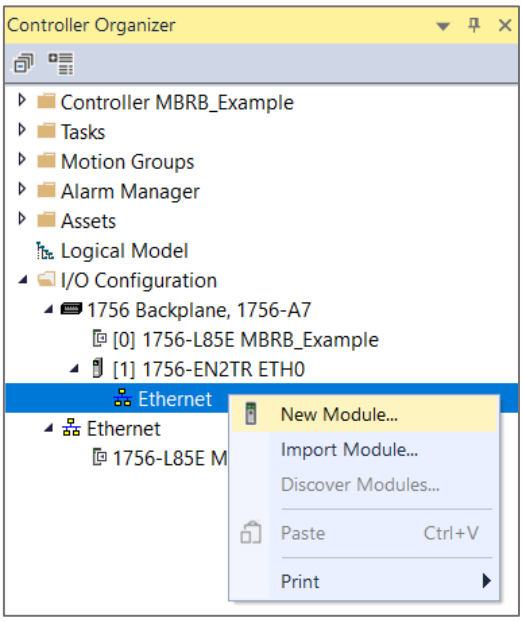


Figure 3.64 – Adding a module.

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.

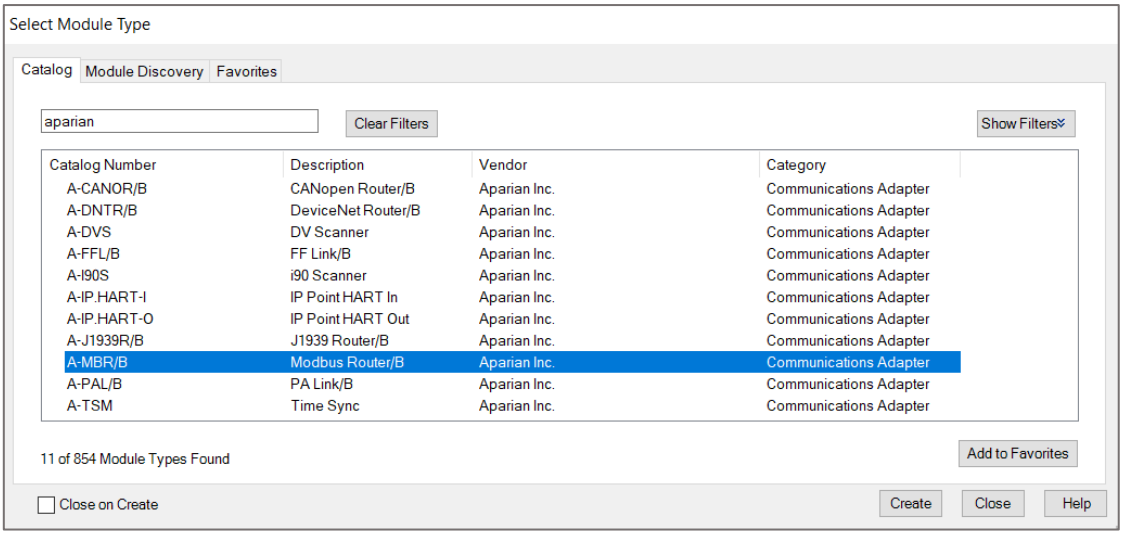


Figure 3.65 – Selecting the module.

Locate and select the Modbus Router/B module and select the ***Create*** option. The module configuration dialog will open, where the user must specify the Name and IP address as a minimum to complete the instantiation.

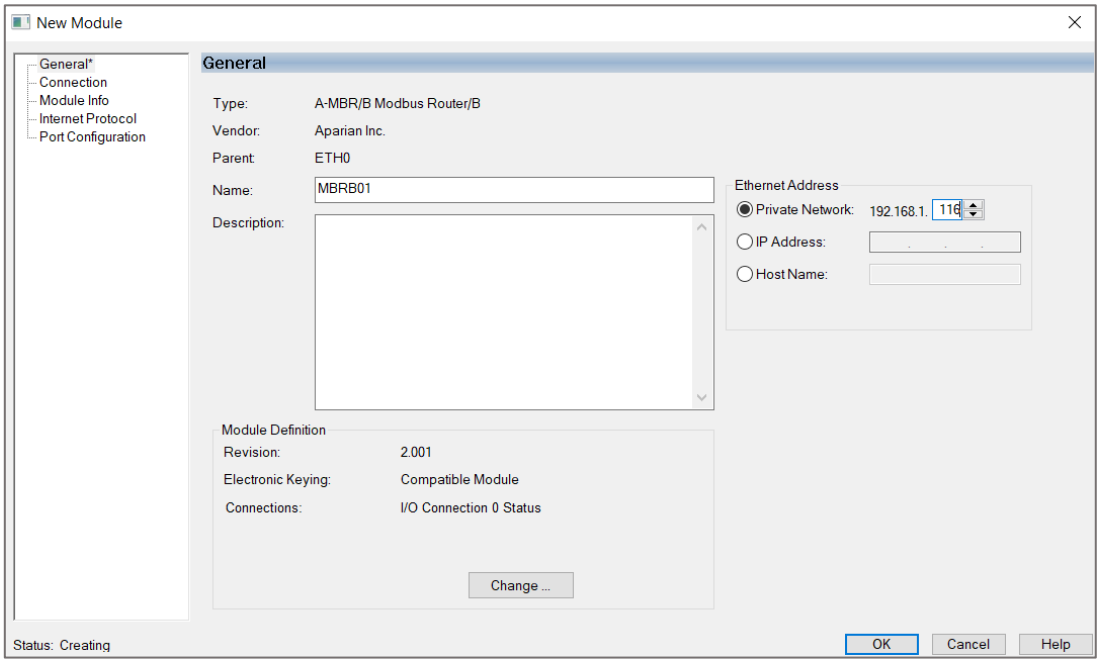


Figure 3.66 – Module instantiation.

The MBR/B supports up to 5 class 1 EtherNet/IP connections. The user will need to **ensure that the number of connections configured in the General tab of the module configuration matches the selected connection count in Logix**.

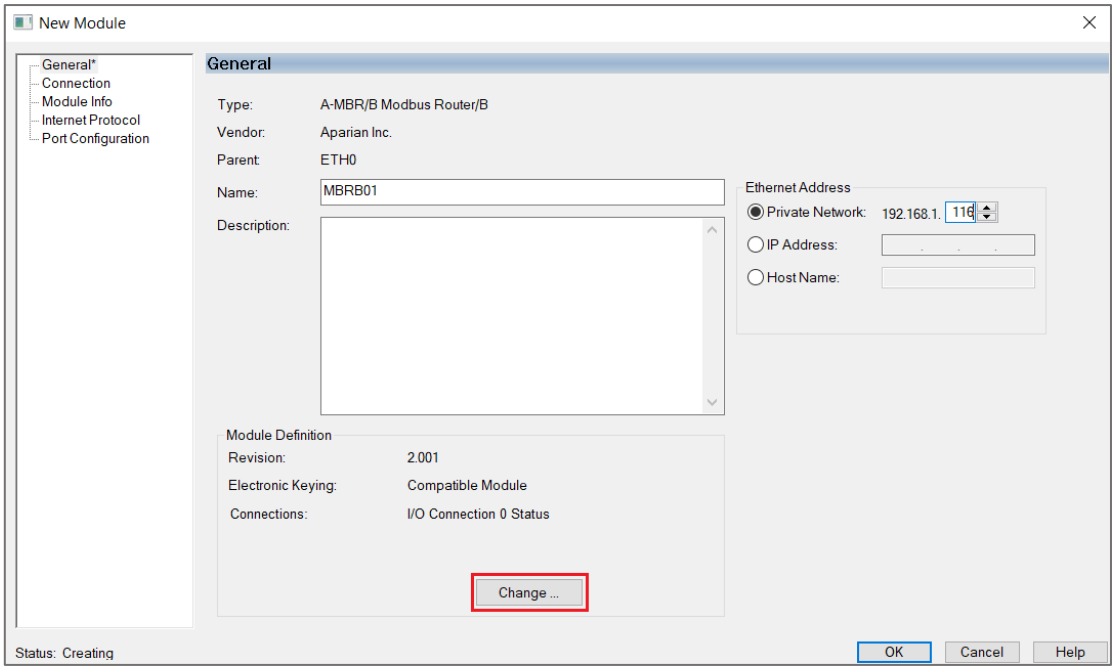


Figure 3.67 – Change number of IO Connections.

Next the user will need to select the number of connections required.

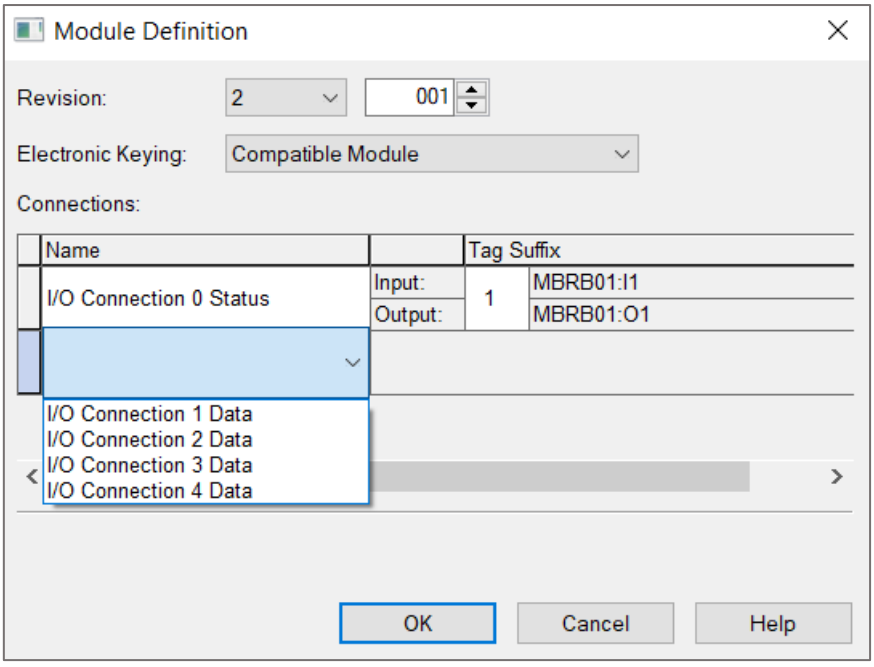


Figure 3.68 – Selection of IO Connections.

Now the MBR/B module will be in the Logix IO tree.

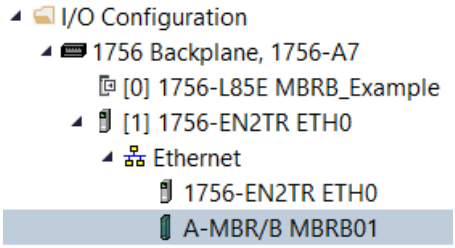


Figure 3.69 – Logix IO tree.

The Module Defined Data Types will automatically be created during the instantiation process. These module-defined tags will need to be copied to and from meaningful structures.

3.10.2.1.2. RSLOGIX 5000 CONFIGURATION (PRE-VERSION 21)

ADD MODULE TO I/O CONFIGURATION

The module can operate in both a Logix "owned" and standalone mode. When the module operates in a Logix "owned" mode the Modbus Router will need to be added to the RSLogix 5000 IO tree. When using Logix versions prior to version 21, then the module will need to be added as a generic Ethernet module. This is done by right clicking on the Ethernet Bridge in the RSLogix 5000 and selecting *New Module* after which the *ETHERNET-MODULE* is selected to be added as shown in the figure below.

NOTE: The MBR/B will need to be added to the Logix tree using an EDS AOP when operating in Enhanced Mode and mapping Modbus data to the Class 1 Input and Output Assemblies.

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

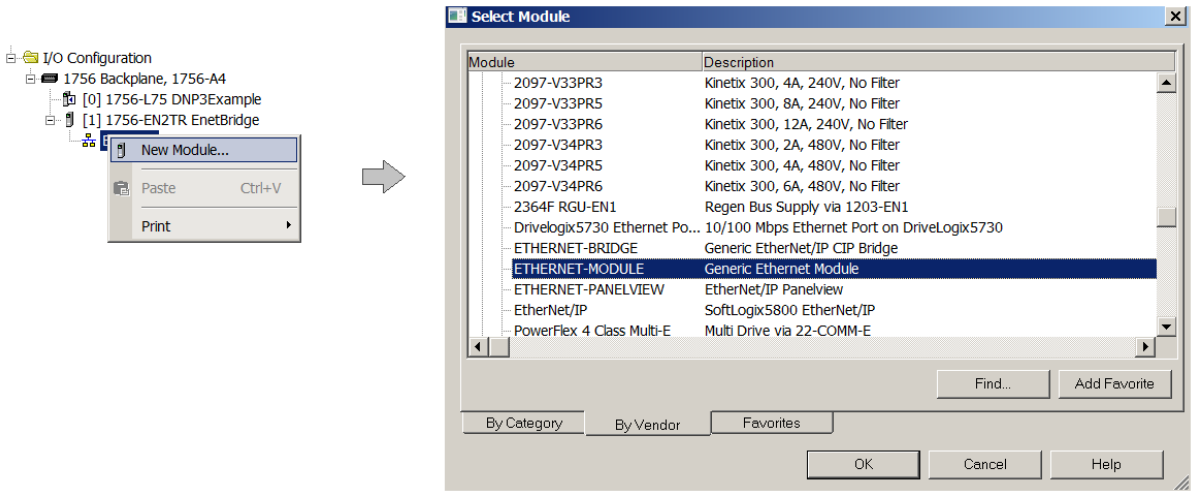


Figure 3.70 - Add a Generic Ethernet Module in RSLogix 5000.

The user must enter the IP address of the Modbus 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	111	34 (32-bit)
Output	101	1 (32-bit)
Configuration	102	0 (8-bit)

Table 3.10 - RSLogix class 1 connection parameters for the Modbus Router.

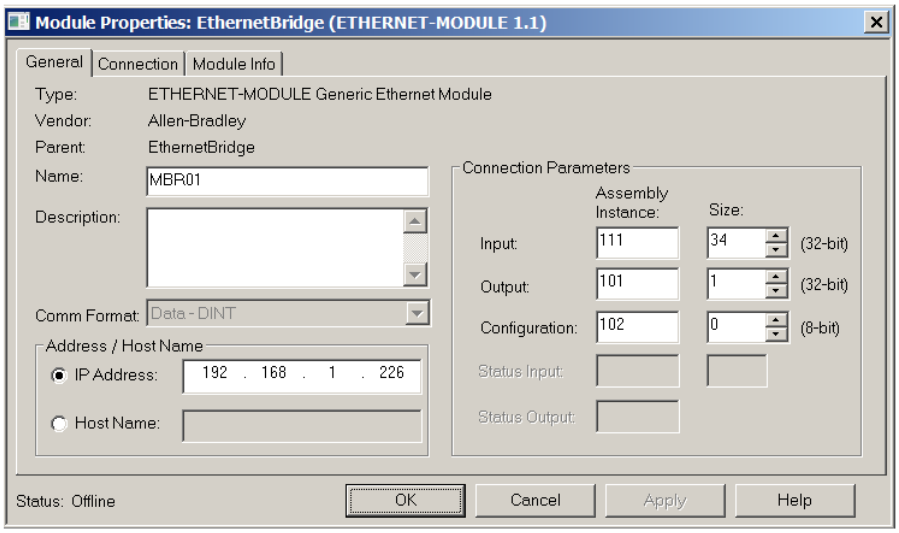


Figure 3.71 - General module properties in RSLogix 5000.

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 add the connection requested packet interval (RPI). This is the rate at which the input and output assemblies are exchanged. The recommended value is 500ms. Refer to the technical specification section in this document for further details on the limits of the RPI.

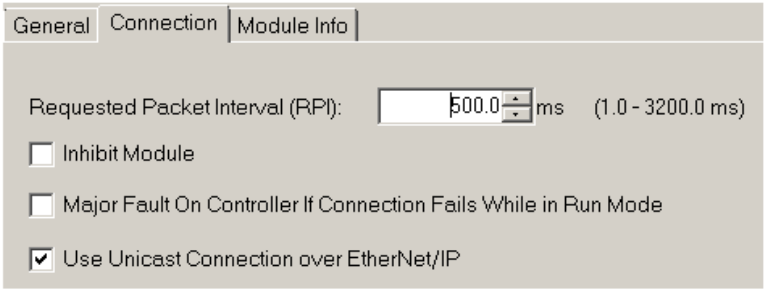


Figure 3.72 - Connection module properties in RSLogix 5000.

Once the module has been added to the RSLogix 5000 IO 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.

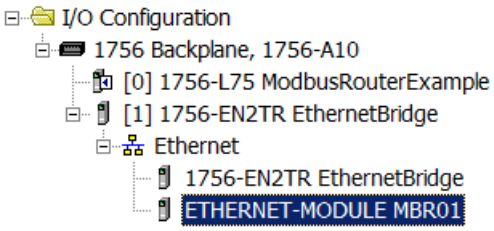


Figure 3.73 – RSLogix 5000 I/O module tree.

IMPORTING UDTS AND MAPPING ROUTINES

To simplify the mapping of the input image, an RSLogix 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.

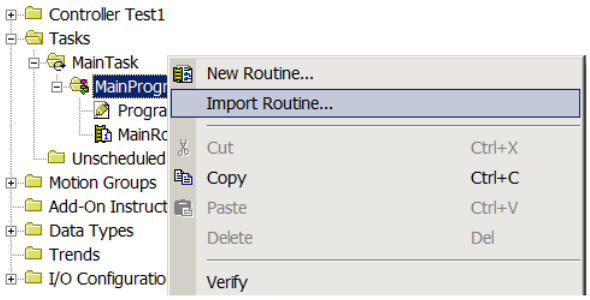


Figure 3.74 – RSLogix 5000 Importing Modbus Router specific routine and UDTs.

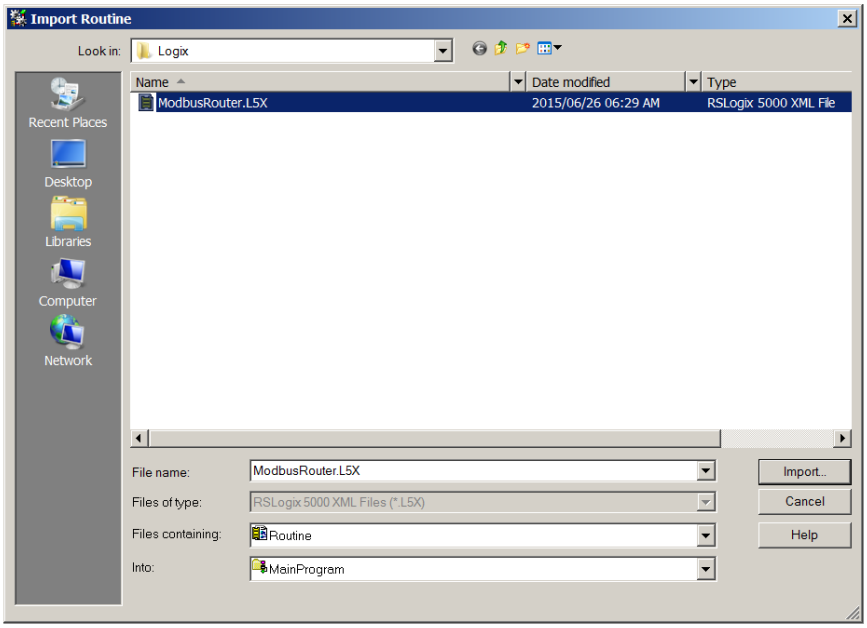


Figure 3.75 - Selecting partial import file.

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 ModbusRouter module to the aforementioned tags.
- An example Unscheduled Message instruction with the associated Tags

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

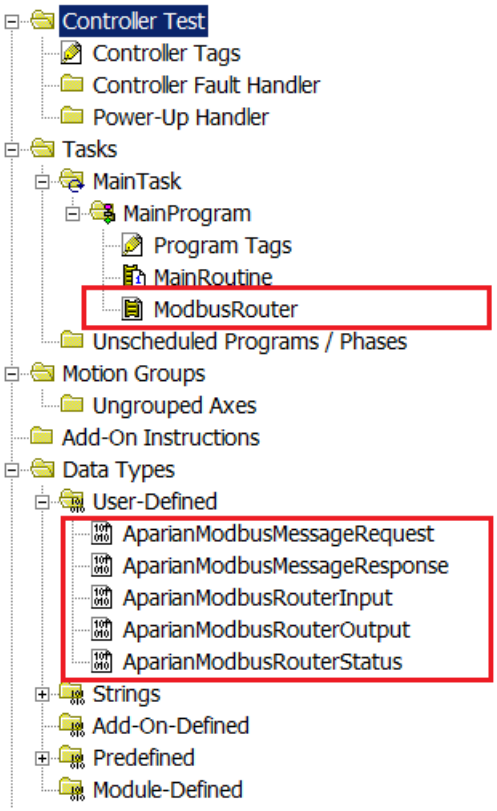


Figure 3.76 - Imported RSLogix 5000 objects.

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

3.10.2.1.3. INTERNAL DATA SPACE MAPPING

When the module is operating as an EtherNet/IP Target, the data from the originator device (e.g. Logix Controller) can be mapped to the Modbus Registers 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.

IDS COPY – ETHERNET/IP TARGET SOURCE

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

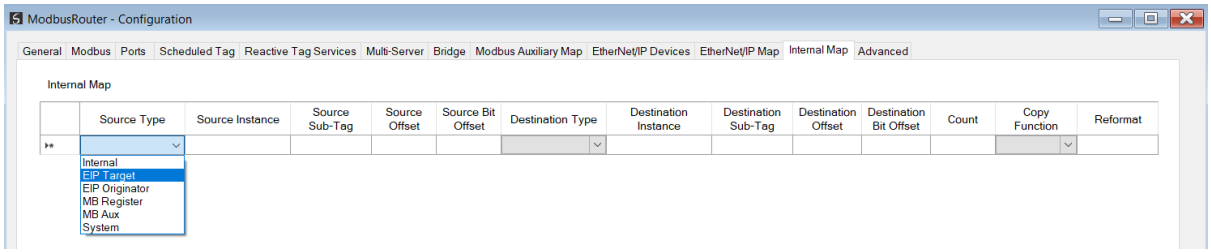


Figure 3.77 – IDS Copy – EtherNet/IP Target Source Type.

The source instance will be the connection number, which can be connection 1 to 4, based on the number of connections configured.

NOTE: Connection 0 cannot be selected because the first connection is used for module status. Connection 1 to 4 is used for Modbus data mapping which will be connection :I2 to :I5 in Logix.

The Source Offset is the offset in the EtherNet/IP output assembly 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.

IDS COPY – ETHERNET/IP TARGET DESTINATION

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

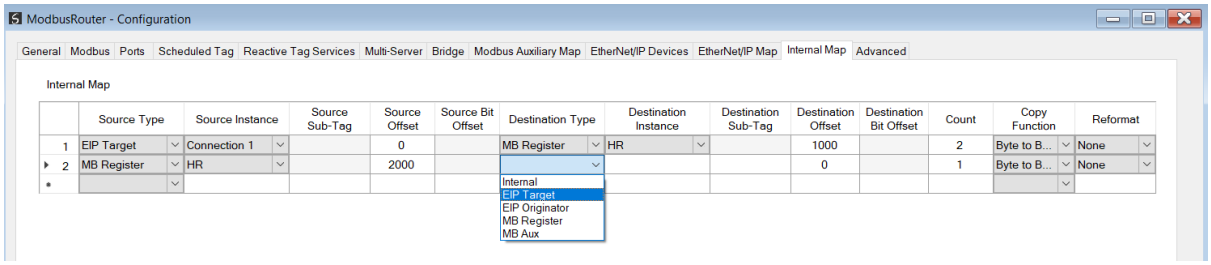


Figure 3.78 – IDS Copy – EtherNet/IP Target Destination Type.

The destination instance will be the connection number, which can be connection 1 to 4, based on the number of connections configured.

NOTE: Connection 0 cannot be selected because the first connection is used for module control. Connection 1 to 4 is used for Modbus data mapping which will be connection :O2 to :O5 in Logix.

The Destination Offset is the offset of the EtherNet/IP input assembly from where the data must be copied. The Count is the number of **bytes** that will be copied.

3.10.2.2. ORIGINATOR

The MBR/B can operate as an 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 main configuration while the cyclic class 1 connections are added to the *EtherNet/IP Connections* node under the module in the Slate project tree.

3.10.2.2.1. ETHERNET/IP CLASS 1 DEVICE CONNECTIONS

The MBR/B can establish up to 10 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.*

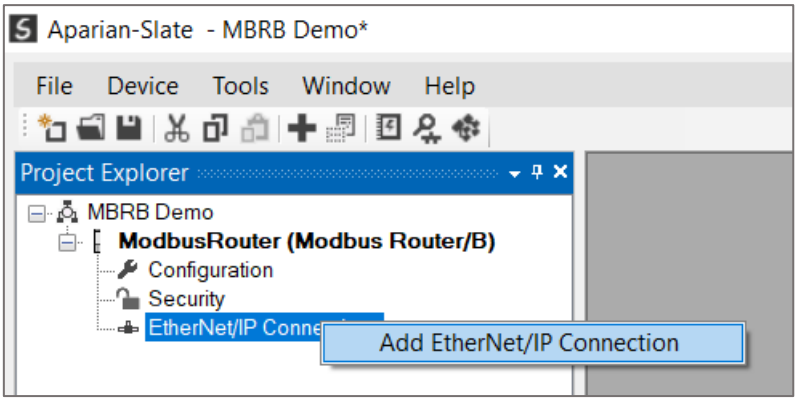


Figure 3.79 – Adding EtherNet/IP Class 1 Connection.

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

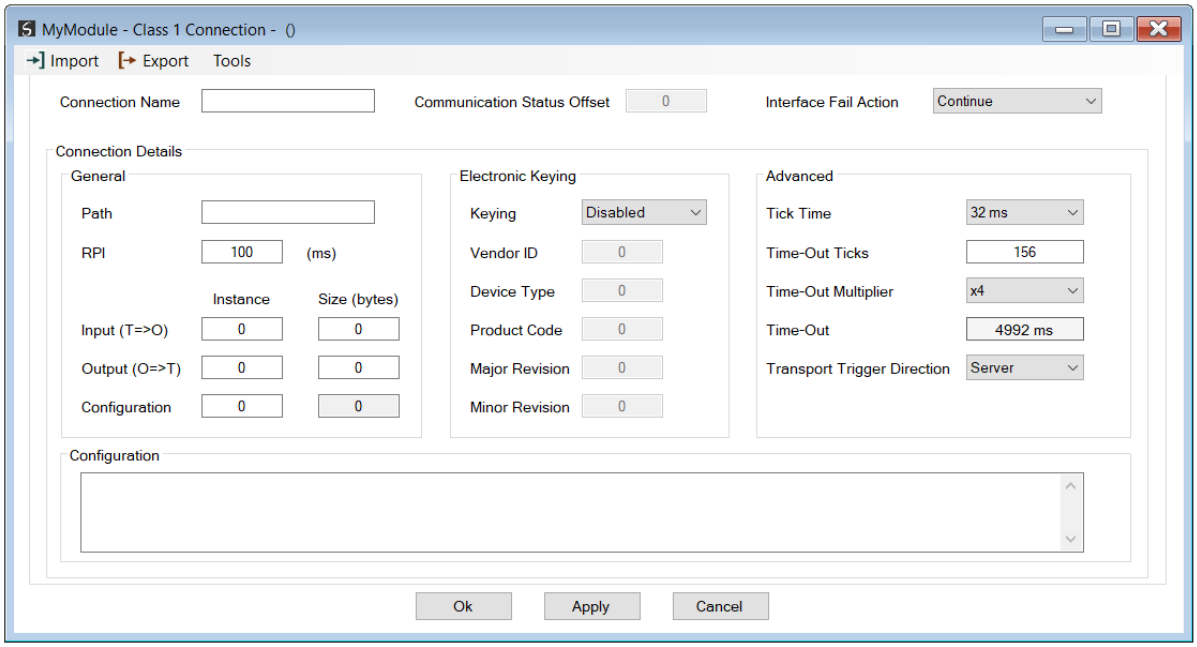


Figure 3.80 – EtherNet/IP Class 1 Connection Parameters.

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.
Interface Fail Action	When the Modbus 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 Modbus communication 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 3.11 – 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.

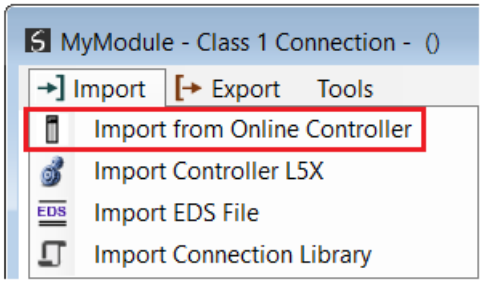


Figure 3.81 – Import from Online Controller.

The Import Connection Parameters form will open.

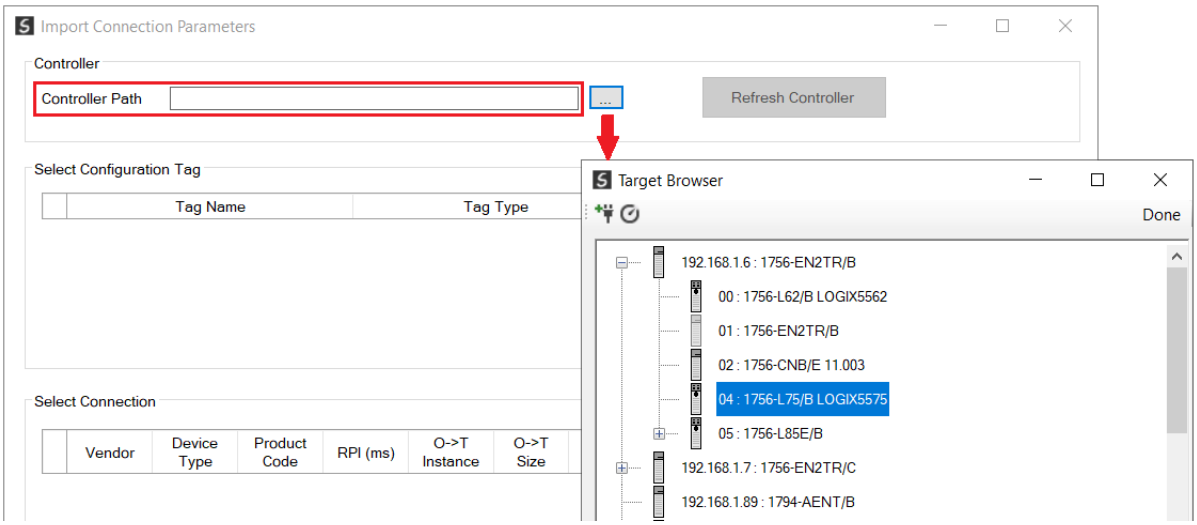


Figure 3.82 – Import Connection Parameters – Controller Path.

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.

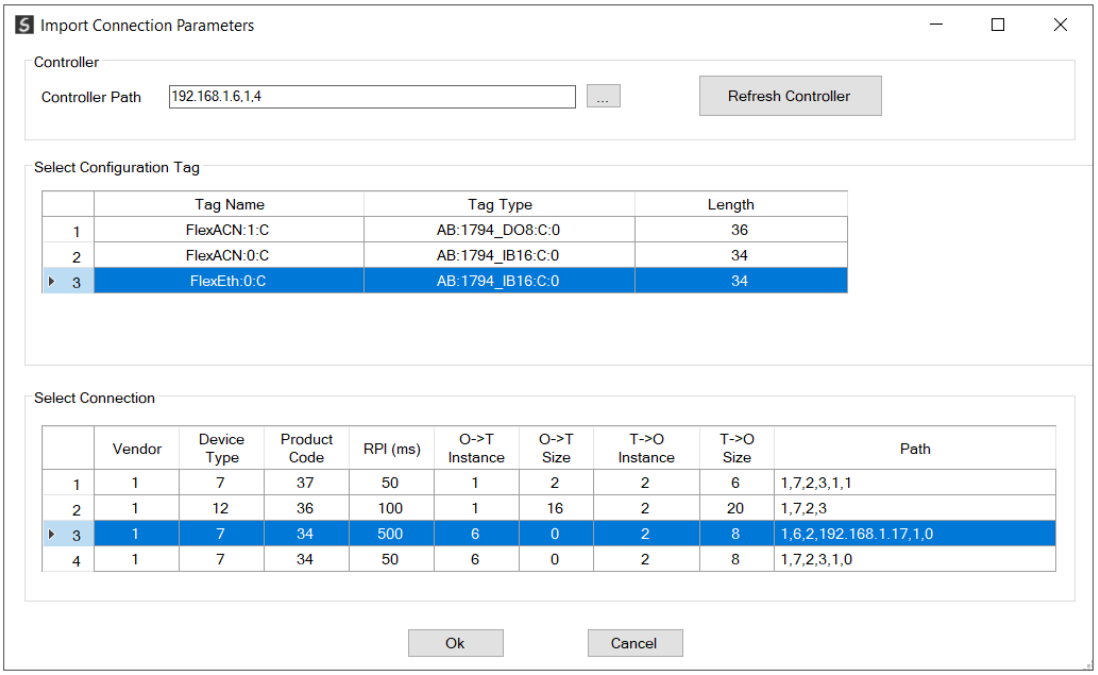


Figure 3.83 – Import Connection Parameters – Select Connection.

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.

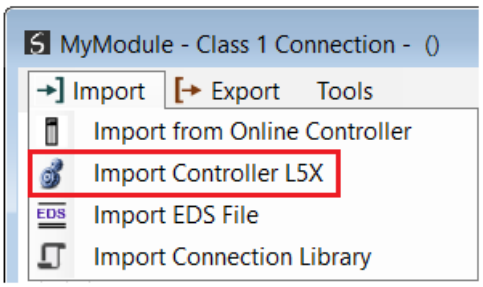


Figure 3.84 – Import from Controller L5X.

The Import Connection Parameters form will open.

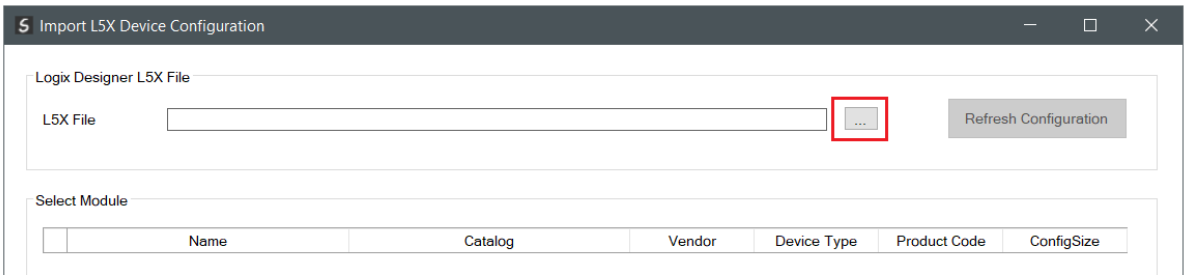


Figure 3.85 – Import L5X Device Configuration – Select L5X.

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.

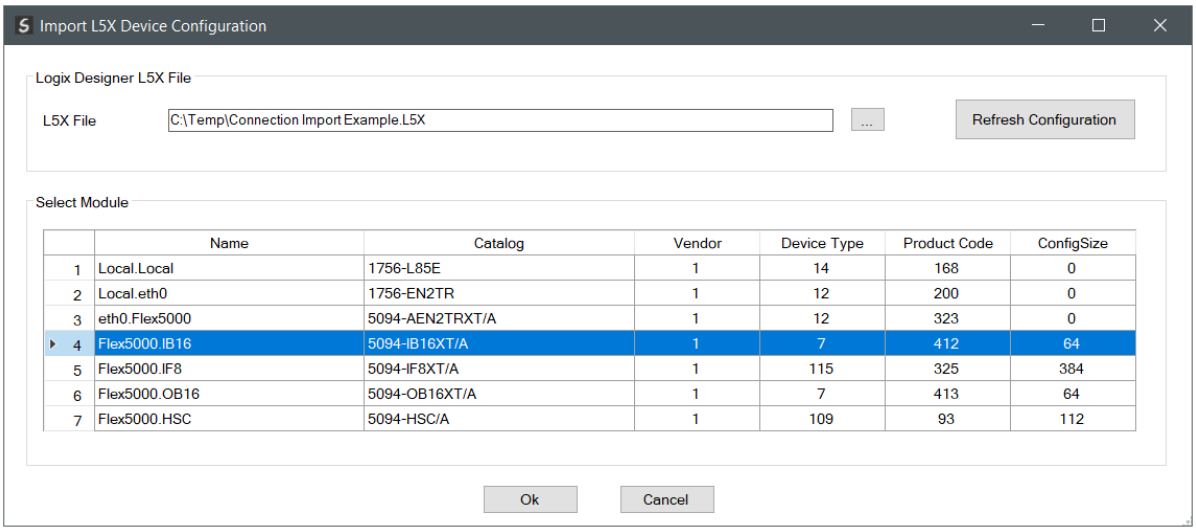


Figure 3.86 – Import L5X Device Configuration.

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.

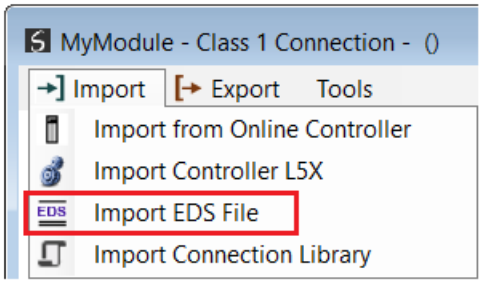


Figure 3.87 – Import EDS File.

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

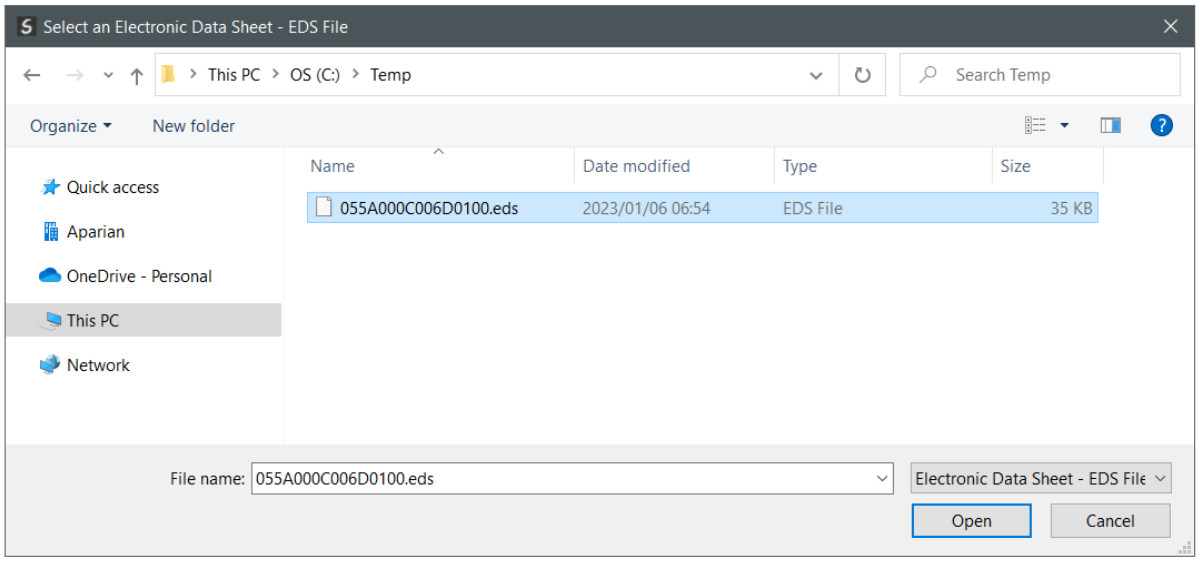


Figure 3.88 – Browse to EDS File.

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.

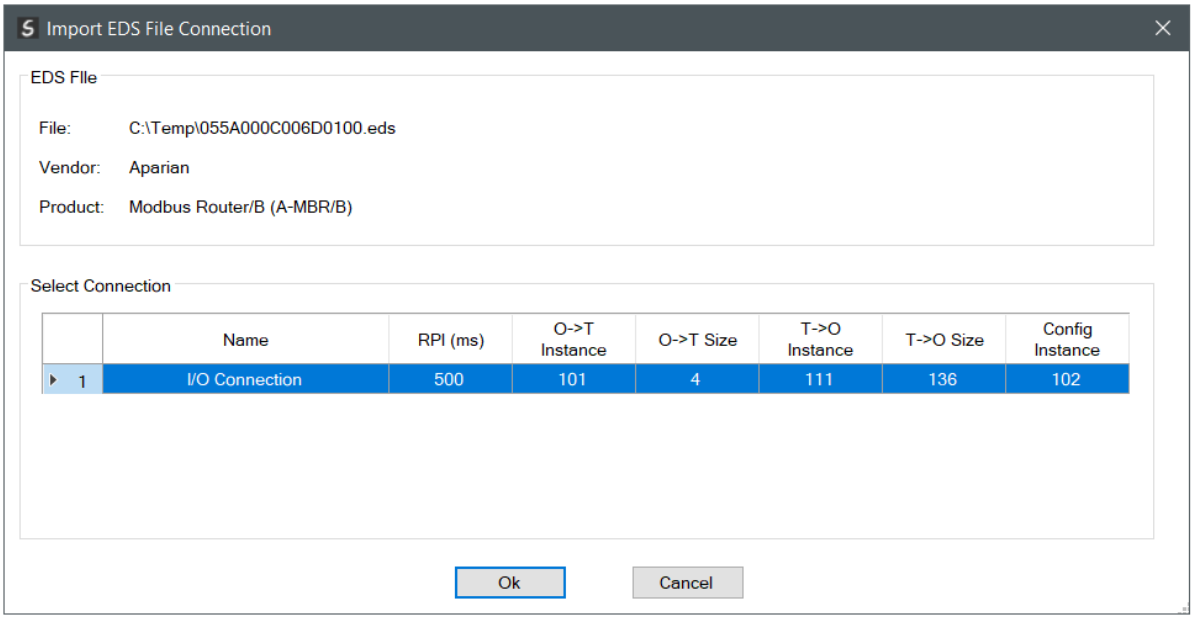


Figure 3.89 – Select Connection.

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.

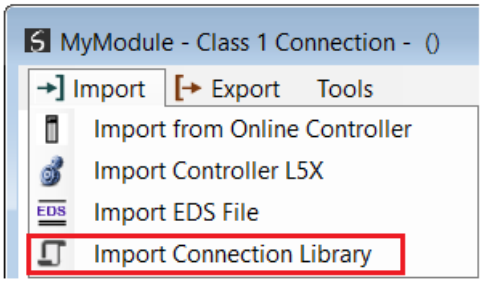


Figure 3.90 – Import Connection Library File.

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.

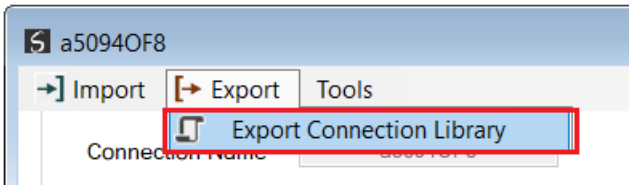


Figure 3.91 – Export Connection Library File.

3.10.2.2.2. 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.

ETHERNET/IP DEVICES

This tab is enabled when the Primary Interface 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 100 Kbytes. 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 by right-clicking the module and selecting *Configuration*.

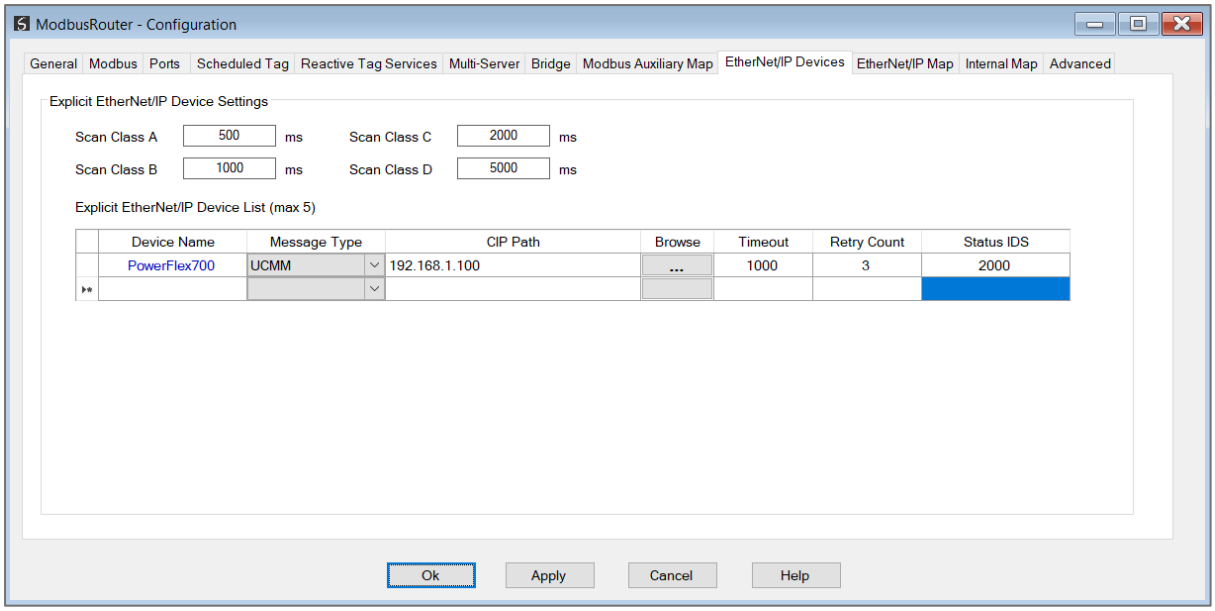


Figure 3.92 – EtherNet/IP Devices – Configuration.

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 3.12 – EtherNet/IP Devices configuration parameters.

ETHERNET/IP MAP

This tab is enabled when the Primary Interface 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 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 by right-clicking the module and selecting *Configuration*.

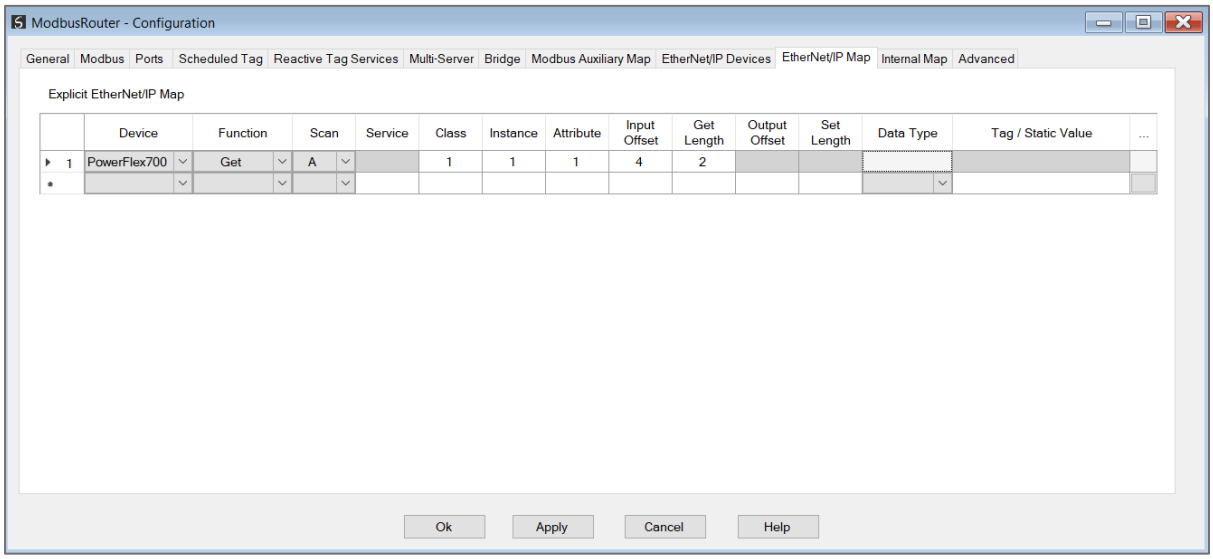


Figure 3.93 – EtherNet/IP Map configuration.

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 MBR/B 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 an EtherNet/IP Device, the MBR/B 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 an EtherNet/IP Device, the MBR/B 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. **NOTE:** When the function is Logix Read, then the Get Length will be the number of elements of the configured data type and **not** the byte count. 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. **NOTE:** When the function is Logix Write, then the Set Length will be the number of elements of the configured data type and **not** the byte count. 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 3.13 – EtherNet/IP Map configuration parameters.

3.10.2.2.3. INTERNAL DATA SPACE MAPPING

When the module is operating as an EtherNet/IP Originator, the data from the EtherNet/IP IO devices can be mapped to the Modbus Registers 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.

IDS COPY – ETHERNET/IP ORIGINATOR SOURCE

When copying data from an EtherNet/IP IO to the Modbus Registers, the source type needs to be EIP Originator.

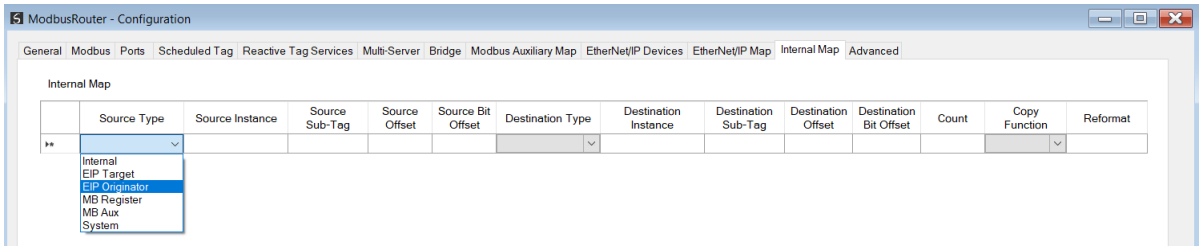


Figure 3.94 – IDS Copy – EtherNet/IP Originator Source Type.

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

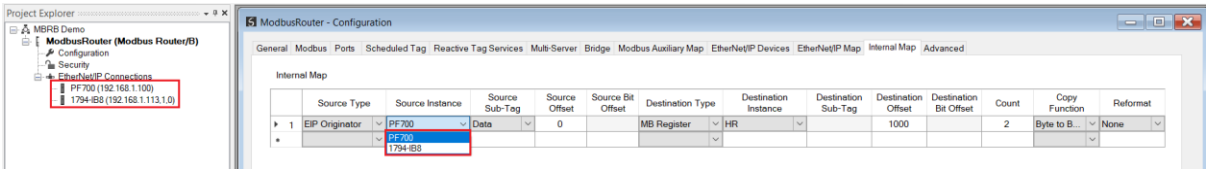


Figure 3.95 – IDS Copy – EtherNet/IP Originator Source Instance.

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.

The user can select to copy either the Data, or Status, from the EtherNet/IP connection.

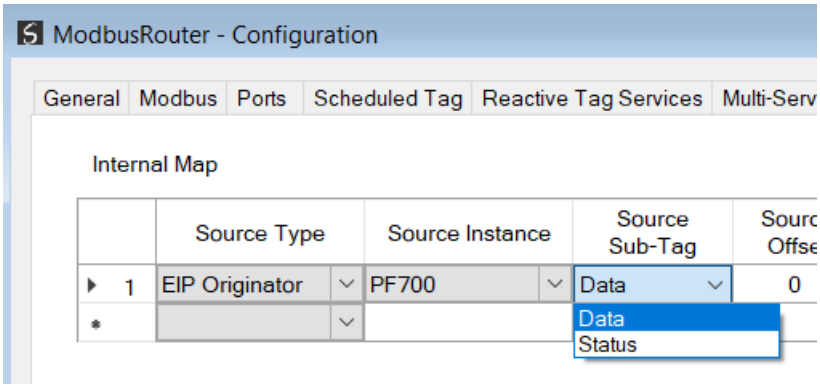


Figure 3.96 – IDS Copy – EtherNet/IP Originator Status.

When selecting the Status, the format of the Status information is shown below:

Parameter	Data Type	Description
EtherNet/IP Originator Connection Status	DINT	Bit 0 – Connection Ok

Table 3.14 – EtherNet/IP Originator Connection Status.

IDS COPY – ETHERNET/IP TARGET DESTINATION

When copying data from the Modbus Registers to an EtherNet/IP IO device **Output** Assembly, the destination type needs to be EIP Originator.

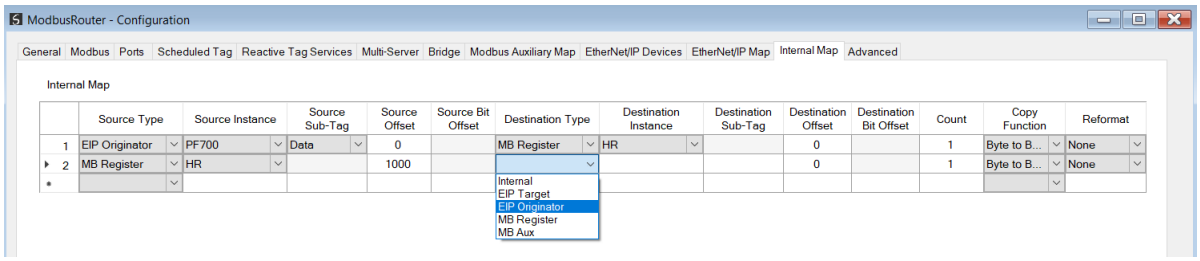


Figure 3.97 – IDS Copy – EtherNet/IP Originator Destination Type.

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

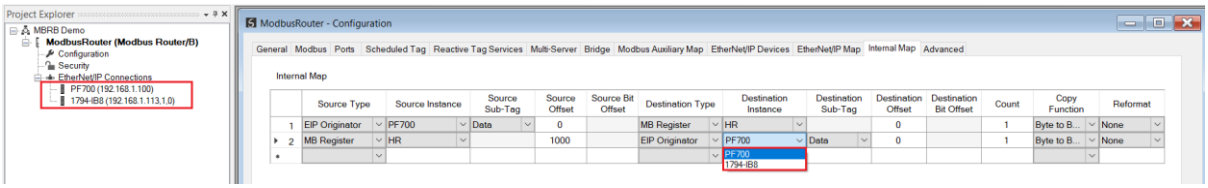


Figure 3.98 – IDS Copy – EtherNet/IP Originator Destination Instance.

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.

3.11. BRIDGE MODE

In Bridge Mode, the MBR/B can operate as a transparent link for Modbus, between different media (TCP, RTU232, and RTU485). This allows, for example, a Modbus client on Modbus TCP to communicate with Modbus devices on RS232 or RS485 and vice versa.

The Bridge mapping will need to be configured once the Bridge Mode has been selected in the *Router Mode*. This will instruct the MBR/B which Modbus requests for a specific Node and from a specific Port (e.g., TCP, RTU232, or RTU485), needs to be routed to which Modbus Server Node and which Port (e.g., TCP, RTU232, or RTU485).

In the below example, any Modbus request received on the EtherNet TCP port for a Modbus Server at Node address 5 (to the IP address of the MBR/B 192.168.1.100), from the Modbus

Client at IP 192.168.1.50, will be routed to the RS232 port and sent to Modbus Server at Node 5. The response will then be routed back to the request originator (i.e., the Modbus Client at IP 192.168.1.50).

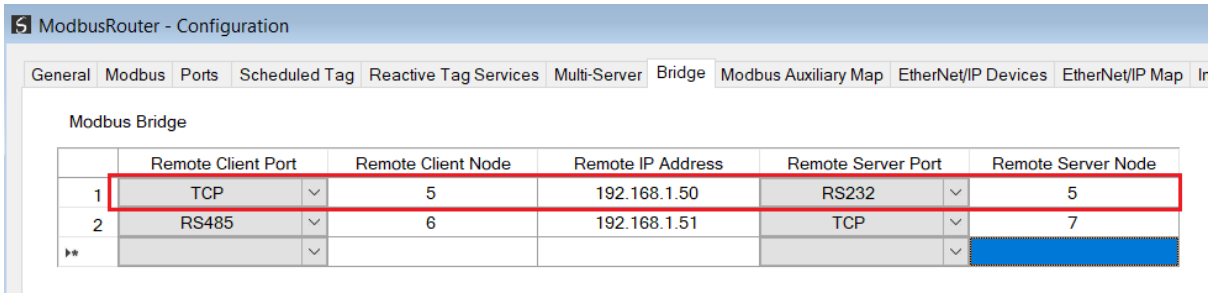


Figure 3.99 – Bridge Mode Map – TCP to RS232

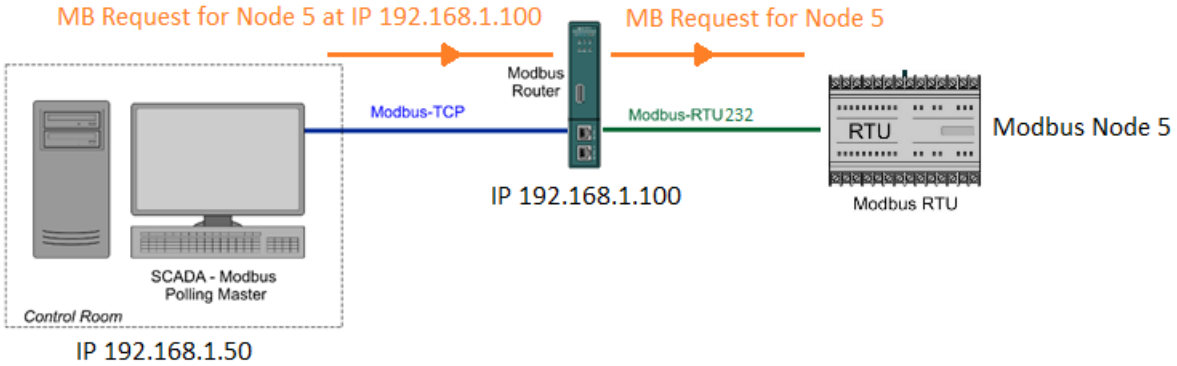


Figure 3.100 – Bridge Mode Diagram – TCP to RS232.

In the next example below, any Modbus request received on the RS485 port for a Modbus Server at Node address 6, will be routed to the Ethernet TCP port and sent to Modbus Server Node 7 with IP address 192.168.1.51. The response will then be routed back to the request originator (the Modbus Client on RS485).

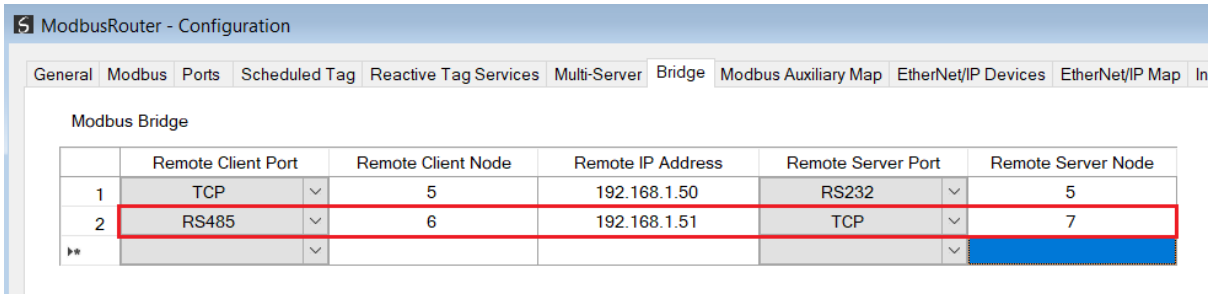


Figure 3.101 – Bridge Mode Map – RS485 to TCP.

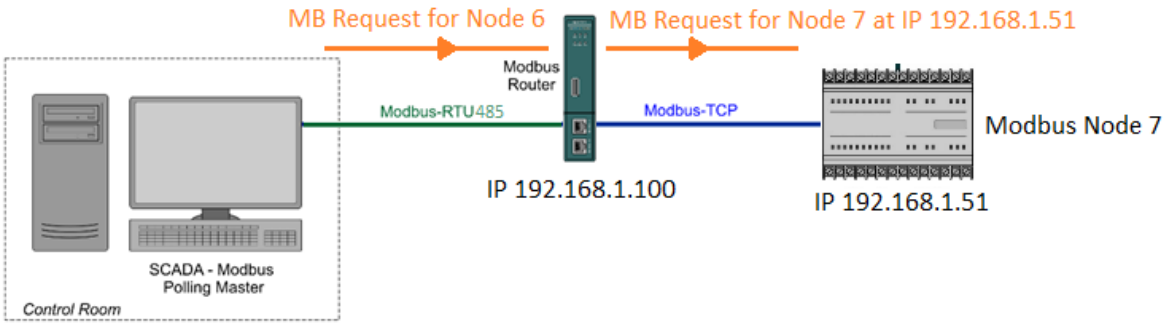


Figure 3.102 – Bridge Mode Diagram – RS485 to TCP.

NOTE: The same remote Client cannot be added more than once.

NOTE: When routing from TCP to Serial (RS232 or RS485), care must be taken when using multiple Modbus Clients to ensure that only one Modbus request is sent at a time on the serial network.

3.12. CONCENTRATOR MODE

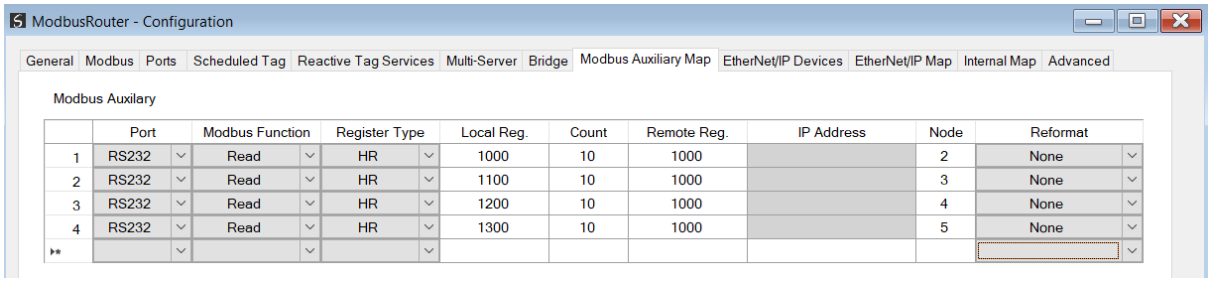
In Concentrator Mode, the MBR/B will collect data from various Modbus devices and provide the consolidated data to one or more Modbus Clients. The MBR/B will be a Modbus Server on one port, and a Modbus Client on the other ports. This will allow a Modbus SCADA to efficiently read the data from many Modbus devices, because the data from the various Modbus devices have been concentrated into the MBR/B.

The Concentrator Server Port will need to be configured in the Modbus parameters from which the Modbus SCADA / Client will read and write. The Modbus Auxiliary Map will then need to be populated to retrieve the data from the various remote Modbus Server devices (similar, to when operating in Enhanced Mode as a Modbus Client).

In the example below the Modbus SCADA is communicating on Modbus TCP to the MBR/B which is serving as a proxy for the Modbus RTU devices in the field. The MBR/B will collect data from the Modbus RTU devices via the radio network and save it in its internal Modbus Registers. Modbus HR 1000 will be read from each node and placed at 100 intervals in the MBR/B internal MB registers (e.g., data from node 2 will be at HR1000, data from node 3 will be at HR1100, data from node 4 will be at HR1200, etc.).

The Modbus SCADA will then be able to read that data from the MBR/B at the specific Modbus Registers.

NOTE: The module will still support an EtherNet/IP connection to a Logix controller to allow monitoring the Modbus data. The internal mapping can be used to map Modbus data to the EIP connection.



	Port	Modbus Function	Register Type	Local Reg.	Count	Remote Reg.	IP Address	Node	Reformat
1	RS232	Read	HR	1000	10	1000		2	None
2	RS232	Read	HR	1100	10	1000		3	None
3	RS232	Read	HR	1200	10	1000		4	None
4	RS232	Read	HR	1300	10	1000		5	None

Figure 3.103 – Concentrator Mode – Modbus Auxiliary Mapping.

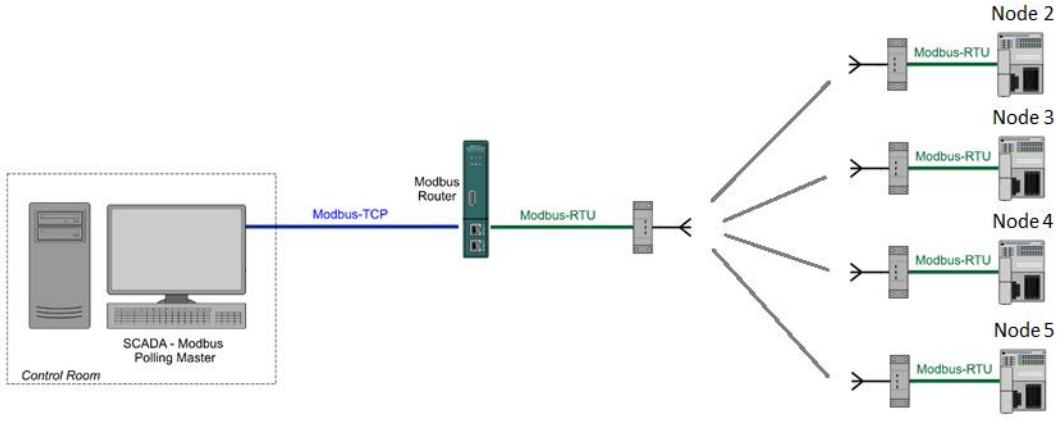


Figure 3.104 – Concentrator Mode Diagram.

3.13. INTERNAL DATA SPACE MAP

The internal data map is used to exchange data from the Ethernet interface to the Modbus Registers and vice versa. Up to 300 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.

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 3.15 – 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 3.16 – Internal Map Reformat Options.

3.13.1. COPY FROM

One of six sources can be selected to copy from:
Internal, EIP Target, EIP Originator, MB Register, MB Aux, and System.

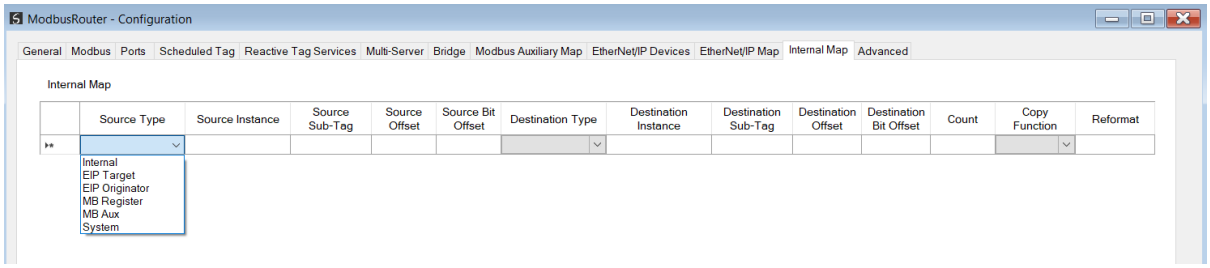


Figure 3.105 – Internal Map – Source Type.

3.13.1.1. INTERNAL

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

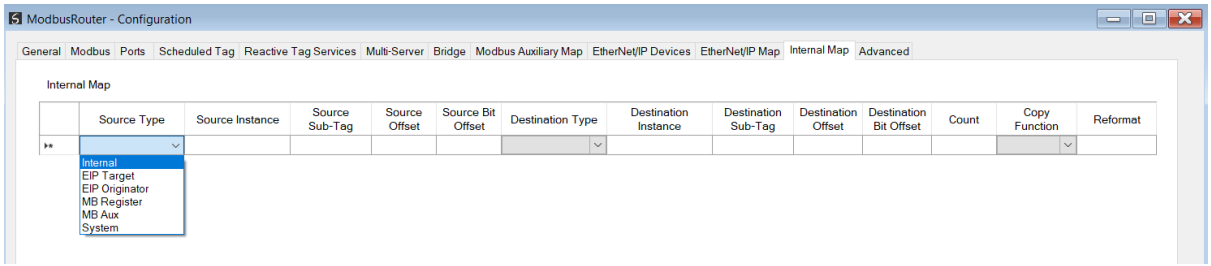


Figure 3.106 – IDS Copy – Internal Source Type.

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.

3.13.1.2. EIP TARGET

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

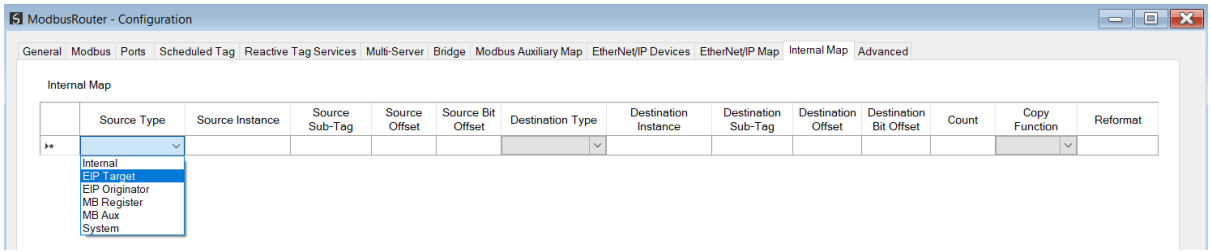


Figure 3.107 – IDS Copy – EtherNet/IP Target Source Type.

The source instance will be the connection number, which can be connection 1 to 4, based on the number of connections configured.

NOTE: Connection 0 cannot be selected because the first connection is used for module status. Connection 1 to 4 is used for Modbus data mapping.

The Source Offset is the offset in the EtherNet/IP output assembly 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.

3.13.1.3. EIP ORIGINATOR

When copying data from an EtherNet/IP IO to the Modbus Registers, the source type needs to be EIP Originator.

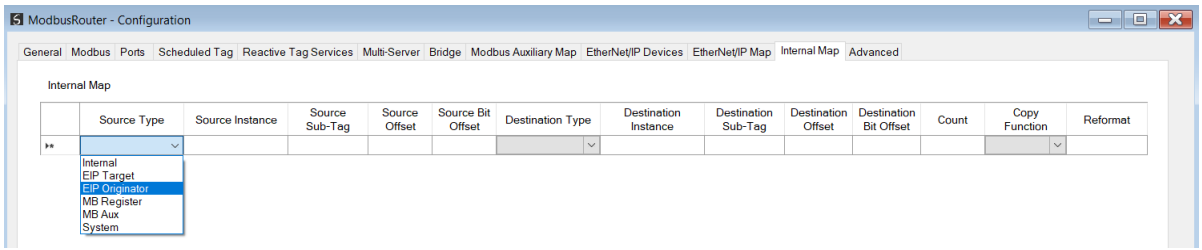


Figure 3.108 – IDS Copy – EtherNet/IP Originator Source Type.

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

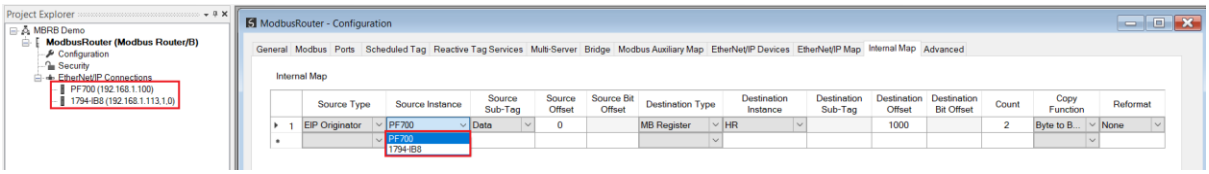


Figure 3.109 – IDS Copy – EtherNet/IP Originator Source Instance.

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.

The user can select to copy either the Data, or Status, from the EtherNet/IP connection.

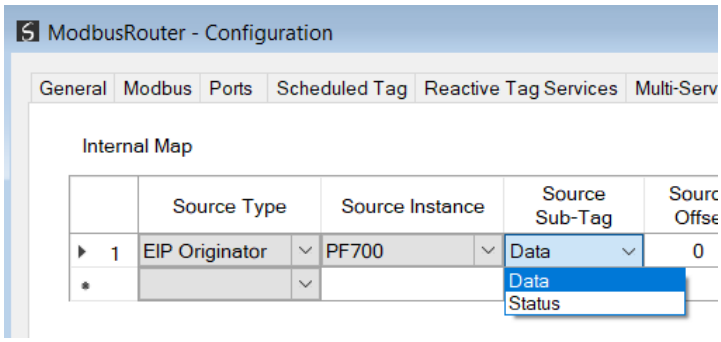


Figure 3.110 – IDS Copy – EtherNet/IP Originator Status.

When selecting the Status, the format of the Status information is shown below:

Parameter	Data Type	Description
EtherNet/IP Originator Connection Status	DINT	Bit 0 – Connection Ok

Table 3.17 – EtherNet/IP Originator Connection Status.

3.13.1.4. MB REGISTER

When copying Modbus Data or Status to the EtherNet/IP interface, the source type needs to be MB Register.

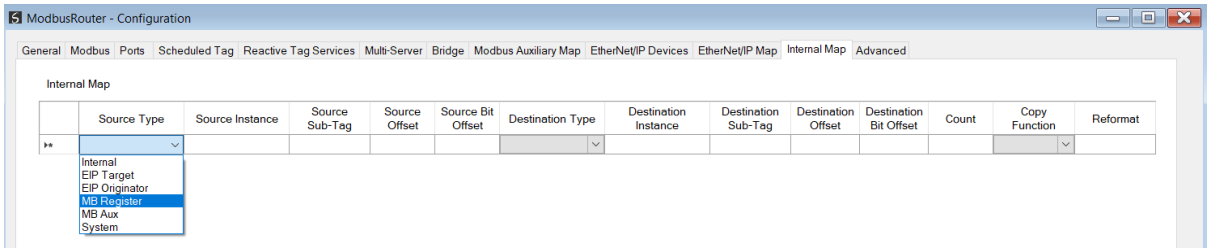


Figure 3.111 – IDS Copy - Modbus Register Source Type.

When using the MB Register source, the source instance will be the Modbus register type required.

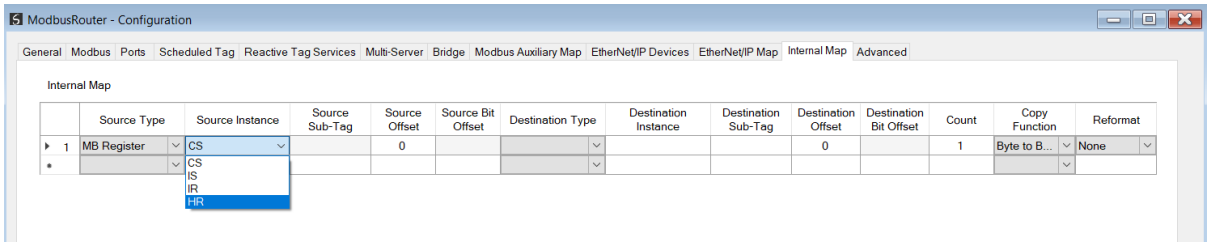


Figure 3.112 – IDS Copy - Modbus Register Source Instance.

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.

3.13.1.5. MB AUX

When using the MB Aux source, the source instance will be an item in the Modbus Auxiliary Map. For example, if there are two items in the Modbus Auxiliary Map, then there will be two Source Instances for the MB Aux Source Type.

NOTE: The Modbus Aux source is only relevant when the MBR/B is operating in Enhanced mode as a Modbus Client.

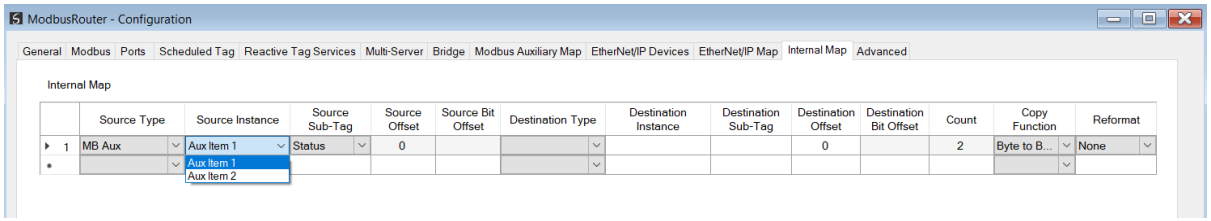


Figure 3.113 – IDS Copy - Modbus Register Source Instance.

The MB Aux format is as shown below:

Parameter	Data Type	Description
MB Aux Status	INT	Bit 0 – Last Transaction Ok

Table 3.18 – Modbus Aux Item Status.

3.13.1.6. SYSTEM

When copying system information, the source type needs to be System.

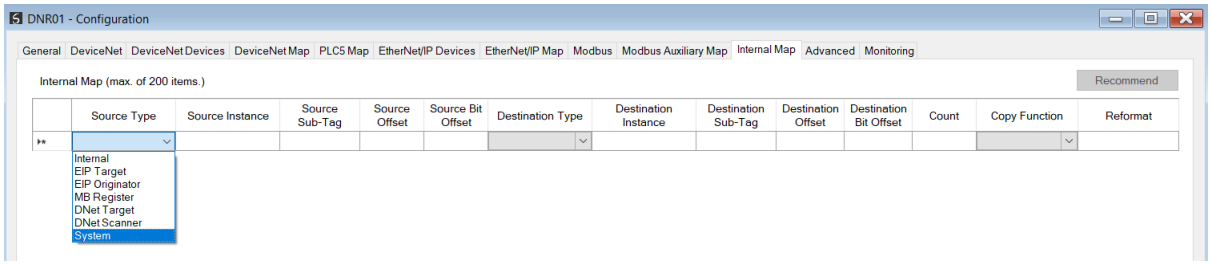


Figure 3.114 – IDS Copy – System Information.

The module's System information has the following format.

Parameter	Data Type	Description
Status	INT	Module Status. Bit 0 – Module Config Valid Bit 1 – Routing Inhibited Bit 2 – Disabled by Logix Standby Bit 3 – EtherNet/IP Originator Comms Ok

		Bit 4 – EtherNet/IP Target Comms Ok Bit 5 – Controller Run (1)/Prog Mode (0) Bit 6 – NTP Ok Bit 7 – Modbus Comms Ok Bit 8 – Modbus Client (0) / Server (1)
ConfigCRC	INT	The module configuration signature.
Mode	SINT	0 – Scheduled 1 – Reactive 2 – Unscheduled 3 – (Reserved) 4 – Enhanced 5 – Bridge 6 – Concentrator
Protocol	SINT	**Only valid for Scheduled, Reactive, and Unscheduled Mode** 0 – Serial 1 – TCP
Transaction Rate	INT	**Only valid for Scheduled and Reactive Mode** The transaction rate is the number of Modbus messages per second that the module is currently routing.
Scheduled Tag Count	INT	**Only valid for Scheduled Mode** The number of items in the Scheduled Tag Map.
Logix Redundant State	INT	**Only valid for Scheduled and Reactive Mode** When the module is operating in Logix – Redundant USB this field will show the redundant status of the local Logix controller connected via USB. **0x02 - Primary SYNC** The local Logix controller is the primary and is synchronized with the secondary. **0x03 - Primary DISQ** The local Logix controller is the primary and the secondary controller is disqualified. **0x04 - Primary ALONE** The local Logix controller is the primary with no secondary controller. **0x10 - Primary Locked** The local Logix controller is the primary and is locked for changes. **0x08 - Secondary SYNC** The local Logix controller is the secondary and is synchronized with the primary.

		0x09 - Secondary DISQ The local Logix controller is the secondary and is disqualified with the primary. **0x0E - No Partner** The local Logix controller is the secondary with no controlling partner. **0x12 - Secondary Locked** The local Logix controller is the secondary and is locked for changes.

Table 3.19 – System Information Format.

3.13.2. COPY TO

One of four destinations can be selected to copy to:
Internal, EIP Target, EIP Originator, and MB Register.

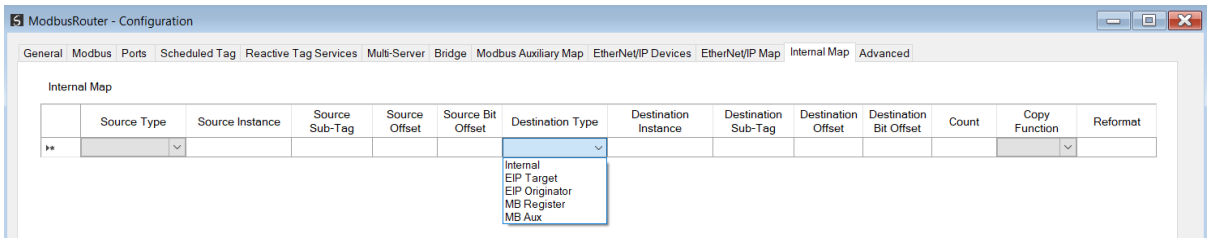


Figure 3.115 – Internal Map – Destination Type.

3.13.2.1. INTERNAL

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

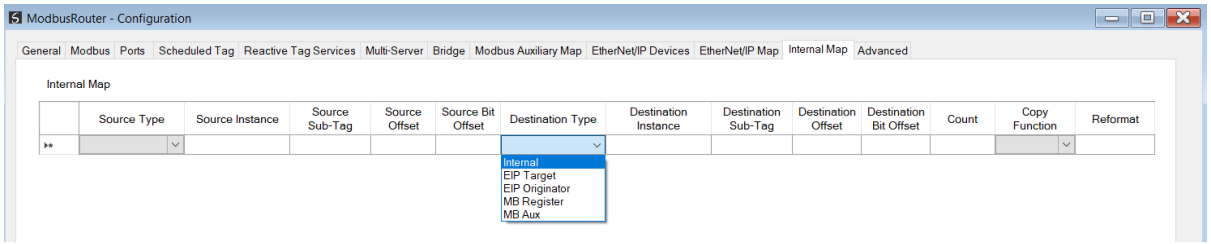


Figure 3.116 – IDS Copy – Internal Source Type.

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.

3.13.2.2. EIP TARGET

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

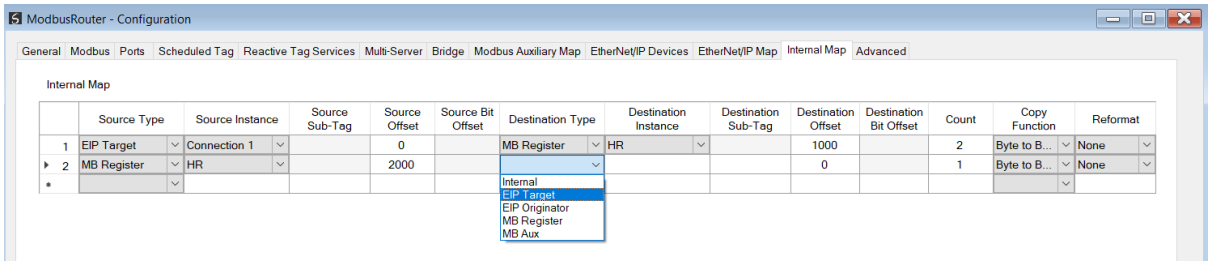


Figure 3.117 – IDS Copy – EtherNet/IP Target Destination Type.

The destination instance will be the connection number, which can be connection 1 to 4, based on the number of connections configured.

NOTE: Connection 0 cannot be selected because the first connection is used for module control. Connection 1 to 4 is used for Modbus data mapping.

The Destination Offset is the offset of the EtherNet/IP input assembly from where the data must be copied. The Count is the number of **bytes** that will be copied.

3.13.2.3. EIP ORIGINATOR

When copying data from the Modbus Registers to an EtherNet/IP IO device **Output** Assembly, the destination type needs to be EIP Originator.

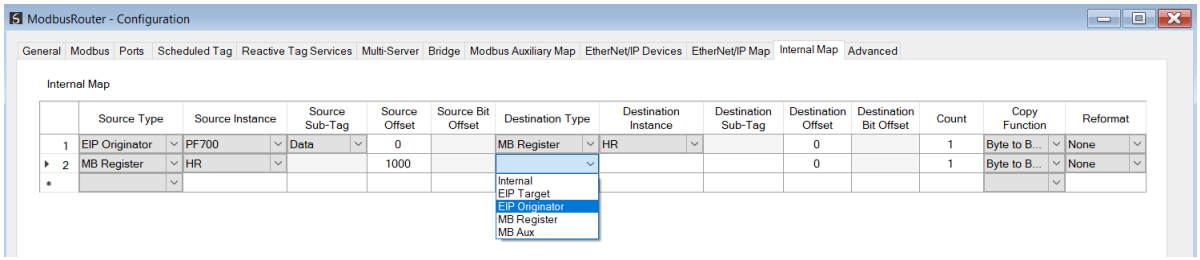


Figure 3.118 – IDS Copy – EtherNet/IP Originator Destination Type.

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

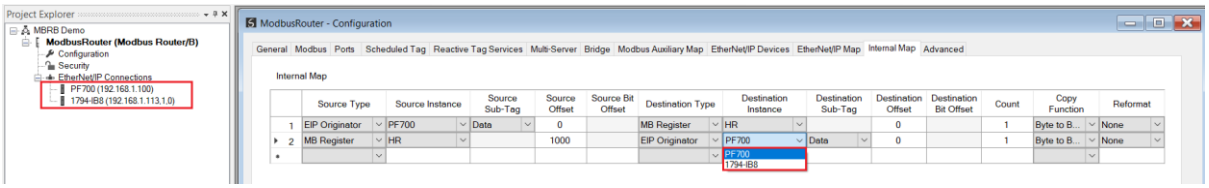


Figure 3.119 – IDS Copy – EtherNet/IP Originator Destination Instance.

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.

3.13.2.4. MB REGISTER

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

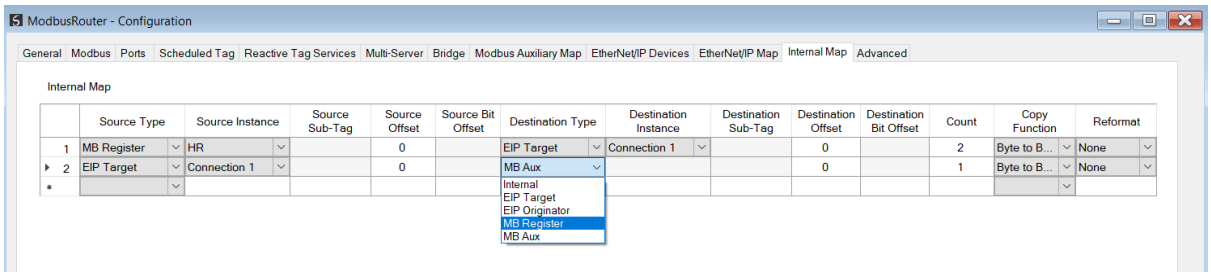


Figure 3.120 – IDS Copy - Modbus Register Destination Type.

The destination instance will be the Modbus register type required.

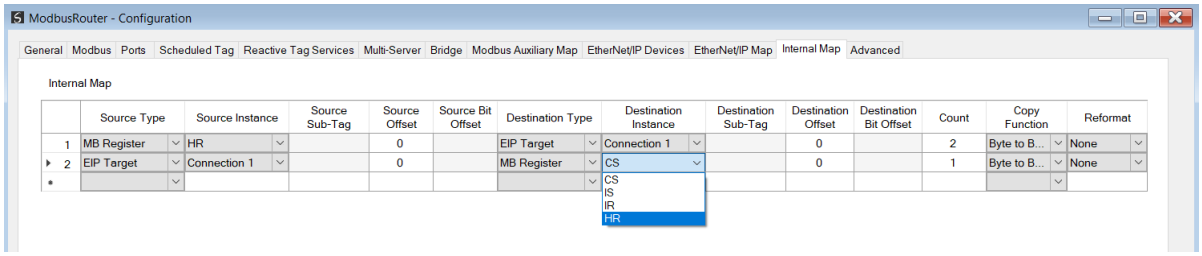


Figure 3.121 – IDS Copy - Modbus Register Destination Instance.

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.

3.14. ADVANCED

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

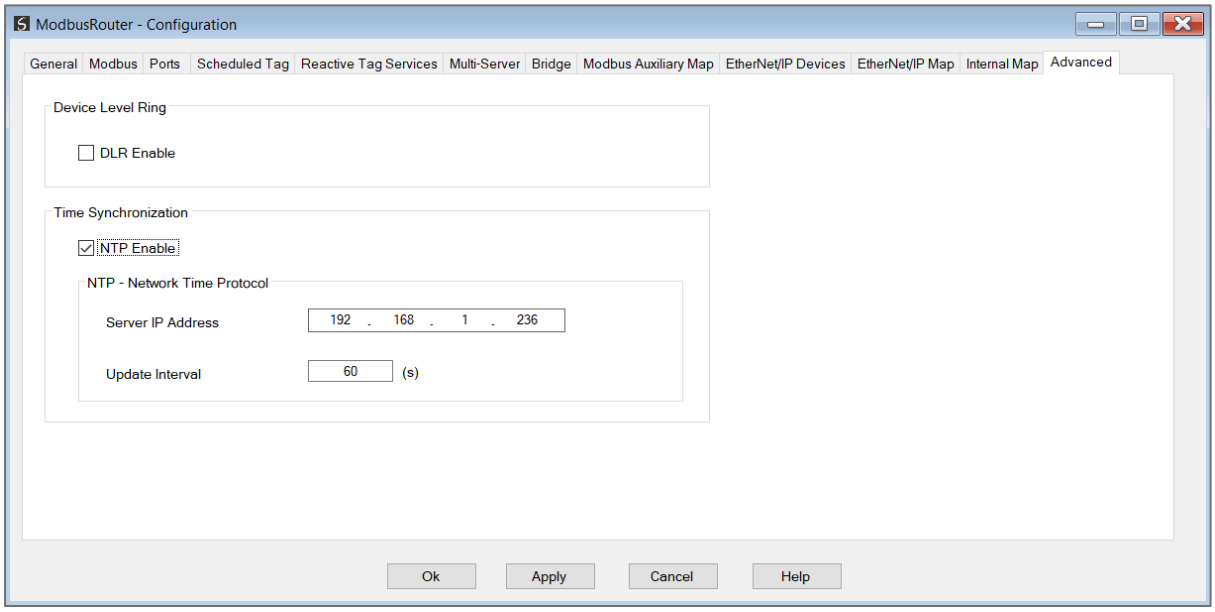


Figure 3.122 – Advanced configuration

The Advanced configuration consists of the following parameters:

Parameter	Description
DLR Enable	**THIS IS ONLY RELEVANT IN SWITCH (LINEAR/DLR) ETHERNET SWITCH MODE** Enable Device Level Ring operation.
NTP Enable	The MBR/B 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 MBR/B will request time from the NTP Server.

Table 3.20 – Advanced configuration parameters.

3.15. MODULE DOWNLOAD

Once the Modbus 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 Modbus 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.

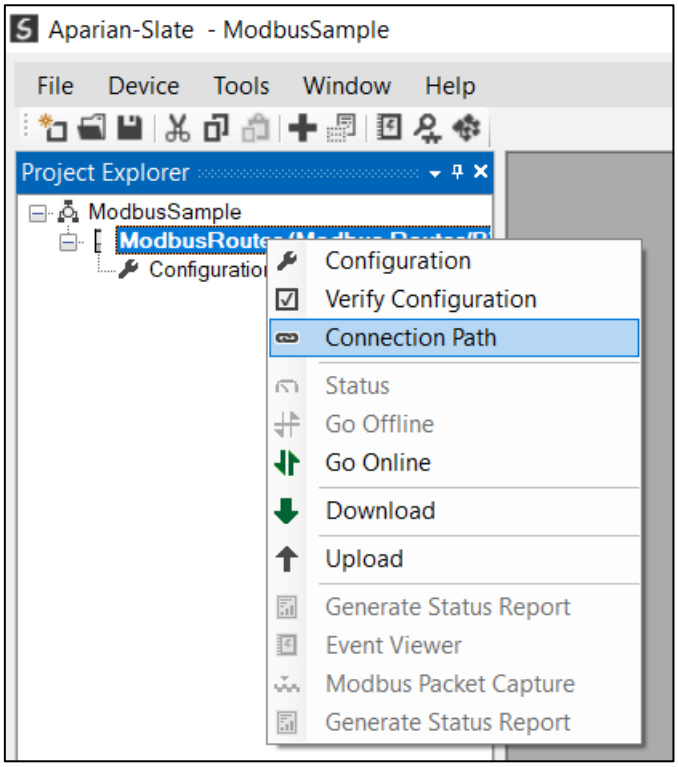


Figure 3.123 - Selecting Connection Path.

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

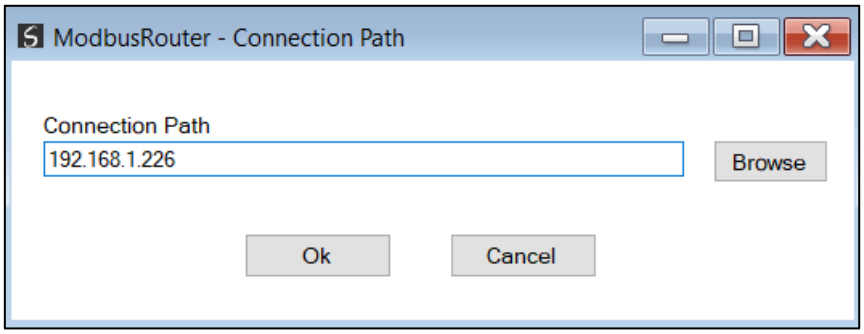


Figure 3.124 - Connection Path.

When the module has been setup with Controller Mode *Logix – USB Redundant* the user will need to enter the connection path for both Modbus Router A and B.

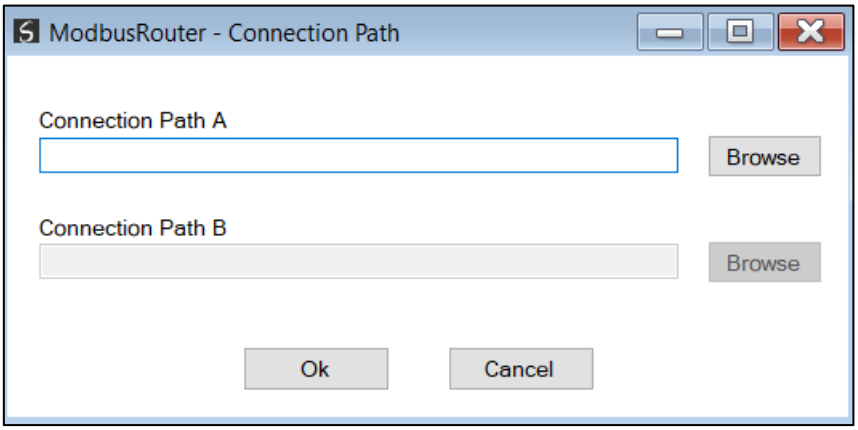


Figure 3.125 - Connection Path (Redundant Modules – see Redundant section).

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

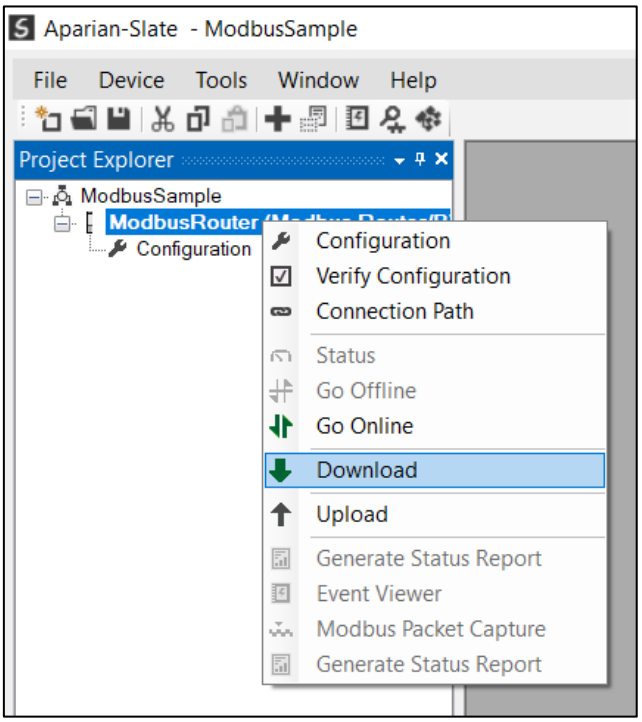


Figure 3.126 - Selecting Download.

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

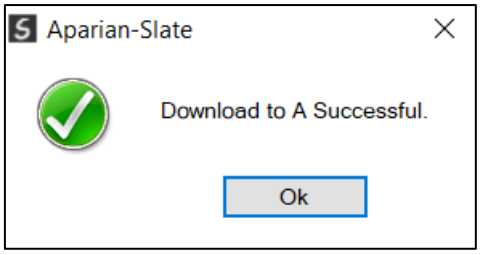


Figure 3.127 - Successful download.

The module is now configured and will start operating immediately.

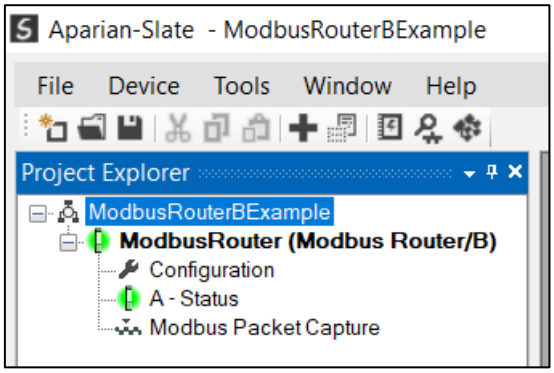


Figure 3.128 - Module online.

4. DEVICE FIRMWARE UPDATE

The MBR/B 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.

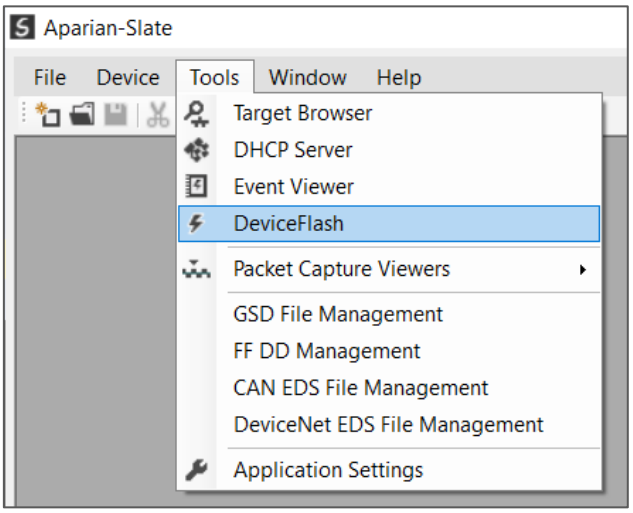


Figure 4.1 – Select DeviceFlash utility from Slate.

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

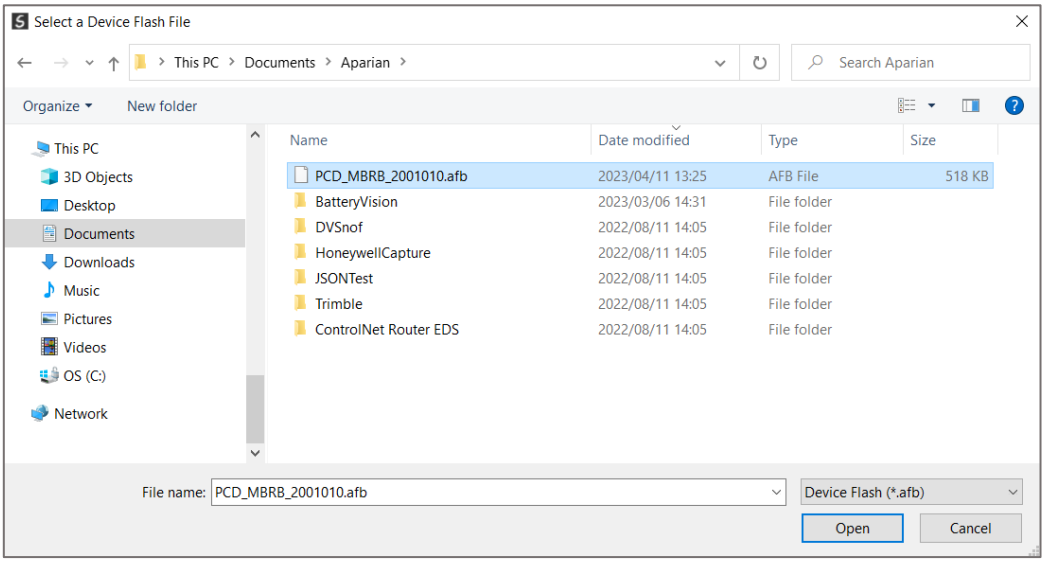


Figure 4.2 – Select the binary file.

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

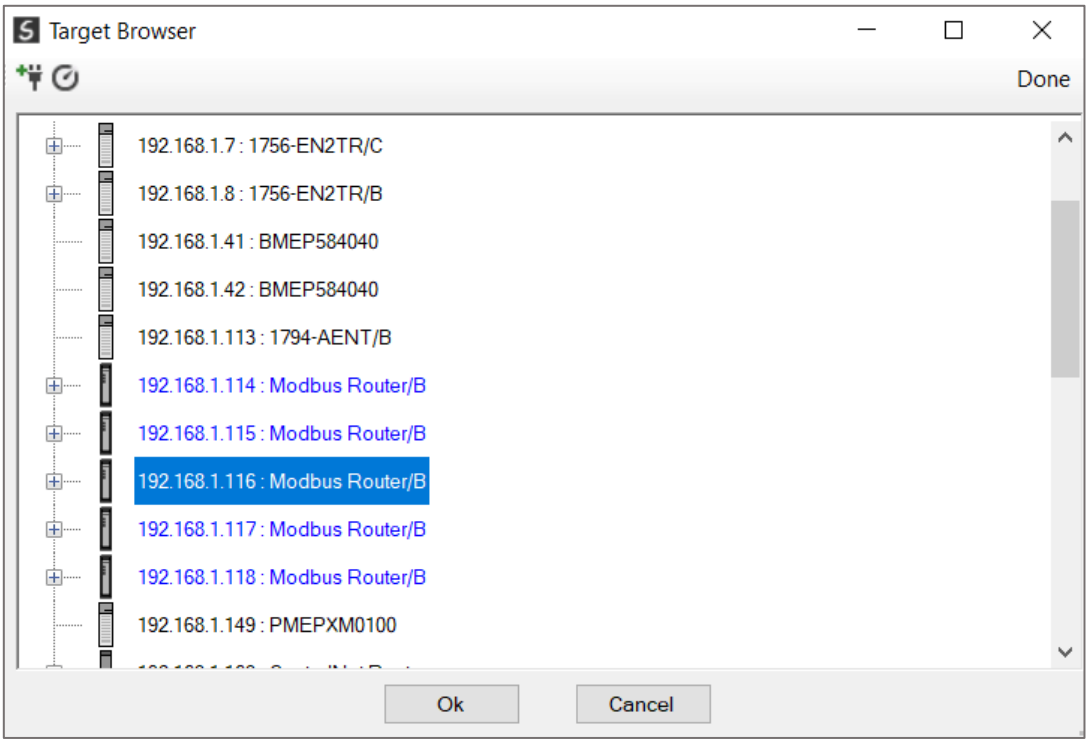


Figure 4.3 – Select the device to be updated.

- 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.

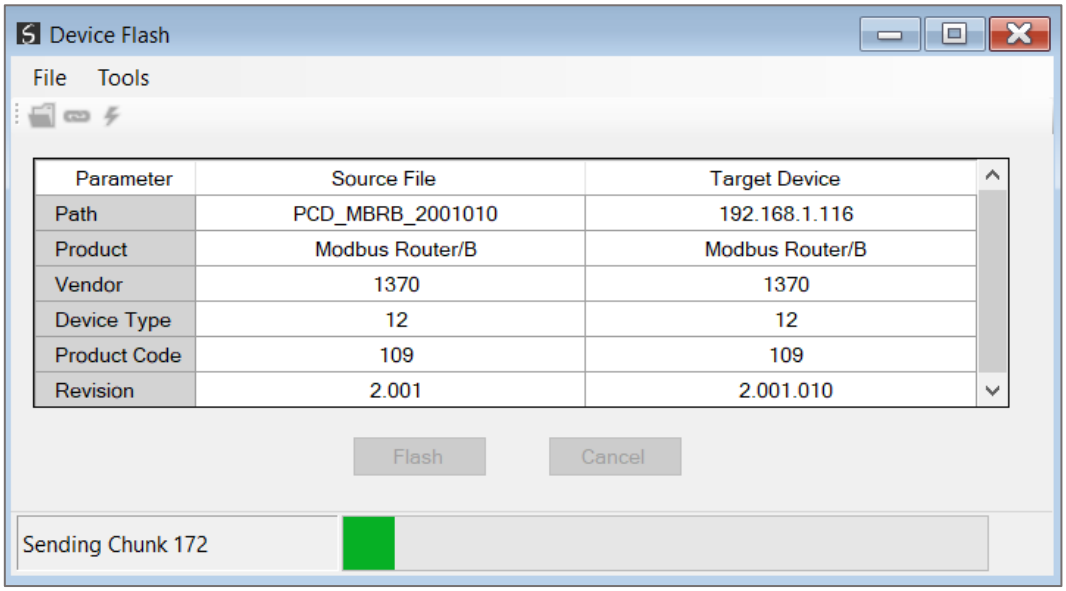


Figure 4.4 – Firmware update busy.

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

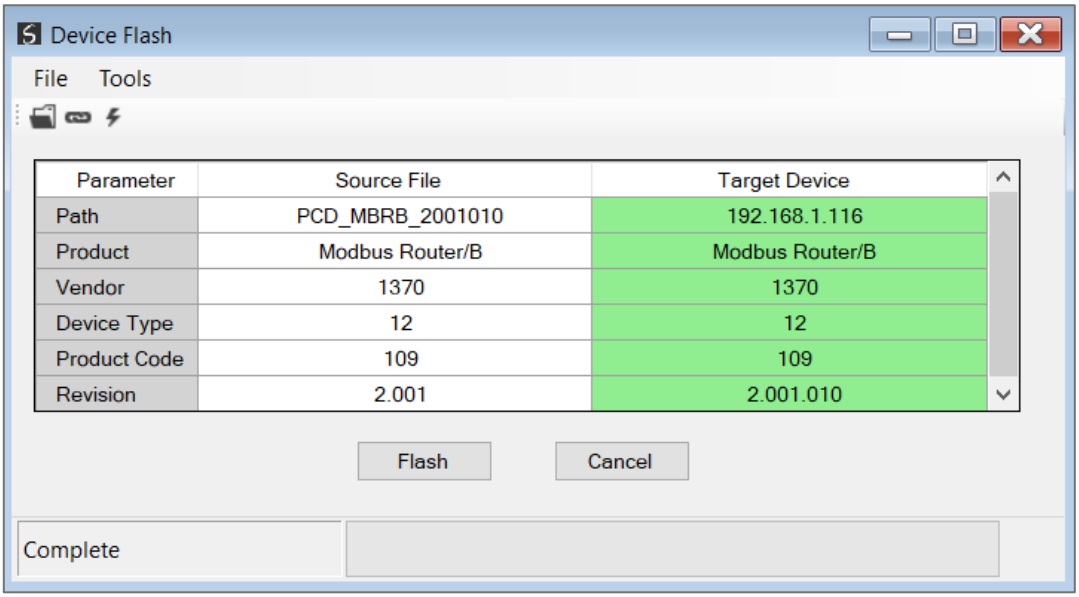


Figure 4.5 – Firmware update successfully completed.

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.

5. SECURITY

The MBR/B supports security services allowing the user to configure various levels of module security.

When security is enabled all EtherNet/IP communication between Slate and the module is encrypted preventing information from being intercepted by a third party.

Security of the module is configured and downloaded separately to the module's primary (application) configuration and can thus be configured online without affecting the running process.

The security services provide the following features:

- Configurable level of security
- Encrypted EtherNet/IP communication
- Fixed User Roles and Custom Users
- Configurable Password rules
- Login Expiry
- Failed Login Cool-off and Account Disable options
- Configurable Global Services and Custom Ports
- Optional IP Address Access Control List
- Optional MAC Access Control List
- Restrictions on Reset command
- Restrictions on Device Flash options
- Security Audit trail in Event Log

Optional user authentication can be enabled, requiring users to login to the module when connecting. In addition to the fixed role users, (viz. Administrator, Engineer, Operator, Auditor and Viewer) up to 5 custom users can be configured. Each custom user is assigned one of the aforementioned fixed roles.

Once user authentication has been initialized, only users with Administrator privileges are able to change the security configuration and reset passwords etc.

Allowed privileges for each user role is tabulated below:

Privileges / Roles	Administrator	Engineer	Operator	Auditor	Viewer	None
System						
Identity Object	✓	✓	✓	✓	✓	✓
Device Flash (Firmware Upgrade)	✓	-	-	-	-	-
Reset Module	✓	✓	-	-	-	-
Change Ethernet Interface Settings	✓	✓	-	-	-	-
Application						
Download/Upload Configuration	✓	✓	-	-	-	-
View Status and Statistics	✓	✓	✓	✓	✓	-
Clear Statistics	✓	✓	✓	-	-	-
Packet Capture	✓	✓	✓	✓	-	-
View Event Log	✓	✓	✓	✓	✓	-
Clear Event Log	✓	-	-	-	-	-
Generate Status Report	✓	✓	✓	✓	-	-
Modbus Pass Through Functions	✓	✓	-	-	-	-
Internal Register Viewer	✓	✓	-	-	-	-
Security						
Download/Upload Security Configuration	✓	-	-	-	-	-
Clear All Security	✓	-	-	-	-	-
View Security Status and Statistics	✓	✓	-	✓	-	-
Clear Security Statistics	✓	-	-	-	-	-
Change Own Password	✓	✓	✓	✓	✓	-
Change Another User's Password	✓	-	-	-	-	-

Table 5.1 – User Role Privileges.

5.1. SECURING PROCESS

The process to secure a module involves the following basic steps:

- Setup Security **Configuration**
- **Download** Security Configuration to the module
- **Initialize** User Authentication

5.1.1. CONFIGURATION

The Security Configuration is accessed by right-clicking on the ***Security*** item in the device tree and selecting the ***Security Configuration*** option.

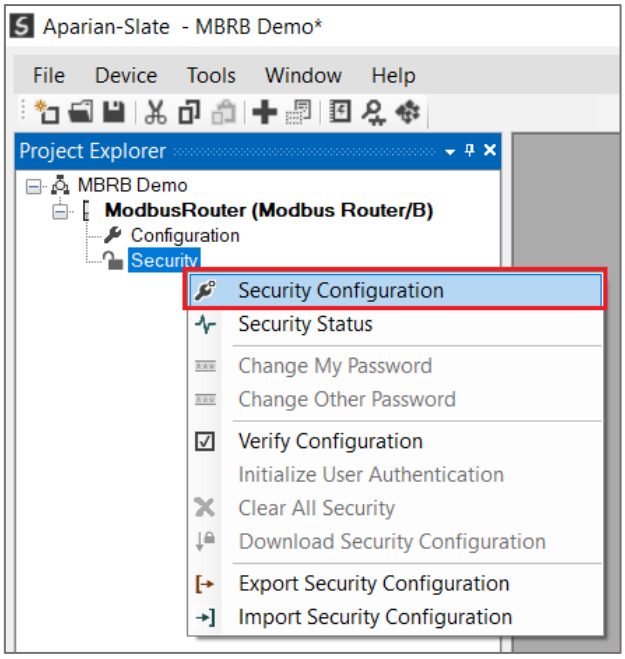


Figure 5.1 – Selecting Security Configuration.

The configuration form will open and display multiple configuration tabs.

5.1.1.1. General Configuration

The ***General*** configuration tab is shown below.

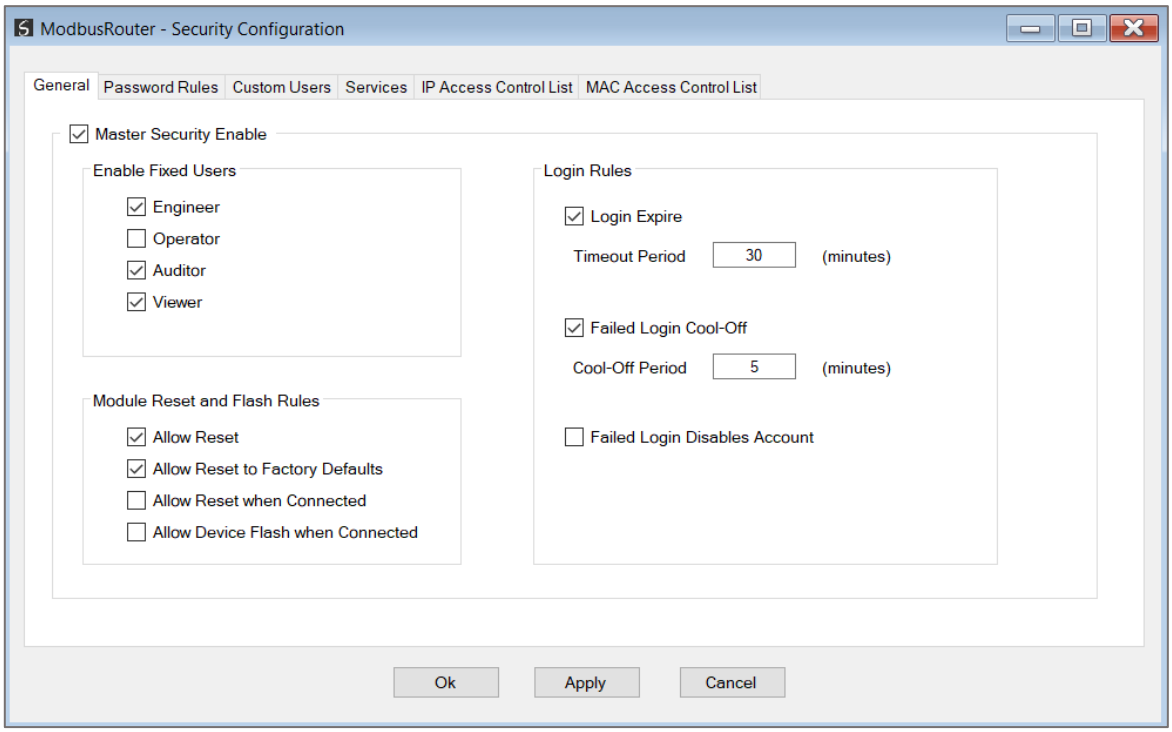


Figure 5.2 – Security Configuration – General.

The ***General*** configuration tab comprises the following parameters:

Parameter	Description
Master Security Enable	Used to enable or disable the master security. **NOTE**: If this option is disabled then all Security measures are disabled.
Enable Fixed Users ***Engineer*** ***Operator*** ***Auditor*** ***Viewer***	Allows each of the fixed users to be enabled or disabled. **NOTE**: The fixed ***Administrator*** user cannot be disabled.
Module Reset and Device Flash Rules	
Allow Reset	Must be checked to allow any type of Reset via EtherNet/IP.
Allow Reset to Factory Defaults	Must be checked to allow Reset to Factory Defaults (including Reset to Factory Defaults except Communication settings).
Allow Reset when Connected	Must be checked to allow any type of Reset when the module is connected, where connected implies an EtherNet/IP Class 1 connection.
Allow Device Flash when Connected	Must be checked to allow Device Flash when the module is connected, where connected implies an EtherNet/IP Class 1 connection.
Login Rules	
Login Expire	Selecting this option causes all users to be automatically logged-out after the Login Expire Timeout Period.
Login Expire Timeout Period	The duration, in minutes, after which all users will be automatically logged-out. Min: 1 minute Max: 1440 minutes (24 hours)
Failed Login Cool-Off	Selecting this option initiates a Cool-off period after a user has entered an incorrect password 3 times in succession. During the cool off period the user will not be able to login. **NOTE**: This option, and the Failed Login Disable Account option, cannot be simultaneously selected.
Failed Login Cool-Off Period	The Cool-Off duration, in minutes. Once a user has entered an incorrect password 3 times in succession, they would need to wait this period before they can login again. Min: 1 minute Max: 1440 minutes (24 hours)

Failed Login Disables Account	Selecting this option causes the user's account to be disabled if they enter an incorrect password 3 times in succession. **NOTE**: Once the user's account has been disabled, an Administrator would need to reset their password. **NOTE**: This option, and the Failed Login Cool-Off option, cannot be simultaneously selected. **NOTE**: This option will not be applicable for the Administrator user account.

Table 5.2 – Security Configuration – General.

5.1.1.2. PASSWORD RULE CONFIGURATION

The ***Password Rules*** tab is shown below.

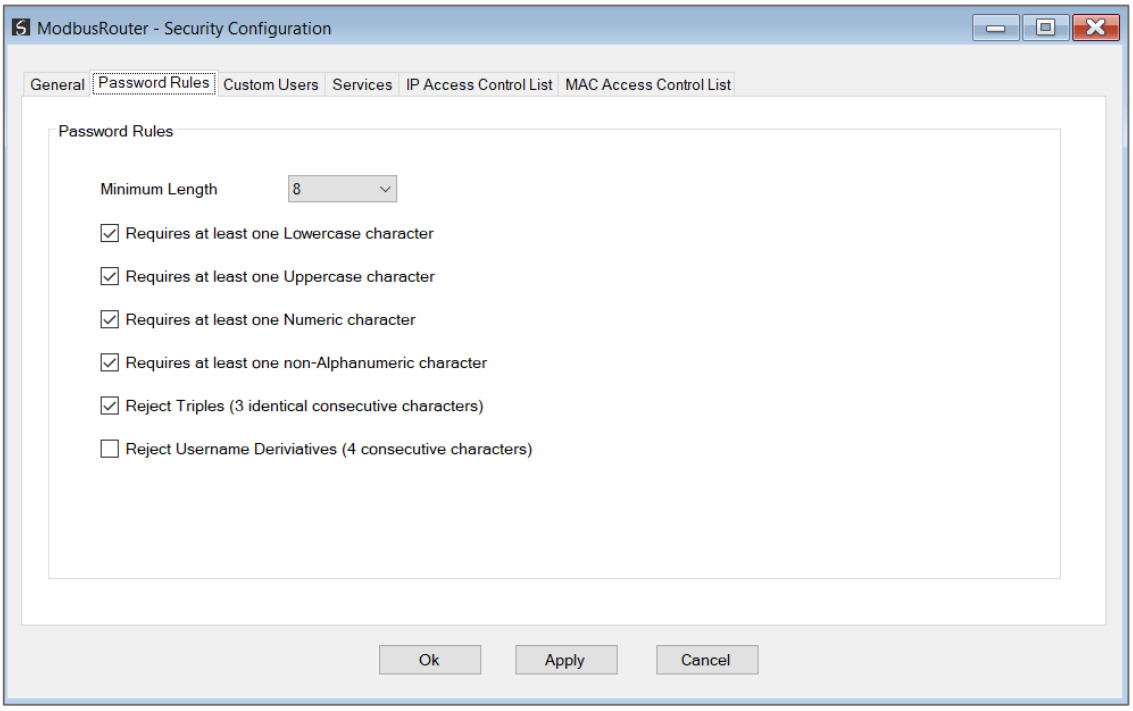


Figure 5.3 – Security Configuration – Password Rules.

The ***Password Rules*** tab comprises the following parameters:

Parameter	Description
Minimum Length	The minimum number of characters the password must contain. Min: 1

	Max: 32
Requires at least one **Lowercase** character	When selected, all passwords will require at least one lowercase character. e.g. a b c d
Requires at least one **Uppercase** character	When selected, all passwords will require at least one uppercase character. e.g. A B C D
Requires at least one **Numeric** character	When selected, all passwords will require at least one non-alphanumeric character. e.g. 1 2 3 4
Requires at least one **Non-Alphanumeric** character	When selected, all passwords will require at least one non-alphanumeric character. e.g. ! @ # $ %
Reject **Triples**	When selected, all passwords may not contain 3 identical (case-insensitive) consecutive characters. e.g. MyDDDog, strangecCcat.
Reject Username Derivative	When selected, all passwords may not contain any derivative of the respective username. A derivative is defined as any 4 consecutive characters of the password case-insensitive-matching any 4 consecutive characters of the username.

Table 5.3 – Security Configuration – Password Rules.

5.1.1.3. CUSTOM USERS CONFIGURATION

The ***Custom Users*** configuration tab is shown below.

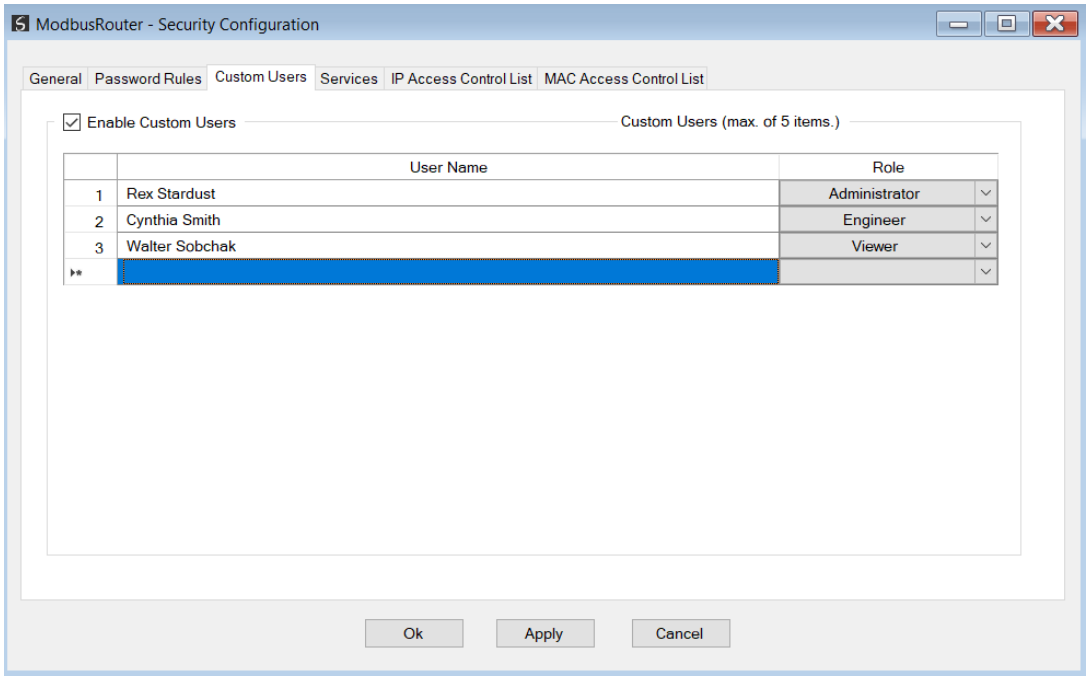


Figure 5.4 – Security Configuration – Custom Users.

The ***Custom Users*** configuration tab comprises the following parameters:

Parameter	Description
Enable Custom Users	Global Enable / Disable for all Custom Users
User Name	User Name of the Custom User. **NOTE**: The User Name cannot exceed 32 characters. **NOTE**: The User Name cannot be the same as any of the fixed roles. (Administrator, Engineer etc.)
Role	The role allocated to this user. Select one of the following: Administrator Engineer Operator Auditor Viewer

Table 5.4 – Security Configuration – Custom Users.

5.1.1.4. SERVICES CONFIGURATION

The ***Services*** configuration tab is shown below.

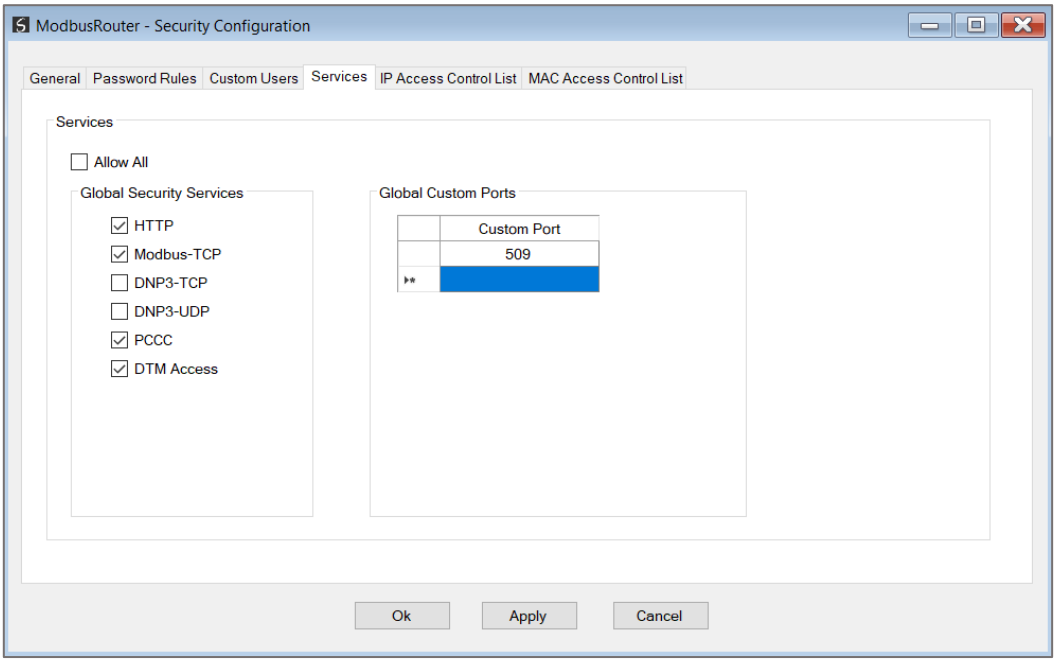


Figure 5.5 – Security Configuration – Services.

The ***Services*** configuration tab comprises the following parameters:

Parameter	Description
Allow All	When selected all services are allowed by default.
HTTP	Allow HTTP web server connections to be accepted.
Modbus-TCP	Allow Modbus-TCP connections to be accepted.
DNP3-TCP	Allow DNP3 TCP connections to be accepted.
DNP3-UDP	Allow DNP3 UDP connections to be accepted.
PCCC	Allow PCCC connections to be accepted.
DTM Access	Allow unencrypted DTM-required passthrough EtherNet/IP transactions. This provides a limited exception to encryption requirement policy for asset management software.
Custom Ports	Up to 4 custom (TCP / UDP) ports can be added to the allowed list. **NOTE**: When the application is using non-standard ports for communication services (e.g. Modus-TCP) then these ports will need to be added to the Custom Ports list.

Table 5.5 – Security Configuration – Services.

5.1.1.5. IP Access Control List Configuration

Enabling the IP Access Control List will block all communication from PCs (and other devices) unless their IP addresses are explicitly included in the list with the allowed services.

The ***IP Access Control List*** configuration tab is shown below.

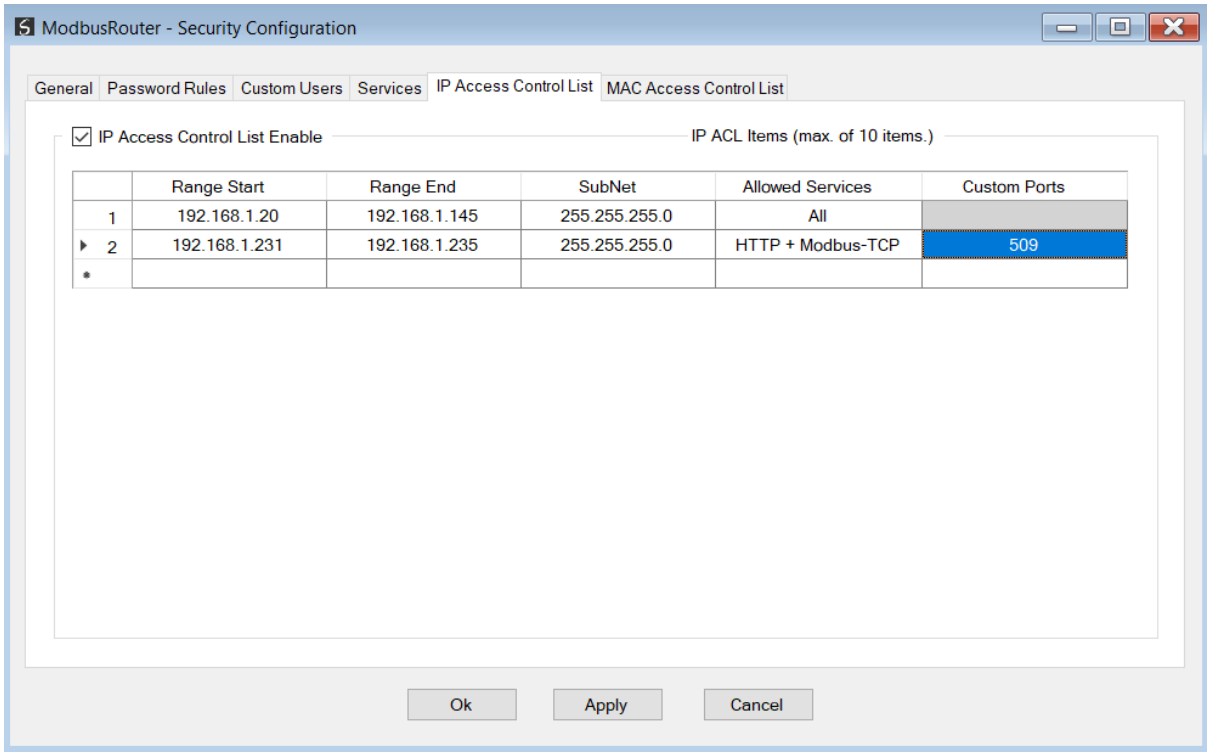


Figure 5.6 – Security Configuration – IP Access Control List.

The ***IP Access Control List*** configuration tab comprises the following parameters:

Parameter	**Description**
IP Access Control List Enable	Enable / Disable for all IP Access Control List
IP Access Control List	
Range Start	First IP address in the range.
Range End	Last IP address in the range.
SubNet	The subnet mask of the range.
Allowed Services	The services allowed for this IP range.
Custom Ports	Additional custom ports allowed for this IP range.

Table 5.6 – Security Configuration – IP Access Control List.

To modify the Allowed Services the user can either click on the specific Allowed Service cell, or right-click on the specific row and select the ***Edit Services*** option.

The Service Selection form will open allowing the user to select each service's checkbox.

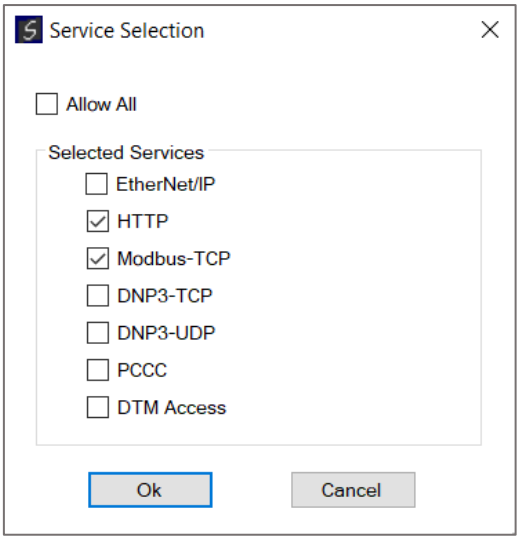


Figure 5.7 – Security Configuration – Service Selection.

To modify the Custom Ports the user can either click on the specific ***Custom Ports*** cell, or right-click on the specific row and select the ***Edit Custom Ports*** option.

NOTE: The Custom Ports option is not available if the ***Allow All*** option has been selected in the Allowed Services.

The ***Custom Ports Selection*** form will open allowing the user to modify the Custom Ports.

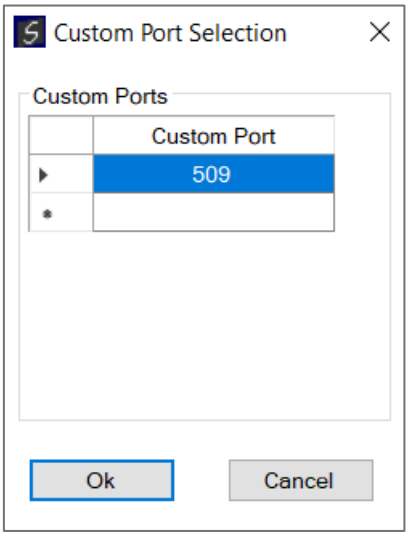


Figure 5.8 – Security Configuration – Custom Ports.

5.1.1.6. MAC Access Control List Configuration

Enabling the MAC Access Control List will block all communication from PCs (and other devices) unless their MAC addresses are explicitly added to the list.

NOTE: Incorrectly configuring the MAC Access Control List may result in the module no longer being able to communicate with Slate.

The ***MAC Access Control List*** configuration tab is shown below.

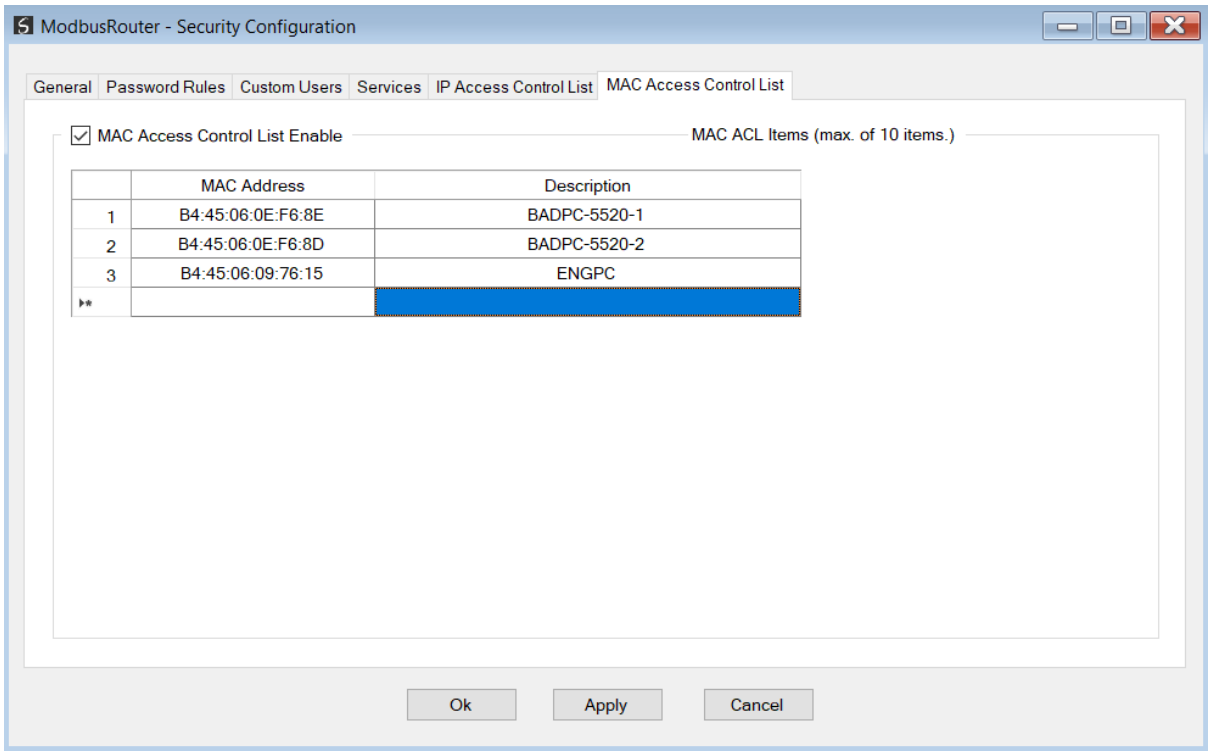


Figure 5.9 – Security Configuration – MAC Access Control List.

The ***MAC Access Control List*** configuration tab comprises the following parameters:

Parameter	Description
MAC Access Control List Enable	Enable / Disable for all MAC Access Control List
MAC Access Control List	
MAC Address	The MAC address which will be allowed to communicate with the module.
Description	An optional description for the Device with the matching MAC address.

Table 5.7 – Security Configuration – MAC Access Control List.

The MAC addresses can be entered manually, or the user can right-click and select the ***Browse*** option.

This will open the ***Network Interface Browser*** displaying all the network interfaces and their corresponding MAC addresses found on the local pc. The user can then select which MAC addresses to be added to the MAC Access Control List.

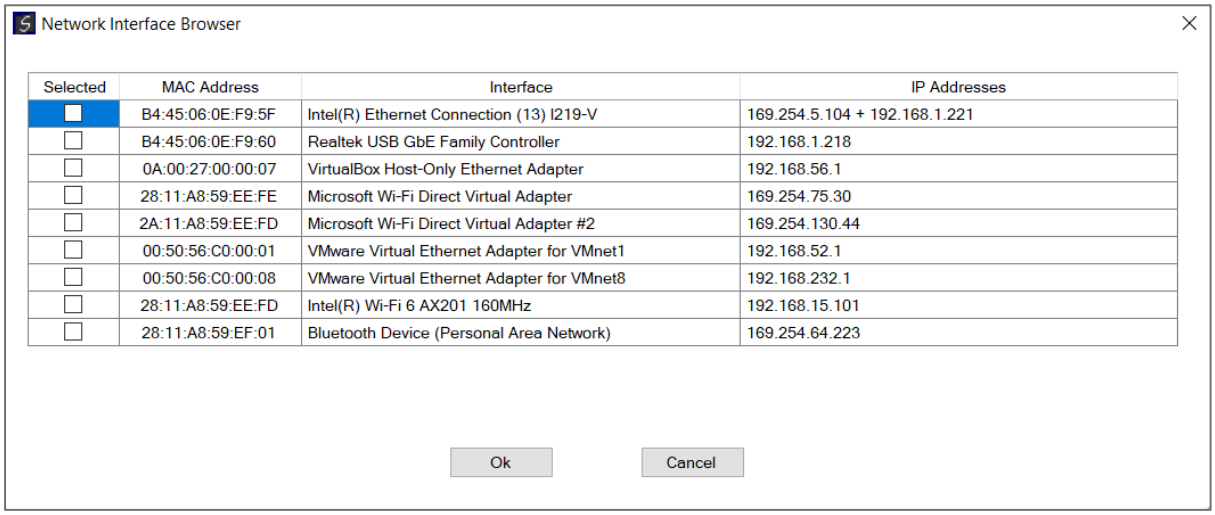


Figure 5.10 – Security Configuration – Network Interface Browser.

5.1.2. Download Security Configuration

Once the security configuration has been completed it must be downloaded to the module.

When online with the module, the current state of the security will be indicated by the icon of the Security node in the project tree as follows:

Icon	Description
	Module Offline – Master Security Disabled
	Module Offline – Master Security Enabled
	Module Online – No Security Configuration Downloaded
	Module Online – Master Security Disabled
	Module Online – Master Security Enabled, User Authentication Uninitialized

	Module Online - Secured

Table 5.8 – Security Status Icons.

To download the security configuration right-click on the Security item in the project tree and select the ***Download Security Configuration*** item.

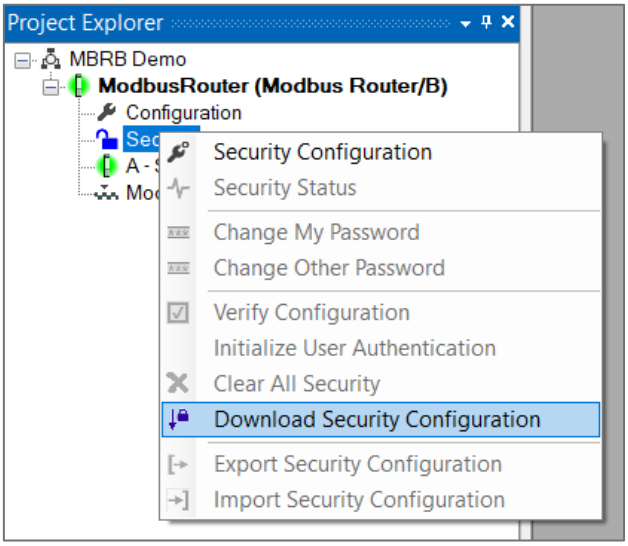


Figure 5.11 – Download Security Configuration.

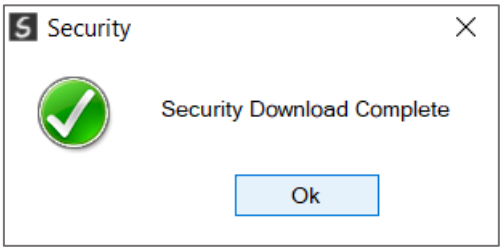


Figure 5.12 – Download Security Configuration Complete.

5.1.3. Initialize User Authentication

The final step in securing the module is to initialize user authentication. This involves creating an Administrator password as well as setting the Default password to be used by all other users for their initial login.

To commence with this step, right-click on the Security node in the project tree and select the ***Initialize User Authentication*** option.

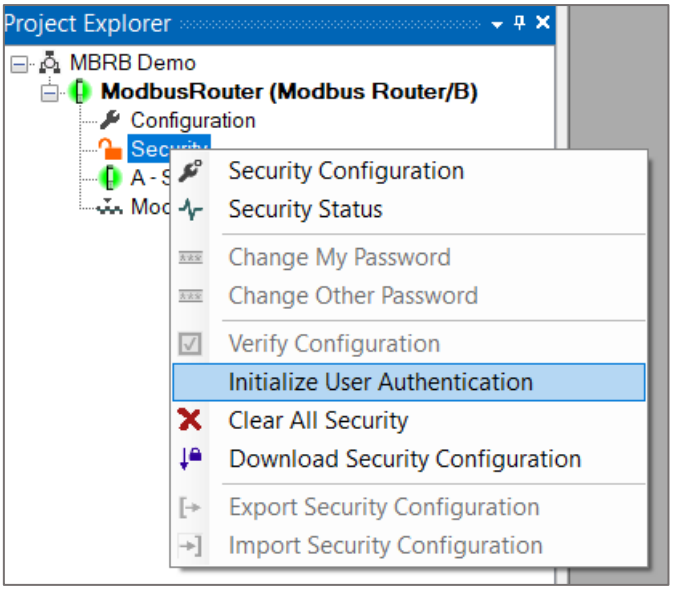


Figure 5.13 – Initialize User Authentication.

The user (now deemed to be the Administrator) will need to configure a password for the fixed ***Administrator*** user.

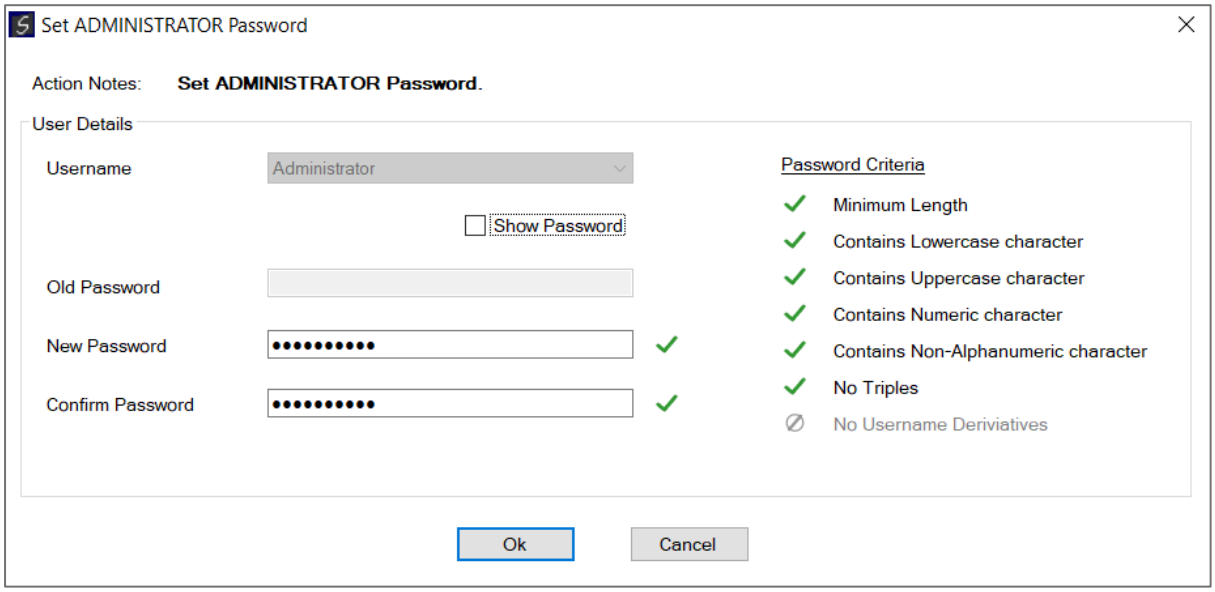


Figure 5.14 – Set Administrator Password.

As the new password is entered, it will be checked against the configured password rules. The icons adjacent to each password rule will indicate if the criteria are fulfilled or not, or not applicable, as shown below:

Icon	Description
	Criteria Not Applicable
	Failed Criteria
	Passed Criteria

Table 5.9 – Password Rule Status Icons.

A green tick icon will be displayed to the right of the ***New Password*** once it meets all the password criteria. A green tick adjacent to the ***Confirm Password*** will indicate that it matches the ***New Password***.

The entered passwords can either be displayed or masked by clicking on the ***Show Password*** check box.

Once the Administrator password has been set, then the ***Default*** password must be set. The ***Default*** password is the password to be used by all other users for their initial login.

The ***Default*** password is configured in similar manner and must also comply with the configured password rules.

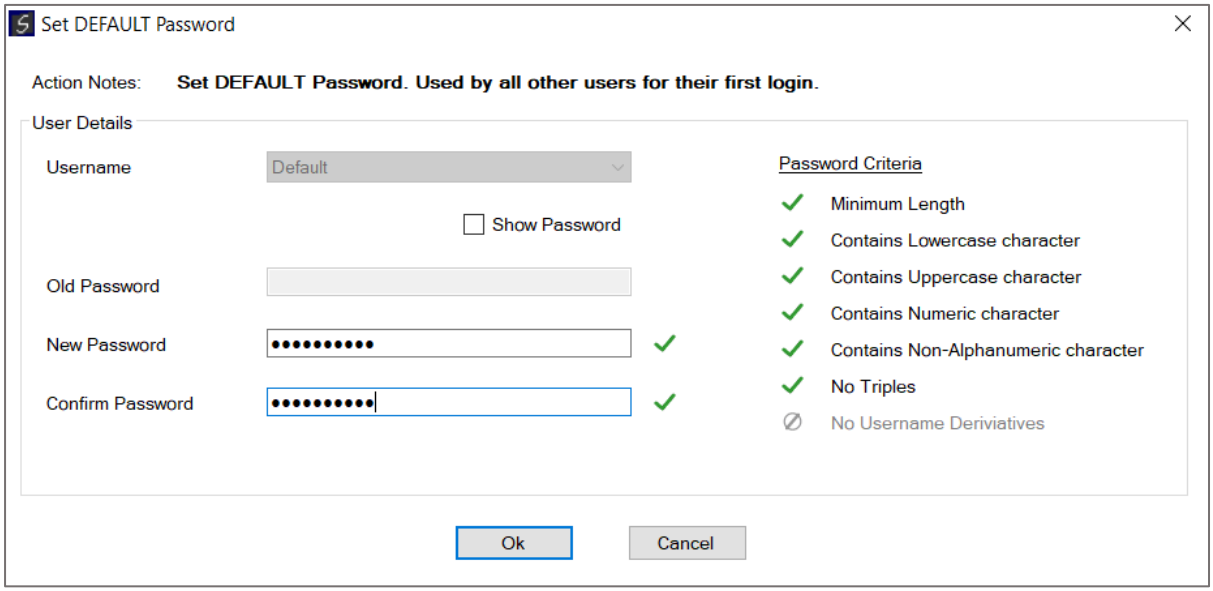


Figure 5.15 – Set Default Password.

NOTE: If required, the Administrator can change the default password individually for each user, by using the ***Change Other Password*** option.

On completion of the User Authentication Initialization step, the module will be secured indicated by the green Locked icon.

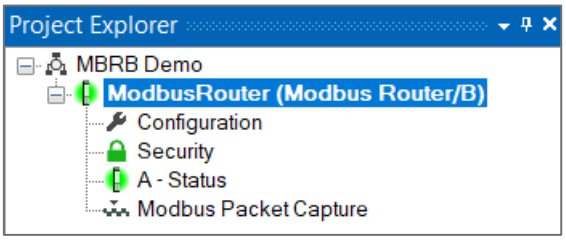


Figure 5.16 – Secured Module.

5.2. OPERATION

To interact with a secured and user initialized module, each user will need to first login.

5.2.1. INITIAL USER LOGIN

Similar, to an unsecured module, a user would open the existing Slate project, right-click on the module, and select ***Go Online***.

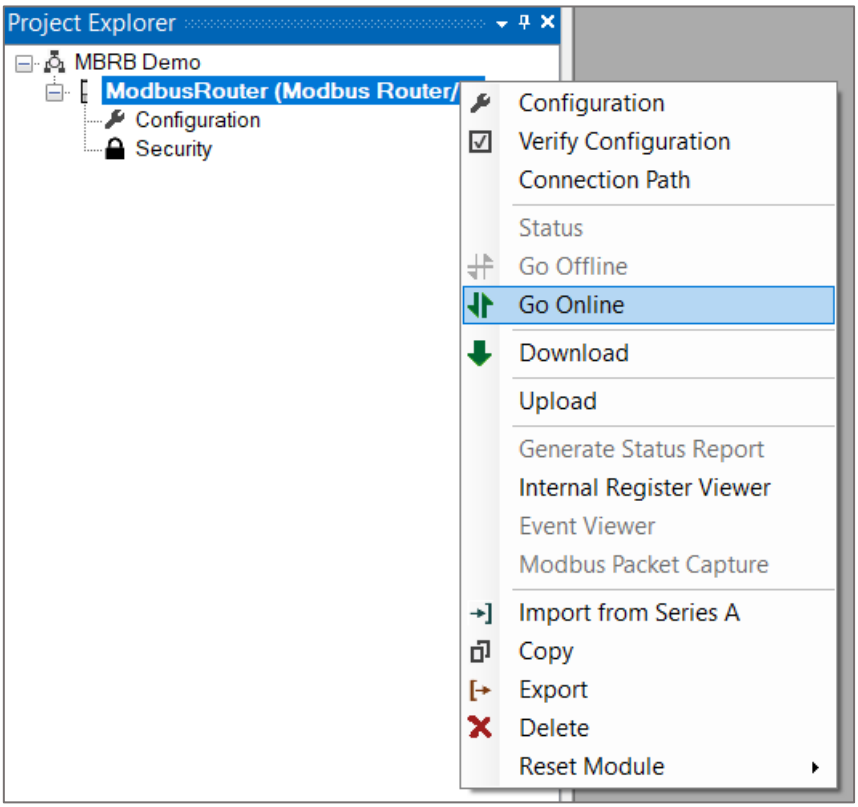


Figure 5.17 – Go Online.

Since the module has been secured, it will then prompt the user to login.

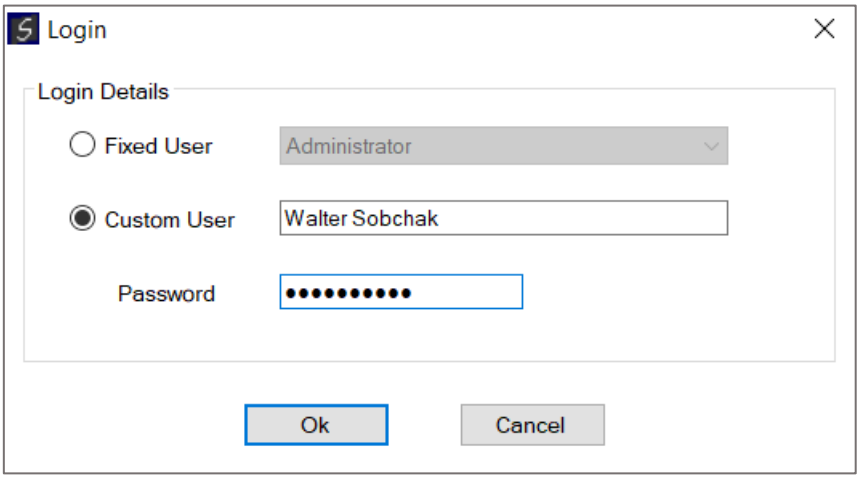


Figure 5.18 – Custom User Login.

The user will need to select either one of the ***Fixed Users*** or select the ***Custom User*** option and then enter the Custom Username.

The user will then need to enter their password.

Should this be the initial login with the Default password, or if the Administrator has reset the password and selected the option for the user to change the password, then the Set Password form will open. The user will then need to set an appropriate password before continuing.

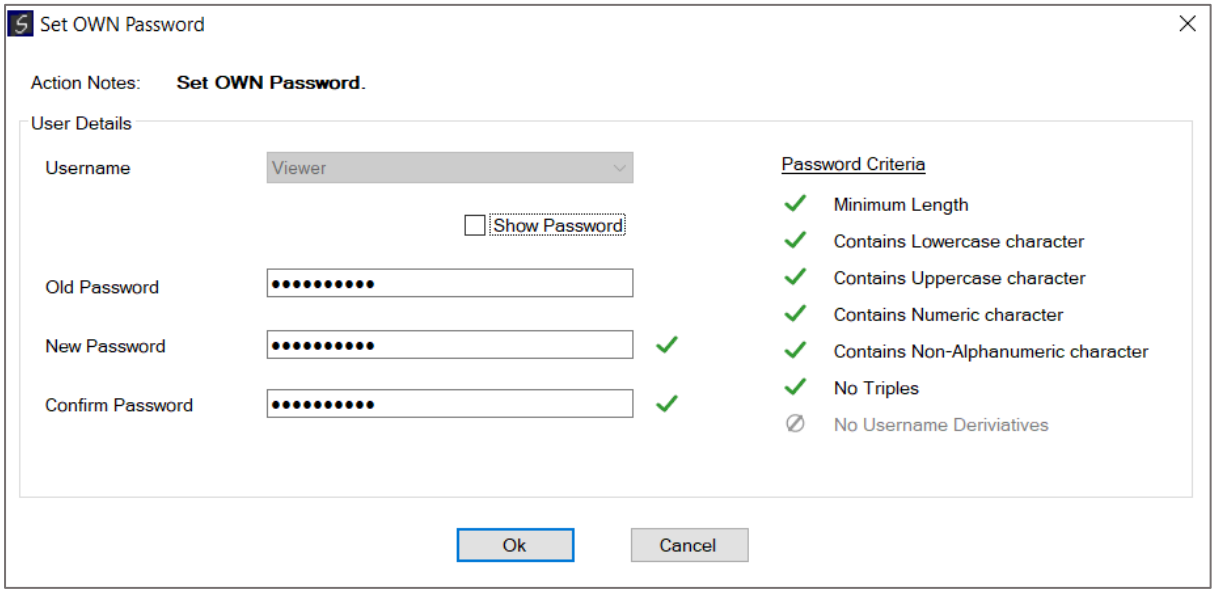


Figure 5.19 – Change Own Password.

NOTE: The *Old Password* that needs to be entered on the Initial User Login, will be the same password that was used to login.

All users are permitted to change their password at any time by right-clicking on the ***Security*** node in the project tree and selecting the ***Change My Password*** option.

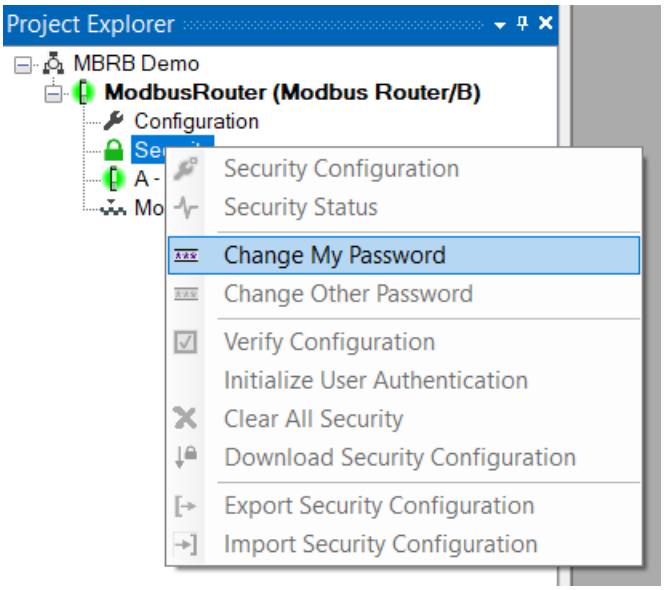


Figure 5.20 – Select Change My Password.

5.2.2. Role Specific Operations

Users with Administrator, Engineer and Auditor roles are permitted to access additional functions not available by other users.

5.2.2.1. Security Status

The Security Status can be viewed by right-clicking on the ***Security*** node in the project tree and selecting the ***Security Status*** option.

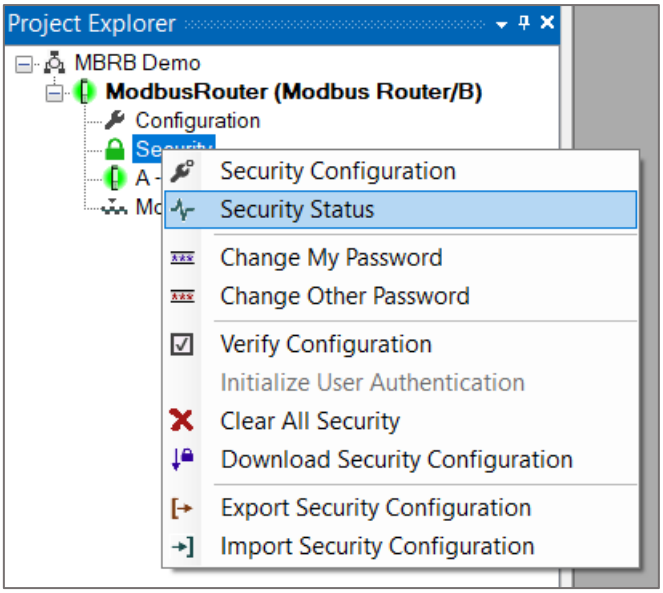


Figure 5.21 – Select Security Status.

The Security Status window will open displaying various status and statistics regarding the security services.

The ***General*** tab displays an overview of the security status.

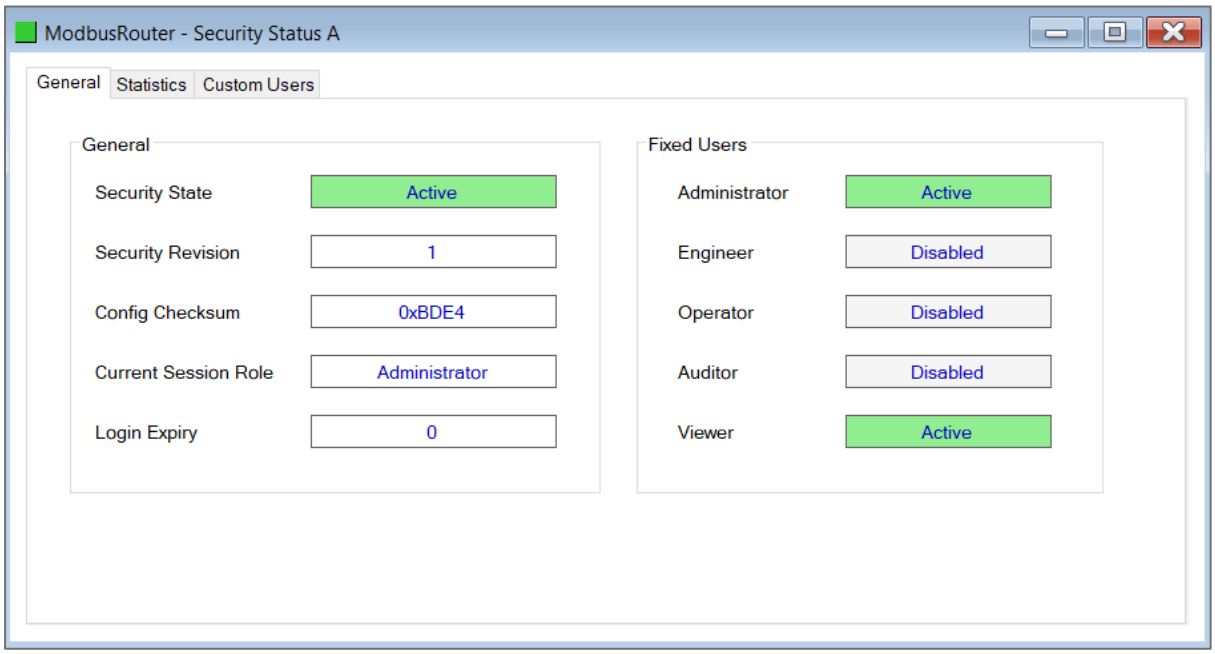


Figure 5.22 – Security Status – General.

The General status tab displays the following:

Parameter	Description
General	
Security State	The current status of the security services: ***No Config*** No security configuration has been downloaded to the module. ***Disabled*** Security configuration has been downloaded to the module, but the master Enable Security option in the configuration has been disabled. ***Uninitialized*** The security configuration has been downloaded and enabled, but the Initialize User Authentication step has not yet been completed. ***Active*** The module has been successfully secured.
Security Revision	The revision of the security services currently running on the module. This is a function of the module's firmware revision.
Config Checksum	The checksum of the current security configuration.
Current Session Role	The role assigned to the user that is currently logged in.
Login Expiry	The time (seconds) until the user is automatically logged out. Only applicable if the ***Login Expire*** option has been enabled in the security configuration.
Fixed Users	
Fixed Users Status Administrator Engineer Operator Auditor Viewer	The status of each fixed user: ***Disabled*** The user has not been enabled in the security configuration. ***Default Password*** The user has not changed from their Default assigned password, or from when an Administrator reset their password. ***No Password*** No password exists for the user. This would typically occur when a fixed user is enabled after ***the User Authentication Initialization*** step has taken place. An Administrator would need to reset the password for this specific user. ***Active*** The user is enabled and has a valid password configured.

Table 5.10 – Security Status – General.

The Statistics tab displays a number of security service statistics.

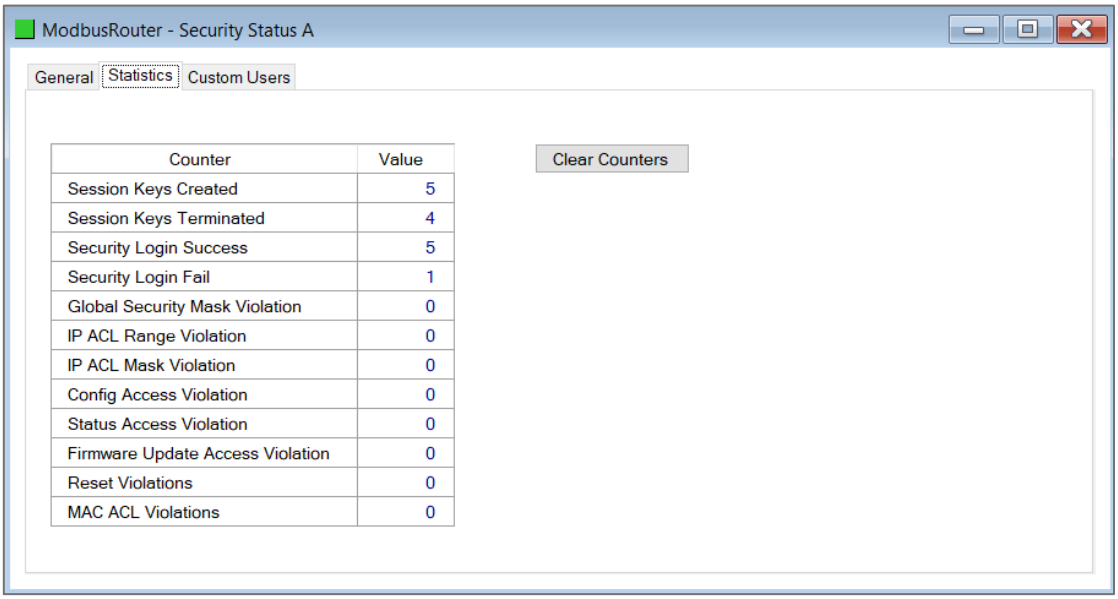


Figure 5.23 – Security Status – Statistics.

NOTE: Only an Administrator is permitted to select the ***Clear Counters*** option.

Parameter	Description
Session Keys Created	The number of security session keys created between the configuration software and the module. Each time a user connects to the module with the configuration software, and the module has been secured, a new session key will be created.
Session Keys Terminated	The number of times when either the connection from the configuration software has been terminated or there was an error in the connection decryption. In both these cases the session keys will be deleted.
Security Login Success	The number of successful logins.
Security Login Fail	The number of failed login attempts.
Global Security Mask Violation	The number of times a connection request is received from another device on the Ethernet network, which has not been enabled in the module's global security mask configuration.
IP ACL Range Violation	Each time a request or packet is received from a device that has an IP address outside the IP ACL range, this number will be incremented.
IP ACL Mask Violation	Each time a request or packet is received from a device that has an IP address outside the Subnet ACL range, this number will be incremented.
Config Access Violation	The number of times a connection with the incorrect role privileges tried to access the module security configuration.

Status Access Violation	The number of times a connection with the incorrect role privileges tried to access the module security status.
Firmware Update Access Violation	The number of times an attempt to flash upgrade the module in an incorrect state (e.g., while running a class 1 EtherNet/IP connection) was made.
Reset Violations	The number of times an attempt to reset the module in an incorrect state (e.g., while running a class 1 EtherNet/IP connection) was made.
MAC ACL Violations	Each time a request or packet is received from a device that does not have a matching MAC address to those in the MAC ACL list, this number will increase.

Table 5.11 – Security Status – Statistics.

The Custom Users status tab displays the following:

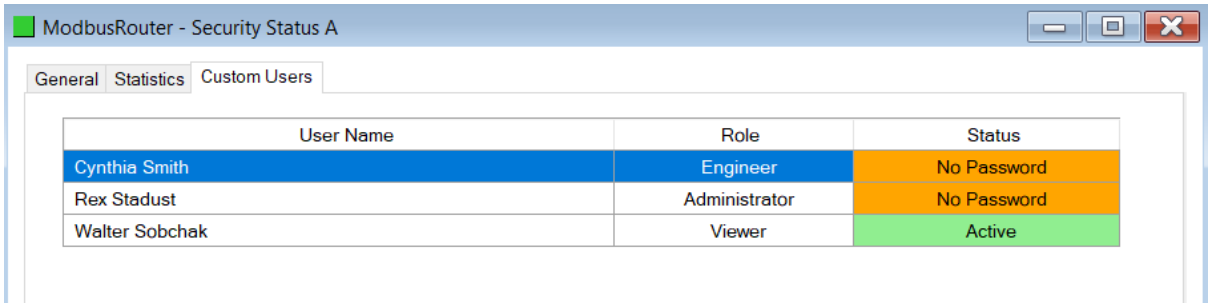


Figure 5.24 – Security Status – Custom Users.

Parameter	Description
User Name	The configured user name.
Role	The configured role for this specific user.
Status	The status of the specific user: ***Default Password*** The user has not changed from their Default assigned password, or from when an Administrator reset their password. ***No Password*** No password exists for the user. This would typically occur when a user is added after the User Authentication Initialization has taken place. An Administrator would need to reset the password for this specific user. ***Active*** The user is enabled and has a valid password configured.

Table 5.12 – Security Status – General.

5.2.3. CHANGE OTHER PASSWORD

An Administrator can change (or reset) any user's password by right-clicking on the Security node in the project tree and selecting the ***Change Other Password*** option.

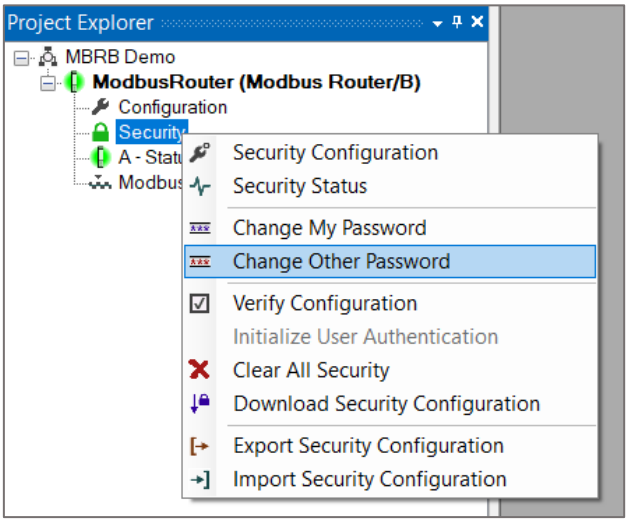


Figure 5.25 – Select Change Other Password.

The Set Password form will open, and the Administrator can then select the User's password to change. The Administrator will need to re-enter their password as well as the new password for the user.

The ***Require User to Change Password*** option forces that user to change the password the next time they login.

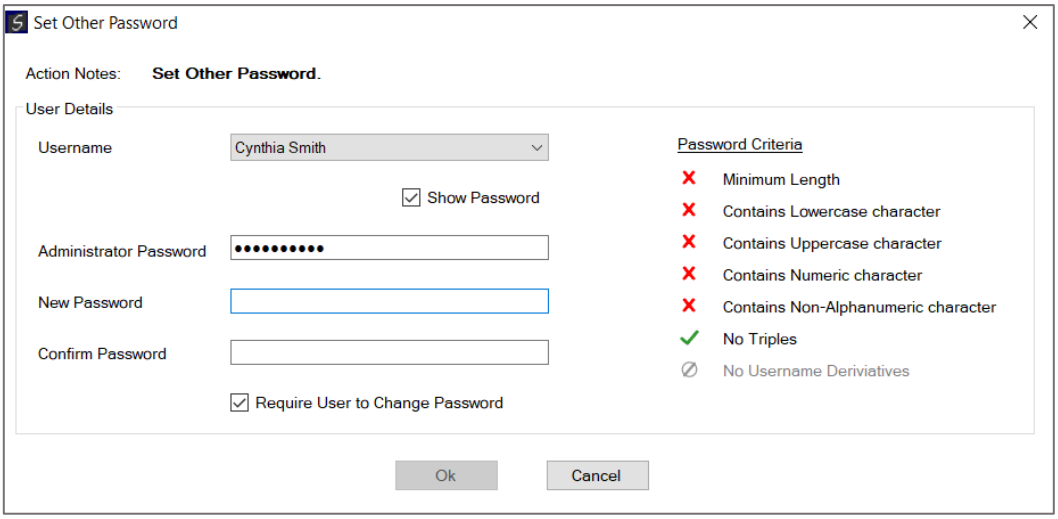


Figure 5.26 – Set Other Password.

5.2.4. CLEAR ALL SECURITY

An Administrator can remove all the security from a module and return it to an unsecured state. This is done by right-clicking on the Security node in the project tree and selecting the ***Clear All Security*** option.

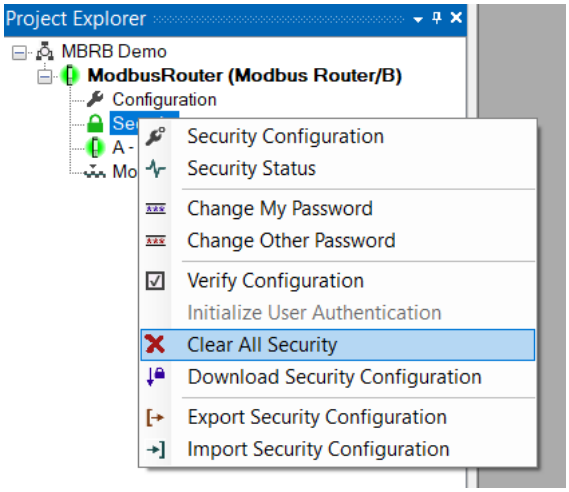


Figure 5.27 – Select Clear All Security.

This action will require the Administrator to enter their password. The security configuration and all passwords will be removed with this action.

NOTE: This process will remove all module security configuration including all passwords.

5.3. EVENT LOG

All security related events are added to the module's event log.

NOTE: Only an Administrator is permitted to select the ***Clear*** the event log.

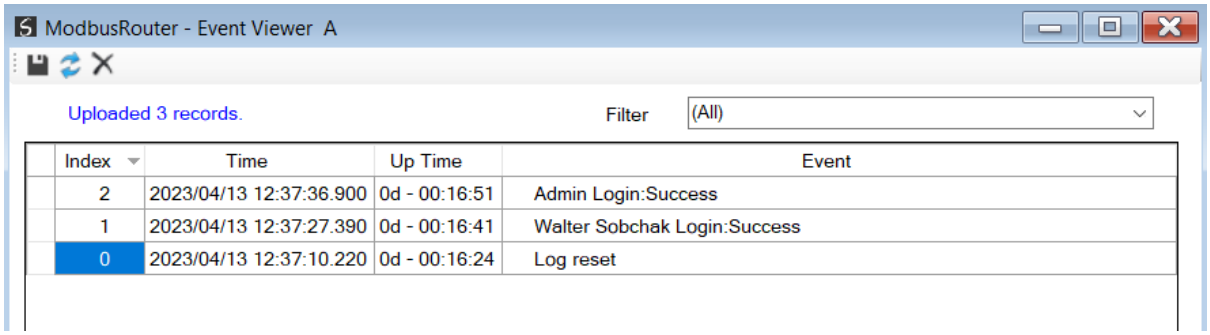


Figure 5.28 – Event Log.

5.4. RESET SECURITY CONFIGURATION

The module's security configuration can be reset (cleared) by powering-up the module with DIP switches 2, 3, and 4 set to the ON position.

NOTE: This process will remove all module application configuration and security configuration including all passwords.

NOTE: Once the module has booted-up, the DIP switches should be returned to normal (all OFF). New security configuration cannot be downloaded with the Reset DIP switch combination active.

6. OPERATION

6.1. SCHEDULED TAG MODE

In Scheduled Tag routing mode the MBR/B is a Modbus Client and transfers data between Modbus devices and Logix controllers. Below are two examples of how data is exchanged in Scheduled Mode.

Example 1

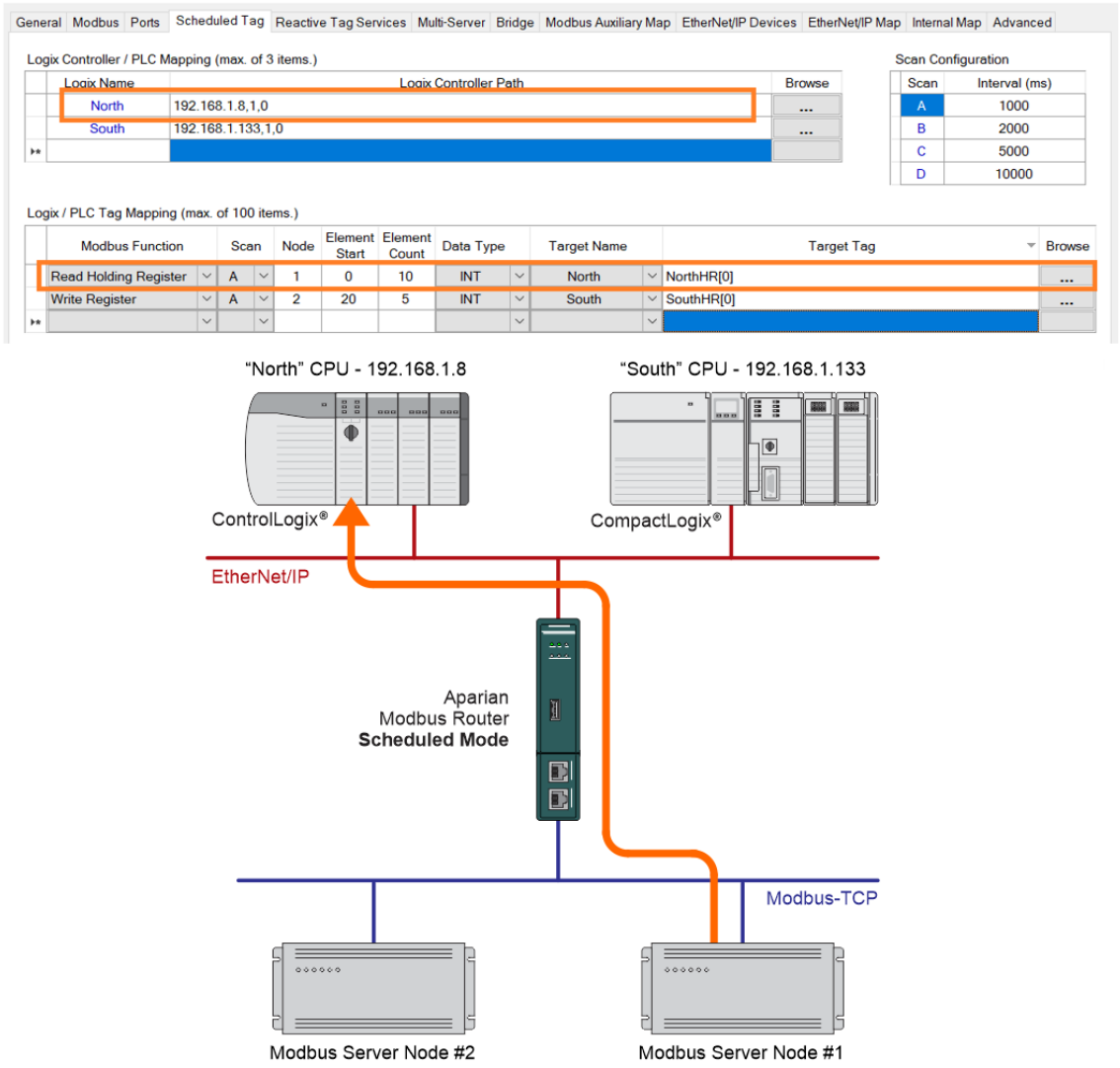


Figure 6.1 – Scheduled Mode – Example 1.

The MBR/B will execute the first item in the Scheduled Tag Map, by sending a Modbus Read Request for Holding Register 0 to Modbus device Node 1. The MBR/B is requesting 10 Modbus

Holding Registers. Once the data is received from the Modbus Device, then it is converted into an EtherNet/IP Tag Write message targeted at the North Logix controller and the received Modbus data is written to Logix tag *NorthHR[0]* which is a Logix INT array.

Example 2

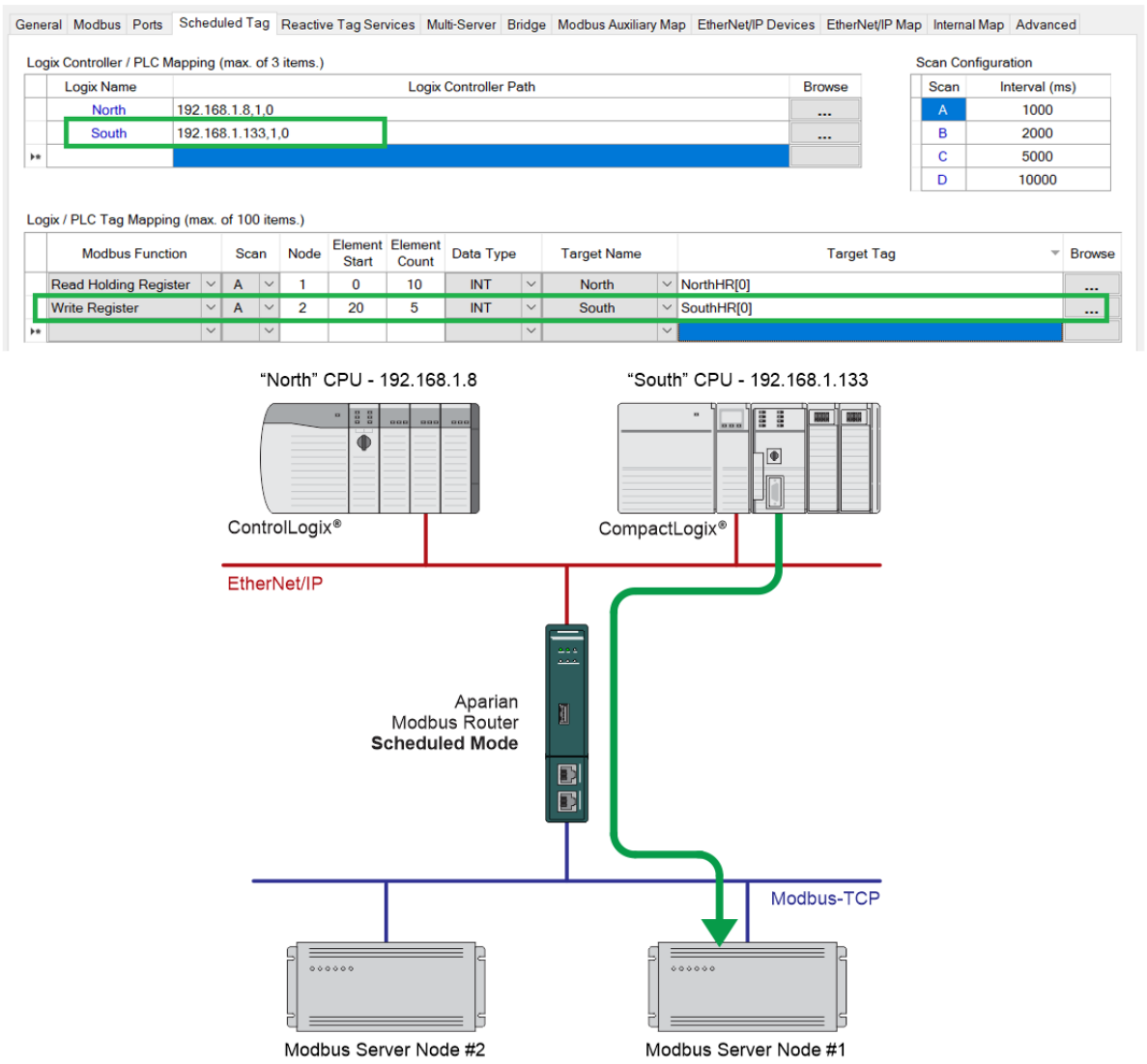


Figure 6.2 – Scheduled Mode – Example 2.

The MBR/B will execute the second item in the Scheduled Tag Map, by sending a Logix Tag read for tag *SouthHR[0]* (INT array) to the South Logix controller. Once the Logix Tag data is received, then the MBR/B will perform a Modbus Write Request for Holding Register 20 to Modbus device Node 2. The MBR/B is writing 10 Modbus Holding Registers.

6.2. REACTIVE TAG MODE

The Reactive Tag routing mode allows mapping of virtual Modbus registers to Logix tags across multiple controllers. In this mode the Modbus Router is a Modbus Server and will redirect a Modbus message to a Logix controller at a preconfigured path.

Thus, the routing of Modbus read and write register requests is managed by the Modbus Router and converted to direct Logix tag read and write functions.

Below are two examples of how data is exchanged in Reactive Mode.

Example 1

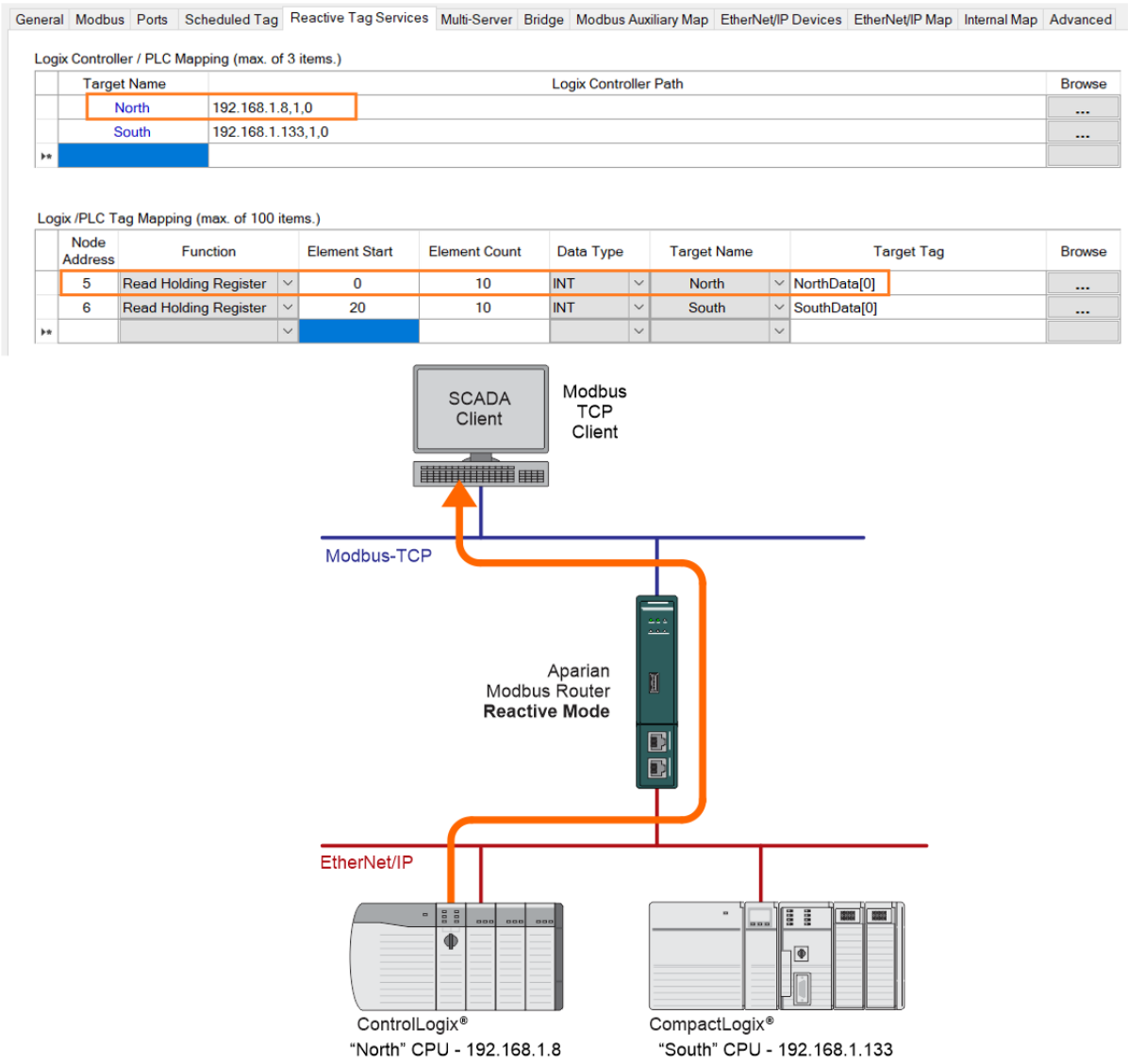


Figure 6.3 – Reactive Mode – Example 1.

When the Modbus Client (i.e., SCADA) sends a Modbus Read to node 5 for Holding Register 0, then the MBR/B will match the request to the first mapped item in the Reactive Tag Map. The MBR/B will read the Logix tag *NorthData[0]* from the North Logix Controller when the request is received. Once the Logix tag data is returned, the MBR/B will respond to the Modbus Client request with the updated Modbus data.

Example 2

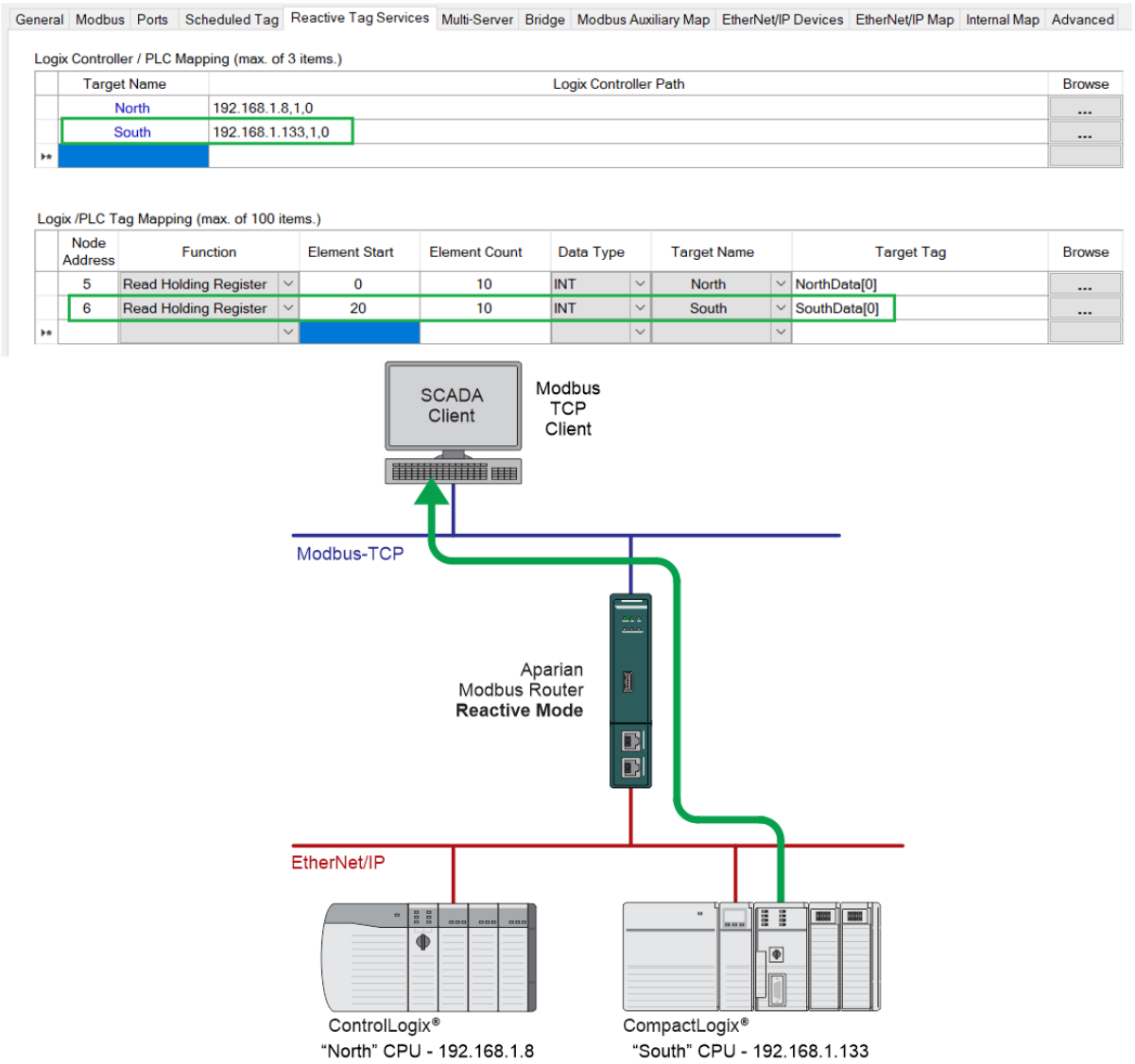


Figure 6.4 – Reactive Mode – Example 2.

When the Modbus Client (i.e., SCADA) sends a Modbus Read to node 6 for Holding Register 20, then the MBR/B will match the request to the second mapped item in the Reactive Tag Map. The MBR/B will read the Logix tag *SouthData[0]* from the South Logix Controller when the request is received. Once the Logix tag data is returned, the MBR/B will respond to the Modbus Client request with the updated Modbus data.

6.3. UNSCHEDULED MODE

When the Modbus Router is configured in Unscheduled Mode, it will process Modbus message requests sent from Logix via a message instruction.

NOTE: The user will need to build some of the Modbus request data of the unscheduled message.

NOTE: When in unscheduled mode the Controller Type selection in the General configuration will not be applicable. The Modbus Router will only support a Logix Controller type in unscheduled mode.

To simplify the configuration of the required message, a number of UDTs have been preconfigured and are available on the Aparian Modbus Router webpage (see the further information section).

The message instruction should be setup as follows:

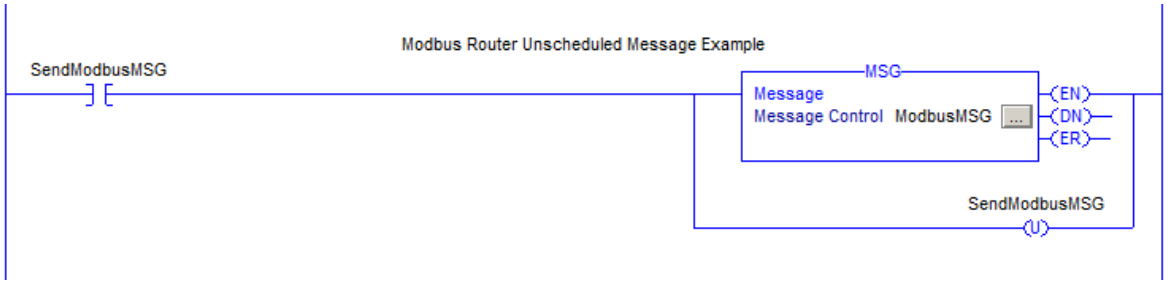


Figure 6.5 - Message Instruction.

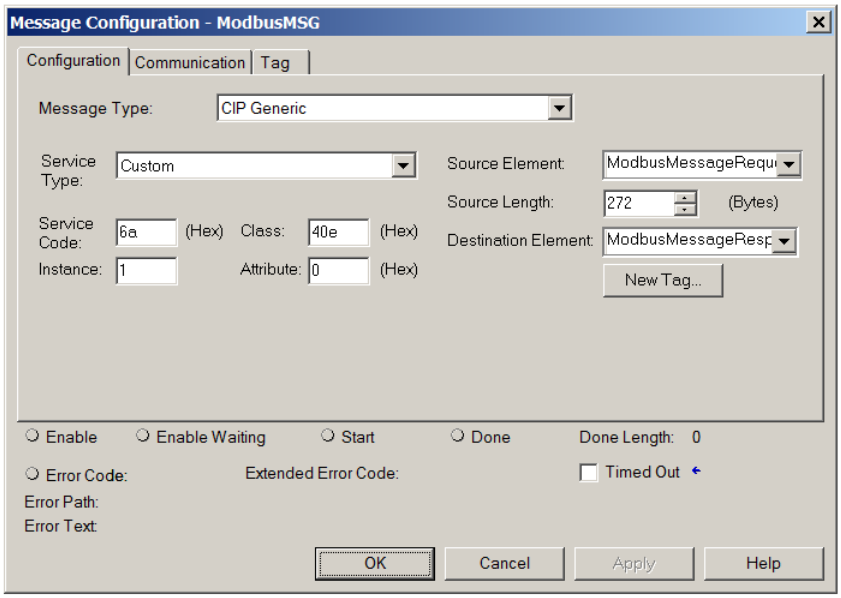


Figure 6.6 - Message Configuration.

Parameter	Description
Message Type	CIP Generic
Service Type	Custom
Service Code	6A (Hex) - Unscheduled Modbus Pass-through
Class	40E (Hex)
Instance	1
Attribute	0
Source Element	The request tag instance. Must follow the structure of the AparianModbusMessageRequest UDT.
Source Length	272
Destination Element	The response tag instance. Must follow the structure of the AparianModbusMessageResponse UDT.

Table 6.1 - Message Configuration Parameters.

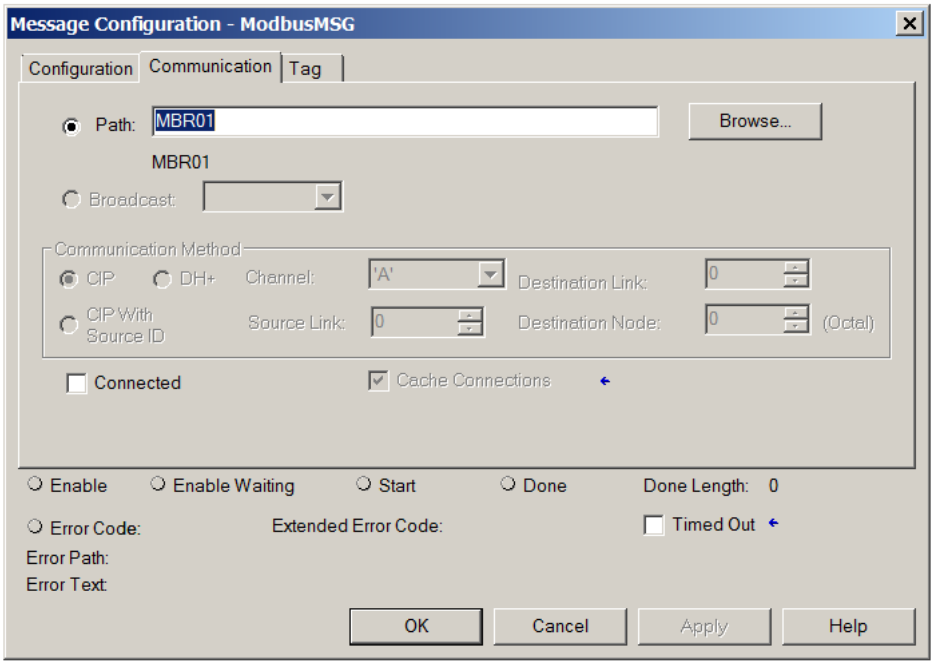


Figure 6.7 - Message Configuration – Communication.

The Path must be configured to that of the Modbus Router. If the Modbus Router has been added in the I/O tree, then the Browse option can be used to select the path.
Alternatively, enter the CIP path in the format:

1,X,2,IP , where

1 represents the backplane port,

X represents the slot of the Ethernet bridge module,

2 represents the Ethernet port of the Ethernet bridge module and
IP represents the IP address of the Modbus Router.
e.g. **1,1,2,192.168.1.41**

The request tag (e.g. ModbusMessageRequest) should be configures as follows:

ModbusMessageRequest	{ . . . }	AparianModbusMessageRequest
ModbusMessageRequest.DestinationAddress	1	INT
ModbusMessageRequest.Function	3	INT
ModbusMessageRequest.Port	0	INT
ModbusMessageRequest.RequestSize	4	INT
ModbusMessageRequest.IPAddress	{ . . . }	INT[4]
ModbusMessageRequest.IPAddress[0]	192	INT
ModbusMessageRequest.IPAddress[1]	168	INT
ModbusMessageRequest.IPAddress[2]	1	INT
ModbusMessageRequest.IPAddress[3]	219	INT
ModbusMessageRequest.RequestData	{ . . . }	SINT[256]

Figure 6.8 - Unscheduled Message Request Tag.

Parameter	Description
Destination Address	The Modbus node address of the destination device.
Function	The Modbus function code. (See the table below for common function codes.)
Port	The Modbus Router supports two ports to interface to a Modbus device: 0 – Serial Port (Modbus-RTU). 1 - Ethernet Port (Modbus-TCP).
Request Data Size	The size of the request data being sent
IP Address	The IP address of the remote device when an Ethernet port was selected above.
Request Data	Modbus request data. All the bytes succeeding the function code but preceding the checksum. See the example further on in the chapter.

Table 6.2 - Unscheduled Message Request Parameters.

ModbusMessageResponse	{ . . . }	AparianModbusMessageResponse
ModbusMessageResponse.Status	0	INT
ModbusMessageResponse.ResponseLength	5	INT
ModbusMessageResponse.Response	{ . . . }	SINT[255]

Figure 6.9 - Unscheduled Message Response Tag.

Parameter	Description
Status	The Modbus exception code returned. A zero represents success. (See the table below for common exception codes.)
Response Length	Length of the response data received.
Response Data	Response to the Modbus application layer object request.

Table 6.3 - Unscheduled Message Response Parameters

After the message has been executed successfully (Msg.DN) the Response Data will hold the Modbus response data.

Below is a table showing common Modbus Functions:

Code	Modbus Function
1	Read Coils
2	Read Discrete Inputs
3	Read Holding Registers
4	Read Input Registers
5	Write Single Coil
6	Write Single Register
7	Read Exception Status
8	Diagnostic
11	Get Comm Event Counter
12	Get Comm Event Log
15	Write Multiple Coils
16	Write Multiple Registers
17	Report Server ID
20	Read File Record
21	Write File Record
22	Mask Write Register
23	Read/Write Multiple Registers
24	Read FIFO Queue

Table 6.4 - Common Modbus Functions.

The following table shows common Status / Exception codes.

Code	Modbus Exception
0	Success
1	Illegal Function
2	Illegal Data Address

3	Illegal Data Value
4	Server Device Failure
5	Acknowledge
6	Server Device Busy
8	Memory Parity Error
10	Gateway Path Unavailable
11	Gateway Target Device Failed to Respond

Table 6.5 - Common Exception Codes.

6.3.1. EXAMPLE

In the following example, the unscheduled message instruction is used to read two Holding Registers (0 and 1), from a remote Ethernet Modbus device located at IP address 192.168.1.219.

The standard ModbusMessageRequest structure is populated as shown in the figure below.

ModbusMessageRequest	{...}	
ModbusMessageRequest.DestinationAddress	1	Modbus node address of destination device
ModbusMessageRequest.Function	3	Function : 03 - Read Holding Register
ModbusMessageRequest.Port	1	Port : 1 - Ethernet (Modbus-TCP)
ModbusMessageRequest.RequestSize	4	Number of bytes in Request Data
ModbusMessageRequest.IPAddress	{...}	
ModbusMessageRequest.IPAddress[0]	192	Modbus IP address of destination device - 192.168.1.219
ModbusMessageRequest.IPAddress[1]	168	
ModbusMessageRequest.IPAddress[2]	1	
ModbusMessageRequest.IPAddress[3]	219	
ModbusMessageRequest.RequestData	{...}	
ModbusMessageRequest.RequestData[0]	0	Register Start - 0000
ModbusMessageRequest.RequestData[1]	0	
ModbusMessageRequest.RequestData[2]	0	Register Count - 0002
ModbusMessageRequest.RequestData[3]	2	

Figure 6.10 – Modbus Message Request Example .

6.4. ENHANCED MODE

In Enhanced Mode the MBR/B can be configured in any of the four combinations:

Modbus	EtherNet/IP
Client	Target
Client	Originator
Server	Target

Server	Originator

Table 6.6 – Enhanced Mode Operating Combinations.

6.4.1. EIP TARGET – MODBUS SERVER

When the MBR/B is operating in Enhanced mode and has the **Modbus Mode** set to **Server** and the **EtherNet/IP Mode** set to **Target**, then the EtherNet/IP data (from, for example, a Logix controller) 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 Client on Modbus TCP, RTU232, or RTU485. The remote Modbus Client can read or write to the configured Modbus addresses to access the EtherNet/IP data that has been mapped to the Modbus Registers.

In the example below the MBR/B, operating in Enhanced mode and has the **Modbus Mode** set to **Server** and the **EtherNet/IP Mode** set to **Target**, will have multiple Modbus Holding Registers written from a Modbus TCP Client and then map the received data to a Logix controller that is owning the MBR/B with an EtherNet/IP Class 1 connection.

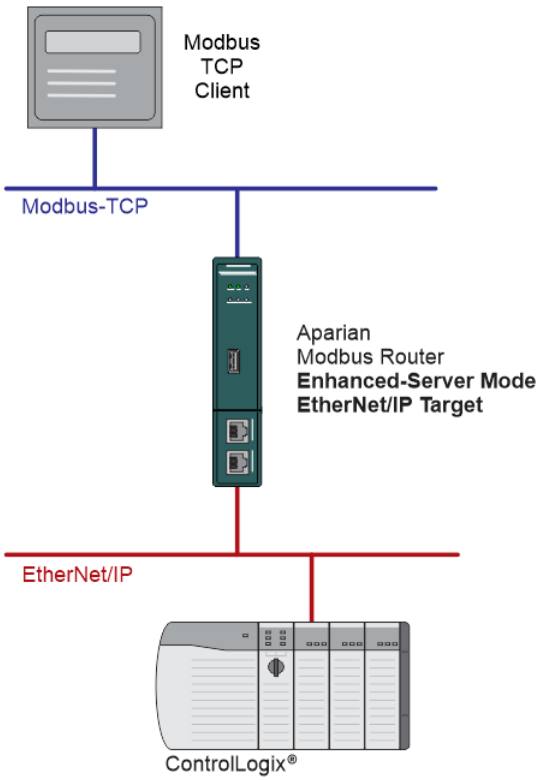


Figure 6.11 – Modbus TCP Client to Logix Assembly operation.

For this example, the remote Modbus TCP Client will write data to Modbus Holding Register 4000 in the MBR/B. The MBR/B will map the received Modbus data (at Holding Register 4000) to the input assembly of the MBR/B in Logix. The data will be available in the second data connection (i.e., Logix connection 3, because the first Logix connection is for module status) at offset 20 (in bytes so word 10).

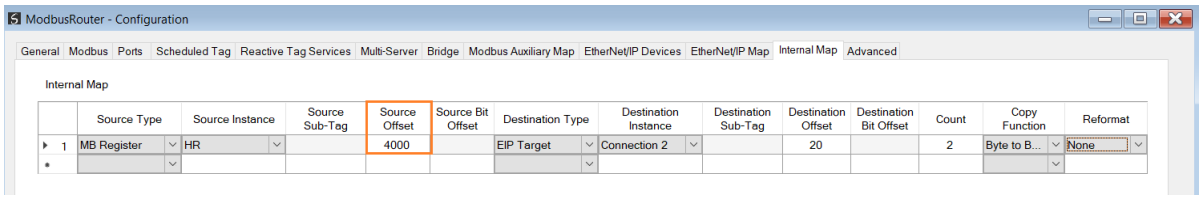


Figure 6.12 – Internal Mapping from Modbus Register to Logix input assembly,

6.4.2. EIP ORIGINATOR – MODBUS SERVER

When the MBR/B is operating in Enhanced mode and has the **Modbus Mode** set to **Server** and the **EtherNet/IP Mode** set to **Originator**, then the EtherNet/IP IO 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 Client on Modbus TCP, RTU232, or RTU485. The remote Modbus Client can read or write to the configured Modbus addresses to access the EtherNet/IP IO data that has been mapped to the Modbus Registers.

In the example below the MBR/B, operating in Enhanced mode and has the **Modbus Mode** set to **Server** and the **EtherNet/IP Mode** set to **Target**, will have multiple Modbus Holding Registers written from a Modbus TCP Client and then map the received data to EtherNet/IP IO that is owned with an EtherNet/IP Class 1 connection.

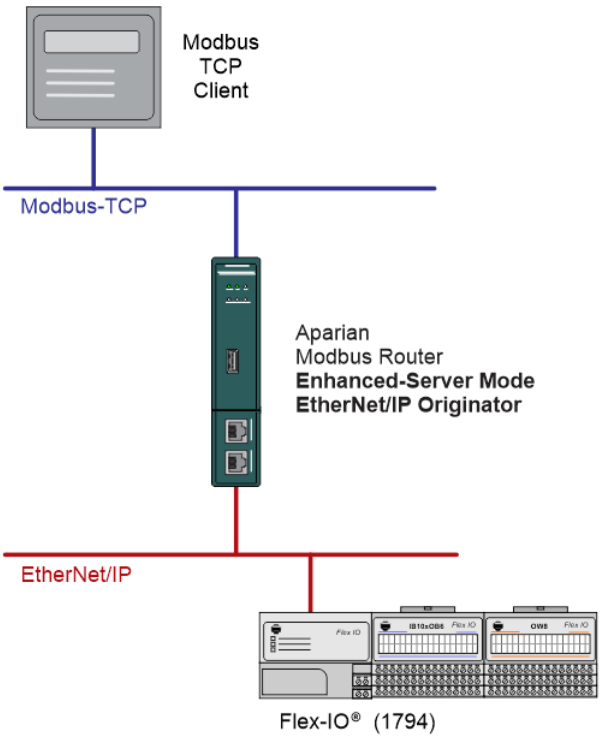


Figure 6.13 – Modbus TCP Client to EtherNet/IP IO operation,

For this example, the remote Modbus TCP Client will write data to Modbus Holding Register 2000 in the MBR/B. The MBR/B will map the received Modbus data (at Holding Register 2000) to the output assembly of EtherNet/IP IO (1794-OW8) being owned (using class 1 EtherNet/IP connection) by the MBR/B.

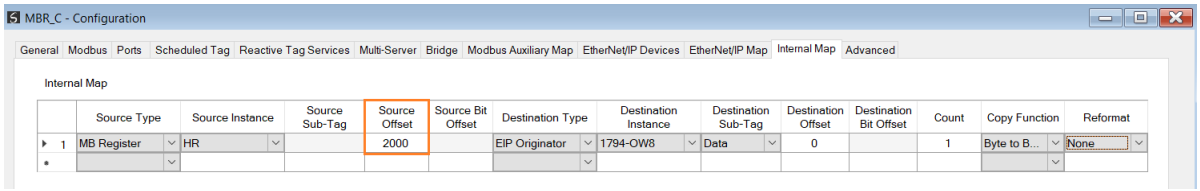


Figure 6.14 – Internal Mapping from Modbus Register to EtherNet/IP IO.

6.4.3. EIP TARGET – MODBUS CLIENT

When the MBR/B is operating in Enhanced mode and has the **Modbus Mode** set to **Client** and the **EtherNet/IP Mode** set to **Target**, then the EtherNet/IP assembly data can be mapped to and from configurable internal Modbus Registers using the Internal Map.

The internal Modbus Registers are then asynchronously exchanged with 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 on Modbus TCP, RTU232, or RTU485.

In the example below the MBR/B (operating in Enhanced mode and has the **Modbus Mode** set to **Client** and the **EtherNet/IP Mode** set to **Target)** will read multiple Modbus Holding Registers from a Modbus Server device and then map the received data to a Logix Input assembly.

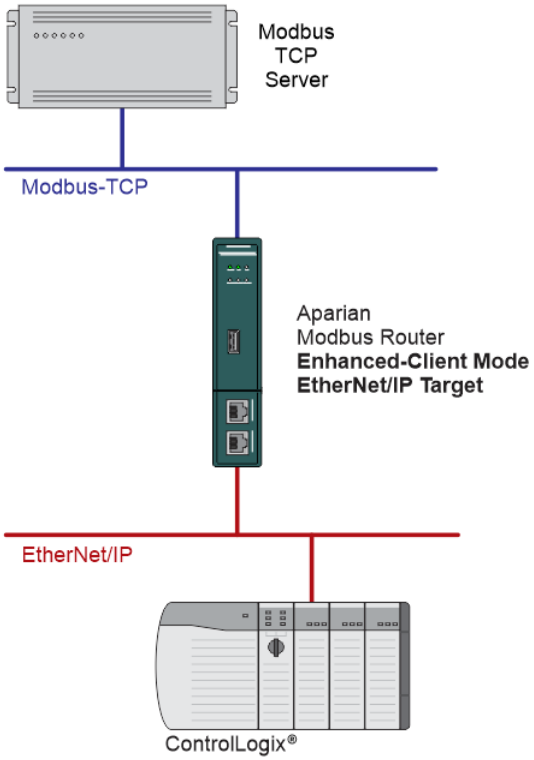


Figure 6.15 – Modbus Client to EtherNet/IP operation.

For this example, the user will configure the Modbus Auxiliary Map to read data from a Modbus Server device on Ethernet. The MBR/B will request data from Modbus Holding Register 4000 (from the external Modbus Server) and write it to the module's internal Modbus Holding Register 3000.

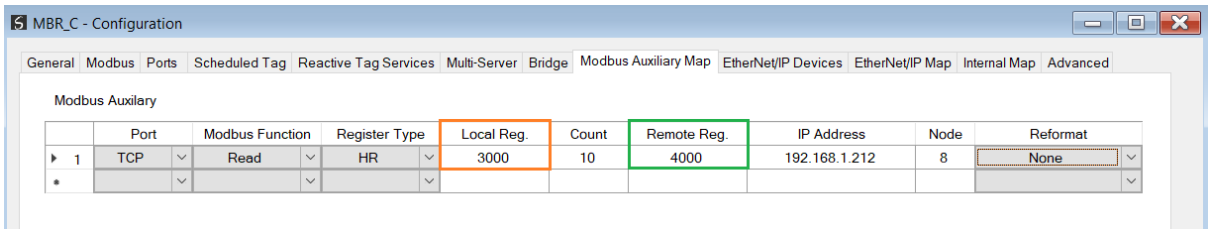


Figure 6.16 – Modbus Client Aux Mapping.

Next the data saved in the internal Modbus Register at MB Holding Register 3000 is mapped to the EtherNet/IP Target connection 1 input assembly with offset 50.

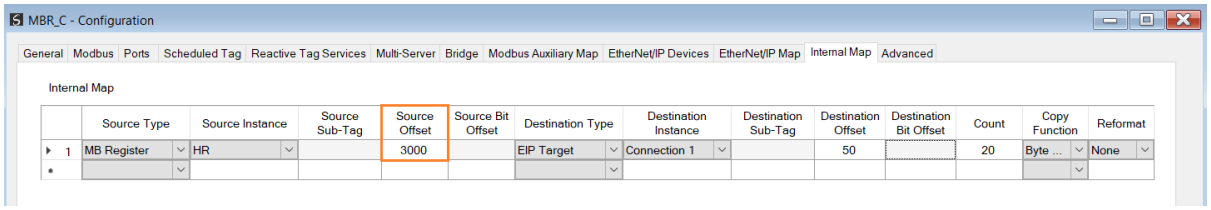


Figure 6.17 – Internal Mapping from Modbus Register to EtherNet/IP.

6.4.4. EIP ORIGINATOR – MODBUS CLIENT

When the MBR/B is operating in Enhanced mode and has the **Modbus Mode** set to **Client** and the **EtherNet/IP Mode** set to **Originator**, then the EtherNet/IP IO data can be mapped to and from configurable internal Modbus Registers using the Internal Map.

The internal Modbus Registers are then asynchronously exchanged with 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 on Modbus TCP, RTU232, or RTU485.

In the example below the MBR/B (operating in Enhanced mode and has the **Modbus Mode** set to **Client** and the **EtherNet/IP Mode** set to **Originator**) will read multiple Modbus Holding Registers from a Modbus Server device and then map the received data to an EtherNet/IP IO Output assembly.

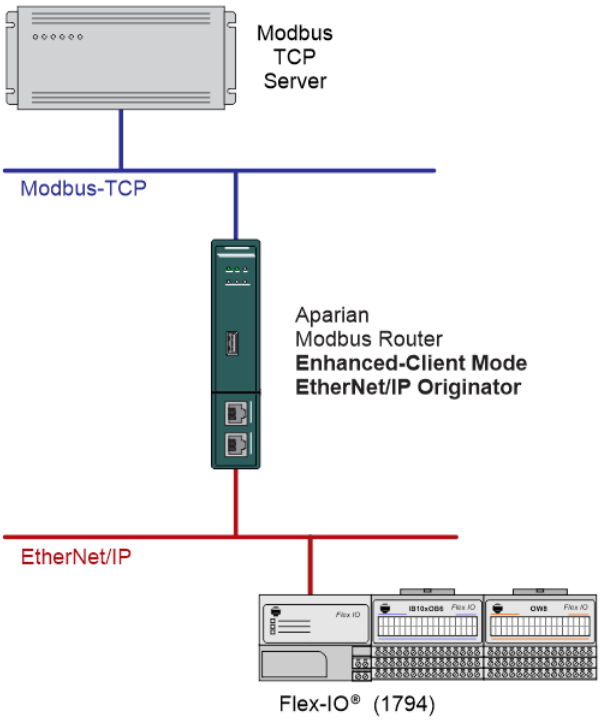


Figure 6.18 – Modbus Client to EtherNet/IP operation.

For this example, the user will configure the Modbus Auxiliary Map to read data from a Modbus Server device on Ethernet. The MBR/B will request data from Modbus Holding Register 4000 (from the external Modbus Server) and write it to the module's internal Modbus Holding Register 3000.

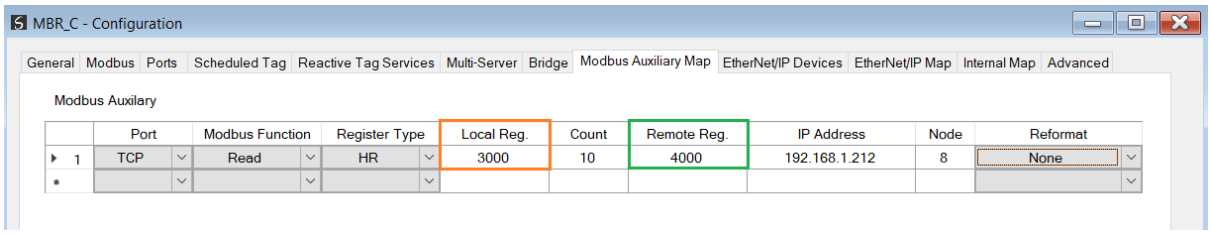


Figure 6.19 – Modbus Client Aux Mapping.

Next the data saved in the internal Modbus Register at MB Holding Register 3000 is mapped to the EtherNet/IP IO connection output assembly with offset 0.

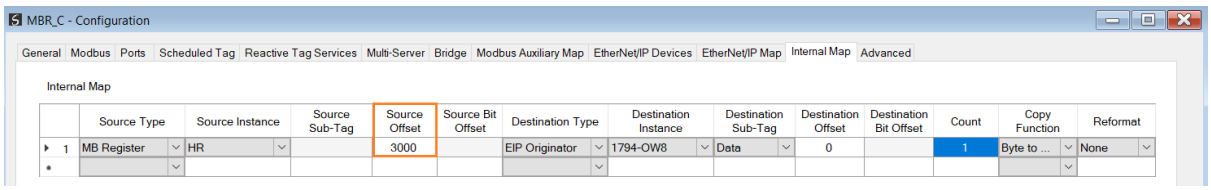


Figure 6.20 – Internal Mapping from Modbus Register to EtherNet/IP.

6.5. BRIDGE MODE

In Bridge Mode, the MBR/B can operate as a transparent link for Modbus between different media (TCP, RTU232, and RTU485). This allows, for example, a Modbus client on Modbus TCP to communicate with Modbus devices on RS232 or RS485 and vice versa.

In the below example, any Modbus request received on the EtherNet TCP port for a Modbus Server at Node address 5 (to the IP address of the MBR/B 192.168.1.100), from the Modbus Client at IP 192.168.1.50, will be routed to the RS232 port and sent to Modbus Server at Node 5. The response will then be routed back to the request originator (i.e., the Modbus Client at IP 192.168.1.50).

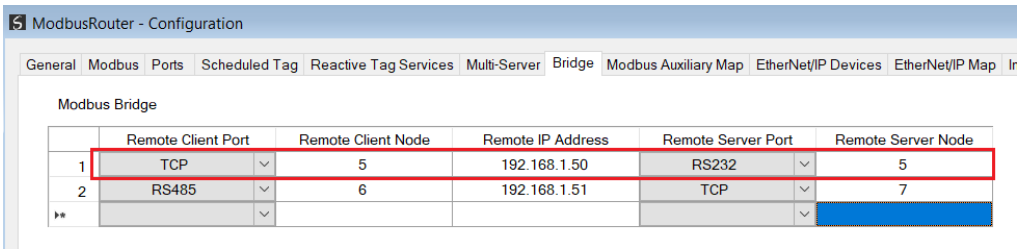


Figure 6.21 – Bridge Mode Map – TCP to RS232.

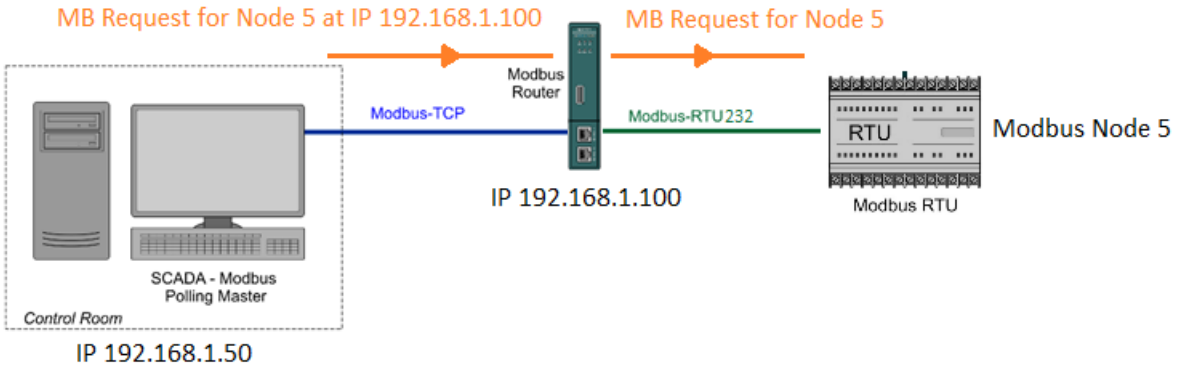


Figure 6.22 – Bridge Mode Diagram – TCP to RS232.

In the next example below, any Modbus request received on the RS485 port for a Modbus Server at Node address 6, will be routed to the Ethernet TCP port and sent to the Modbus Server at Node 7 with IP address 192.168.1.51. The response will then be routed back to the request originator (the Modbus Client on RS485).

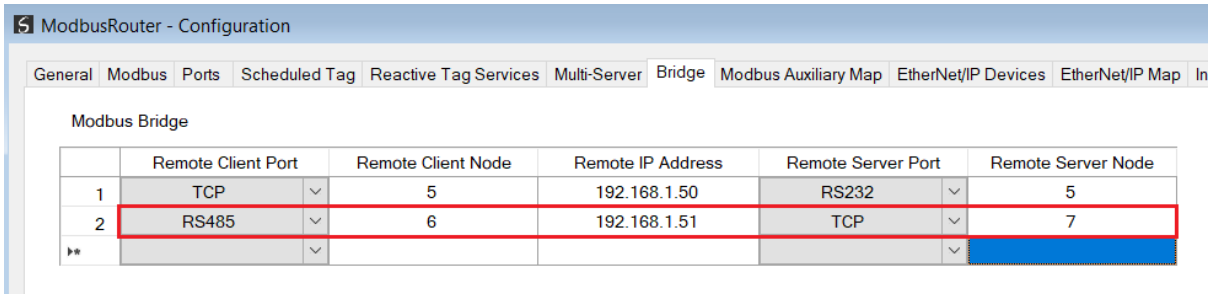


Figure 6.23 – Bridge Mode Map – RS485 to TCP.

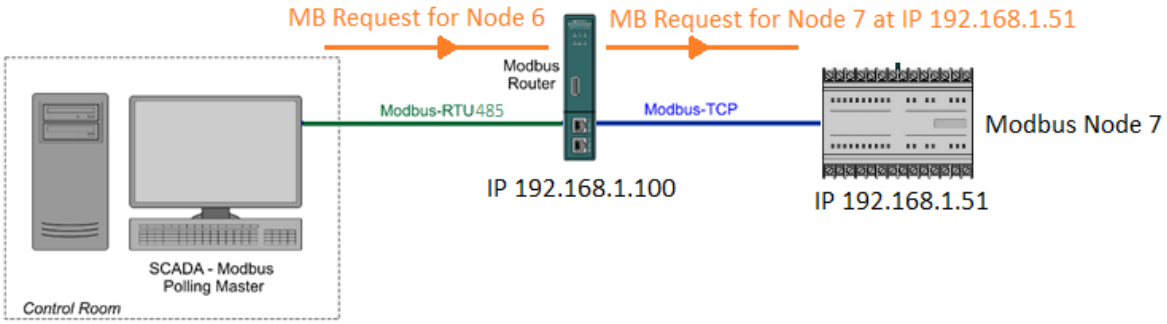


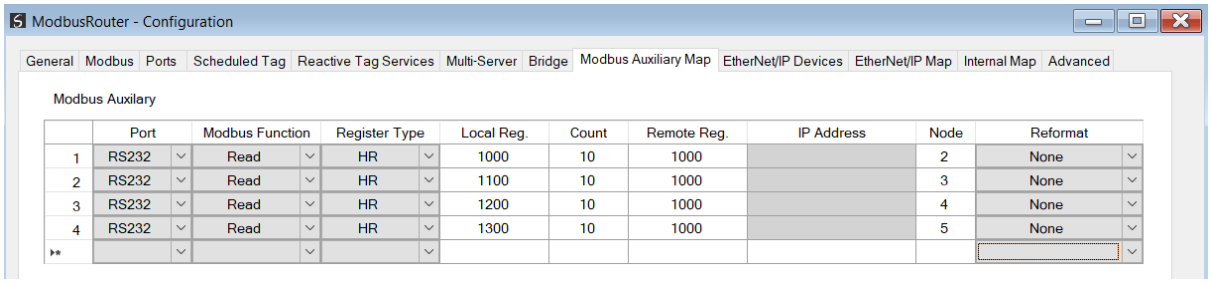
Figure 6.24 – Bridge Mode Diagram – RS485 to TCP.

6.6. CONCENTRATOR MODE

In Concentrator Mode, the MBR/B will collect data from various Modbus devices and provide the consolidated data to one or more Modbus Clients. The MBR/B will be a Modbus Server on one port, and a Modbus Client on the other ports. This will allow a Modbus SCADA to efficiently read the data from many Modbus devices, because the data from the various Modbus devices have been concentrated into the MBR/B.

In the example below the Modbus SCADA is communicating on Modbus TCP to the MBR/B which is serving as a proxy for the Modbus RTU devices in the field. The MBR/B will collect data from the Modbus RTU devices via the radio network and save it in its internal Modbus Registers. Modbus HR 1000 will be read from each node and placed at 100 intervals in the MBR/B internal MB registers (e.g., data from node 2 will be at HR1000, data from node 3 will be at HR1100, data from node 4 will be at HR1200, etc.).

The Modbus SCADA will then be able to read that data from the MBR/B at the specific Modbus Registers.



	Port	Modbus Function	Register Type	Local Reg.	Count	Remote Reg.	IP Address	Node	Reformat
1	RS232	Read	HR	1000	10	1000		2	None
2	RS232	Read	HR	1100	10	1000		3	None
3	RS232	Read	HR	1200	10	1000		4	None
4	RS232	Read	HR	1300	10	1000		5	None
▸*									

Figure 6.25 – Concentrator Mode – Modbus Auxiliary Mapping.

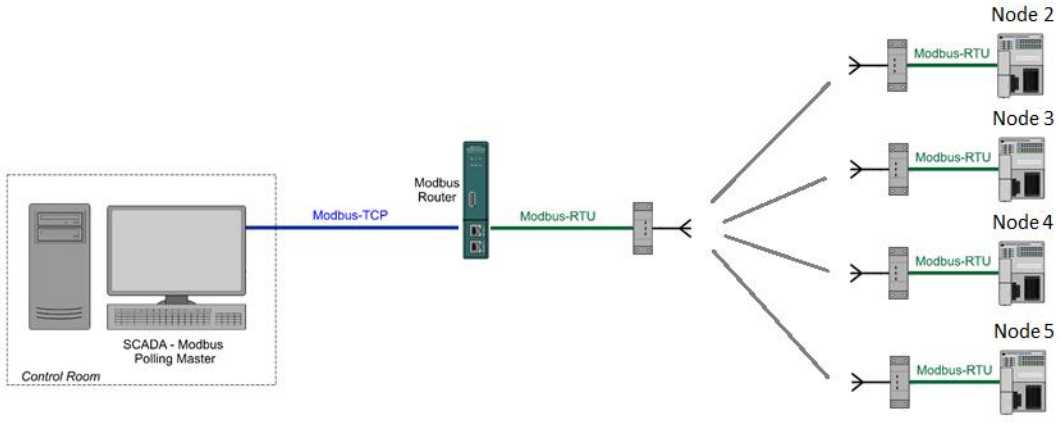


Figure 6.26 – Concentrator Mode Diagram.

6.7. INTERNAL MAP DATA FORMATS

The following tables describe the raw format of the data structures that can be mapped in the Internal Map table configuration.

6.7.1. SYSTEM STATUS

When copying system information, the source type needs to be System.

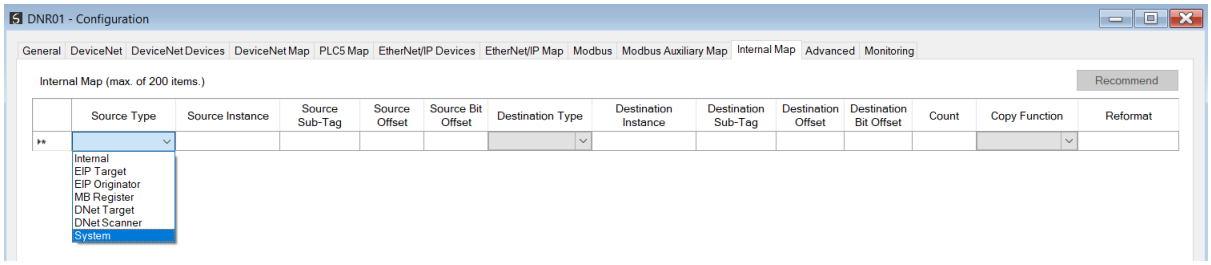


Figure 6.27 – IDS Copy – System Information.

The module's system information has the following format.

Parameter	Data Type	Description
Status	INT	Module Status. Bit 0 – Module Config Valid Bit 1 – Routing Inhibited Bit 2 – Disabled by Logix Standby Bit 3 – EtherNet/IP Originator Comms Ok Bit 4 – EtherNet/IP Target Comms Ok Bit 5 – Controller Run (1)/Prog Mode (0) Bit 6 – NTP Ok Bit 7 – Modbus Comms Ok Bit 8 – Modbus Client (0) / Server (1)
ConfigCRC	INT	The module configuration signature.
Mode	SINT	0 – Scheduled 1 – Reactive 2 – Unscheduled 3 – Reserved 4 – Enhanced 5 – Bridge 6 – Concentrator
Protocol	SINT	**Only valid for Scheduled, Reactive, and Unscheduled Mode**

		0 – Serial 1 – TCP
Transaction Rate	INT	**Only valid for Scheduled and Reactive Mode** The transaction rate is the number of Modbus messages per second that the module is currently routing.
Scheduled Tag Count	INT	**Only valid for Scheduled Mode** The number of items in the Scheduled Tag Map.
Logix Redundant State	INT	**Only valid for Scheduled and Reactive Mode** When the module is operating in Logix – Redundant USB this field will show the redundant status of the local Logix controller connected via USB. **0x02 - Primary SYNC** The local Logix controller is the primary and is synchronized with the secondary. **0x03 - Primary DISQ** The local Logix controller is the primary and the secondary controller is disqualified. **0x04 - Primary ALONE** The local Logix controller is the primary with no secondary controller. **0x10 - Primary Locked** The local Logix controller is the primary and is locked for changes. **0x08 - Secondary SYNC** The local Logix controller is the secondary and is synchronized with the primary. **0x09 - Secondary DISQ** The local Logix controller is the secondary and is disqualified with the primary. **0x0E - No Partner** The local Logix controller is the secondary with no controlling partner. **0x12 - Secondary Locked** The local Logix controller is the secondary and is locked for changes.

Table 6.7 – System Information Format

6.7.2. MB AUX ITEM STATUS

When using the MB Aux source, the source instance will be an item in the Modbus Auxiliary Map. For example, if there are two items in the Modbus Auxiliary Map, then there will be two Source Instances for the MB Aux Source Type.

NOTE: The Modbus Aux source is only relevant when the MBR/B is operating in Enhanced mode as a Modbus Client.

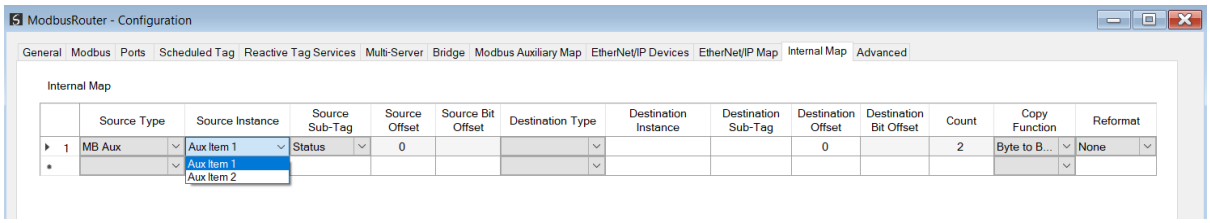


Figure 6.28 – IDS Copy - Modbus Register Source Instance.

The MB Aux format is as shown below:

Parameter	Data Type	Description
MB Aux Status	INT	Bit 0 – Last Transaction Ok

Table 6.8 – Modbus Aux Item Status.

6.7.3. ETHERNET/IP IO DEVICE STATUS

The user can select to copy the Data, or Status, from the EtherNet/IP connection.

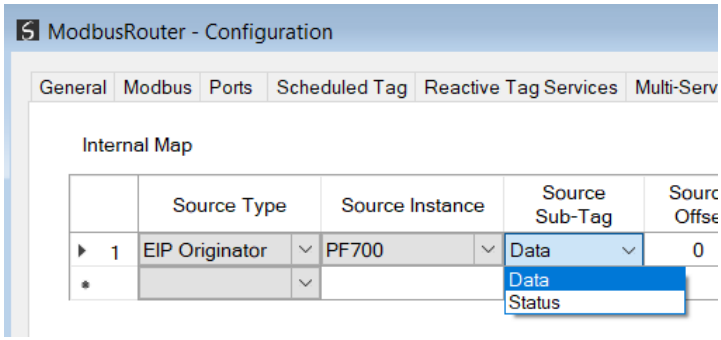


Figure 6.29 – IDS Copy – EtherNet/IP Originator Status.

When selecting the Status, the format of the Status information is shown below:

Parameter	Data Type	Description
EtherNet/IP Originator Connection Status	DINT	Bit 0 – Connection Ok

Table 6.9 – EtherNet/IP Originator Connection Status.

7. LOGIX REDUNDANCY

The Modbus Router will allow the user to create a Logix redundant architecture where the active module will follow the Primary Logix controller and use a shared (Modbus) IP address. When the previously Secondary Logix becomes the Primary, the Modbus Router connected to the new Primary Logix will take-over and assume the shared (Modbus) IP address. The Modbus Router connected to the new Secondary Logix, will disable its Modbus functionality.

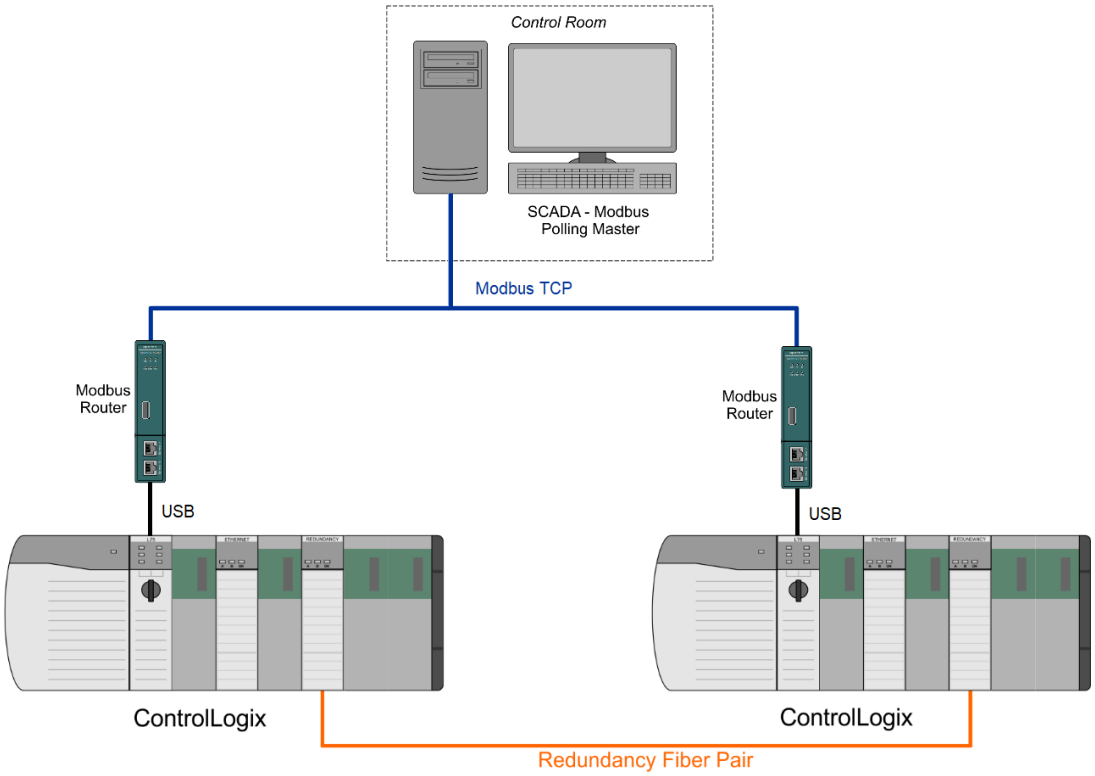


Figure 7.1 - Redundant Logix Controller communicating with Modbus Router via USB.

7.1. SETUP

In this architecture the user will need to set the *Router Mode* to ***Reactive***, and the *Controller Mode* to ***Logix – USB Redundant*** (see below). This will enable IP Address B which is used for the second Modbus Router connected to the other pair of the Logix Controllers.

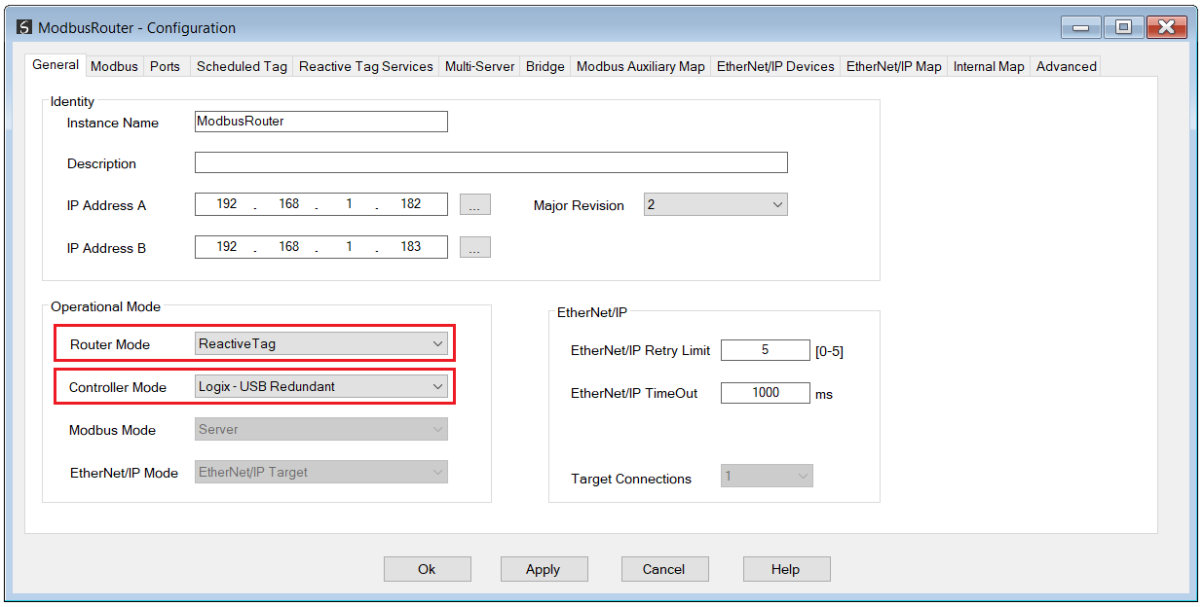


Figure 7.2 – Selecting Logix – USB Redundant Controller Mode.

Next the user will need to enter the IP address of each Modbus Router used in the redundant pair:

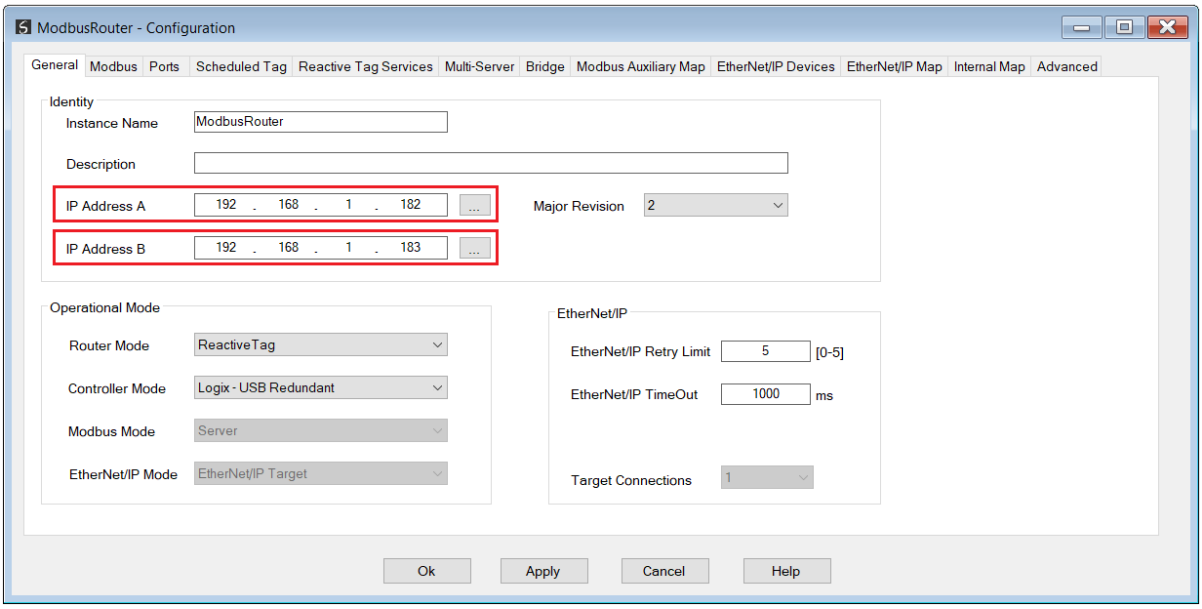


Figure 7.3 – Setting redundant pair IP addresses.

Next the user will need to set the Modbus protocol to Modbus TCP and set the shared Modbus TCP IP address between the two Modbus Routers used for the redundant architecture.

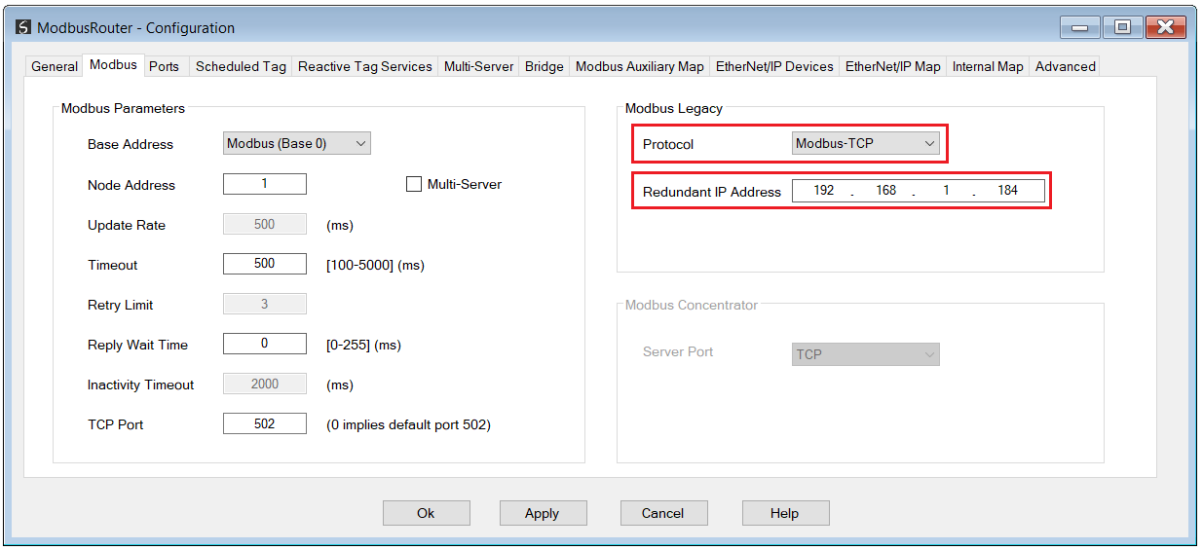


Figure 7.4 – Setting Modbus Interface.

The Modbus Client needs to be configured to communicate with the Modbus ***Redundant IP Address*** via Modbus TCP (e.g. 192.168.1.184).

7.2. OPERATION

The Modbus Router connected to the Primary Logix controller (in the redundant system) will activate the shared Modbus TCP IP address (which in the example above is 192.168.1.184). This will allow the Modbus Client to communicate with the Primary Logix controller via the Modbus Router connected to it.

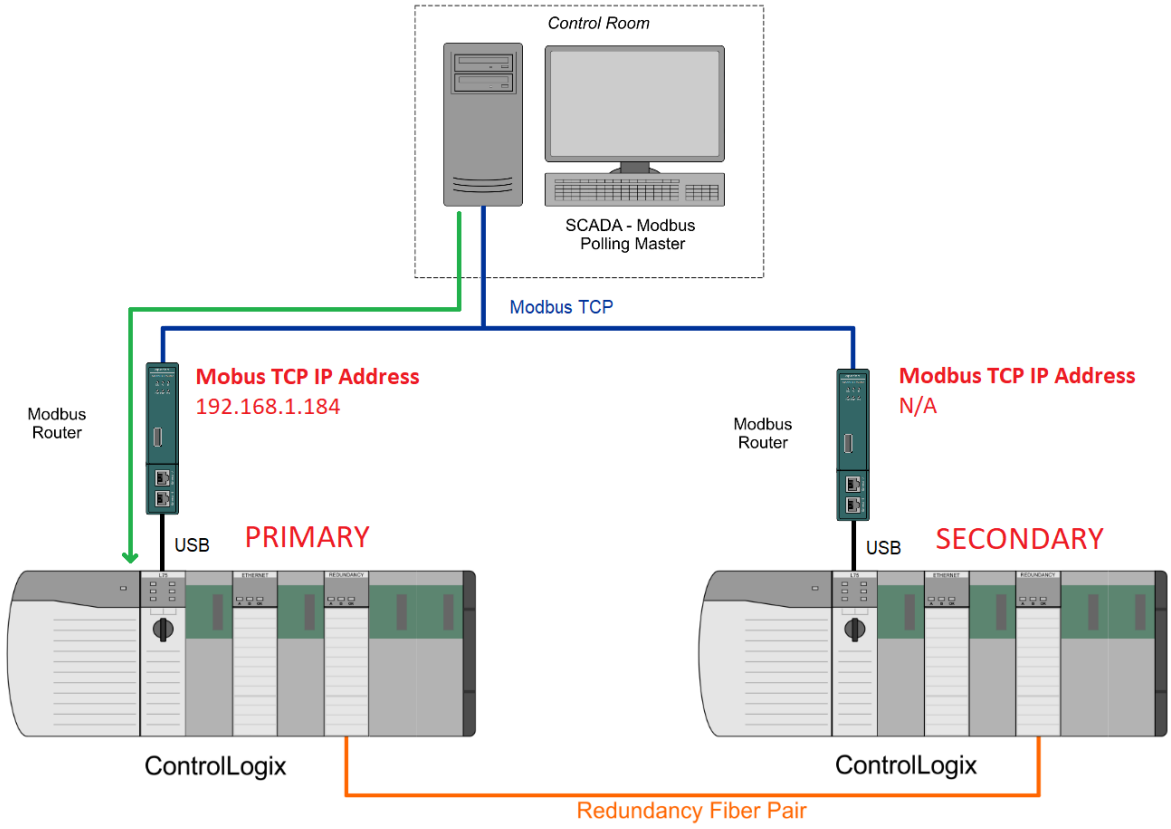


Figure 7.5 – Modbus Client connected to Primary Logix Controller.

In the event of a failure on the Primary Logix Controller the Secondary Logix Controller will take over operation. At that moment the Modbus Router connected to the “new” Primary Logix Controller will activate the shared Modbus TCP IP address (192.168.1.184) and the Modbus Router previously connected to the Primary Logix controller will release the shared Modbus Router IP Address.

At this moment the Modbus Client will have seamlessly switched Modbus communication to the “new” Primary Logix controller.

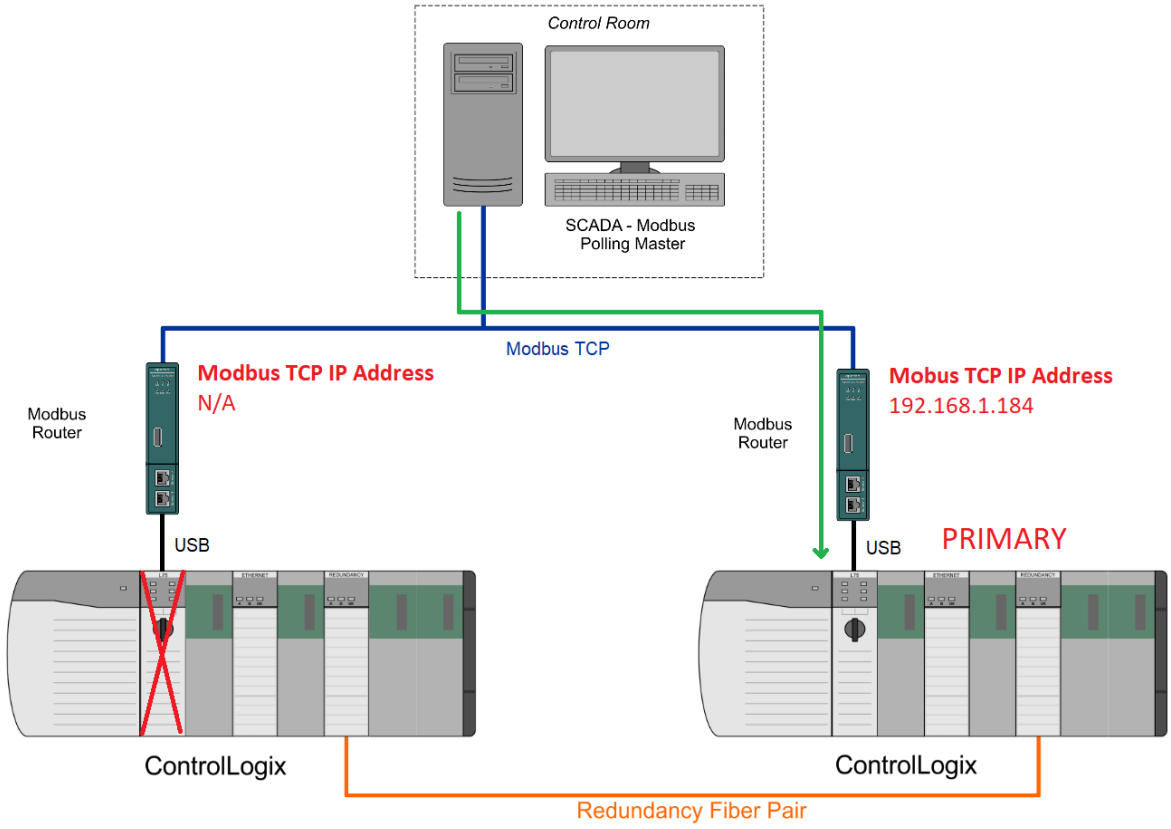


Figure 7.6 – Modbus Client connected to “new” Primary Logix Controller.

For the status of Modbus Router’s Logix Redundancy, the user can launch the Status window in Slate for each of the Modbus Router pair.

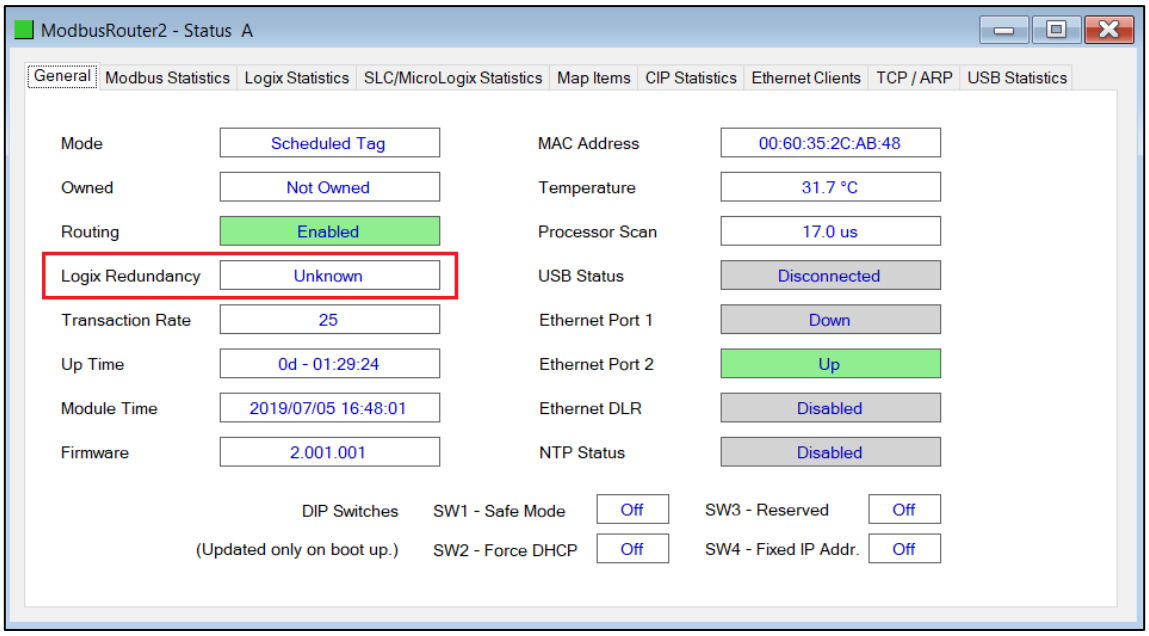


Figure 7.7 – Modbus Router’s Logix Redundancy in Status Window.

Parameter	Description
Logix Redundancy	When the module is operating in Logix – Redundant USB this field will show the redundant status of the local Logix controller connected via USB. **Primary SYNC** The local Logix controller is the primary and is synchronized with the secondary. **Primary DISQ** The local Logix controller is the primary and the secondary controller is disqualified. **Primary ALONE** The local Logix controller is the primary with no secondary controller. **Primary Locked** The local Logix controller is the primary and is locked for changes. **Secondary SYNC** The local Logix controller is the secondary and is synchronized with the primary. **Secondary DISQ** The local Logix controller is the secondary and is disqualified with the primary. **No Partner** The local Logix controller is the secondary with no controlling partner. **Secondary Locked** The local Logix controller is the secondary and is locked for changes. **Unknown** Redundancy state is unknown. Either incorrect mode or not connected via USB to a Logix controller.

Table 7.1 - Parameters displayed in the Status Monitoring – General Tab.

NOTE: The user will need to ensure that the Modbus Client destination Modbus TCP IP address is the shared IP address configured (as shown previously). Using the Modbus Router's main EtherNet/IP IP address can result in the Modbus Client reading values from the Secondary Logix Controller.

8. LOGIX USB CONNECTIVITY

The Modbus Router will allow the user to connect to a Logix controller with Studio 5000 over USB (as shown below):

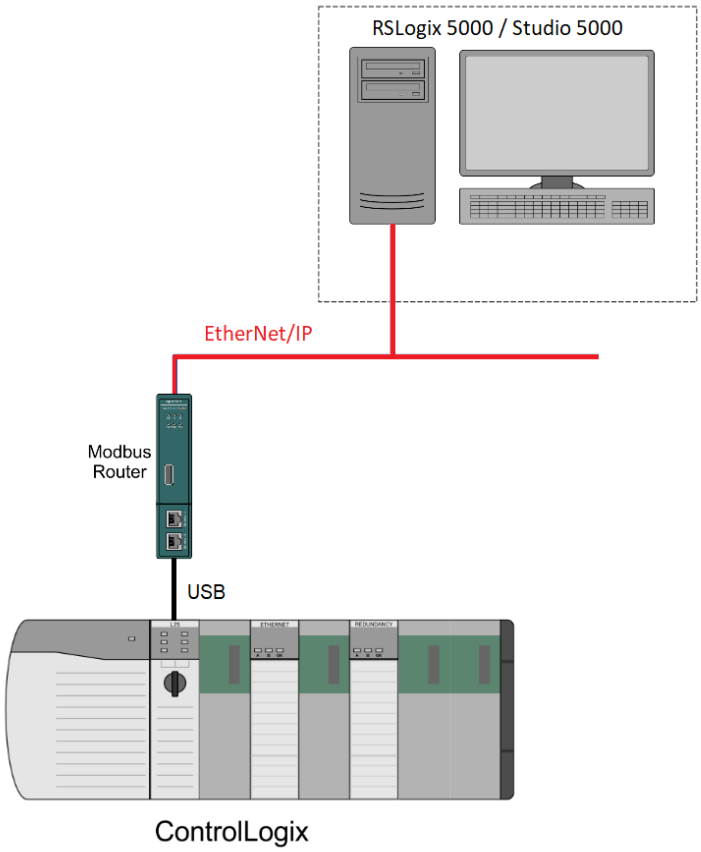


Figure 8.1 – Connect to Logix with Studio 5000 over USB.

NOTE: The Controller Mode selected in the Modbus Router Slate configuration must be USB.

Once the user is connected to the USB port of either a Logix controller (e.g. L7x) or an Ethernet bridge (e.g. EN2TR) and the Modbus Router/B EDS file has been installed, then the user can browse the local Logix backplane by expanding the USB Port network under the Modbus Router/B in RSLinx.

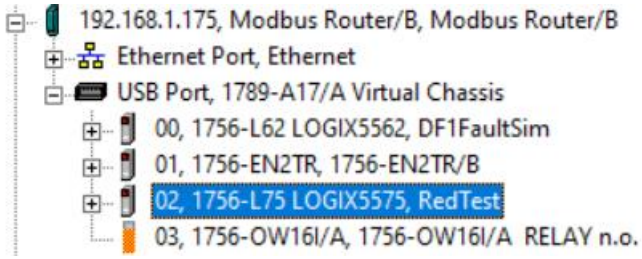


Figure 8.2 – Logix Controller seen in RSLinx via Modbus Router/B USB Port.

Once the Logix controller is seen under the USB port of the Modbus Router/B, the user can simply select this Logix controller in Studio 5000 to go online, download, or upload the Studio 5000 project.

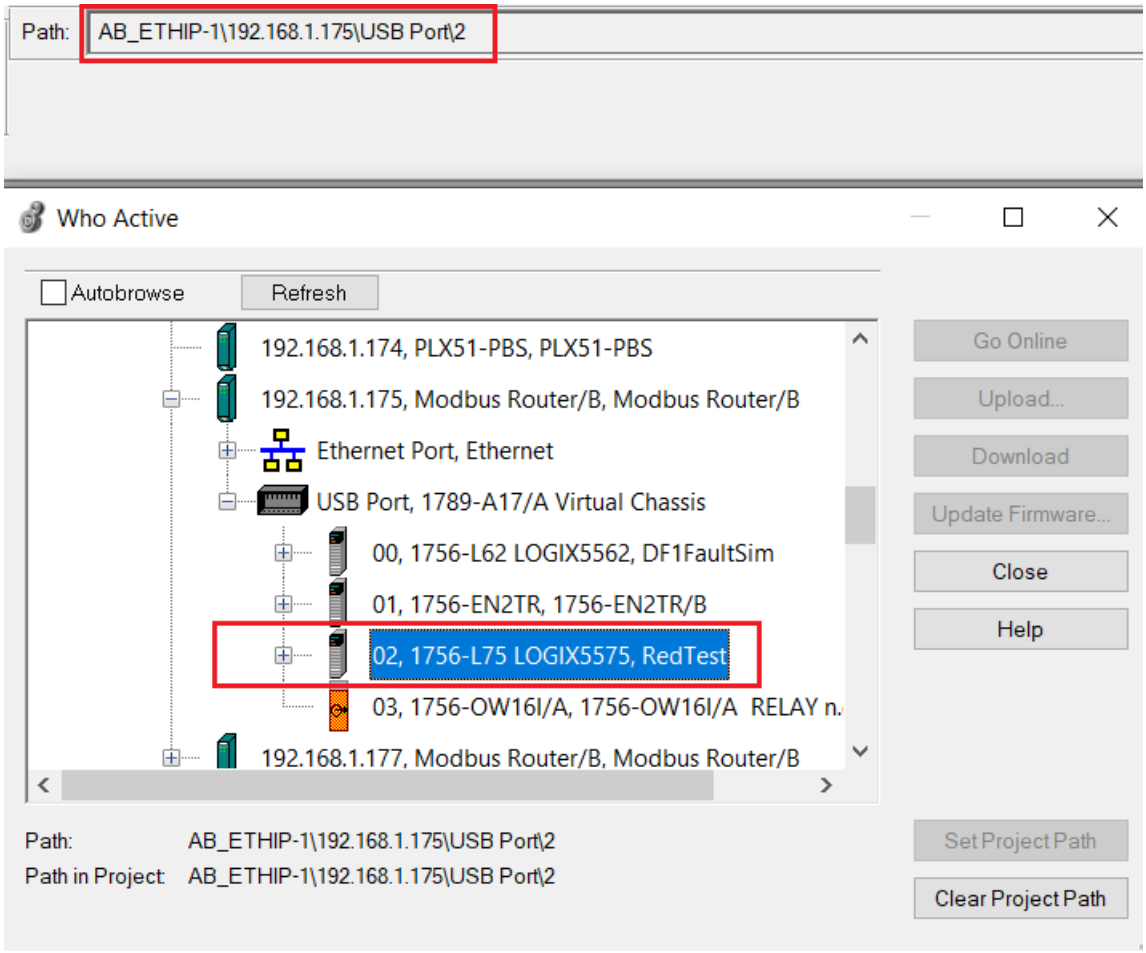


Figure 8.3 – Logix Controller seen in Studio 5000 via Modbus Router/B USB Port.

9. DIAGNOSTICS

9.1. 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 9.1 - Modbus 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 is detected. This module has two Ethernet ports A and B. Each LEDs represents each specific port.
Act	The activity LED is used for the Modbus Routing. Every time there is a successful Modbus routing transaction the LED will flash green. The LED will flash red if the routing was unsuccessful (e.g. Logix Tag does not exist).
232	The 232 LED is used for the RS232 port. Every time there is a successful Modbus packet on RS232 the LED will flash green. The LED will flash red if the Modbus packet failed (e.g. checksum failure).
485	The 485 LED is used for the RS485 port. Every time there is a successful Modbus packet on RS485 the LED will flash green. The LED will flash red if the Modbus packet failed (e.g. checksum failure).

Table 9.1 - Module LED operation.

9.2. MODULE STATUS MONITORING IN SLATE

The Modbus Router can provide a range of 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.

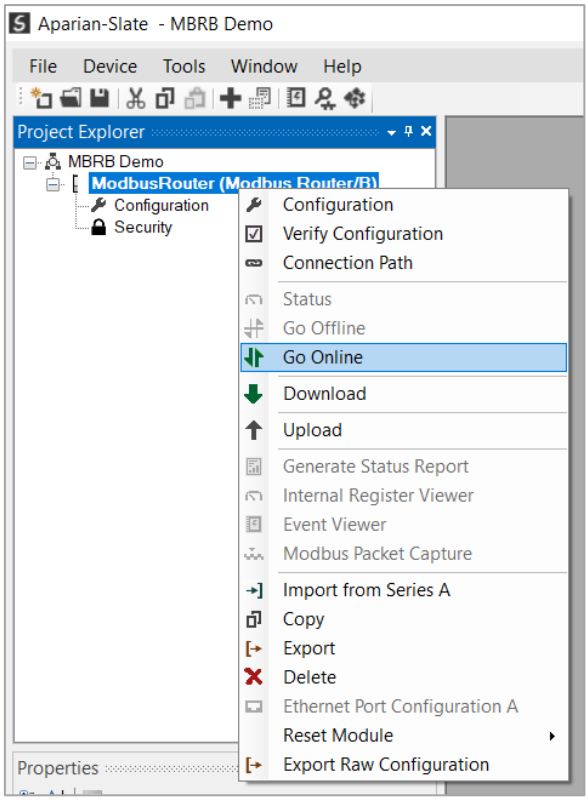


Figure 9.2 - Selecting to Go Online.

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

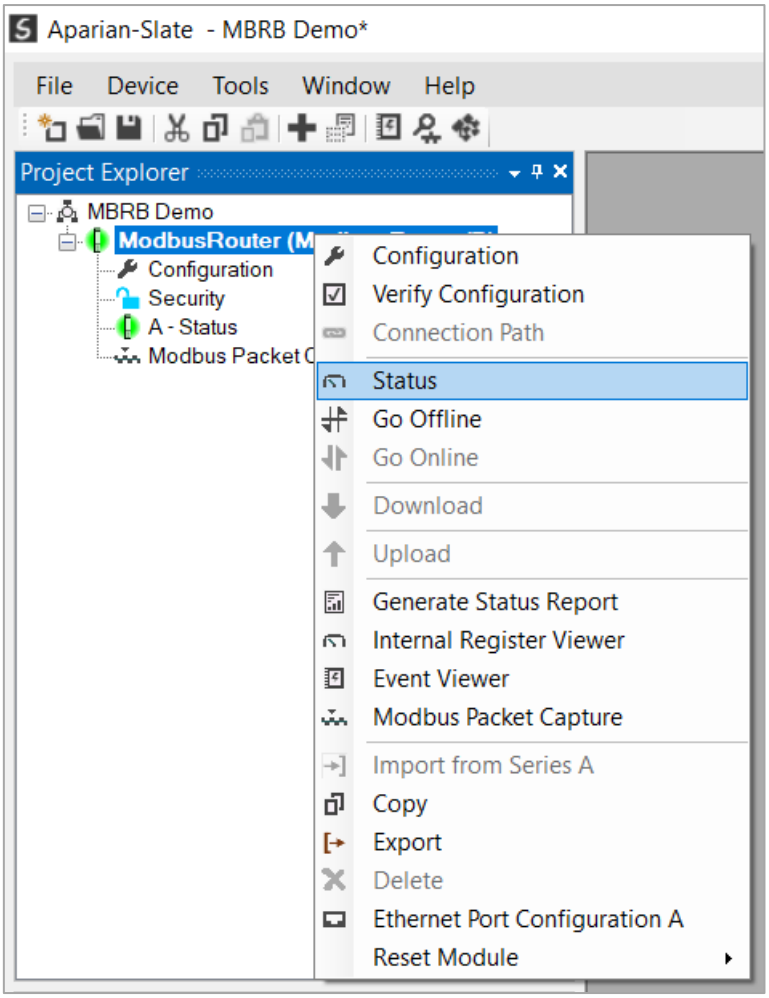


Figure 9.3 - Selecting online Status.

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.

9.2.1. GENERAL

The General tab displays the general status for the local MBR/B module.

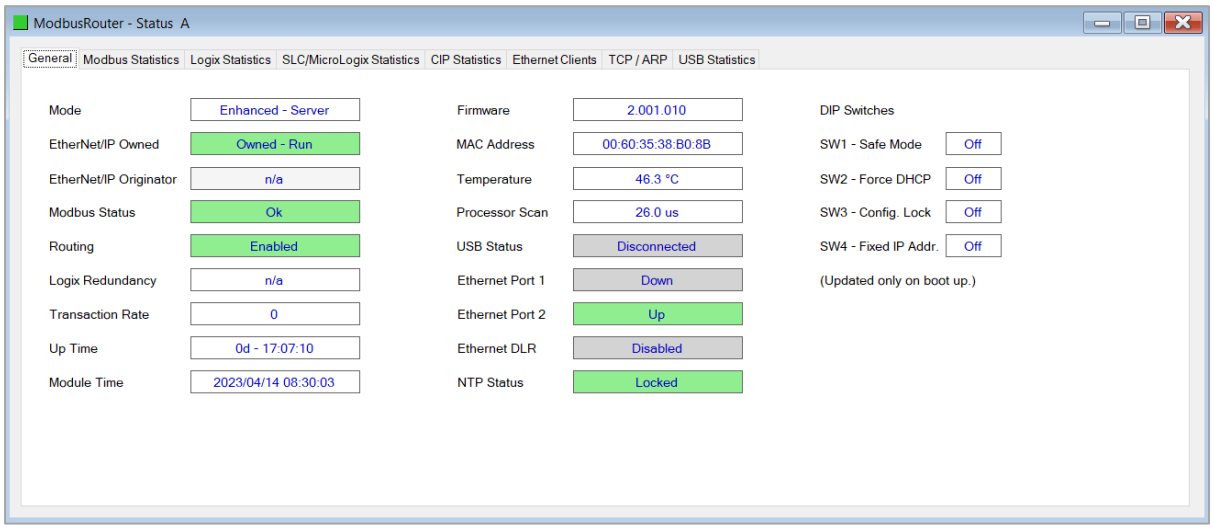


Figure 9.4 - Status monitoring – General.

The General tab displays the following general parameters:

Parameter	Description
Mode	Indicates the current operating mode: Reactive Tag Scheduled Tag Unscheduled Enhanced – Server Enhanced – Client Bridge Concentrator
EtherNet/IP Owned	Indicates whether or not the module is currently owned (Class 1) by a Logix controller. **NOTE:** The number of class 1 connections in Logix needs to match the configured EtherNet/IP Connection count in Slate.
EtherNet/IP Originator	**ONLY RELEVANT IN ENHANCED MODE** 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	**ONLY RELEVANT IN ENHANCED MODE** When the module is operating as a **Modbus Server**, 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 Client**, this parameter will indicate that all the mapping items in the Modbus Auxiliary Map are executing correctly.
Routing	**ONLY RELEVANT IN SCHEDULED AND REACTIVE MODE** Indicates whether the routing of module is enabled or inhibited. The routing operation can be inhibited in the output assembly of the module.
Logix Redundancy	**ONLY RELEVANT IN SCHEDULED AND REACTIVE MODE** When the module is operating in Logix – Redundant USB this field will show the redundant status of the local Logix controller connected via USB. **Primary SYNC** The local Logix controller is the primary and is synchronized with the secondary. **Primary DISQ** The local Logix controller is the primary and the secondary controller is disqualified. **Primary ALONE** The local Logix controller is the primary with no secondary controller. **Primary Locked** The local Logix controller is the primary and is locked for changes. **Secondary SYNC** The local Logix controller is the secondary and is synchronized with the primary. **Secondary DISQ** The local Logix controller is the secondary and is disqualified with the primary. **No Partner** The local Logix controller is the secondary with no controlling partner. **Secondary Locked** The local Logix controller is the secondary and is locked for changes. **Unknown** Redundancy state is unknown. Either incorrect mode or not connected via USB to a Logix controller.
Transaction Rate	**ONLY RELEVANT IN SCHEDULED AND REACTIVE MODE** The transaction rate is the number of Modbus messages per second that the module is currently routing.
Up Time	Indicates the elapsed time since the module was powered-up.
Module Time	Indicates the module's internal time. The module time is stored in UTC (Universal Coordinate Time) but displayed on this page according to the local PC Time Zone settings.
Firmware	Current firmware revision of the module

MAC Address	Displays the module's unique Ethernet MAC address.
Temperature	The internal temperature of the module.
Processor Scan	The amount of time (microseconds) taken by the module's processor in the last scan.
USB Status	The status of the USB connection **Connected** The USB cable is connected to the Modbus Router and partner. **Disconnected** The USB cable is not connected to either the Modbus Router or partner.
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	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.

	Note that this status will not change if the DIP switches are altered when the module is running.

Table 9.2 - Parameters displayed in the Status Monitoring – General Tab.

9.2.2. MODBUS STATISTICS

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

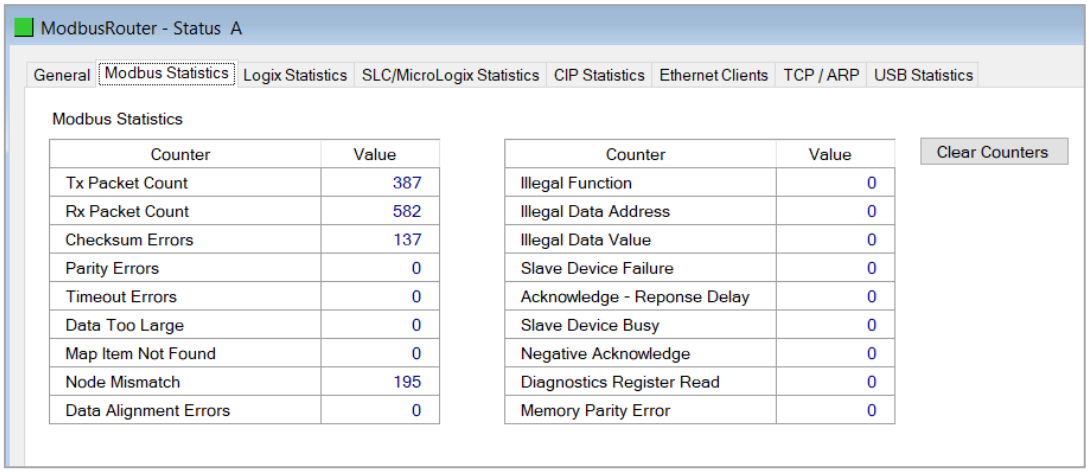


Figure 9.5 - Status monitoring – Modbus Statistics.

The Modbus Statistics tab displays the statistics associated with the Modbus communication and mapping.

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 9.3 – Modbus statistics.

9.2.3. LOGIX STATISTICS

The following Logix statistics are only relevant when the module is running in either Reactive Tag or Scheduled Tag mode and the Logix controller type has been selected.

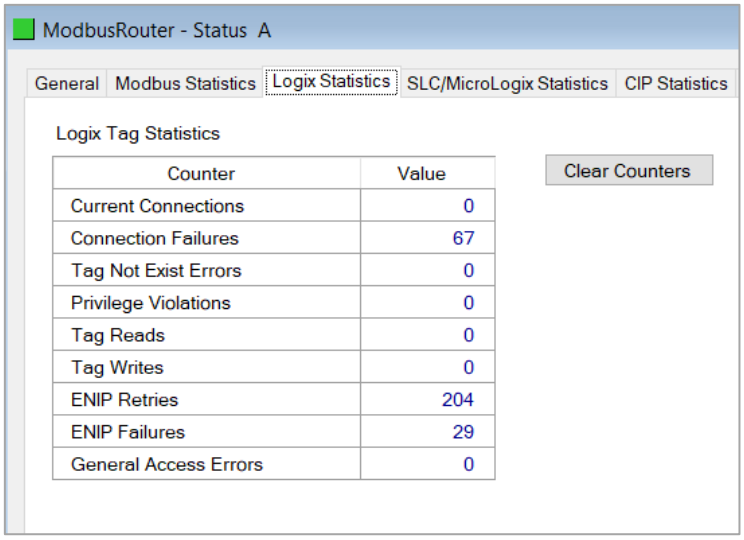


Figure 9.6 - Status monitoring – Logix Statistics.

Statistic	**Description**
Current Connections	The number of current open class 3 connections.

Connection Failures	The number of failed attempts at establishing a class 3 connections 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 Violation Errors	The number of tag read and tag write transactions that failed due to a privilege violation error. 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 Modbus Router module.
Tag Writes	The number of tag write transactions executed by the Modbus 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.
General Access Error	This count increases when a tag cannot be accessed for any other reason not reported above.

Table 9.4 - Tag Mapping statistics.

9.2.4. SLC/MicroLogix Statistics

The following SLC/MicroLogix statistics are only relevant when the module is running in either Reactive Tag or Scheduled Tag mode and the MicroLogix/SLC controller type has been selected.

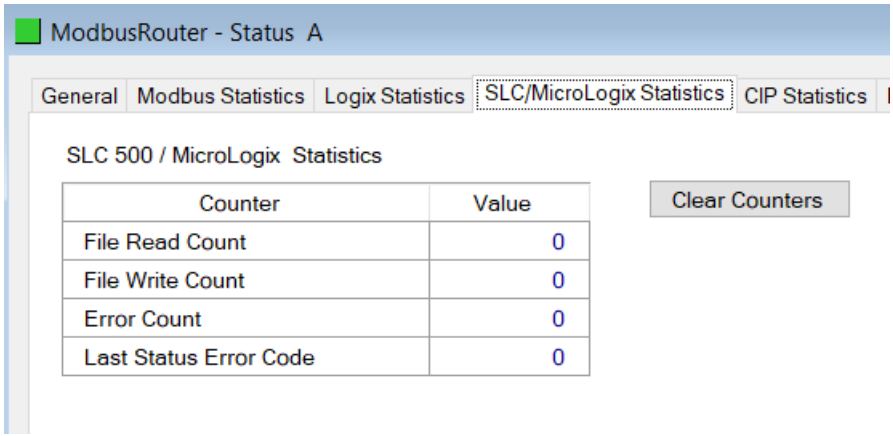


Figure 9.7 - Status monitoring – SLC/MicroLogix Statistics.

Statistic	**Description**
File Read Count	The number of file read transactions executed by the Modbus Router module.

File Write Count	The number of file write transactions executed by the Modbus Router module.
Error Count	The number of status errors received from the SLC/MicroLogix by the Modbus Router.
Last Status Error Code	The value of the last status error received.

Table 9.5 – SLC/MicroLogix statistics.

9.2.5. MAP ITEMS

The Map Items tab will display the successful packet counts processed by each mapping item. If an item count changes, then the success count field will be displayed with a green background for approximately 3 seconds. This provides quick visual feedback as to which items are currently active.

The fields in the map items will adjust to suite the appropriate mode. No items are displayed in Unscheduled mode.

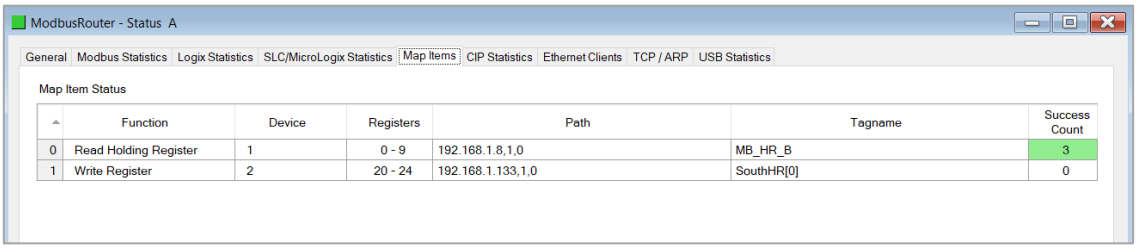


Figure 9.8 - Map Item status.

9.2.6. AUX. ITEMS

The Aux. Items tab will display the last status of the transaction by each mapping item in the Modbus Auxiliary Map. If the last transaction failed, then the status will show **Fail** and if it was successful then it will show **Ok**. This provides quick visual feedback as to which items are currently active.

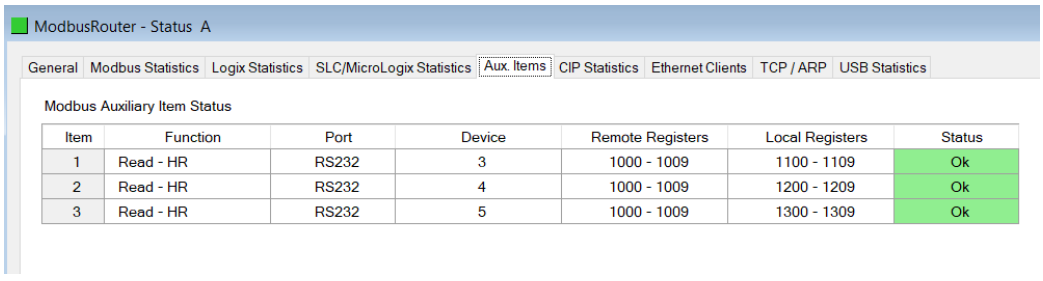


Figure 9.9 - Auxiliary Item status.

9.2.7. CIP STATISTICS

Each Modbus Router provides a set of Common Industrial Protocol (CIP) communication statistics as shown below:

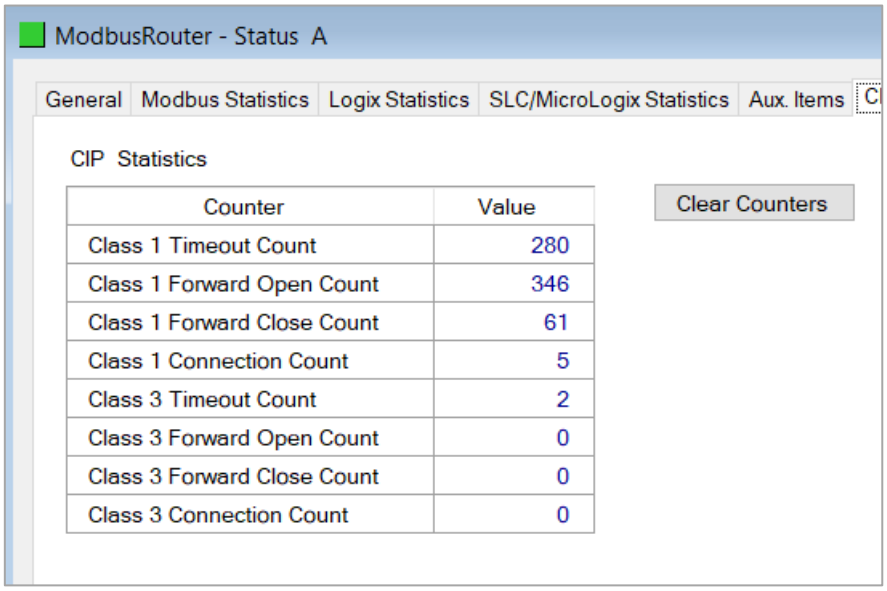


Figure 9.10 - Status monitoring – CIP Statistics.

Statistic	Description
Class 1 Timeout Count	Number of times a Class 1 connection has timed out
Class 1 Forward Open Count	Number of Class 1 Connection establish attempts
Class 1 Forward Close Count	Number of Class 1 Connection close attempts
Class 1 Connection Count	Number of Class 1 Connections currently active
Class 3 Timeout Count	Number of times a Class 3 connection has timed out
Class 3 Forward Open Count	Number of Class 3 Connection establish attempts
Class 3 Forward Close Count	Number of Class 3 Connection close attempts
Class 3 Connection Count	Number of Class 3 Connections currently active

Table 9.6 – CIP statistics.

9.2.8. 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 is operating in Enhanced mode with EtherNet/IP mode set to Originator.

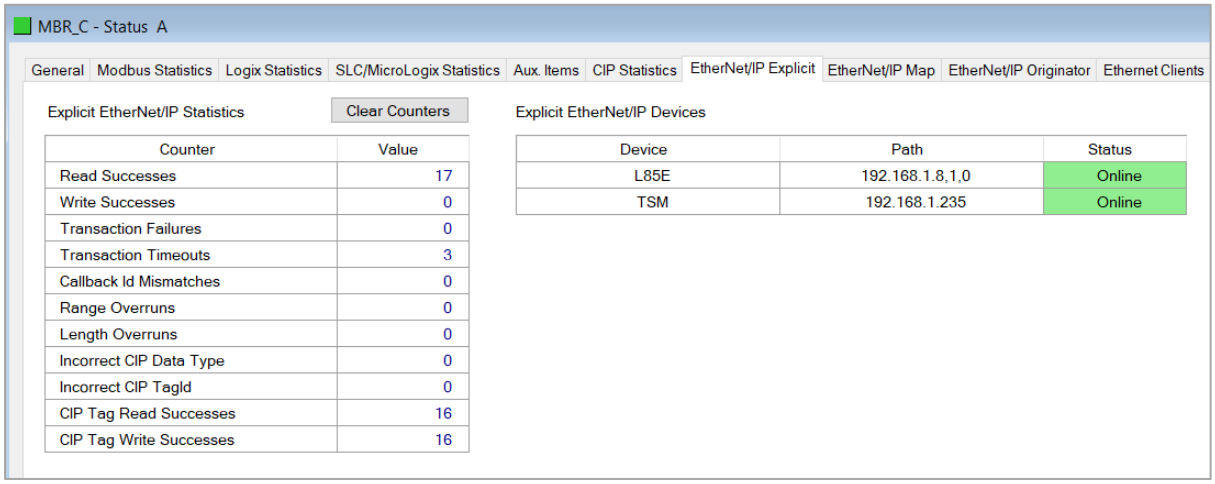


Figure 9.11 - Status monitoring – EtherNet/IP Explicit.

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 amount runs over the max Internal Data Space.
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 9.7 – EtherNet/IP Explicit Statistics.

9.2.9. 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 is operating in Enhanced mode with EtherNet/IP mode set to Originator.

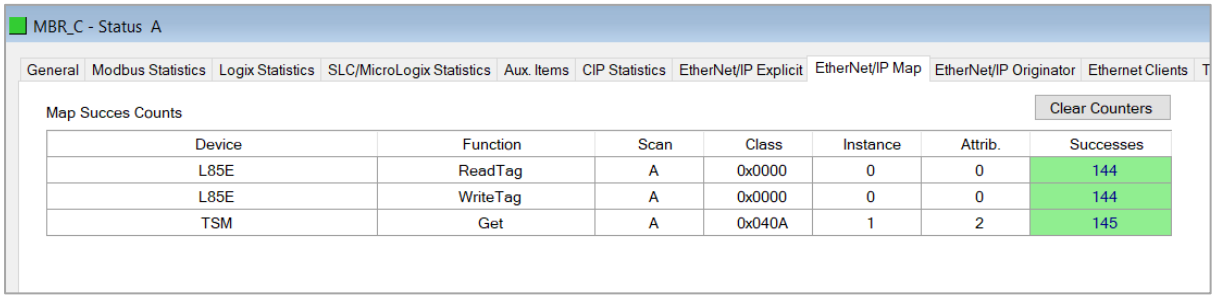


Figure 9.12 - Status monitoring – EtherNet/IP Map.

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.

9.2.10. 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 is operating in Enhanced mode with EtherNet/IP mode set to Originator.

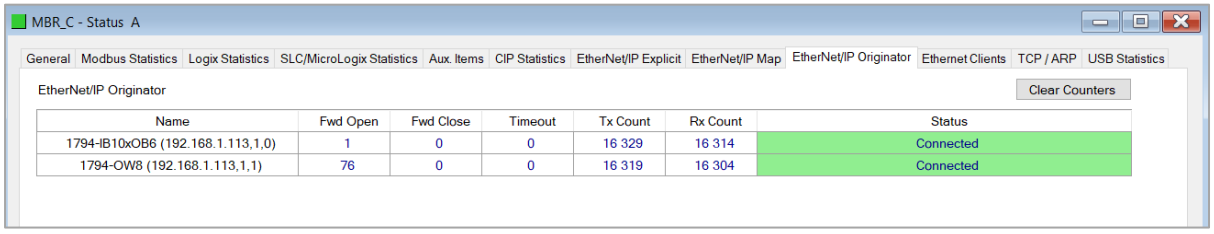


Figure 9.13 - Status monitoring – EtherNet/IP Originator.

Statistic	Description
Status	The current connection status of the module. **Connected** The device is connected and is exchanging data using Class 1 cyclic communication.

	Offline The device is offline and is 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 9.8 – EtherNet/IP Class 1 status and statistics

9.2.11. ETHERNET CLIENTS

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

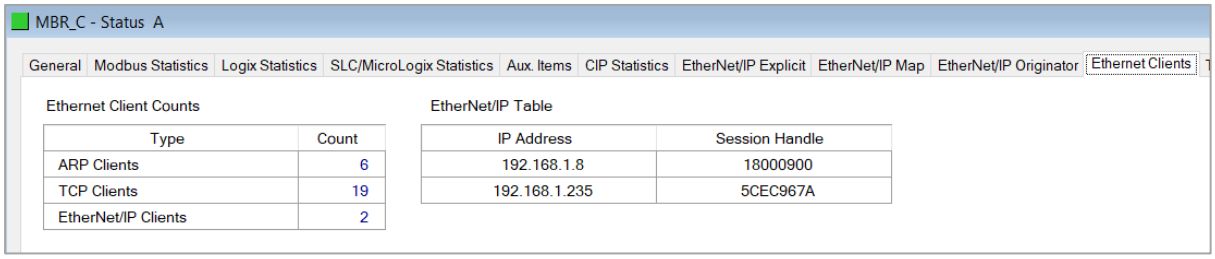


Figure 9.14 –Status monitoring – Ethernet Client Statistics.

9.2.12. TCP / ARP

The TCP/ARP tab displays details of the internal Ethernet ARP and TCP lists of the MBR/B.

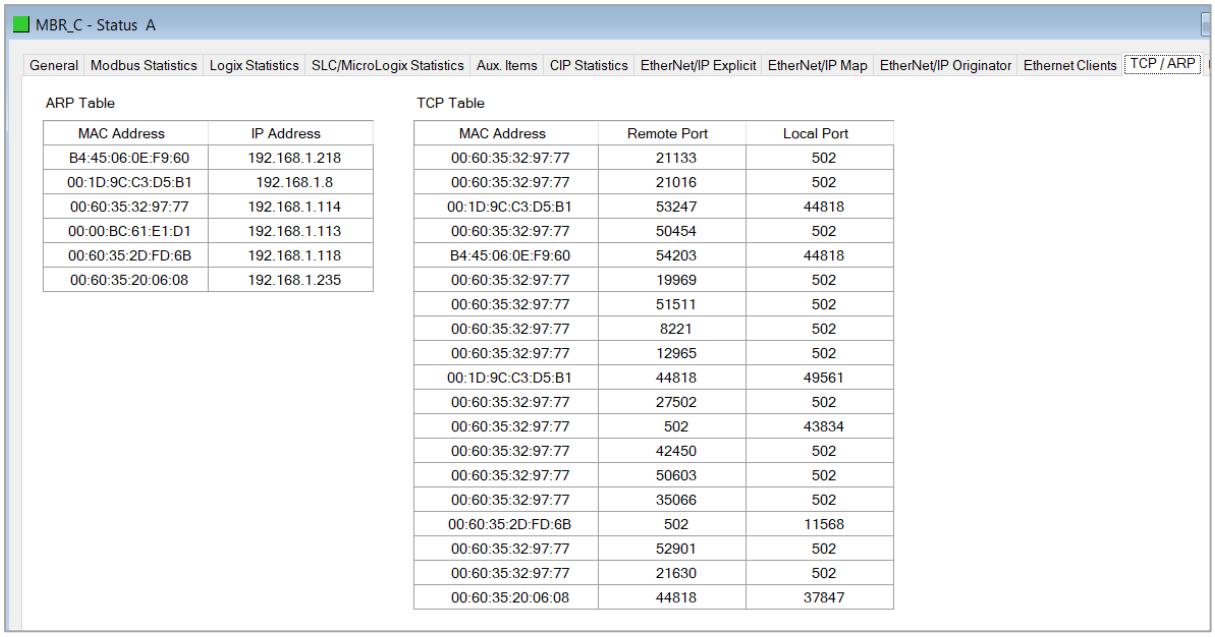


Figure 9.15 – Status monitoring – Ethernet TCP / ARP Statistics.

9.2.13. USB STATISTICS

The following USB statistics are only relevant when the *Logix – USB* or *Logix – Redundant USB* controller type has been selected.

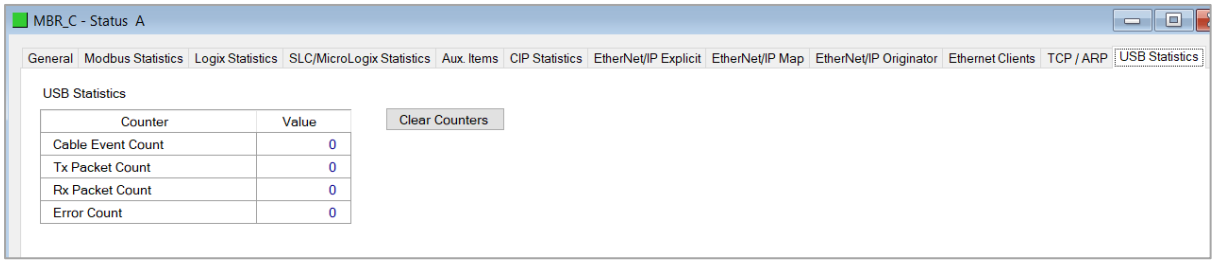


Figure 9.16 - Status monitoring – USB Statistics.

Statistic	Description
Cable Event Count	This is the number of times the USB cable has either been removed or inserted from the Host or Client.
Tx Packet Count	The number of packets sent on the USB port.
Rx Packet Count	The number of packets received on the USB port.
Error Count	The number of errors that have occurred on the USB port. The user will need to open the module event log for more detail regarding the error.

Table 9.9 – USB statistics.

9.3. TARGET DEVICE STATUS MONITORING IN SLATE

The MBR/B can also provide individual statistics and status for each for each of the EtherNet/IP Class 1 IO devices when the module is operating in Enhanced mode with EtherNet/IP mode set to Originator.

9.3.1. 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*.

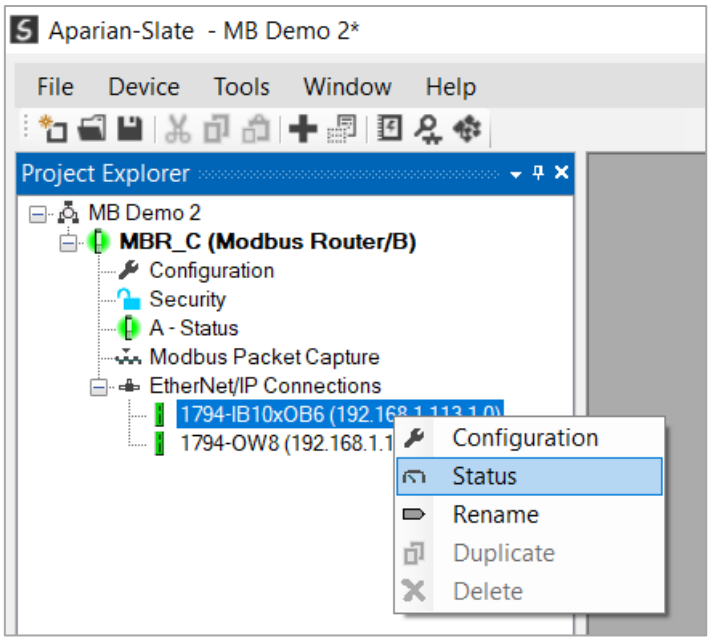


Figure 9.17 – EtherNet/IP Device Status – Status selection.

9.3.1.1. GENERAL

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

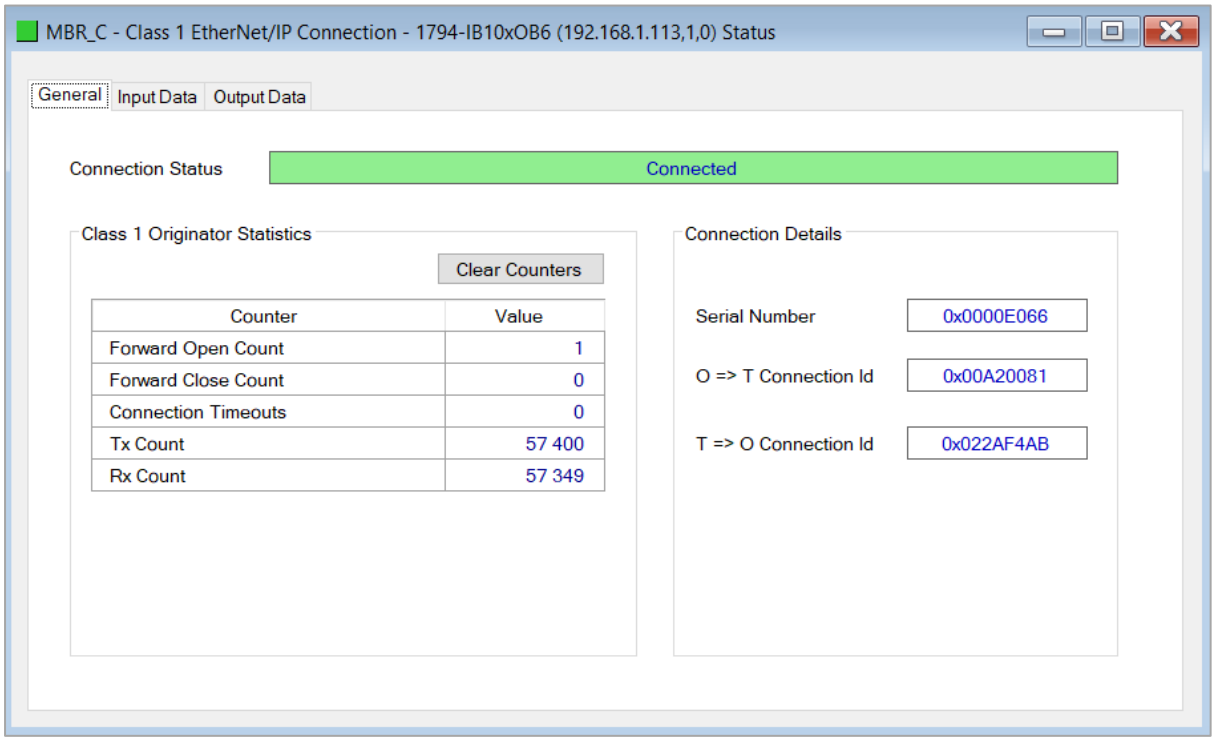


Figure 9.18 – EtherNet/IP Device Status – General Status.

Statistic	Description
Connection Status	The current connection status of the module. **Connected** The device is connected and is exchanging data using Class 1 cyclic communication. **Offline** The device is offline and is 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 9.10 – EtherNet/IP Class 1 Device status and statistics .

9.3.1.2. INPUT DATA

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

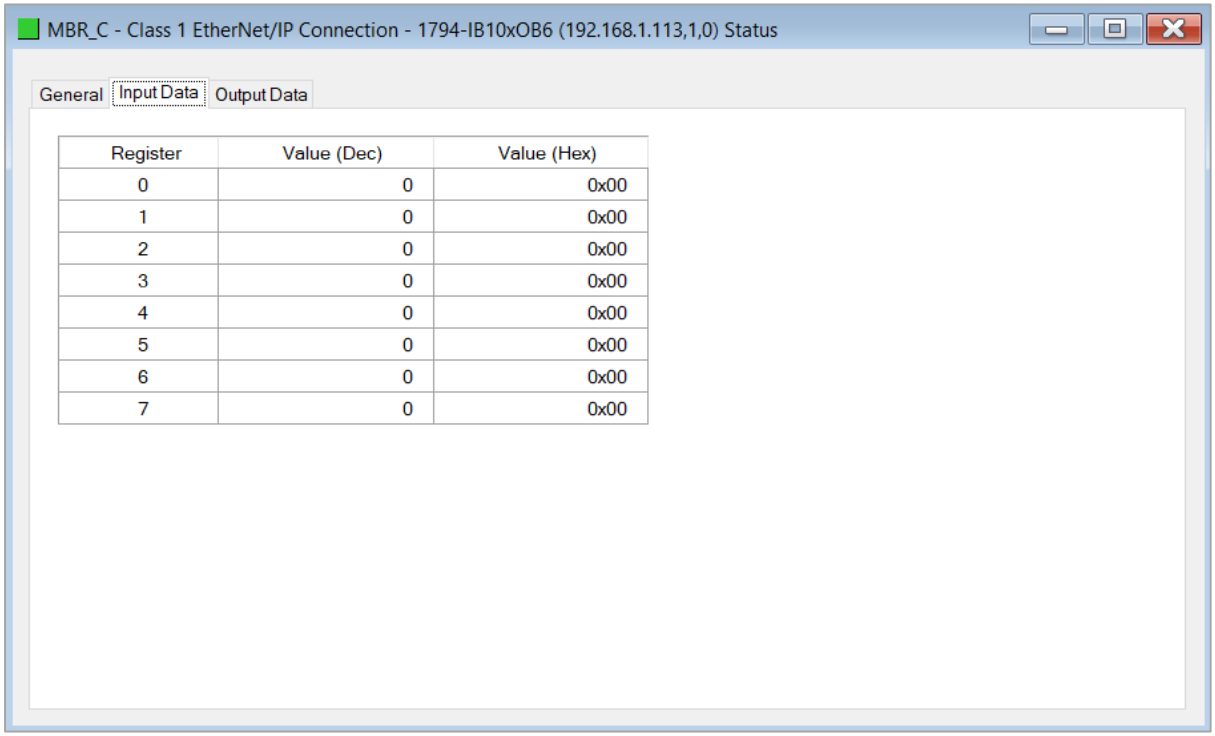


Figure 9.19 – EtherNet/IP Device Status – Input Data.

9.3.1.3. OUTPUT DATA

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

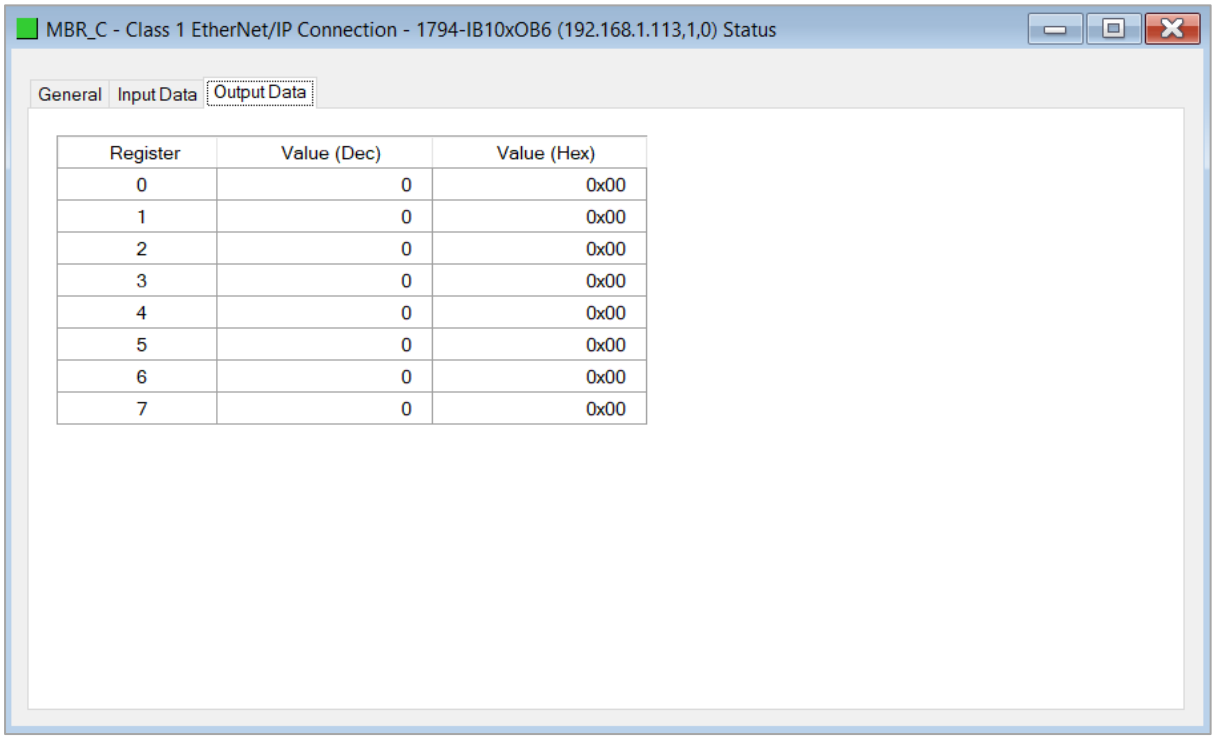


Figure 9.20 – EtherNet/IP Device Status – Output Data.

9.4. MODBUS DIAGNOSTICS

The Modbus Router supports Modbus diagnostics (function code 8) which can be read from the Modbus Client. This functionality will only be available when the Modbus Router is running in Reactive Mode. This will provide some basic tests for checking the communication between the Client and Server. The following sub-functions are supported for Modbus Diagnostics:

Sub-function code	Description
0	Return Query Data
1	Restart Communication
2	Return Diagnostic Register
10	Clear Counters and Diagnostics Register
11	Return Bus Message Count

12	Return Bus Communication Error Count
13	Return Bus Exception Error Count
14	Return Server Message Count
15*	Return Server No Response Count
16*	Return Server NAK Count
17*	Return Server Busy Count
18*	Return Bus Character Overrun Count
20*	Clear Overrun Counter and Flag

Table 9.11 - Tag Mapping statistics.

* These sub-functions are static objects which will always return a value of zero. They are supported in the Modbus Router for certain legacy systems requiring access to these sub-functions.

9.5. MODBUS PACKET CAPTURE

The module provides the capability to capture the Modbus traffic for analysis. They will allow the user and the support team to resolve any possible issue on site.

To invoke the capture of the module, double-click on the Modbus Packet Capture item in the Project Explorer tree.

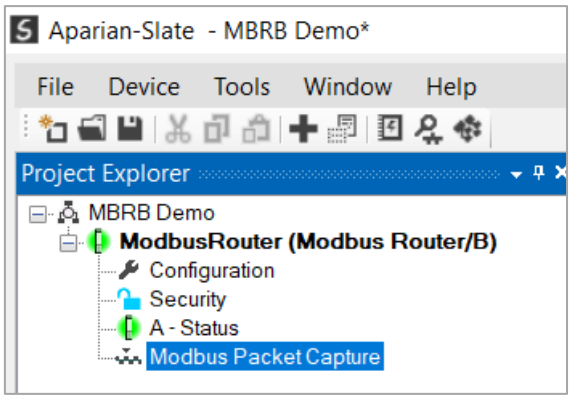


Figure 9.21 - Selecting Modbus Packet Capture.

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

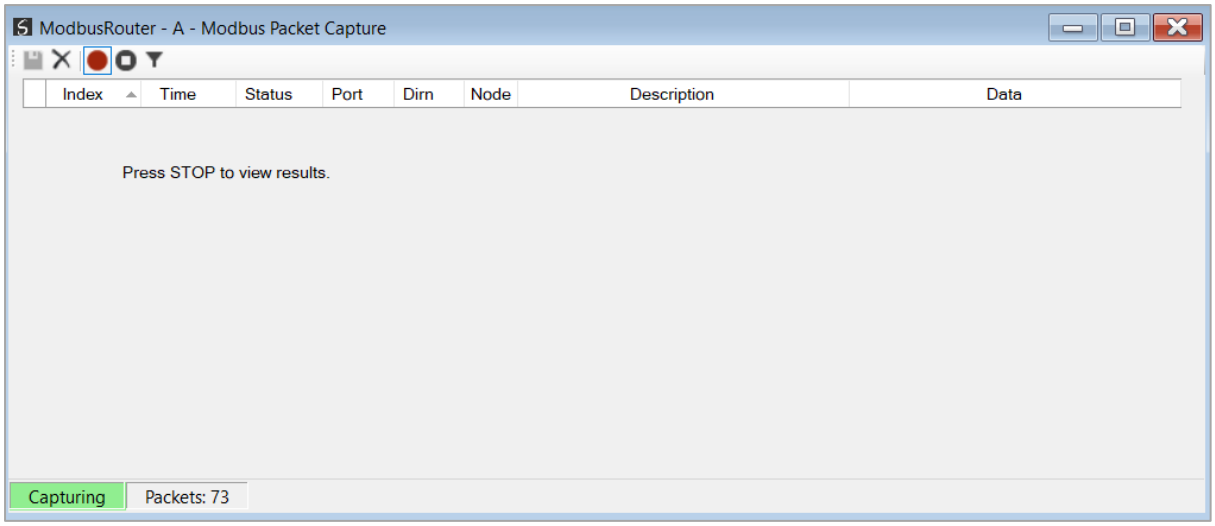


Figure 9.22 – Modbus packet capture.

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

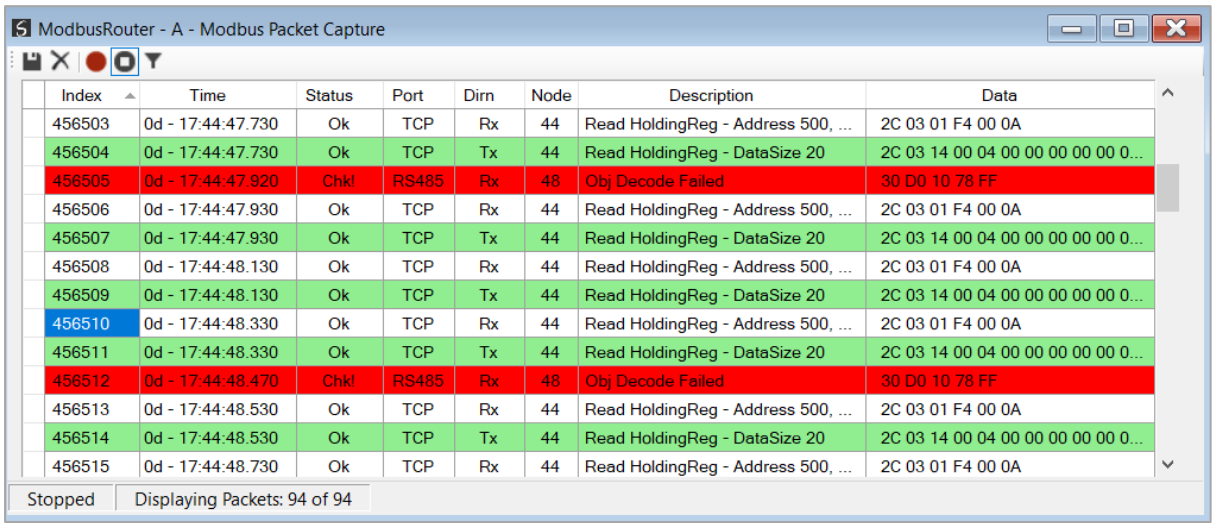


Figure 9.23 – Modbus Packet Capture complete.

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	Modbus node address of the message destination.
Description	A brief description of the packet, showing the function and register range if applicable.
Data	The raw packet data.

Table 9.12 – 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.

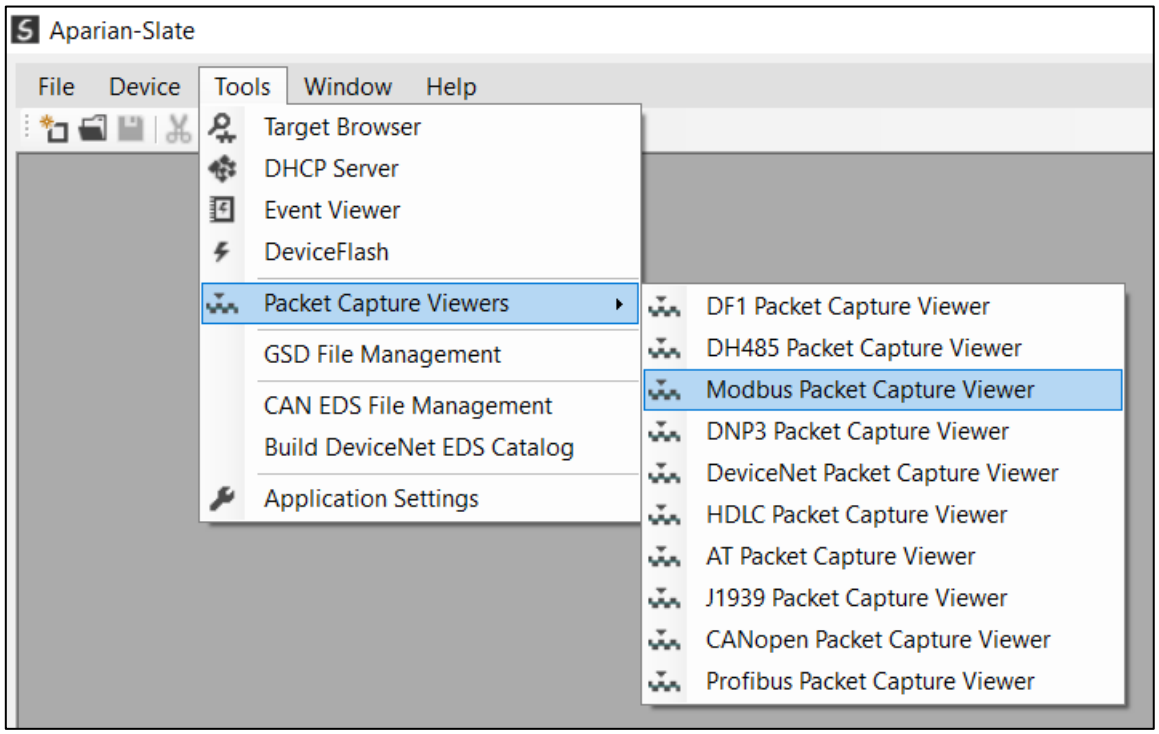


Figure 9.24 - Selecting the Modbus Packet Capture Viewer.

9.6. MODULE EVENT LOG

The Modbus 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.

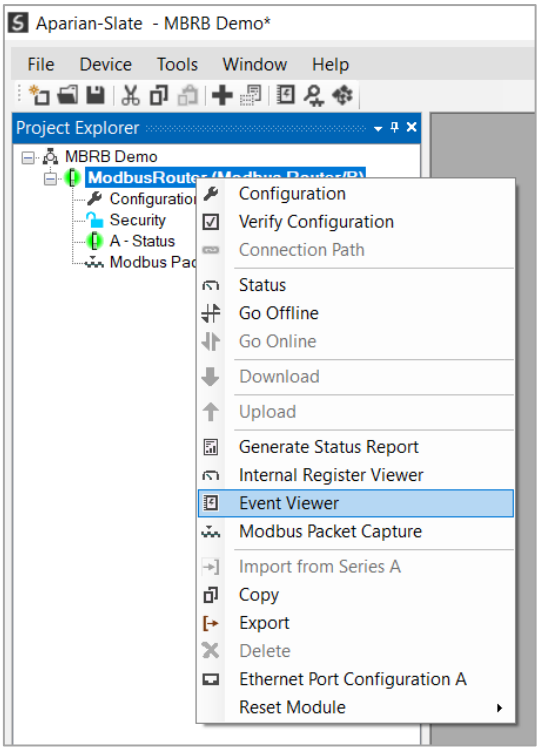
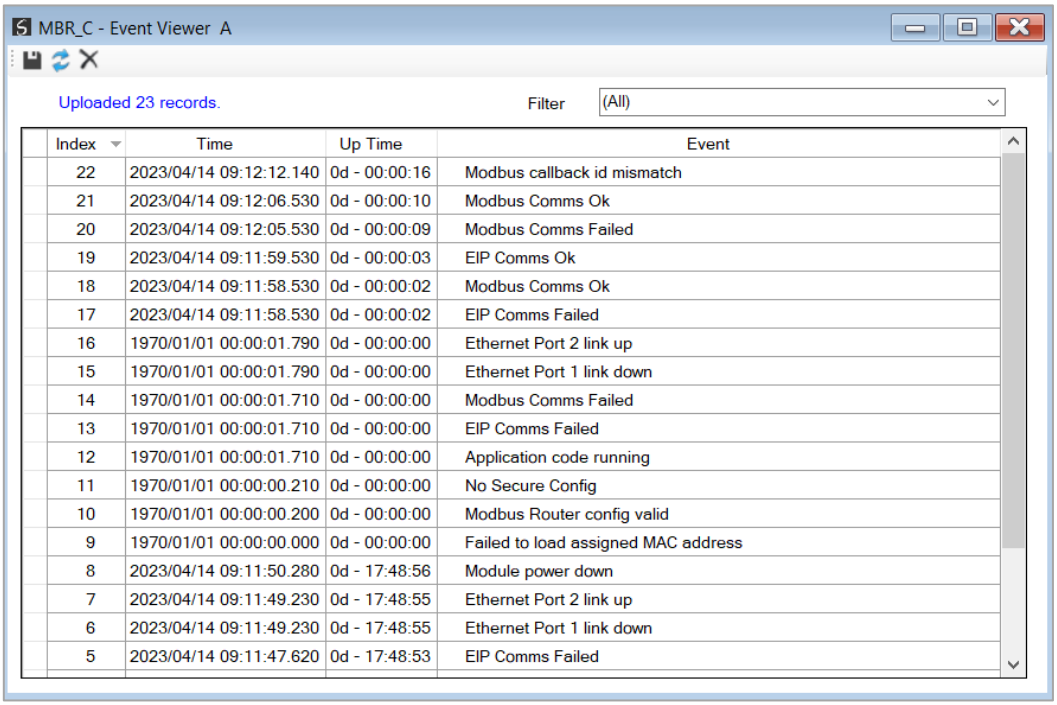


Figure 9.25 - Selecting the module Event Log.

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



Index	Time	Up Time	Event
22	2023/04/14 09:12:12.140	0d - 00:00:16	Modbus callback id mismatch
21	2023/04/14 09:12:06.530	0d - 00:00:10	Modbus Comms Ok
20	2023/04/14 09:12:05.530	0d - 00:00:09	Modbus Comms Failed
19	2023/04/14 09:11:59.530	0d - 00:00:03	EIP Comms Ok
18	2023/04/14 09:11:58.530	0d - 00:00:02	Modbus Comms Ok
17	2023/04/14 09:11:58.530	0d - 00:00:02	EIP Comms Failed
16	1970/01/01 00:00:01.790	0d - 00:00:00	Ethernet Port 2 link up
15	1970/01/01 00:00:01.790	0d - 00:00:00	Ethernet Port 1 link down
14	1970/01/01 00:00:01.710	0d - 00:00:00	Modbus Comms Failed
13	1970/01/01 00:00:01.710	0d - 00:00:00	EIP Comms Failed
12	1970/01/01 00:00:01.710	0d - 00:00:00	Application code running
11	1970/01/01 00:00:00.210	0d - 00:00:00	No Secure Config
10	1970/01/01 00:00:00.200	0d - 00:00:00	Modbus Router config valid
9	1970/01/01 00:00:00.000	0d - 00:00:00	Failed to load assigned MAC address
8	2023/04/14 09:11:50.280	0d - 17:48:56	Module power down
7	2023/04/14 09:11:49.230	0d - 17:48:55	Ethernet Port 2 link up
6	2023/04/14 09:11:49.230	0d - 17:48:55	Ethernet Port 1 link down
5	2023/04/14 09:11:47.620	0d - 17:48:53	EIP Comms Failed

Figure 9.26 – 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.

9.7. WEB SERVER

The Modbus Router provides a web server allowing a user without Slate or RSLogix 5000 to view various diagnostics of the module. This includes Ethernet parameters, system event log, advanced diagnostics, and application diagnostics (Modbus diagnostics).

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

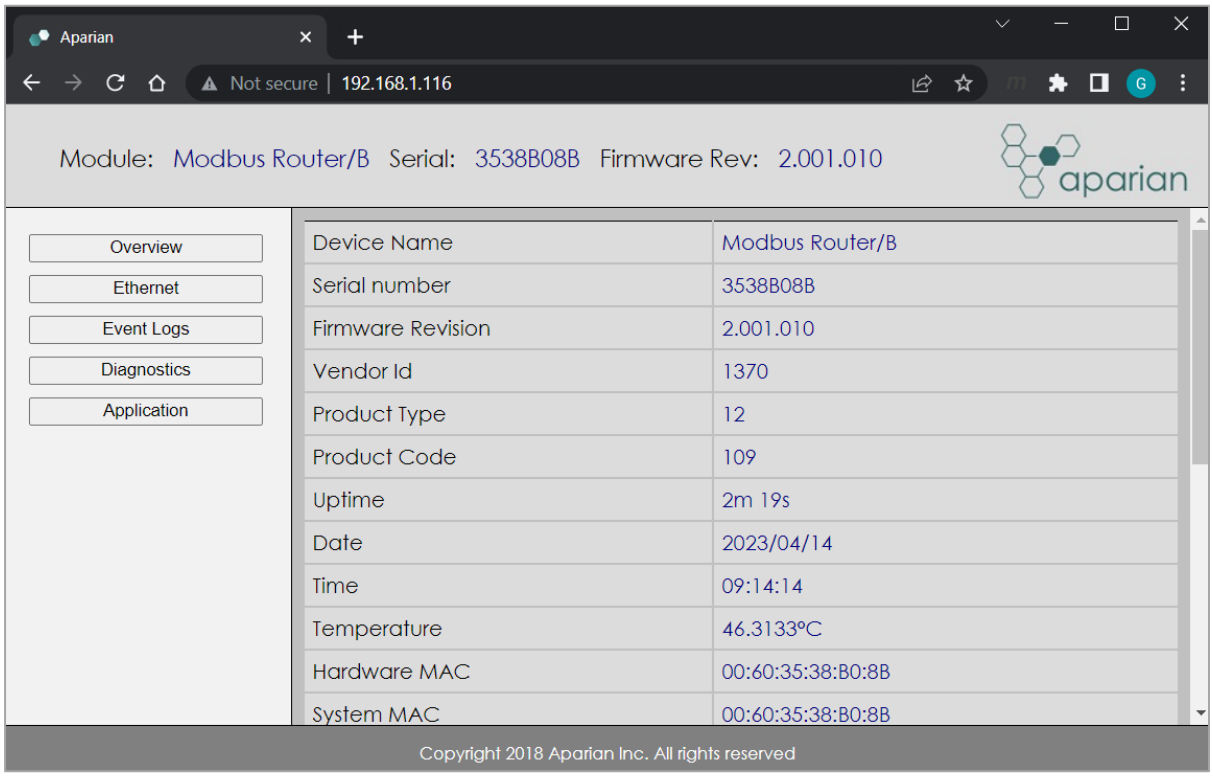


Figure 9.27 - Web interface.

9.8. 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*** option.

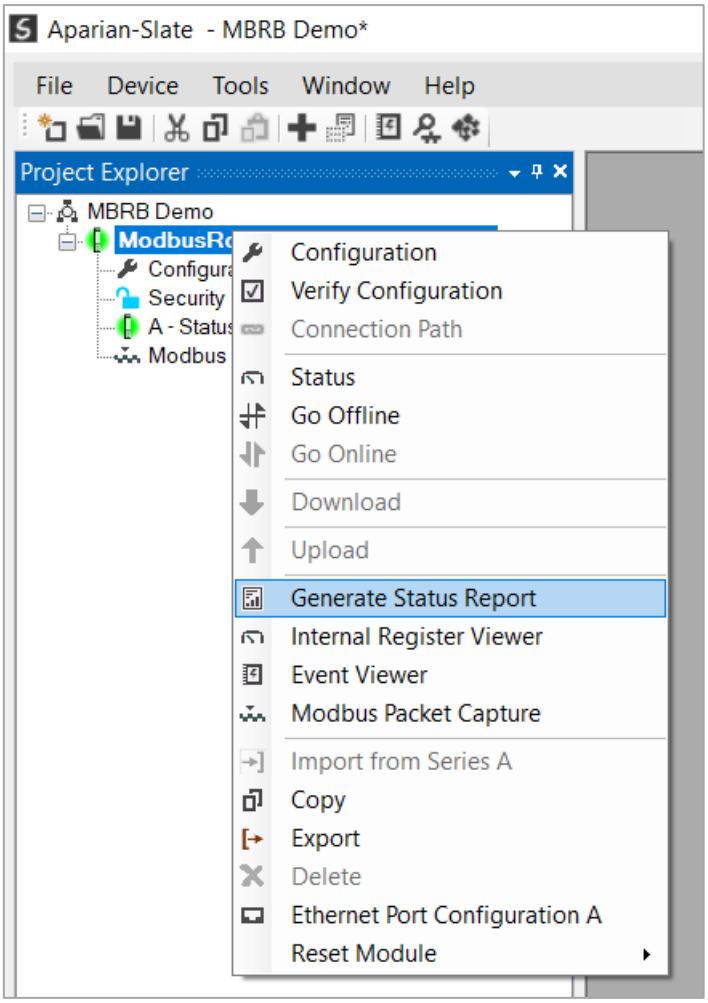


Figure 9.28 – Module Status Report.

10.TECHNICAL SPECIFICATIONS

10.1. DIMENSIONS

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

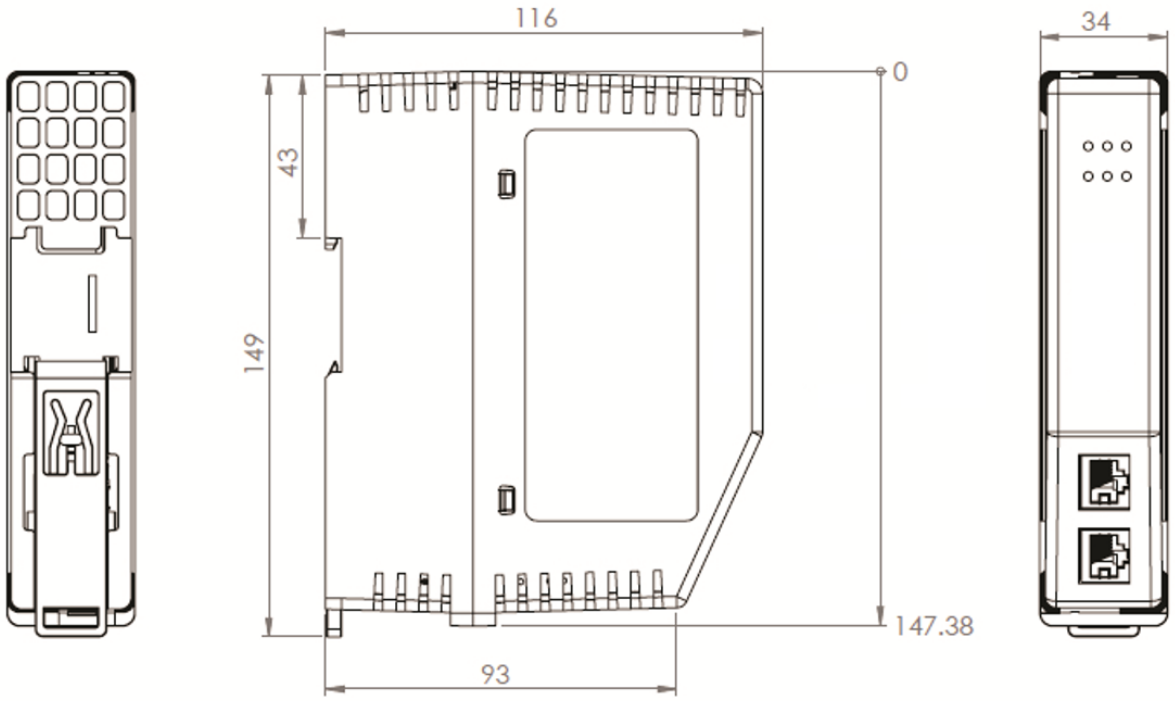


Figure 10.1 – Modbus Router enclosure dimensions.

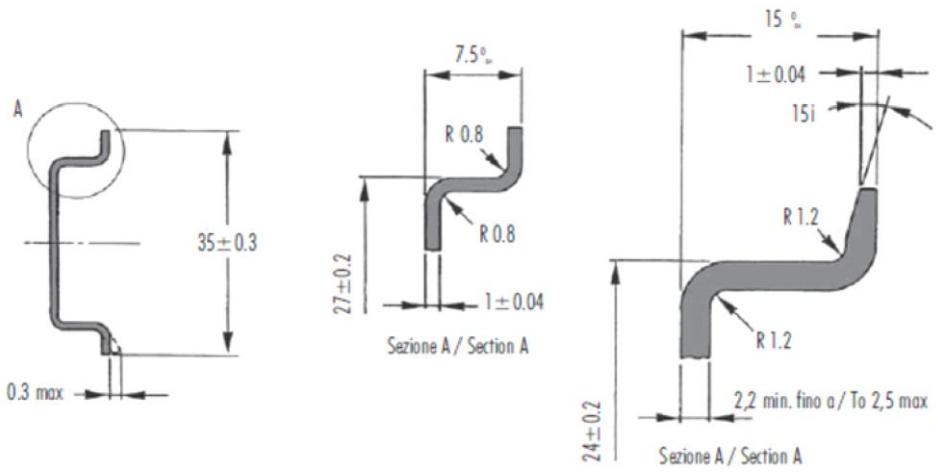


Figure 10.2 - Required DIN dimensions.

10.2. ELECTRICAL

Specification	Rating
Power requirements	Input: 10 – 32V DC, (121 mA @ 24 VDC)
Voltage Fluctuations	Voltage fluctuations < ±10% Transient Over-voltages up to the levels of OVERVOLTAGE CATEGORY I
Power consumption	3.1 W (Including full load on USB of 200mA) 300 mA maximum
USB Power	5V, maximum load of 200 mA (1W).
Connector	3-way terminal
Conductors	24 – 18 AWG
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 10.1 - Electrical specification.

10.3. ENVIRONMENTAL

Specification	Rating
Enclosure rating	IP20, NEMA/UL Open Type Indoor use only
Temperature	-20 – 70 °C
Relative Humidity	5% to 90% - No condensation
Pollution Degree	2
Altitude	< 2000 m

Table 10.2 - Environmental specification.

10.4. ETHERNET

Specification	Rating
Connector	RJ45
Conductors	CAT5 STP/UTP
ARP connections	Max 200
TCP connections	Max 200
CIP connections	Max 20
Communication rate	10/100Mbps
Duplex mode	Full/Half
Auto-MDIX support	Yes
Controller Support	ControlLogix, CompactLogix, MicroLogix, SLC
Embedded switch	Yes, 2 x Ethernet ports
Device Level Ring (DLR)	Supported
Network Time Protocol (NTP)	Supported

Table 10.3 - Ethernet specification.

10.5. SERIAL PORT (RS232)

Specification	Rating
RS232 Connector	9-way terminal (shared with RS485)
RS232 Conductor	24 – 18 AWG
Electrical Isolation	1000 Vdc
BAUD	1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200
Parity	None, Even, Odd
Data bits	8
Stop bits	1

Table 10.4 – RS232 Serial Port specification.

10.6. SERIAL PORT (RS485)

Specification	Rating
RS485 Connector	9-way terminal (shared with RS485)
RS485 Conductor	24 – 18 AWG
Electrical Isolation	1500 Vrms for 1 minute.
BAUD	1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200
Parity	None, Even, Odd
Data bits	8
Stop bits	1

Table 10.5 – RS485 Serial Port specification.

10.7. USB PORT

Specification	Rating
USB supported	USB2.0
Module USB Connector	Type-A
Recommended USB Cable	Type-A (male) to Type-B (male)

Table 10.6 – USB Port specification.

10.8. MODBUS – REACTIVE / SCHEDULED MODE

Specification	Rating
Reactive Tag mode	Max 100 mapping items
Scheduled Tag mode	Max 100 mapping items
Application Functions Supported	Read Coil, Read Discrete Input, Read Holding Register, Read Input Register, Write Coil, Write Register
Maximum Logix Controller support	3 (when using Ethernet) 1 (when using USB)
Protocols	Modbus RTU (RS232 or RS485), Modbus TCP

Table 10.7 – Modbus specification.

10.9. MODBUS CLIENT – ENHANCED MODE

Specification	Rating
Modes Supported	Modbus TCP, Modbus RTU232, Modbus RTU485
Modbus RTU485 Termination	125 Ω - Software Enabled
Max. Modbus Server Devices	100
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 Reformatting Supported	BB AA BB AA DD CC CC DD AA BB DD CC BB AA

Table 10.8 – Modbus Client specification.

10.10. MODBUS SERVER – ENHANCED MODE

Specification	Rating
Modes Supported	Modbus TCP, Modbus RTU232, Modbus RTU485 (simultaneous)
Modbus RTU485 Termination	Software set
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 10.9 – Modbus Server specification.

10.1. ETHERNET/IP TARGET

Specification	Rating
Class 1 Cyclic connection count	5
Logix Direct-to-Tag Supported	Yes

Table 10.10 – EtherNet/IP Target specification.

10.2. ETHERNET/IP ORIGINATOR

Specification	Rating
Class 1 Cyclic Connections Supported	Yes
Class 3 / UCMM Connections Supported	Yes
Class 1 Connection Count	10
Class 3 / UCMM Target Device Count	10
Class 3 / UCMM Mapping Count	50

Table 10.11 – EtherNet/IP Originator specification.

10.3. CERTIFICATIONS

Certification	Mark
CE Mark	CE
RoHS2 Compliant	RoHS2
UL Mark File: E494895	c UL us LISTED CLASS 1, DIV 2, GROUPS A, B, C, D
ODVA Conformance	EtherNet/IP™

UKCA	

Table 10.12 – Certifications.

INDEX

A

assembly instance, 78

C

class 1, 78
CompactLogix, 8, 17
Connection path, 112
Contact Us, 19
ControlLogix, 8, 16, 17

D

DC power, 20
DHCP, 21, 26, 27, 28, 29
dimensions, 198
DIN rail, 22, 198
DIP, 21

E

Ethernet Bridge, 77
Ethernet connector, 24
Ethernet TCP, 8, 38, 149
Ethernet UDP, 11, 38, 43, 47, 149
Ethernet/IP, 41, 45, 52, 57, 93
EtherNet/IP, 8, 17, 29, 41, 45, 49, 52, 56, 60, 93
EtherNet/IP Devices, 92
EtherNet/IP Devices configuration, 92, 93
EtherNet/IP MAP, 94
EtherNet/IP Map configuration, 94, 111

F

File Number, 50, 52, 53, 55, 57, 59, 61
firmware upgrade, 33

G

General parameters, 33

I

input assembly, 179
input voltage, 23

L

LED, 172, 173
Logix controller, 40, 41, 42, 45, 46, 49, 51, 52, 53, 56, 57, 60, 61, 78, 145, 176, 181
Logix platform, 8, 16, 17
Logix tag, 35, 41, 42, 46, 50, 52, 53, 55, 57, 59, 61, 148
Logix Tag Map, 55, 59

M

Mapping Mode, 34
MODBUS, 1, 8, 11, 16, 17, 19, 20, 21, 24, 31, 32, 33, 34, 35, 37, 38, 39, 40, 41, 42, 44, 46, 48, 50, 51, 52, 53, 54, 55, 57, 58, 59, 61, 63, 77, 78, 80, 107, 111, 112, 143, 145, 147, 148, 149, 150, 151, 161, 172, 173, 177, 179, 180, 181, 182, 192, 193, 194, 196, 198, 201
MODBUS general configuration, 33
Modbus Router, 8, 11, 16, 17, 19, 20, 21, 24, 31, 32, 35, 38, 40, 51, 77, 78, 80, 112, 145, 147, 148, 149, 172, 173, 181, 182, 194, 196, 198
Modbus Router general configuration, 65, 68, 70, 81, 96
Modbus Router Internal Data Space, 101

O

output assembly, 177

P

Port parameters, 40
Protocol, 26, 38

R

Reactive Tag, 17, 35, 50, 51, 52, 53, 55, 57, 59, 61, 145, 176, 180, 181, 201
requested packet interval (RPI), 78
Rockwell Automation, 30
RS232, 11, 20, 23, 24, 39, 40
RS232/RS485, 23
RSLinx, 30
RSLogix 5000, 77, 79, 81, 196

S

Safe Mode, 21
Schedule Tag, 41
Scheduled Tag, 16, 35, 40, 42, 46, 143, 176, 180, 181, 201
Serial, 11, 149, 200, 201
Slate, 11, 19, 26, 29, 31, 32, 33, 40, 44, 48, 54, 59, 173, 188, 194, 196
SLATE Configuration Utility, 120, 123, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 138, 139, 140, 141
statistics, 173, 182, 186, 190, 192
Support email, 19

T

Target Browser, 29, 41, 45, 49, 52, 56, 60, 93, 112
Target Tag, 44, 48, 54, 59

U

UDTs, 19, 79, 80, 81, 147
Unscheduled, 35, 63, 80, 147, 148, 149, 150, 176, 182
User Defined Types (UDTs), 79

W

web server, 173, 196