Modbus IPBlock

User Manual

A-MBIPB

Document No. D170-010
07/2026
Revision 0.5

CONTENTS

1. Preface 7
1.1. Introduction to the Modbus IPBlock 7
1.2. Features 10
1.3. Architecture 11
1.4. Additional Information 14
1.5. Support 14
2. Installation 15
2.1. Module Layout 15
2.2. Module Mounting 16
2.3. Power 16
2.4. RS485 Ports 17
2.5. RS485 Termination 17
2.6. Ethernet Ports 18
2.7. North American Hazardous Location Approval 19
3. Setup 20
3.1. Install Configuration Software 20
3.2. Network Parameters 20
3.3. Creating a New Project 24
3.4. General Parameters 27
3.5. Modbus Parameters 30
3.6. Enhanced Mode 32
3.6.1. Modbus Mode 32
3.6.1.1. Server 32
3.6.1.1.1. Multi-Server 32
3.6.1.1.2. Internal Data Space Mapping 34
3.6.1.2. Client 36
3.6.1.2.1. Modbus Auxiliary Map 36
3.6.1.2.2. Internal Data Space Mapping 38
3.6.2. EtherNet/IP Mode 40
3.6.2.1. Target 40
3.6.2.1.1. Studio 5000 Configuration (Version 21+) 41

3.6.2.1.2. RSLogix 5000 Configuration (Pre-Version 21) 45
3.6.2.1.3. Importing UDTs and Mapping Routines 49
3.6.2.1.4. Internal Data Space Mapping 52
3.6.2.2. Originator 54
3.6.2.2.1. EtherNet/IP Class 1 Device Connections 54
3.6.2.2.2. EtherNet/IP Explicit Message Device Connections 63
3.6.2.2.3. Internal Data Space Mapping 67
3.7. Bridge Mode 69
3.8. Concentrator Mode 71
3.9. Internal Data Space Map 72
3.9.1. Copy From 73
3.9.1.1. Internal 74
3.9.1.2. EIP Target 74
3.9.1.3. EIP Originator 75
3.9.1.4. MB Register 76
3.9.1.5. MB Aux 76
3.9.1.6. System 77
3.9.2. Copy To 78
3.9.2.1. Internal 78
3.9.2.2. EIP Target 79
3.9.2.3. EIP Originator 79
3.9.2.4. MB Register 80
3.10. Advanced 81
3.11. Module Download 81
4. Device Firmware Update 84
5. Security 87
5.1. Securing Process 88
5.1.1. Configuration 88
5.1.1.1. General Configuration 89
5.1.1.2. Password Rule Configuration 91
5.1.1.3. Custom Users Configuration 93
5.1.1.4. Services Configuration 94
5.1.1.5. IP Access Control List Configuration 95

5.1.1.6. MAC Access Control List Configuration 96
5.1.2. Download Security Configuration 98
5.1.3. Initialize User Authentication 99
5.2. Operation 102
5.2.1. Initial User Login 102
5.2.2. Role Specific Operations 104
5.2.2.1. Security Status 104
5.2.3. Change Other Password 109
5.2.4. Clear All Security 110
5.3. Event Log 111
6. Operation 112
6.1. Enhanced Mode 112
6.1.1. EIP Target – Modbus Server 112
6.1.2. EIP Originator – Modbus Server 113
6.1.3. EIP Target – Modbus Client 115
6.1.4. EIP Originator – Modbus Client 116
6.2. Bridge Mode 118
6.3. Concentrator Mode 119
6.4. Internal Map Data Formats 120
6.4.1. System Status 120
6.4.2. MB Aux Item Status 121
6.4.3. EtherNet/IP IO Device Status 121
7. Diagnostics 123
7.1. LEDs 123
7.2. Module Status Monitoring in Slate 124
7.2.1. General 126
7.2.2. Modbus Statistics 128
7.2.3. Serial Statistics 129
7.2.4. Logix Statistics 130
7.2.5. Aux. Items 131
7.2.6. CIP Statistics 132
7.2.7. EtherNet/IP Explicit 132
7.2.8. EtherNet/IP Map 133

7.2.9. EtherNet/IP Originator 134
7.2.10. Ethernet Clients 135
7.2.11. TCP / ARP 135
7.3. Target Device Status Monitoring In Slate 136
7.3.1. EtherNet/IP 136
7.3.1.1. General 136
7.3.1.2. Input Data 138
7.3.1.3. Output Data 138
7.4. Modbus Diagnostics 139
7.5. Modbus Packet Capture 140
7.6. Module Event Log 142
7.7. Module Status Report 143
7.8. Internal Register Viewer 144
8. Technical Specifications 146
8.1. Dimensions 146
8.2. Electrical 146
8.3. Environmental 147
8.4. Ethernet 147
8.5. Serial Port (RS485) 148
8.6. Modbus Client – Enhanced Mode 148
8.7. Modbus Server – Enhanced Mode 149
8.1. EtherNet/IP Target 149
8.2. EtherNet/IP Originator 149
8.3. Certifications 150
9. Index 151

Revision History

Revision	Date	Comment
0.4	16 July 2026	Initial document

1. PREFACE

1.1. INTRODUCTION TO THE MODBUS IPBLOCK

This manual describes the installation, operation, and diagnostics of the Aparian Modbus IPBlock (A-MBIPB). The Modbus IPBlock provides intelligent data routing between EtherNet/IP and Modbus (Modbus-RTU485 and Ethernet Modbus-TCP). The Modbus IPBlock allows the user to integrate Modbus devices into a Rockwell Automation Logix platform (ControlLogix or CompactLogix).

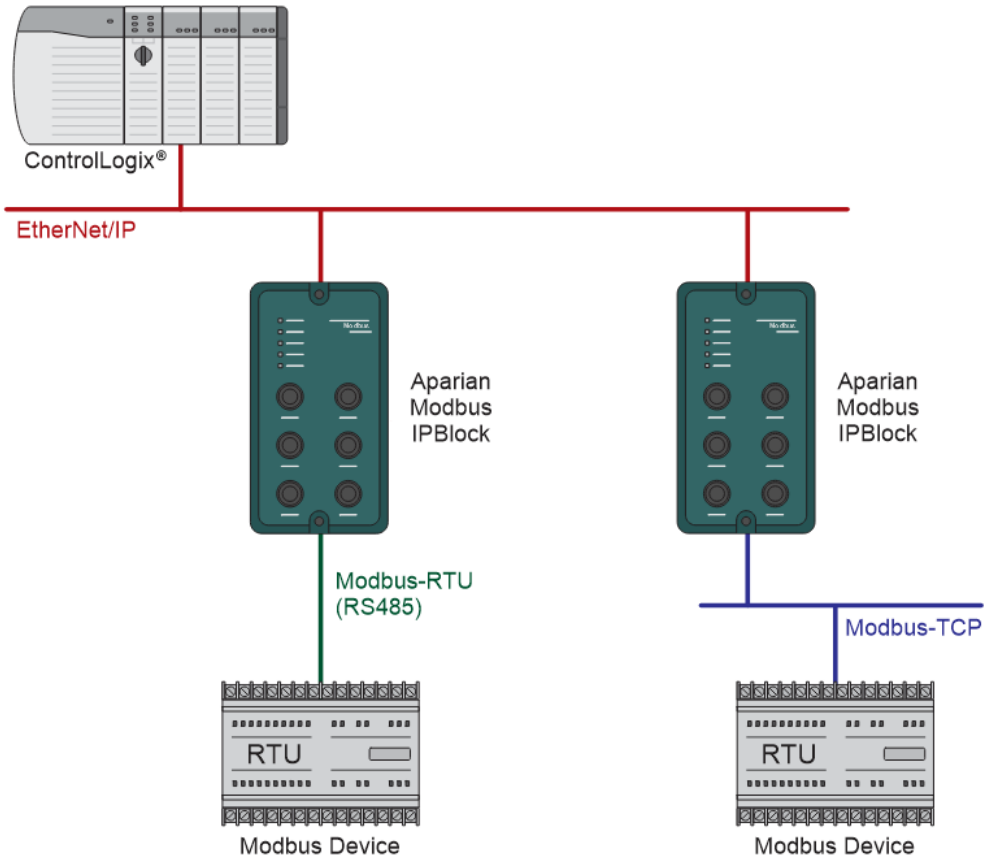


Figure 1.1 – Typical Setup.

The module can be configured in one of three modes:

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

In Enhanced Mode, the A-MBIPB 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 RTU485, or both.

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 RTU485, or both.

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 A-MBIPB 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 A-MBIPB 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 A-MBIPB to "own" the EtherNet/IP target device and exchange Modbus data using the EtherNet/IP device's input and output assemblies.

Bridge
The A-MBIPB can operate as a transparent link for Modbus between different media (TCP and RTU485). This allows a Modbus client on Modbus TCP to communicate with Modbus devices on RS485 and vice versa.

Concentrator
In Concentrator Mode, the A-MBIPB will collect data from multiple Modbus devices and provide the consolidated data to one or more Modbus Clients. The A-MBIPB will be a Modbus Server on one port, and Modbus Clients on the other ports. This will allow, for example, 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 A-MBIPB.
NOTE: The module will still support an EtherNet/IP connection to a Logix controller to allow monitoring the Modbus data.

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

The Modbus IPBlock uses isolated RS485 for Modbus serial communication providing better noise immunity.

The A-MBIPB supports security services allowing the user to configure various levels of module security.

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

1.2. FEATURES

- IP67 Enclosure with M12 connectors
- Can operate as a Modbus Client or Modbus Server.
- Modbus Client mode can communicate with up to 100 Modbus Servers.
- Modbus Server mode supports full Modbus data range.
- Modbus Server mode supports emulation of 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).
- 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 RTU485).
- Slate provides a Modbus packet capture utility for better diagnosis.
- Three Ethernet ports, two of which support Linear, Device-Level-Ring (DLR) or Split Dual-IP Modes.
- Network Time Protocol (NTP) supported for external time synchronization.

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 EtherNet/IP.

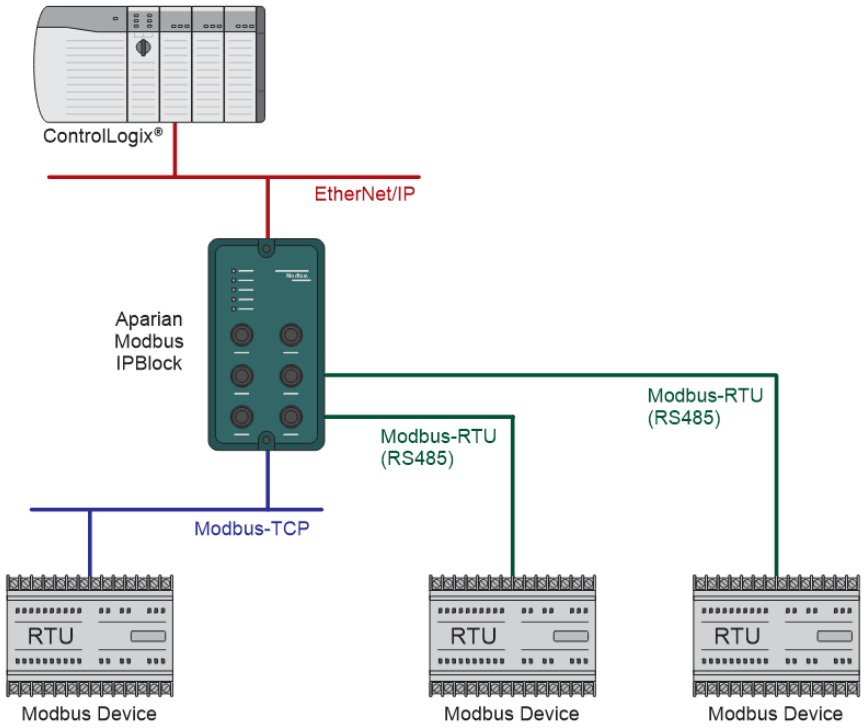


Figure 1.2 - Enhanced mode as a Modbus Client and EtherNet/IP Target.

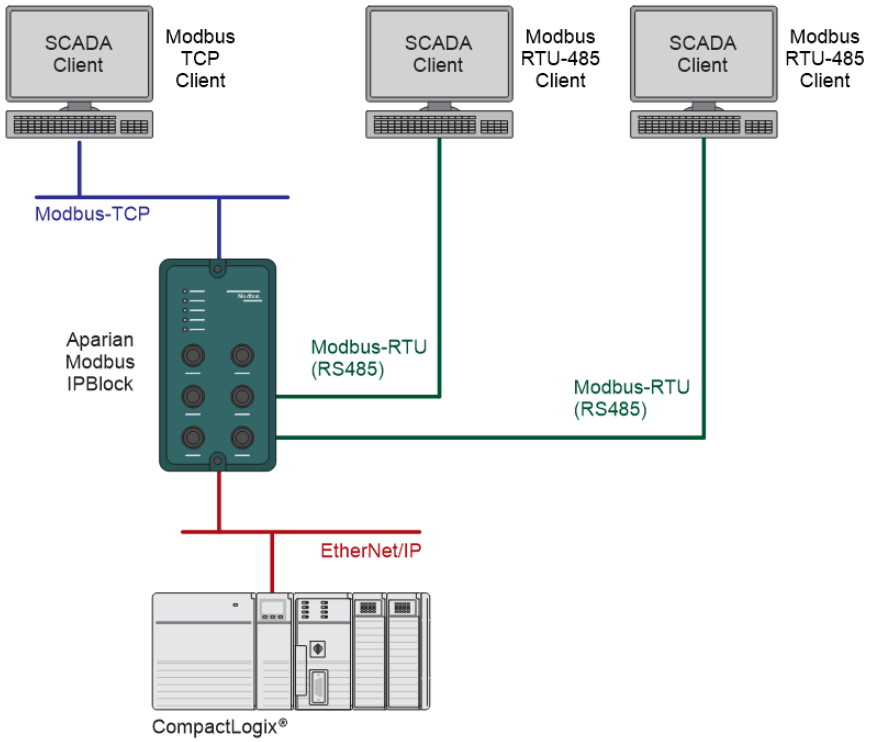


Figure 1.3 - Enhanced mode as a Modbus Server and EtherNet/IP Target.

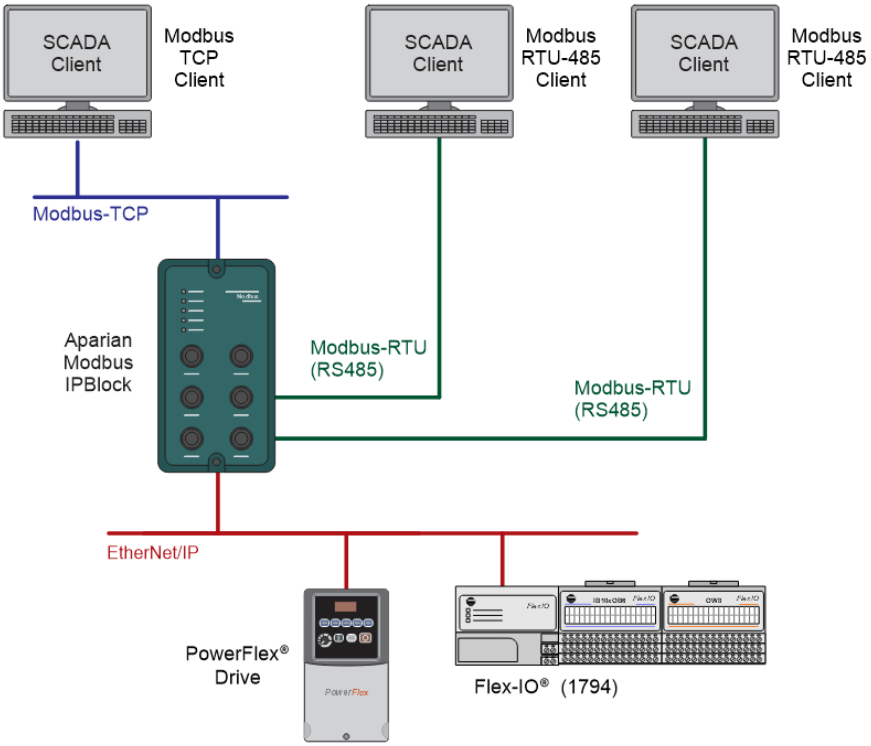


Figure 1.4 - Enhanced mode as a Modbus Server and EtherNet/IP Originator.

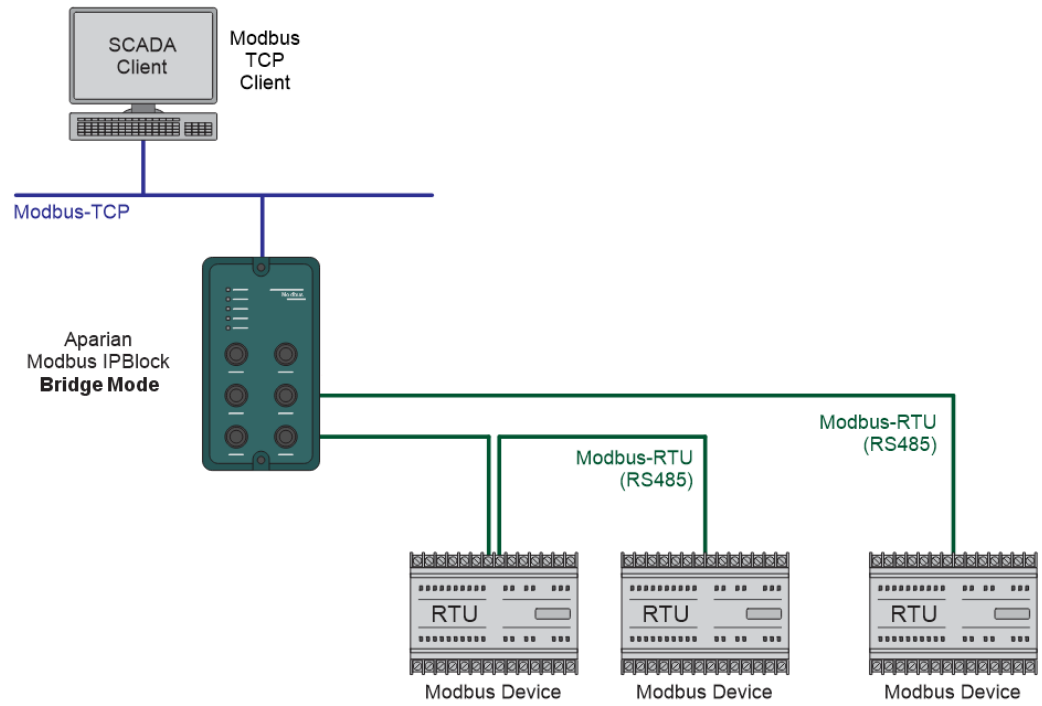


Figure 1.5 – Bridge Mode - from Modbus TCP to Modbus RTU485.

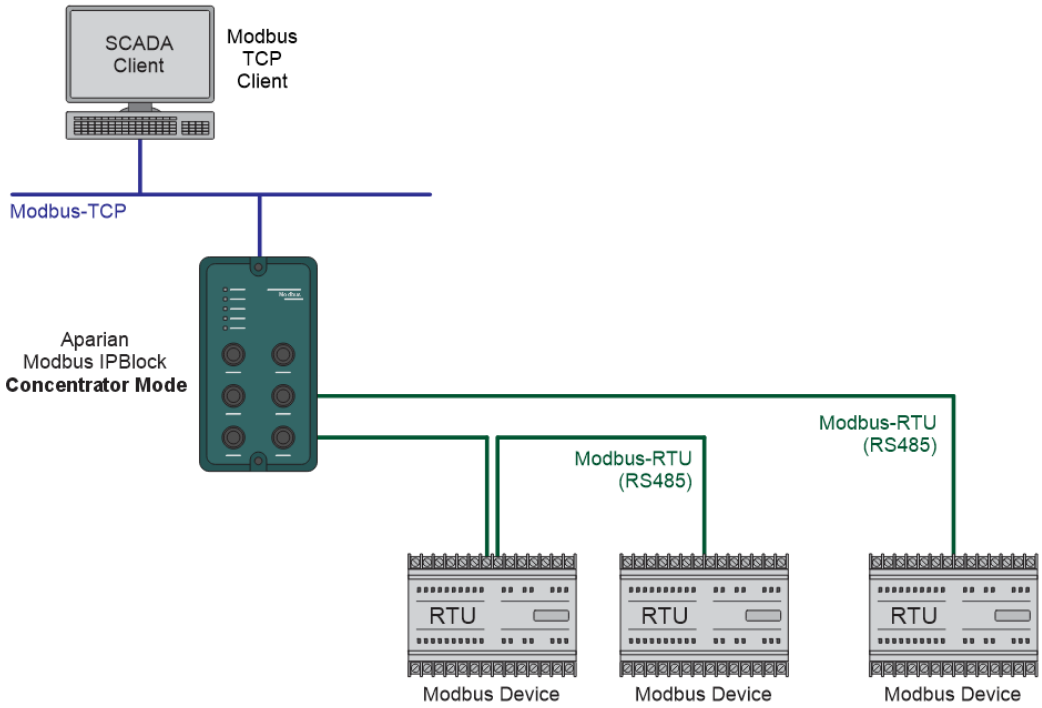


Figure 1.6 – Concentrator Mode – Modbus TCP SCADA interfacing with Modbus RTU devices.

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 IPBlock User Manual Modbus IPBlock Datasheet Example Code & UDTs	https://www.aparian.com/products/modbusIPBlockb
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 six (M12) ports on the top of the enclosure as shown in the figure below. The ports are used for Ethernet, RS485 serial, and power.

The power port uses a 4-way M12 connector which is used for the DC power supply positive and negative (or ground) voltage as well as the earth connection.

The Ethernet ports uses 4-way M12 connectors and must be wired according to industry standards which can be found in the additional information section of this document.

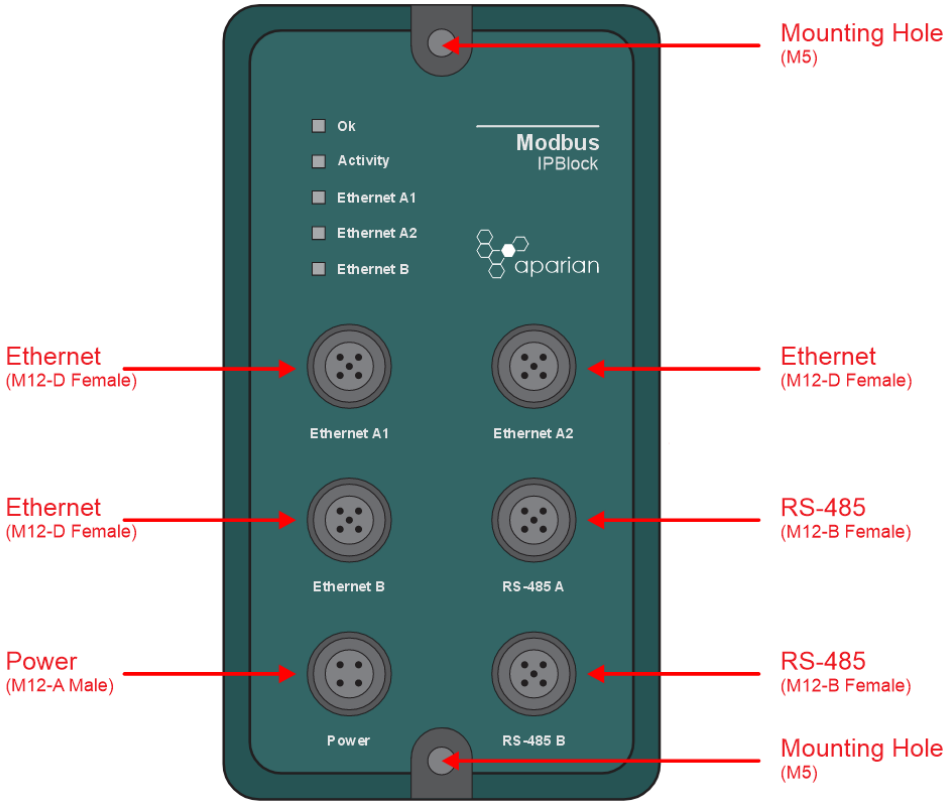


Figure 2.1 – Modbus IPBlock top view.

The module provides five diagnostic LEDs as shown in the top view figure above. These LEDs are used to provide information regarding the module system operation, the Ethernet interface status and communication status.

2.2. MODULE MOUNTING

The module can be mounted using two M5 screws as shown in the diagram below.

NOTE: Do not tighten the mounting screws excessively as this may lead to damage to the enclosure and compromise the IP67 rating.

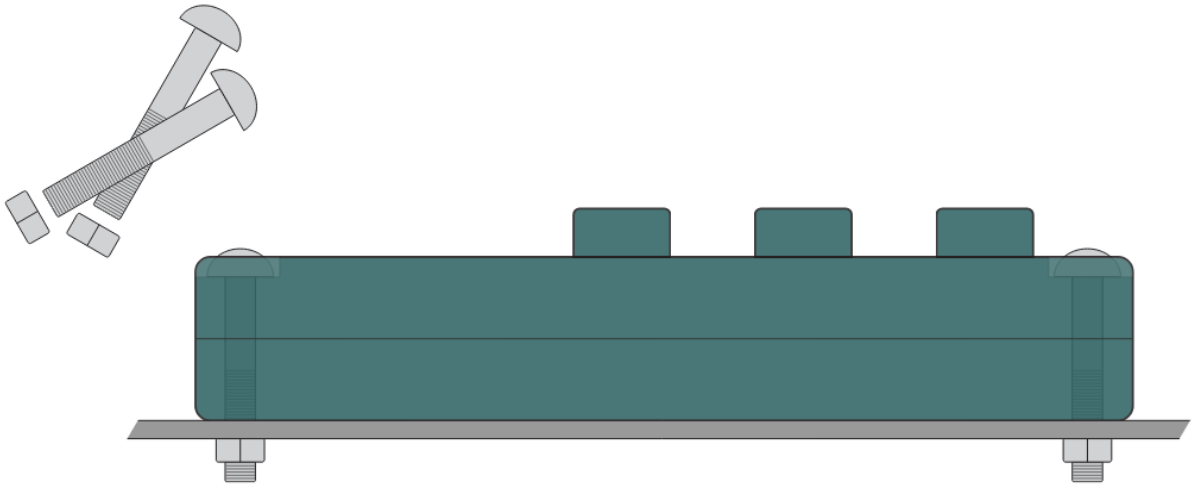


Figure 2.2 - Mounting.

2.3. POWER

A 4-pin M12 code-A female power connector is required to be connected to the module's 4-pin M12 code-A male power port. The module requires an input voltage of 10 – 32Vdc. **Refer** to the technical specifications section in this document.

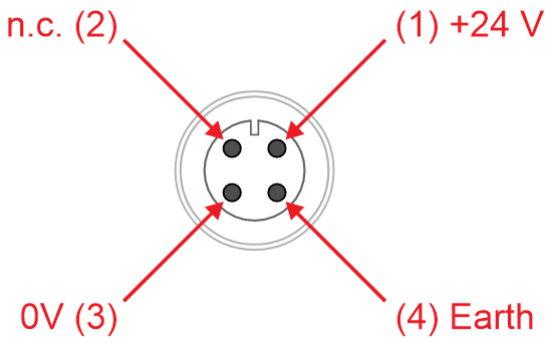


Figure 2.3 - Power port (M12 4-pin A-code male)

2.4. RS485 PORTS

The module has two RS485 M12 5-pin B-code female ports. A 5-pin (or 4-pin) M12 code-B male connector is required for each port in use.

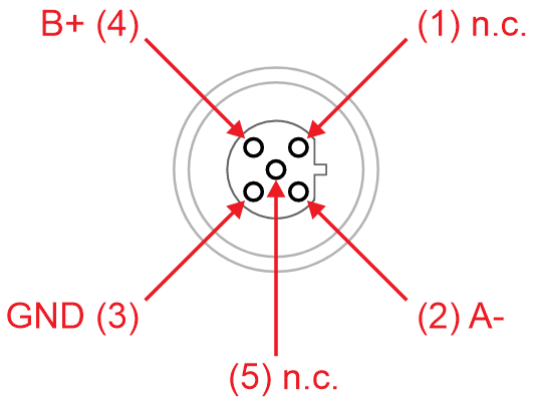


Figure 2.4 - RS485 connector (M12 5-pin B-Code Female)

NOTE: Unused RS485 ports must be sealed with the M12 covers provided to ensure the IP67 rating.

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

For long RS485 networks, it is recommended to also connect the RS485 Ground (GND) reference.

NOTE: The RS485 ports are electrically isolated from the remainder of the IPBlock module, but not from each other. The (isolated) GND connector of RS485 Port A is internally connected to that of RS485 Port B.

NOTE: In noisy environments it is recommended to use a shielded cable for the RS485 communication. The cable shield should be connected to a clean earth, and at only one point to avoid ground loops. It is recommended that the cable shield be earthed at the first remote RS485 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 module has three M12 4-pin D-coded female Ethernet ports.

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

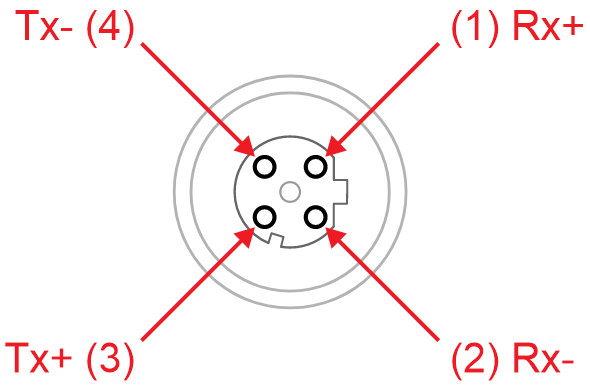


Figure 2.5 - Ethernet connectors (M12 4-pin D-Code Female)

The module has an embedded switch connecting the three Ethernet ports. The data flow between the ports depends on the Ethernet configuration.

2.7. NORTH AMERICAN HAZARDOUS LOCATION APPROVAL

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

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

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

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

WARNING - EXPLOSION HAZARD - DO NOT DISCONNECT EQUIPMENT WHILE THE CIRCUIT IS LIVE OR UNLESS THE AREA IS KNOW TO BE FREE OF IGNITABLE CONCENTRATIONS.

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

.

3. SETUP

3.1. INSTALL CONFIGURATION SOFTWARE

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



Figure 3.1 - Aparian Slate Environment.

3.2. NETWORK PARAMETERS

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

Within the Slate environment, the DHCP server can be found under the Tools menu.

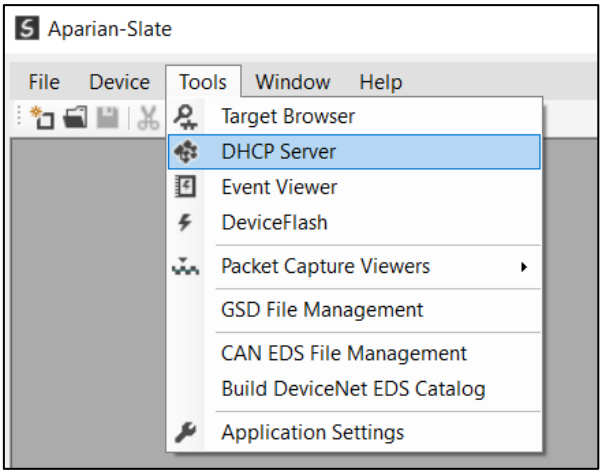


Figure 3.2 - Selecting DHCP Server.

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

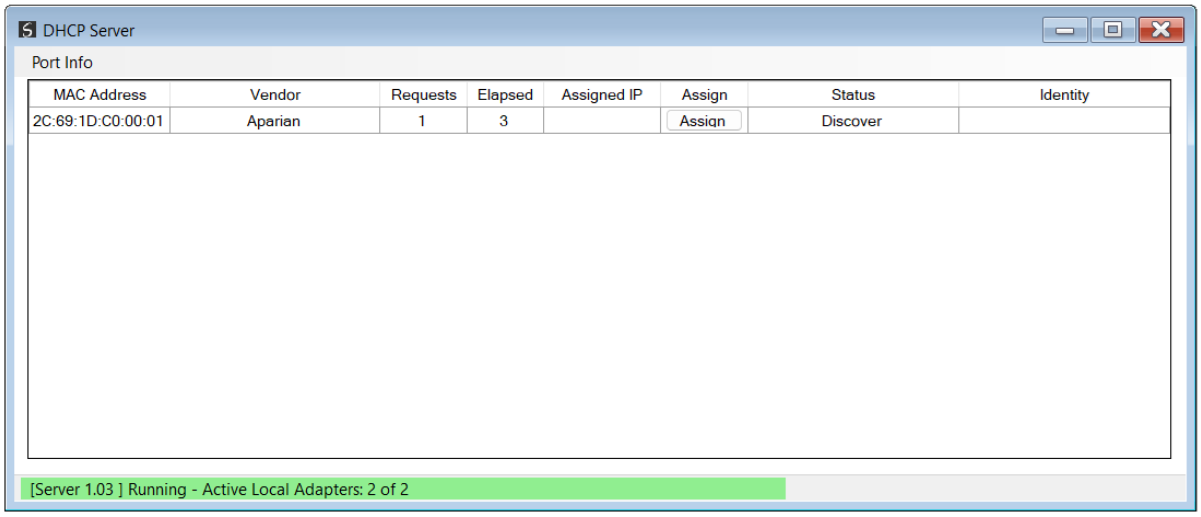


Figure 3.3 - DHCP Server.

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

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

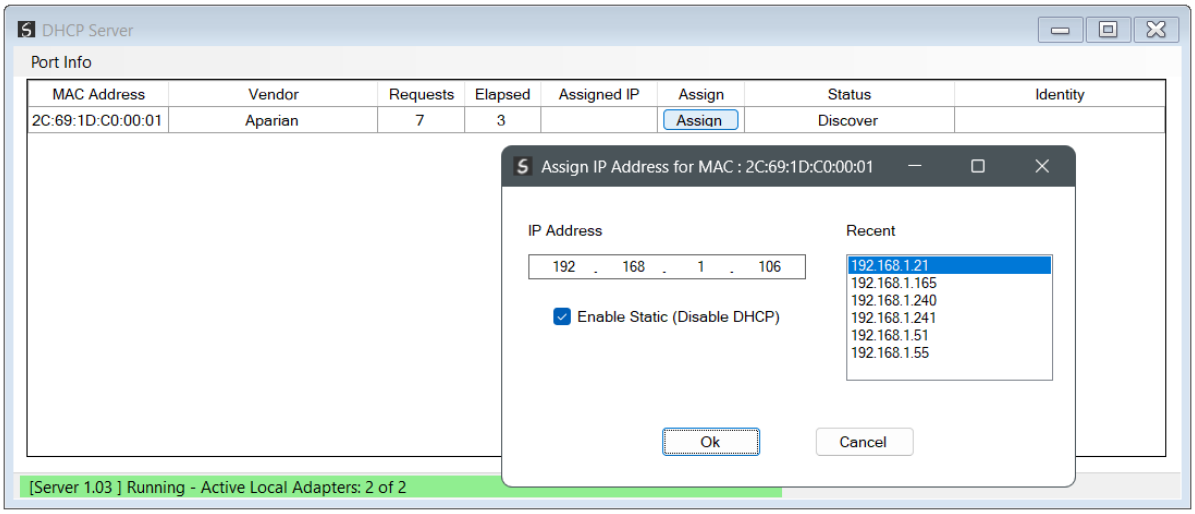


Figure 3.4 - Assigning IP Address.

The required IP address can then be either entered, or a recently used IP address can be selected by clicking on an item in the Recent List.

If the ***Enable Static*** checkbox is checked, then the IP address will be set to static after the IP assignment, thereby disabling future DHCP requests.

Once the IP address window has been accepted, the DHCP server will automatically assign the IP address to the module and then read the Identity Object Product name from the device.

The successful assignment of the IP address by the device is indicated by the green background of the associated row.

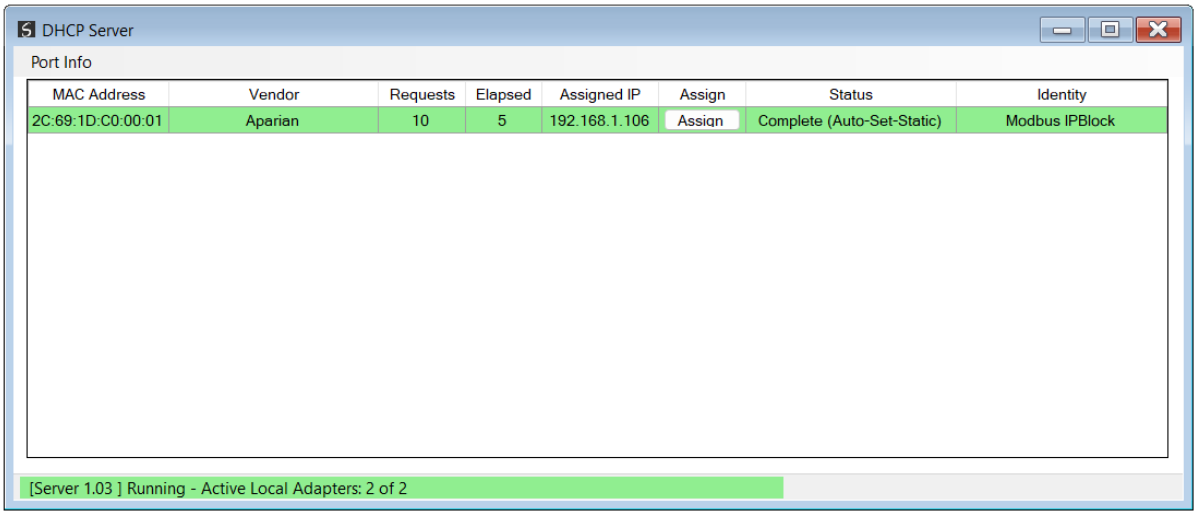


Figure 3.5 - Successful IP address assignment.

In addition to 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.

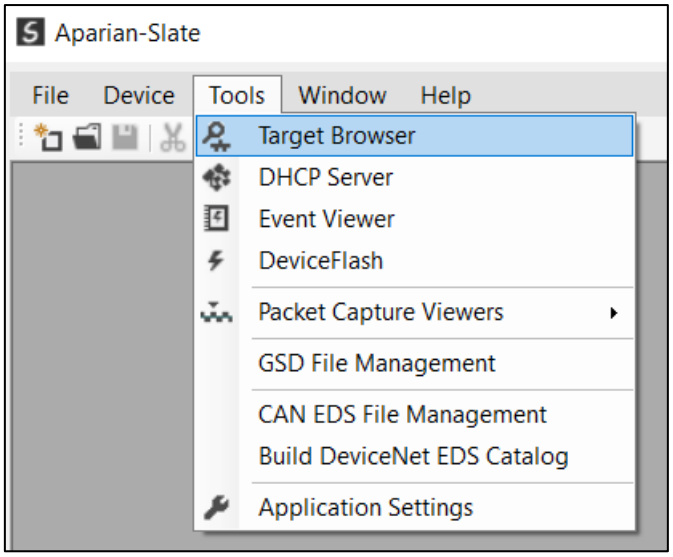


Figure 3.6 - Selecting the Target Browser.

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

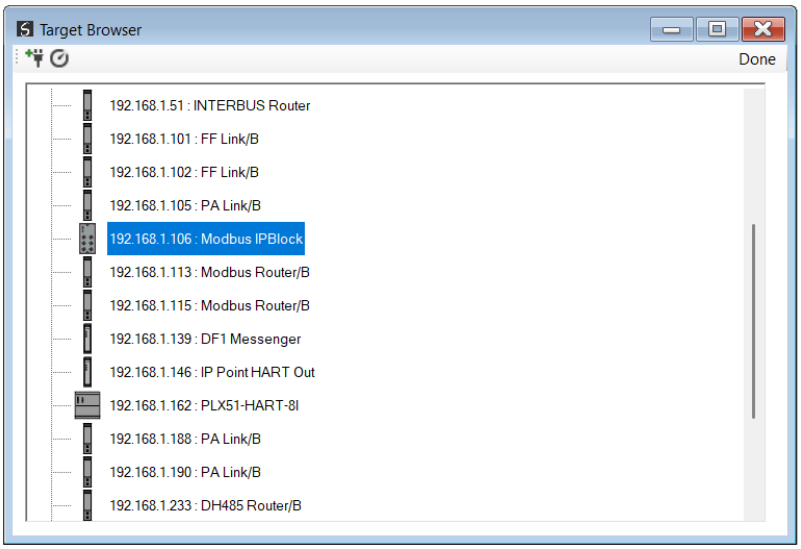


Figure 3.7 - Target Browser.

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

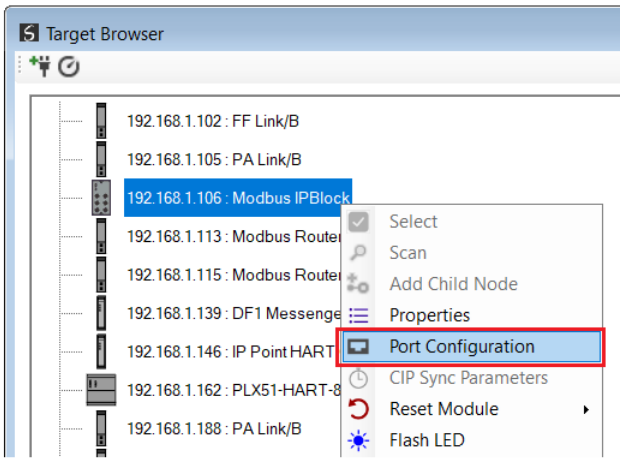


Figure 3.8 - Selecting Port Configuration.

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

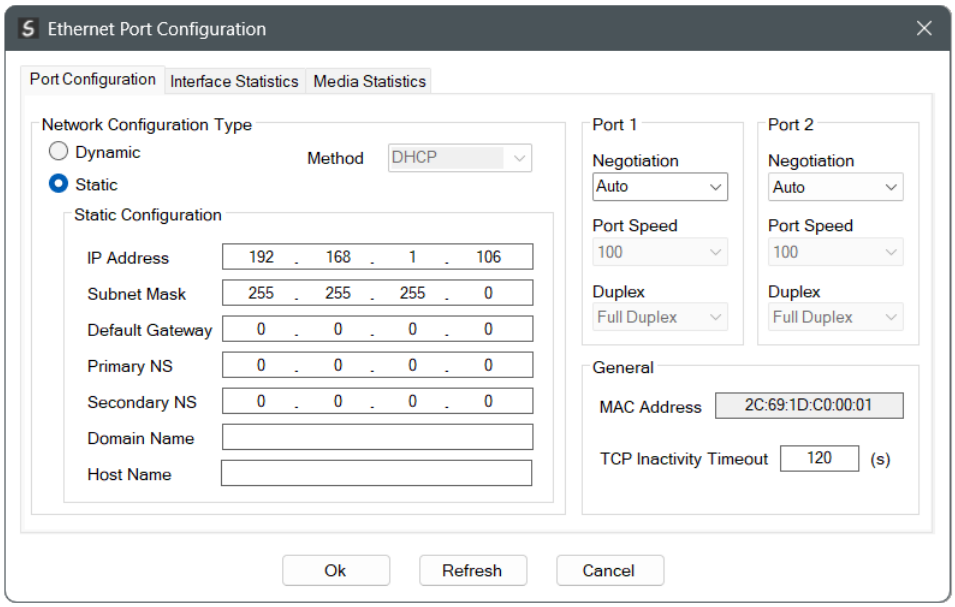


Figure 3.9 - Port Configuration.

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.

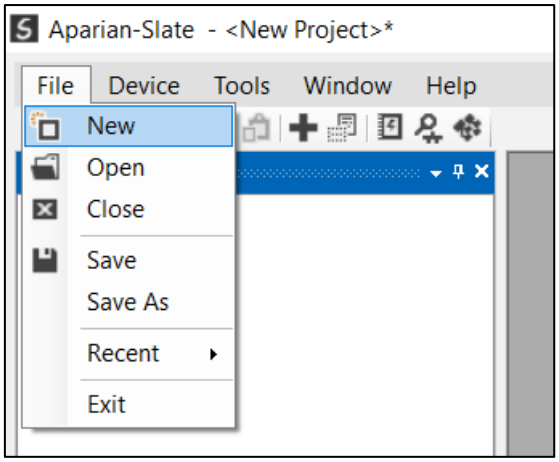


Figure 3.10 - Creating a new project.

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.

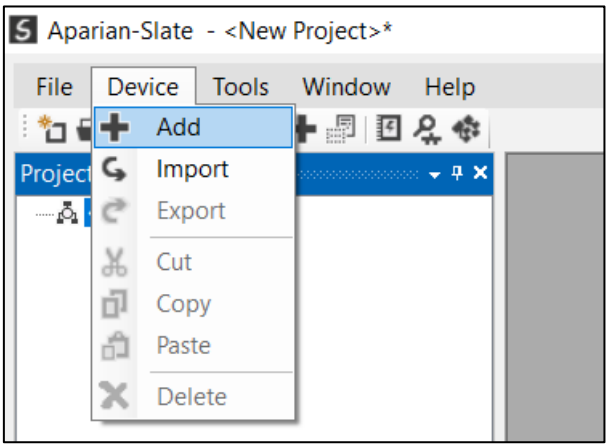


Figure 3.11 - Adding a new device.

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

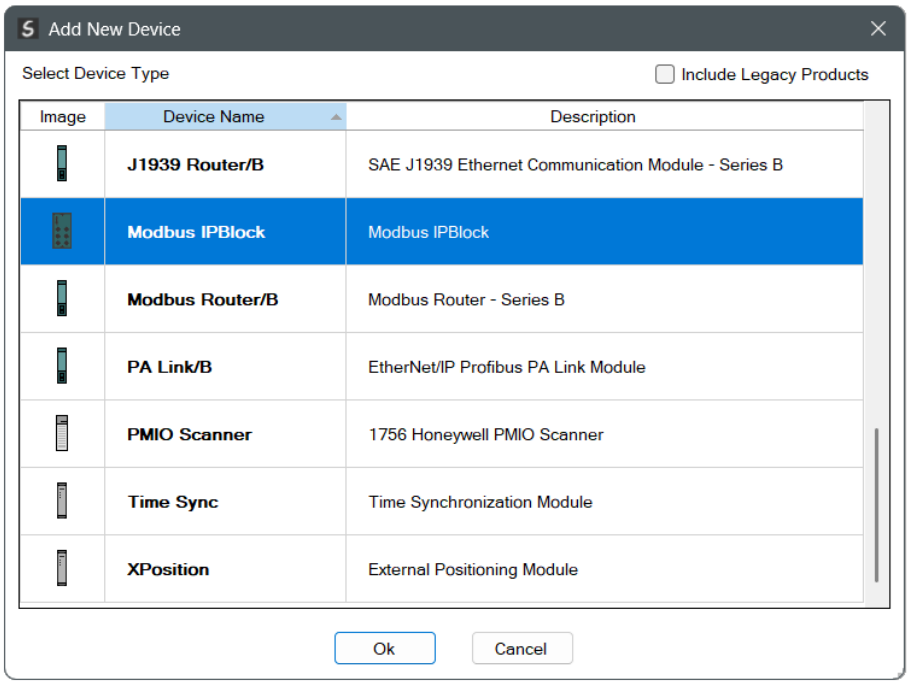


Figure 3.12 – Selecting a new Modbus IPBlock.

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***.

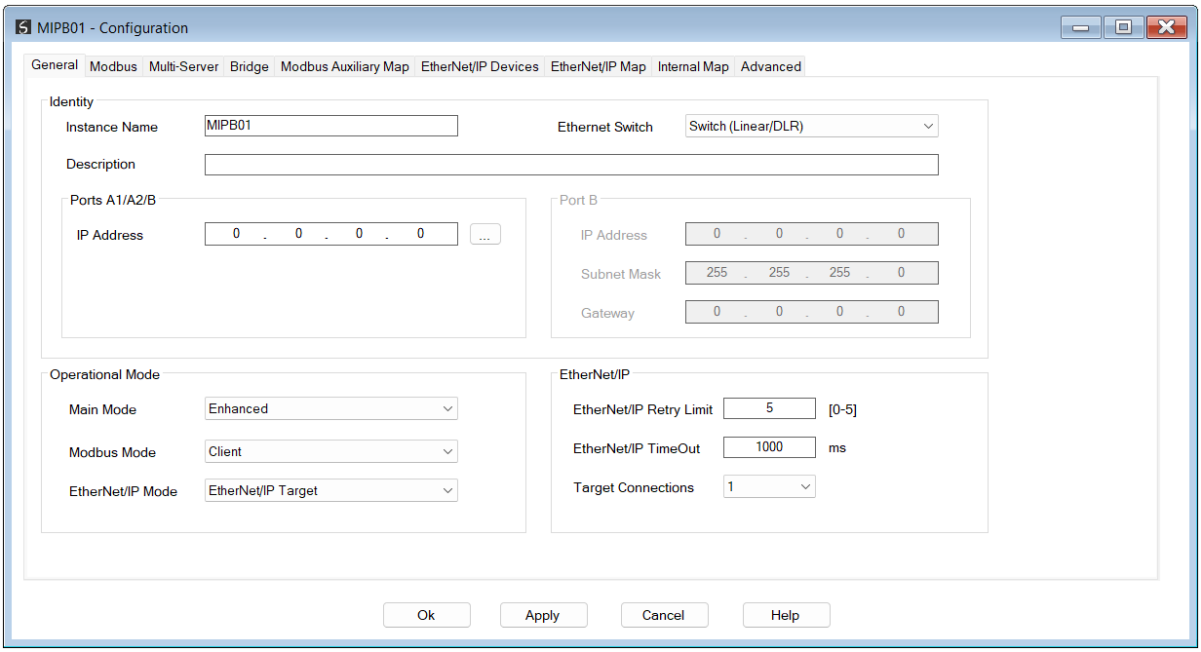


Figure 3.13 – Modbus IPBlock configuration.

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

3.4. GENERAL PARAMETERS

The A-MBIPB 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*.

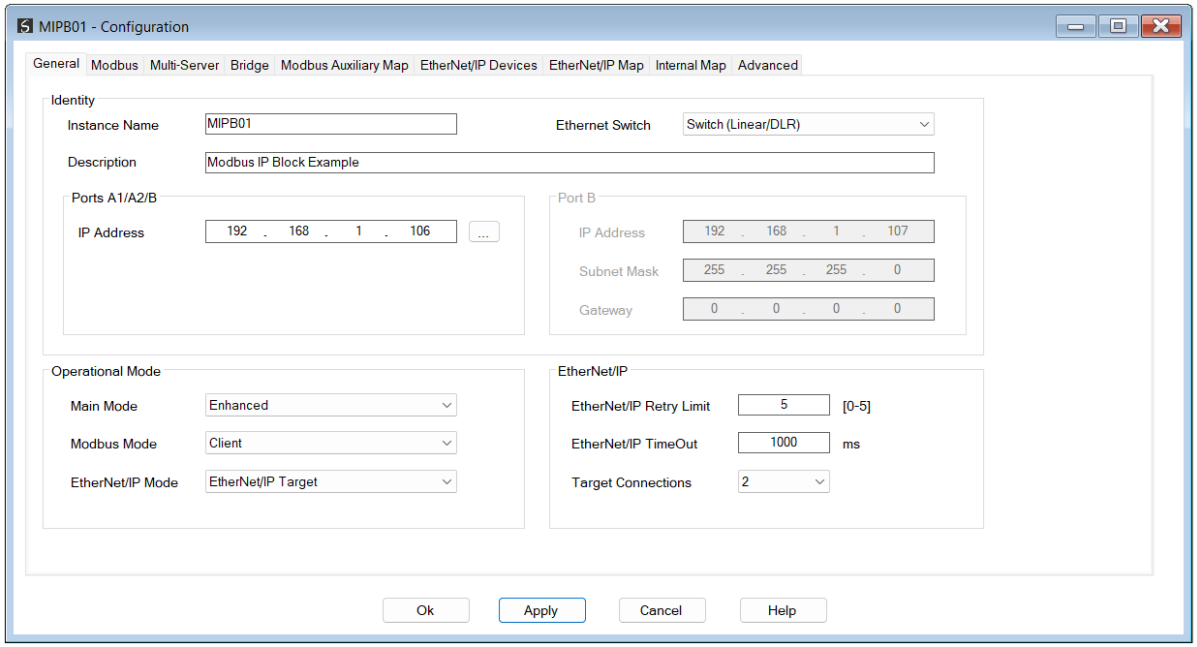


Figure 3.14 - General Configuration

The general configuration consists of the following parameters:

Parameter	Description
Instance Name	This parameter is a user defined name to identify between various Modbus IPBlocks.
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 three Ethernet ports are identical in functionality and (foreign) traffic received on one port is transmitted out on the other ports. All ports use the IP address configured using DHCP or set the Target browser. **Split (Dual-IP)** In this mode the embedded switch is disabled for Port B. Traffic on ports A1 and A2 is isolated from port B and vice versa. Port A1 & A2: 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 B: The IP address is the one contained within the configuration below (Port B). EtherNet/IP requests and connections will not be processed. Modbus TCP requests will be processed.
Port A1/A2 IP Address	IP address of the local Modbus IPBlock.
Port B IP Address	**THIS IS RELEVANT ONLY IN SPLIT (DUAL-IP) ETHERNET SWITCH MODE** The IP address used on Port B.
Port B Subnet Mask	**THIS IS RELEVANT ONLY IN SPLIT (DUAL-IP) ETHERNET SWITCH MODE** The Subnet Mask used on Port B.
Port B Gateway	**THIS IS RELEVANT ONLY IN SPLIT (DUAL-IP) ETHERNET SWITCH MODE** The Default Gateway address used on Port B. **NOTE**: If the IP address ranges for Port A1/A2 and Port B overlap, then only one Port's Gateway address can be non-zero.
Main Mode	The mapping mode will determine how the Modbus messages are routed. **Enhanced mode** In Enhanced mode, the user can decide if the A-MBIPB is operating as a Modbus Client or Server as well as if the A-MBIPB 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 A-MBIPB can operate as a transparent link for Modbus between different media (TCP and RTU485). This allows, for example, a Modbus client on Modbus TCP to communicate with multiple Modbus devices on RS485 and vice versa. **Concentrator mode**

	In Concentrator Mode, the A-MBIPB will collect data from multiple Modbus devices and provide the consolidated data to one or more Modbus Client. The A-MBIPB 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 A-MBIPB.
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 RTU485, or both. **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 RTU485, or both.
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 A-MBIPB 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 A-MBIPB 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 A-MBIPB to "own" the EtherNet/IP target device and exchange Modbus data using the EtherNet/IP device's input and output assemblies.

EtherNet/IP Retry Limit	The amount of EtherNet/IP retries the module will make once no response was received from the Logix Controller.
EtherNet/IP 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 EtherNet/IP 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 (Studio 5000)

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*.

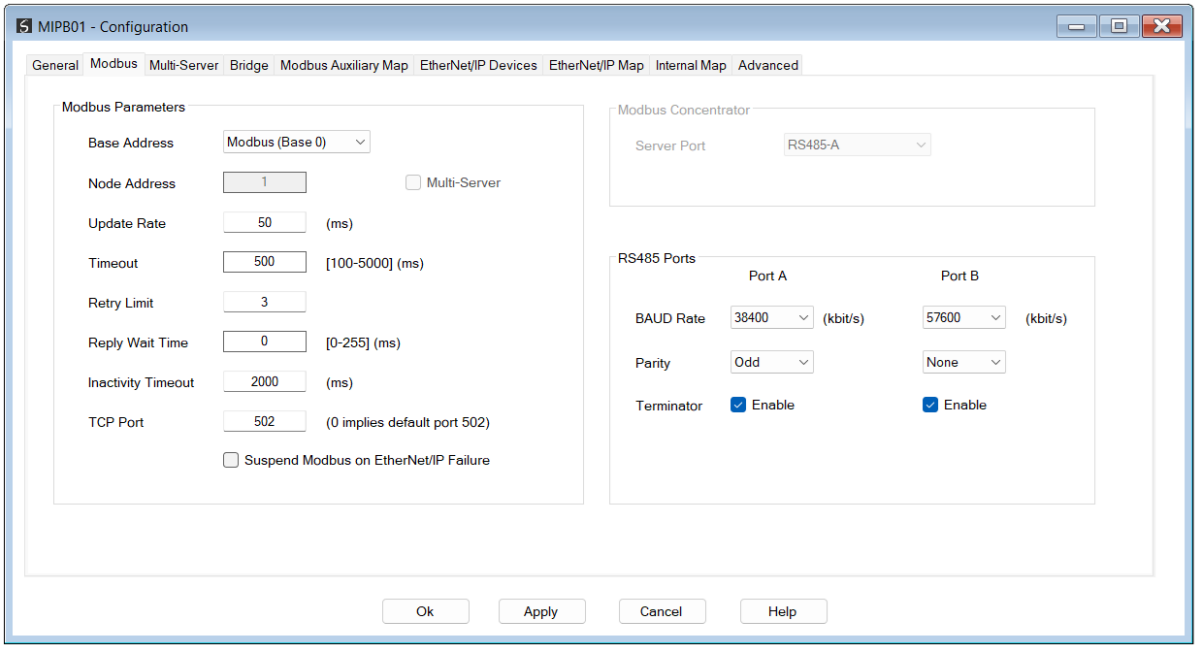


Figure 3.15 – Modbus Configuration

The Modbus configuration consists of the following parameters:

Parameter	Description
Modbus Parameters	
Base Address	The Modbus IPBlock 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 IPBlock.
Multi-Server	When this option is selected the user can select multiple nodes for the mapping items in Enhanced - Modbus Server mode. This will allow the Modbus IPBlock 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.**
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 A-MBIPB 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 Concentrator – Only Relevant for Concentrator Mode	
Server Port	The Modbus port which will be communicating with the Modbus Client. **Only valid for Concentrator Mode.**
RS485 Ports (Ports A and B)	
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
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.
Terminator	Enables the on-board 125Ω RS485 terminating resistor.

Table 3.2 – Modbus configuration parameters.

NOTE: The Modbus IPBlock RS485 communication supports 8 data bits and 1 stop bit.

3.6. ENHANCED MODE

In Enhanced Mode the A-MBIPB can be configured in any of the four combinations:

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

Table 3.3 – Enhanced Mode Operating Combinations.

3.6.1. MODBUS MODE

3.6.1.1. SERVER

The A-MBIPB can operate as a Modbus Server for Modbus TCP and RTU485. One or more remote Modbus Clients can read and write to the full Modbus Register range in the A-MBIPB. The A-MBIPB can operate as a Modbus Server for Modbus TCP and Modbus RTU485 simultaneously. The user will need to configure the relevant Modbus Parameters as described in section 3.5.

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

3.6.1.1.1. MULTI-SERVER

The A-MBIPB supports emulating multiple Modbus Server nodes when operating as a Modbus Server in Enhanced Mode.

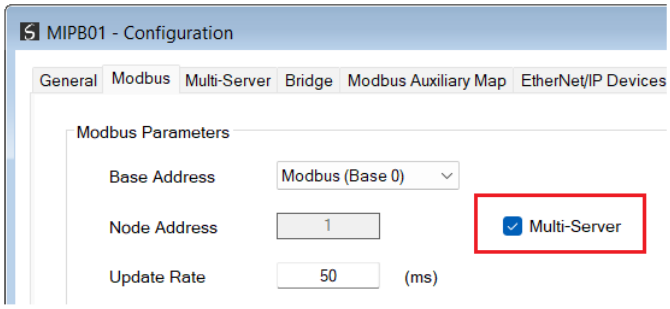


Figure 3.16 – Multi-Server selection.

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

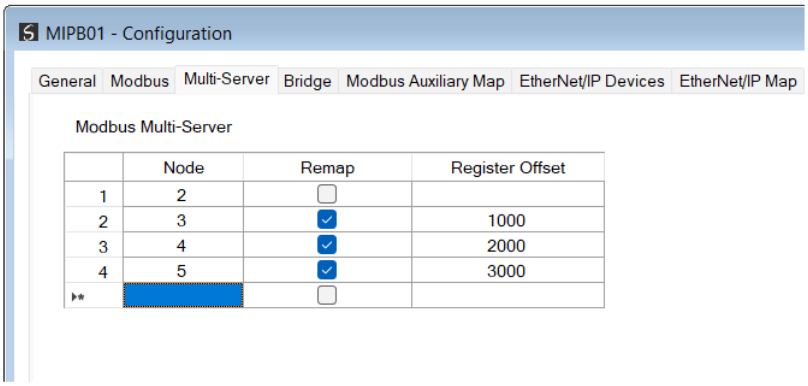


Figure 3.17 – Multi-Server Nodes and Remap.

In the above example, the A-MBIPB 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 A-MBIPB 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 A-MBIPB will respond with the data in Modbus Holding Register 2100 (request offset of 100 + Remap offset of 2000) in its internal memory.

3.6.1.1.2. 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.

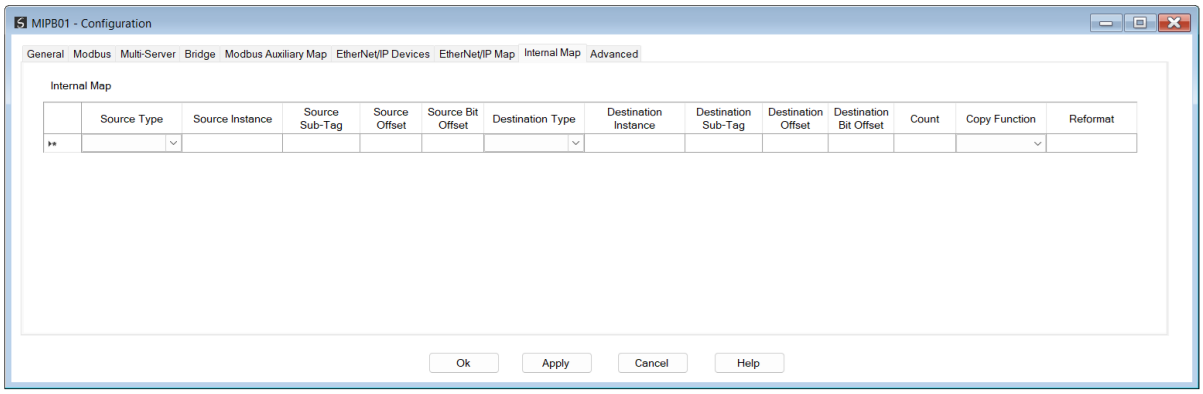


Figure 3.18 – Modbus Server – Internal Mapping

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 A-MBIPB is operating in Enhanced mode as a Modbus Client.

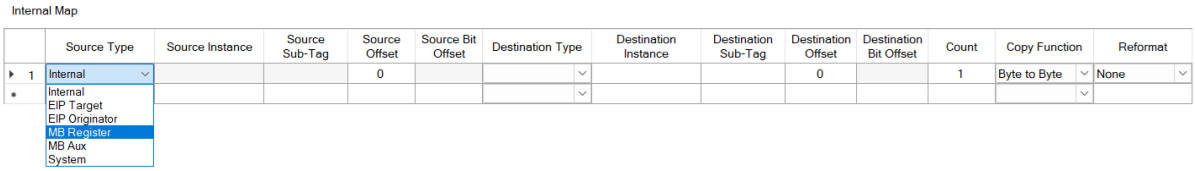


Figure 3.19 – IDS Copy - Modbus Register Source Type.

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

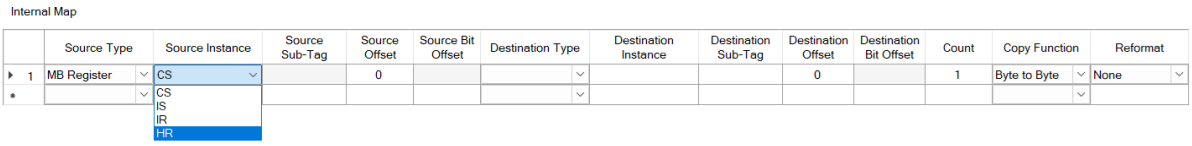


Figure 3.20 – IDS Copy - Modbus Register Source Instance.

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.

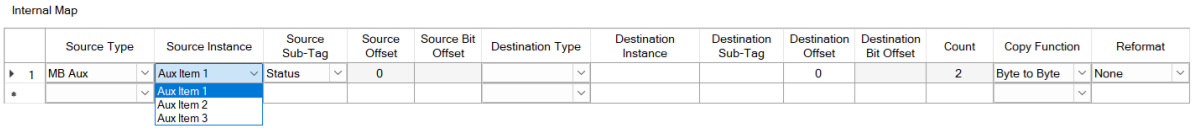


Figure 3.21 – IDS Copy - Modbus Register Source Instance.

The MB Aux format is as shown below:

Parameter	Data Type	Description
MB Aux Status	SINT	Bit 0 – Last Successful (0=Failed, 1=Success) Bit 1 – Waiting (0=Not waiting, 1= Waiting for Trigger) Bit 2 – Not Started (0=Started, 1=Not Started)
MB Aux Count	USINT	Successful transaction count (rolls-over at 255)

Table 3.4 – 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.

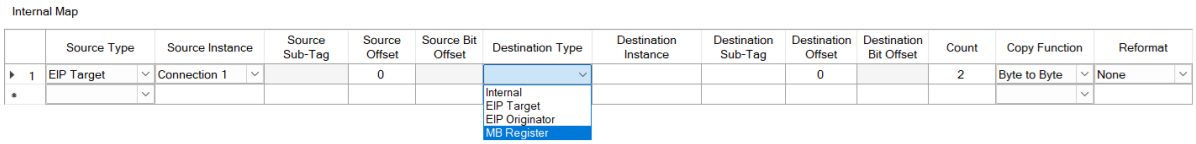


Figure 3.22 – IDS Copy - Modbus Register Destination Type.

The destination instance will be the Modbus register type required.

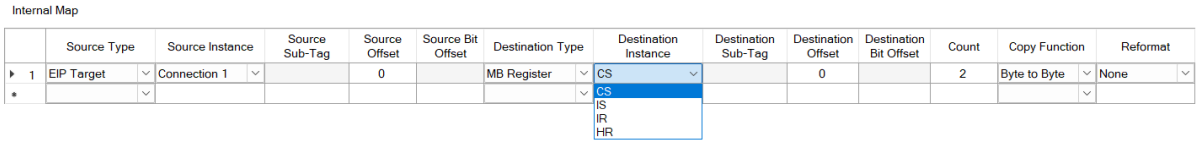


Figure 3.23 – IDS Copy - Modbus Register Destination Instance.

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.

3.6.1.2. CLIENT

The A-MBIPB can operate as a Modbus Client for Modbus TCP and RTU485. The user will need to configure the relevant Modbus Parameters as described in section 3.5.

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.

3.6.1.2.1. 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***.

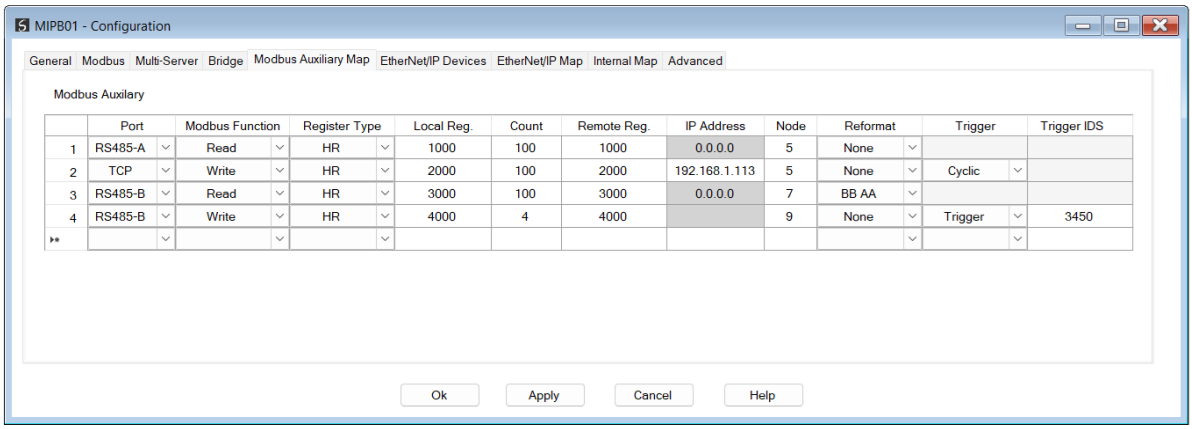


Figure 3.24 – Modbus Auxiliary Map Configuration.

The Modbus Auxiliary Map configuration consists of the following parameters:

Parameter	Description
Port	The external port to be used: **TCP** – Modbus TCP (Ethernet) **RS485-A** – Modbus RTU485 on physical Port A **RS485-B** – Modbus RTU485 on physical Port B
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) A-MBIPB 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
Trigger	For Modbus **Write** Functions, the user can select one of three triggers for the write operation. **Cyclic** The specific mapped item will be executed at the update rate (similar to all the read functions). **Change** When any of the data has been changed in the Modbus register range used to write the mapping, then this mapped item will be executed. **Trigger** When Trigger is selected, then the user will need to provide a IDS register which is used to trigger the specific Modbus Auxiliary option. When the provided IDS register changes, then the mapped item transaction will execute.
Trigger IDS	The IDS Register Offset used to trigger the respective Mapped item.

Table 3.5 – Modbus Auxiliary Map parameters

3.6.1.2.2. 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.

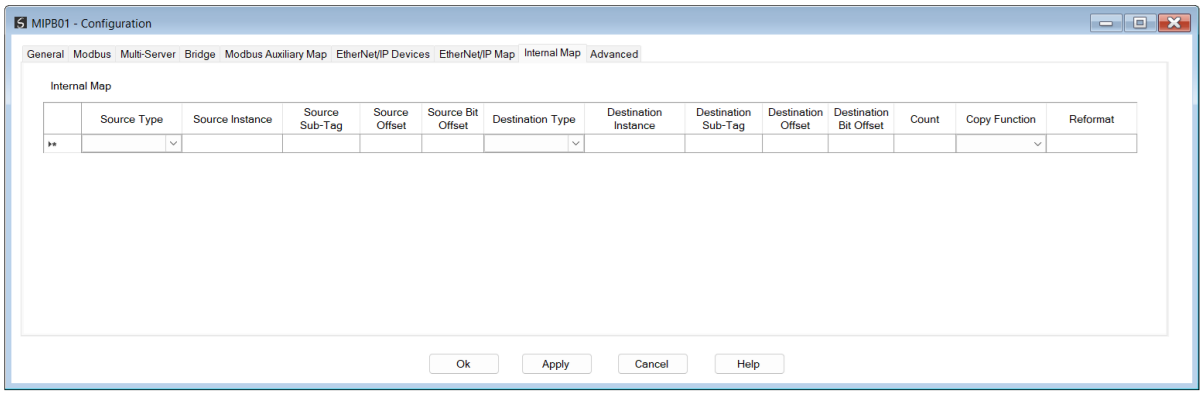


Figure 3.25 – Modbus Clients – Internal Mapping.

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 A-MBIPB is operating in Enhanced mode as a Modbus Client.



Figure 3.26 – IDS Copy - Modbus Register Source Type.

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



Figure 3.27 – 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.

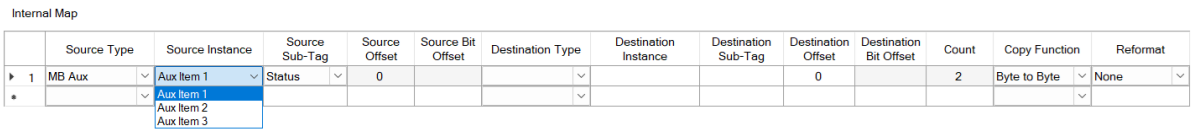


Figure 3.28 – IDS Copy - Modbus Register Source Instance.

The MB Aux format is as shown below:

Parameter	Data Type	Description
MB Aux Status	SINT	Bit 0 – Last Successful (0=Failed, 1=Success) Bit 1 – Waiting (0=Not waiting, 1= Waiting for Trigger) Bit 2 – Not Started (0=Started, 1=Not Started)

MB Aux Count	USINT	Successful transaction count (rolls-over at 255)

Table 3.6 – 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.

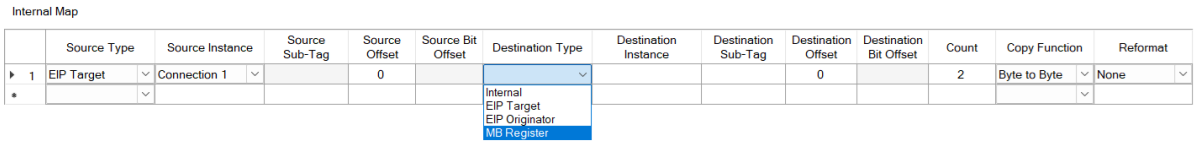


Figure 3.29 – IDS Copy - Modbus Register Destination Type.

The destination instance will be the Modbus register type required.

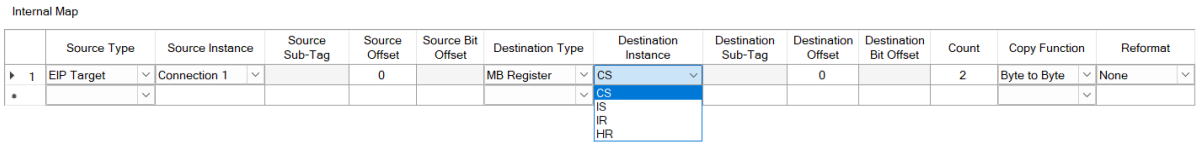


Figure 3.30 – IDS Copy - Modbus Register Destination Instance.

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.

3.6.2. EtherNet/IP Mode

3.6.2.1. Target

A controller (e.g. Logix controller) can own the A-MBIPB 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 A-MBIPB 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 A-MBIPB 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 3.6.2.1.1.

For older versions (19 and below), the module must be added using a Generic Profile which is described in section 3.6.2.1.2.

NOTE: The A-MBIPB 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.

3.6.2.1.1. STUDIO 5000 CONFIGURATION (VERSION 21+)

ADD MODULE TO ETHERNET/IP I/O CONFIGURATION

The A-MBIPB 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.

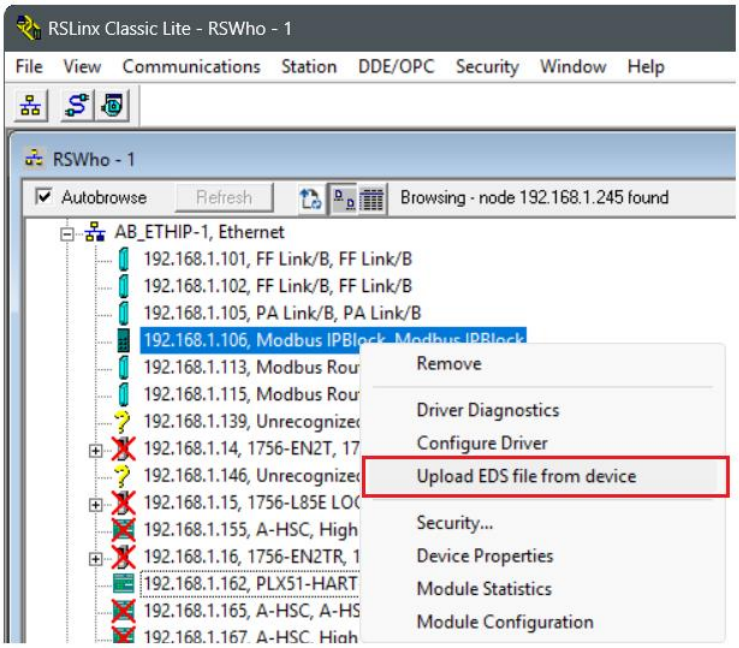


Figure 3.31 – EDS file upload from A-MBIPB.

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.

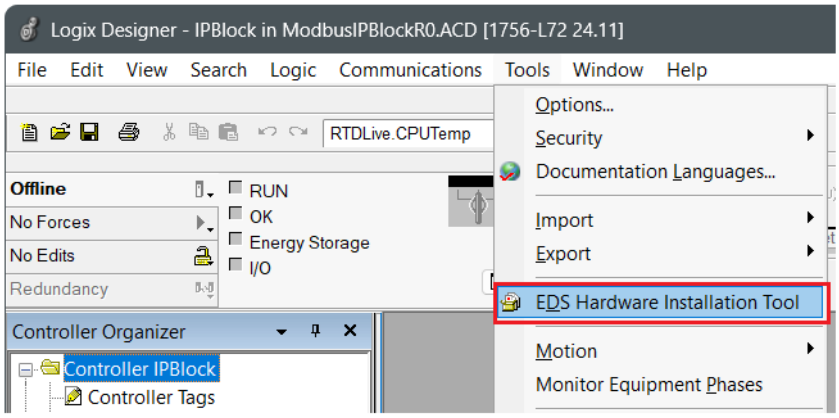


Figure 3.32 - EDS Hardware Installation Utility.

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.

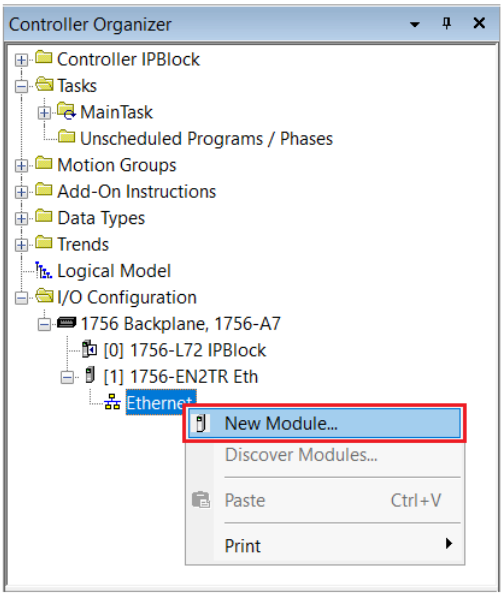


Figure 3.33 – Adding a module.

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.

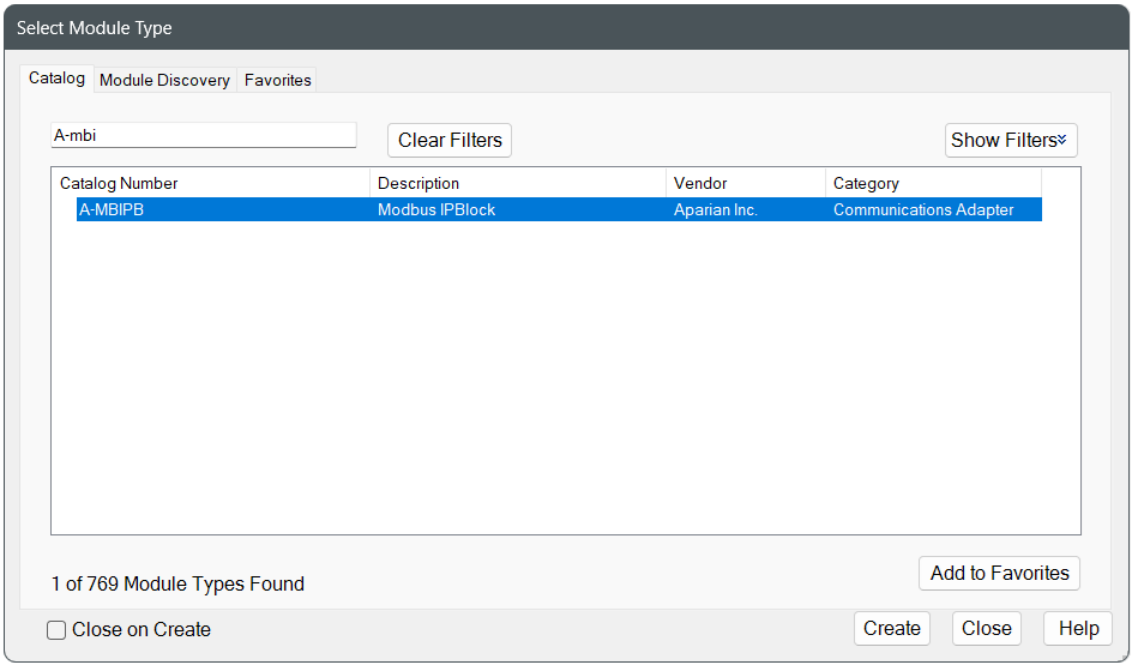


Figure 3.34 – Selecting the module.

Locate and select the Modbus IPBlock 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.

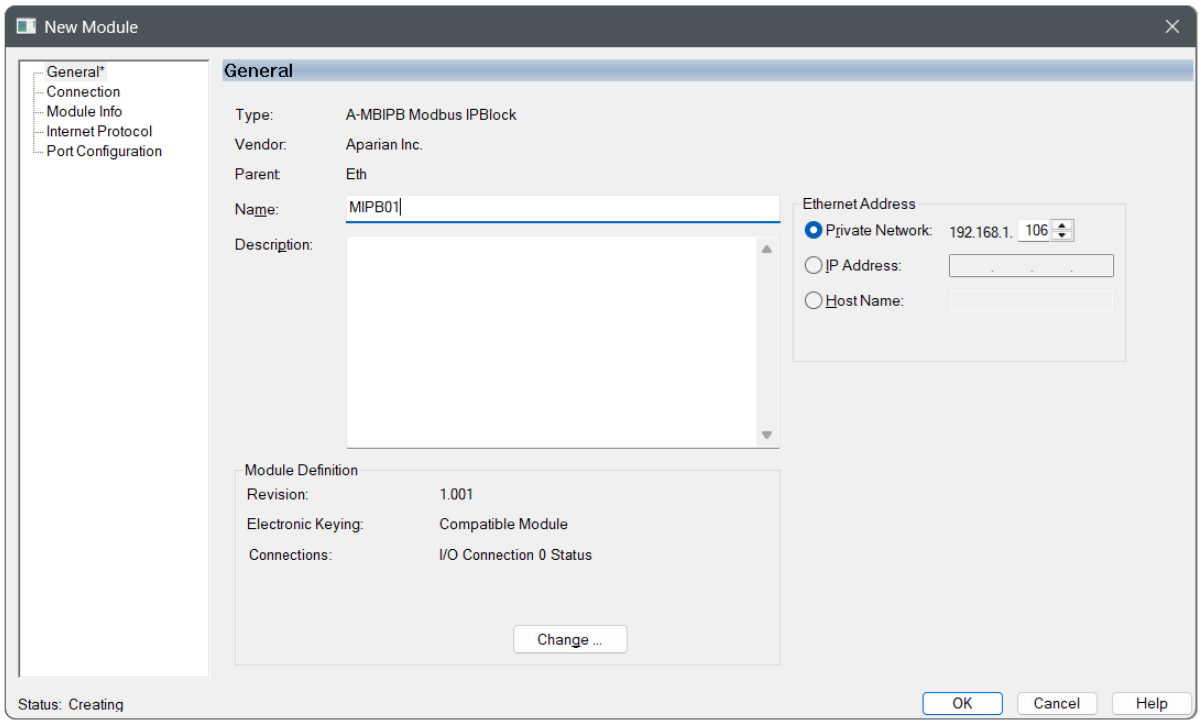


Figure 3.35 – Module instantiation.

The A-MBIPB 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**.

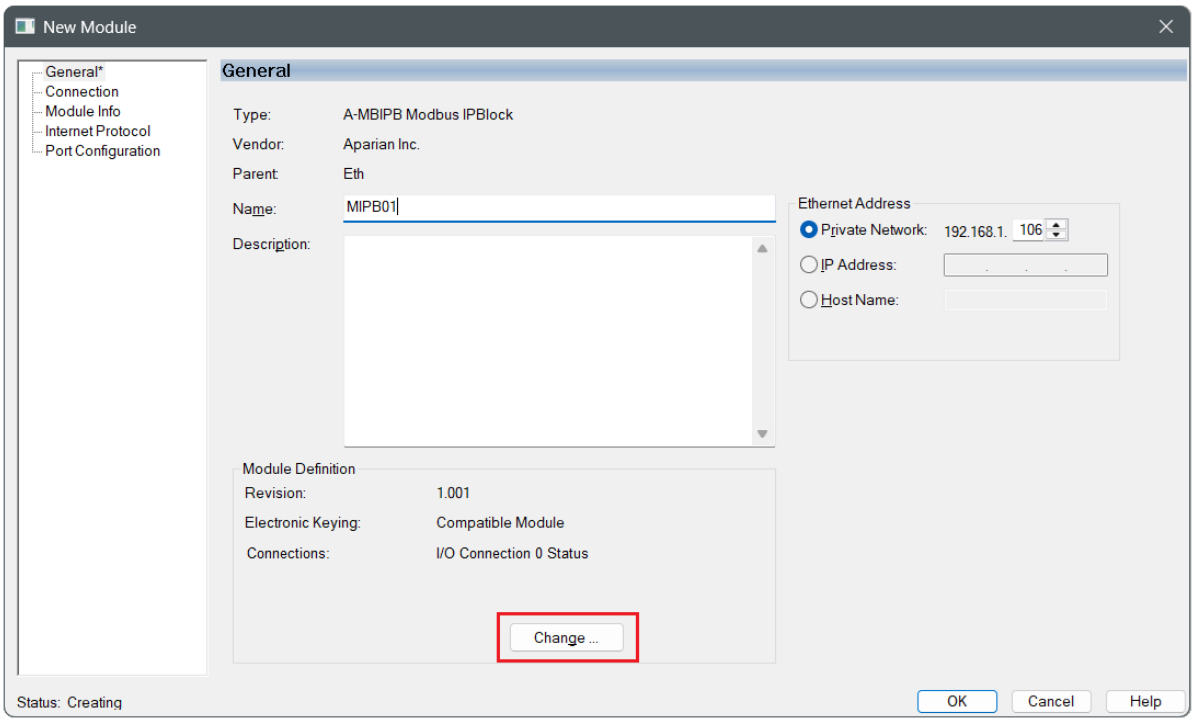


Figure 3.36 – Change number of IO Connections.

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

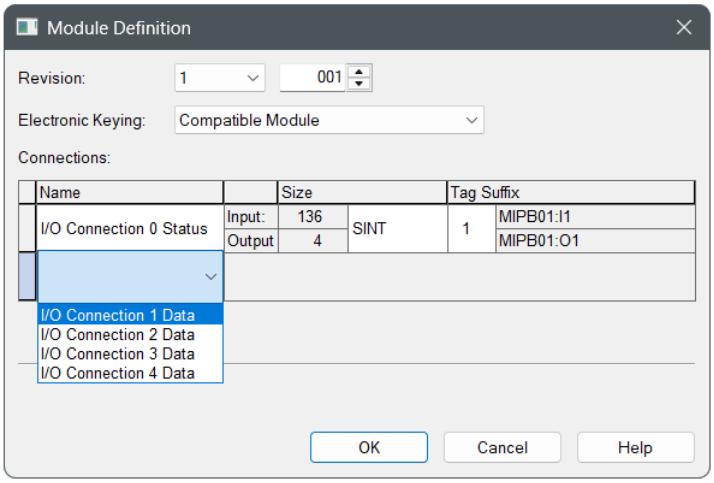


Figure 3.37 – Selection of IO Connections.

Now the A-MBIPB module will be in the Logix IO tree.

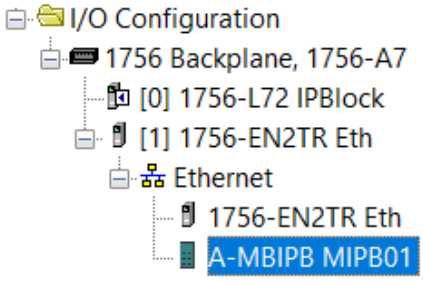


Figure 3.38 – Logix IO tree.

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.

3.6.2.1.2. RSLOGIX 5000 CONFIGURATION (PRE-VERSION 21)

The module can operate in both a Logix "owned" and standalone mode. When the module operates in a Logix "owned" mode the Modbus IPBlock 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: See the next section for importing the configuration (L5X).

NOTE: When using the generic profile, only one connection can be established.

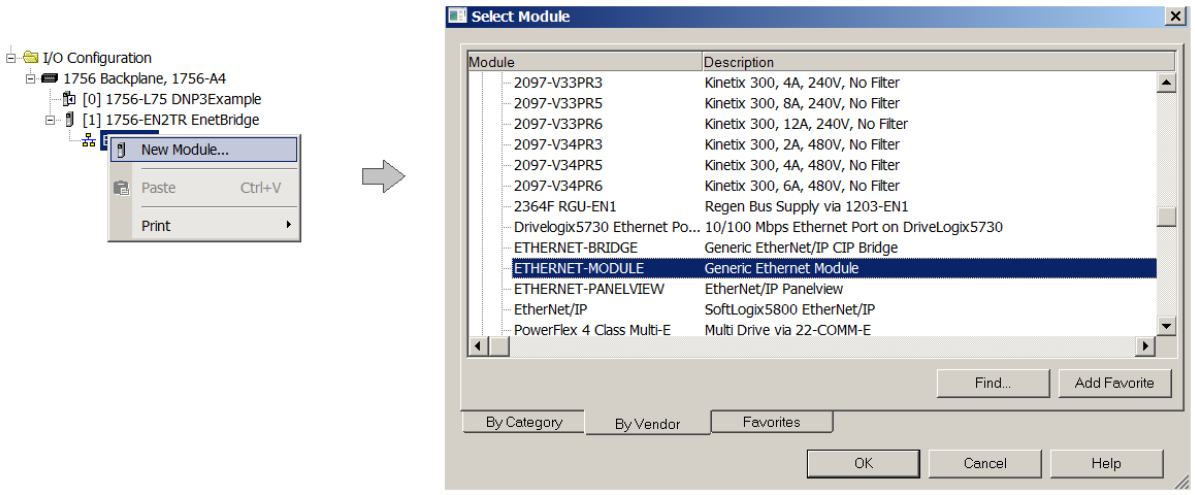


Figure 3.39 - Add a Generic Ethernet Module in RSLogix 5000.

Only one class 1 connection can be established to the module when using the generic profile. The user will need to decide if the status connection or a data connection will be used. The status connection will provide status information about the module (with no Modbus data mapping), while the data connection can be used for mapping (status and/or Modbus data mapping).

STATUS CONNECTION - ADD MODULE TO I/O CONFIGURATION

The user must enter the IP address of the Modbus IPBlock 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	136 (8-bit)
Output	101	4 (8-bit)
Configuration	102	0 (8-bit)

Table 3.7 - RSLogix class 1 connection parameters for the Modbus IPBlock.

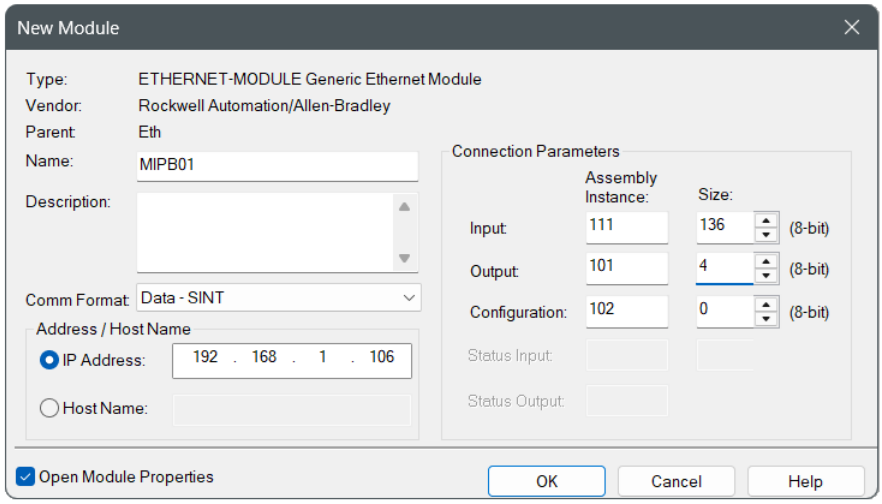


Figure 3.40 - General module properties in RSLogix 5000.

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.

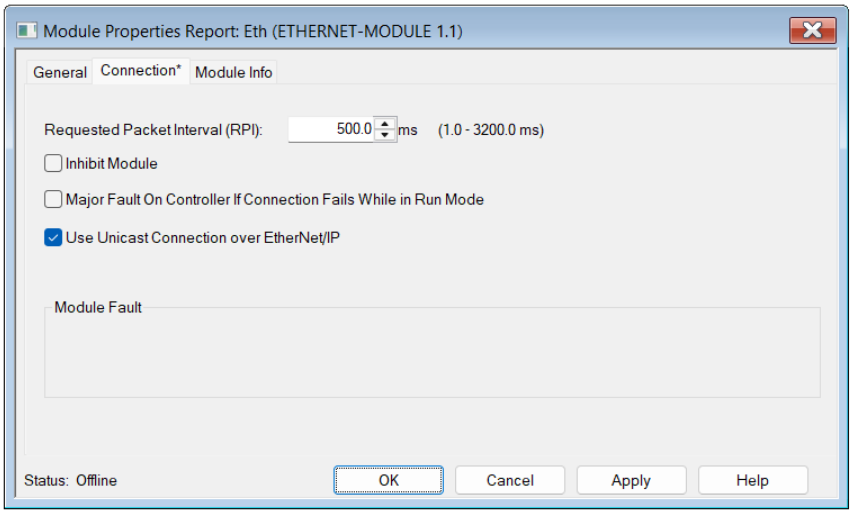


Figure 3.41 - Connection module properties in RSLogix 5000.

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.

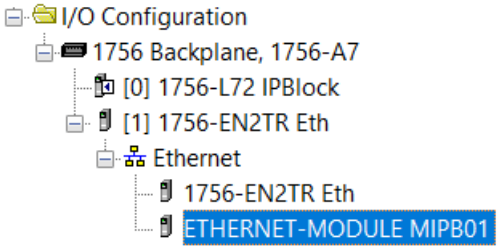


Figure 3.42 – RSLogix 5000 I/O module tree.

DATA CONNECTION - ADD MODULE TO I/O CONFIGURATION

The user must enter the IP address of the Modbus IPBlock 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	188	500 (8-bit)
Output	189	496 (8-bit)
Configuration	102	0 (8-bit)

Table 3.8 - RSLogix class 1 connection parameters for the Modbus IPBlock.

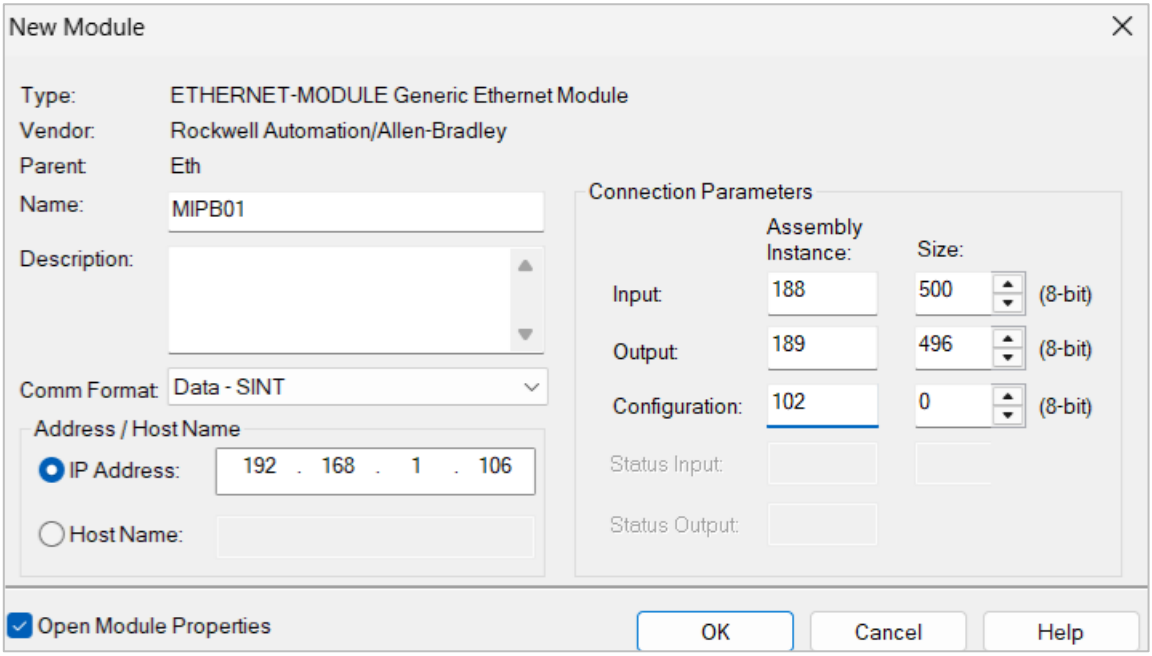


Figure 3.43 - General module properties in RSLogix 5000.

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.

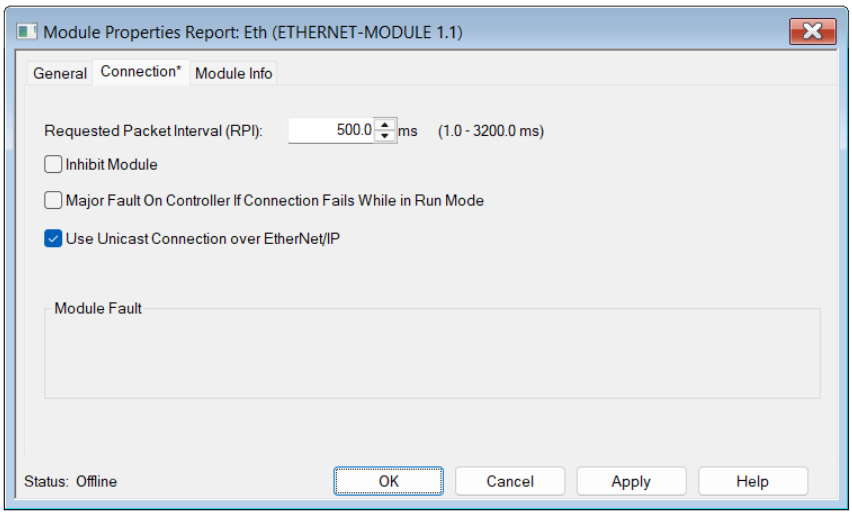


Figure 3.44 - Connection module properties in RSLogix 5000.

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.

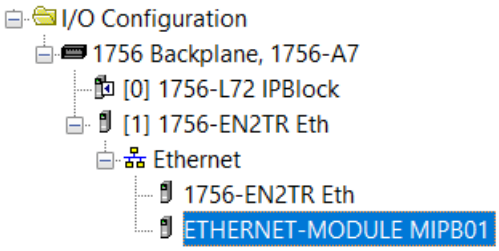


Figure 3.45 – RSLogix 5000 I/O module tree.

3.6.2.1.3. IMPORTING UDTS AND MAPPING ROUTINES

To simplify the mapping of the input image, a Studio 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.

NOTE: The module input and output assembly of the connection will be an undecorated array of bytes. The imported Logix routine will copy this data to the input and output assemblies.

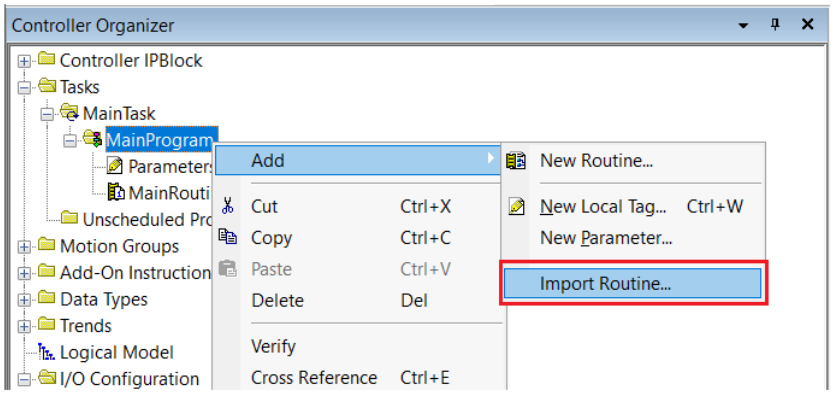


Figure 3.46 – Studio 5000 Importing Modbus IPBlock specific routine and UDTs.

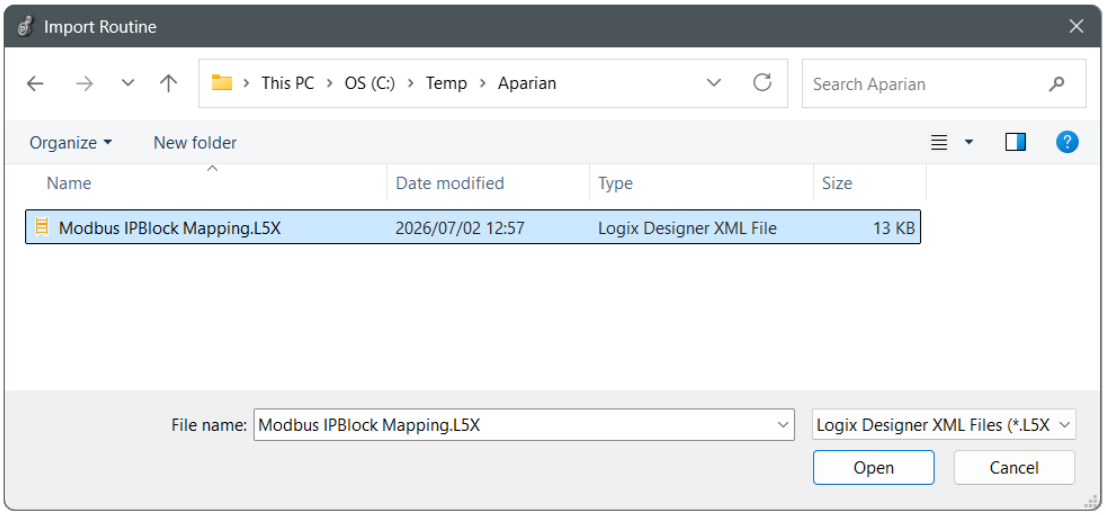


Figure 3.47 - Selecting partial import file.

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 Modbus IPBlock input and output assemblies to the aforementioned tags.

The user may need to change the routine to map to the correct Modbus IPBlock module instance name.

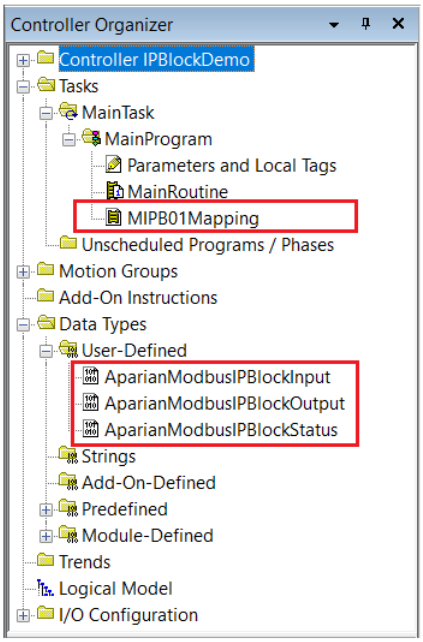


Figure 3.48 - Imported Studio 5000 objects.

The imported mapping routine must be called by the main routine.



Figure 3.49 – Calling mapping routine

An example of the Input Assembly Status tag (e.g. MIPB01StatusInput) is shown below.

Name	Value	Style	Data Type	Description
MIPB01StatusInput	{...}		AparianModbusIPBlockInput	
MIPB01StatusInput.Status	{...}		AparianModbusIPBlockStatus	General Status
MIPB01StatusInput.Status.ConfigValid	1	Decimal	BOOL	Configuration Valid (0=Invalid, 1=Valid)
MIPB01StatusInput.Status.EIPOwned	1	Decimal	BOOL	EtherNet/IP Target (0=Not-Owned, 1=Owned)
MIPB01StatusInput.Status.ControllerMode	1	Decimal	BOOL	Controller Mode (0=Program, 1=Run)
MIPB01StatusInput.Status.EIPOriginatorCommsOk	0	Decimal	BOOL	EtherNet/IP Originator (0=Failures, 1=AllConnectedOk)
MIPB01StatusInput.Status.ModbusOk	1	Decimal	BOOL	Modbus Status (0=Failures,1=AllOk)
MIPB01StatusInput.Status.ModbusMode	0	Decimal	BOOL	Modbus Mode (0=Client, 1=Server)
MIPB01StatusInput.Status.ModbusDisabledByController	0	Decimal	BOOL	Modbus Disabled By Controller (0=Normal, 1=Disabled)
MIPB01StatusInput.Status.NTPOk	1	Decimal	BOOL	NTP Status (0=Fail, 1=Ok)
MIPB01StatusInput.Status.EthernetPortA1	1	Decimal	BOOL	Link Status (0=Down, 1=Up)
MIPB01StatusInput.Status.EthernetPortA2	1	Decimal	BOOL	Link Status (0=Down, 1=Up)
MIPB01StatusInput.Status.EthernetPortB	1	Decimal	BOOL	Link Status (0=Down, 1=Up)
MIPB01StatusInput.Status.DLREnabled	1	Decimal	BOOL	Device Level Ring Enabled (0=Disabled, 1=Enabled)
MIPB01StatusInput.Status.DLRRing	1	Decimal	BOOL	Device Level Ring State (0=Linear, 1=Ring)
MIPB01StatusInput.Status.DLRFault	0	Decimal	BOOL	Device Level Ring Status (0=Ok, 1=Fault)
MIPB01StatusInput.ConfigCRC	16#c361	Hex	INT	Configuration Checksum
MIPB01StatusInput.TransactionRate	21	Decimal	INT	Transaction Rate (transactions per second)
MIPB01StatusInput.Mode	4	Decimal	SINT	Operation Mode (4 = Enhanced, 5 = Bridge, 6 = Concentrator)
MIPB01StatusInput.Temperature	43.7	Float	REAL	Device CPU Temperature (deg. C)
MIPB01StatusInput.AuxMapItemStatus	{...}	Decimal	SINT[100]	Auxiliary Mapping Item Status (Bit0: LastSuccessful , Bit1: Waiti...

Figure 3.50 – Input Assembly Status Tag

The decorated Input Assembly Status tag is structured as follows:

Parameter	Data Type	Description
Status	DINT	Module Status. Bit 0 - Configuration Valid (0=Invalid, 1=Valid) Bit 1 - EtherNet/IP Target (0=Not-Owned, 1=Owned) Bit 2 - Controller Mode (0=Program, 1=Run) Bit 3 - EtherNet/IP Originator (0=Failures, 1=All Connected Ok) Bit 4 - Modbus Status (0=Failures,1=All Ok) Bit 5 - Modbus Mode (0=Client, 1=Server) Bit 6 - Modbus Disabled By Controller (0=Normal, 1=Disabled) Bit 7 - NTP Status (0=Fail, 1=Ok) Bit 8 - Link Status (0=Down, 1=Up) Bit 9 - Link Status (0=Down, 1=Up) Bit 10 - Link Status (0=Down, 1=Up) Bit 11 - Device Level Ring Enabled (0=Disabled, 1=Enabled) Bit 12 - Device Level Ring State (0=Linear, 1=Ring)

		Bit 13 - Device Level Ring Status (0=Ok, 1=Fault)
ConfigCRC	INT	The module configuration signature.
Transaction Rate	INT	The transaction rate is the number of Modbus messages per second that the module is currently routing.
Mode	DINT	Operational Mode: 4 – Enhanced 5 – Bridge 6 – Concentrator
Temperature	REAL	Temperature of CPU
AuxMapItemStatus[100]	SINT[100]	Auxiliary Mapping Item Status for each mapped item. Bit 0 – Last Successful (0=Failed, 1=Success) Bit 1 – Waiting (0=Not waiting, 1= Waiting for Trigger) Bit 2 – Not Started (0=Started, 1=Not Started)

Table 3.9 – Input Assembly Status Tag Structure

An example of the Output Assembly Control tag (e.g. MIPB01ControlOutput) is shown below.

Name	Value	Style	Data Type	Description
MIPB01ControlOutput	{...}		AparianModbusIPBlockOutput	
MIPB01ControlOutput.DisableModbusCommunication	0	Decimal	BOOL	ModbusCommunication (0=Normal, 1=Disable)

Figure 3.51 – Output Assembly Control Tag

The decorated Output Assembly Control tag is structured as follows:

Parameter	**Data Type**	**Description**
Control	DINT	Module Control. Bit 0 - Modbus Disabled By Controller (0=Normal, 1=Disabled)

Table 3.10 – Output Assembly Control Tag Structure

3.6.2.1.4. 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.

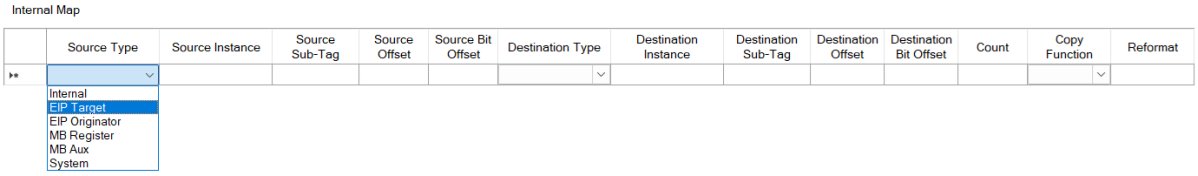


Figure 3.52 – IDS Copy – EtherNet/IP Target Source Type.

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.

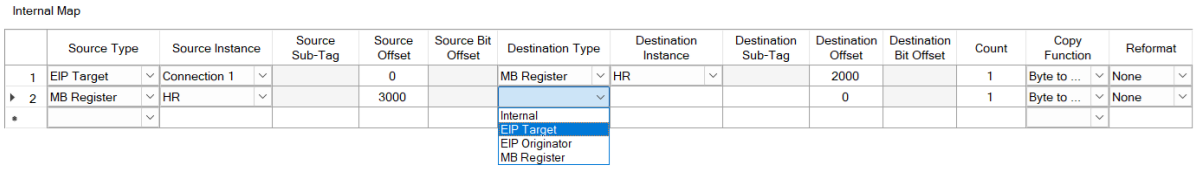


Figure 3.53 – IDS Copy – EtherNet/IP Target Destination Type.

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.

3.6.2.2. ORIGINATOR

The A-MBIPB 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.

3.6.2.2.1. ETHERNET/IP CLASS 1 DEVICE CONNECTIONS

The A-MBIPB 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.*

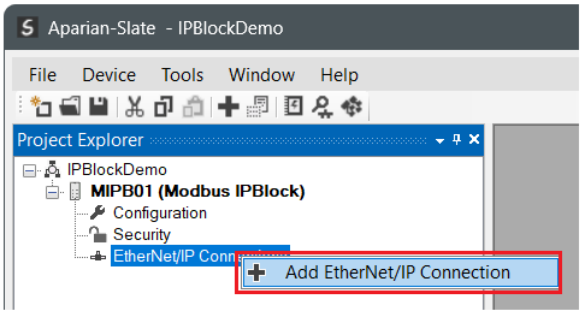


Figure 3.54 – Adding EtherNet/IP Class 1 Connection.

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

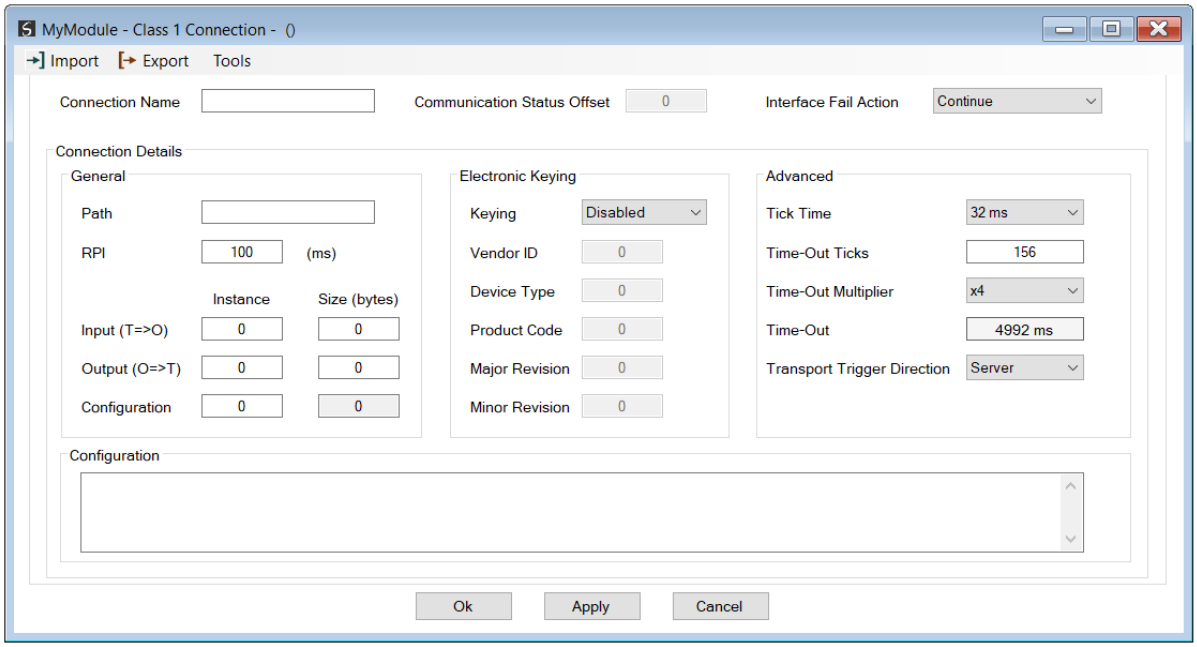


Figure 3.55 – EtherNet/IP Class 1 Connection Parameters.

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.

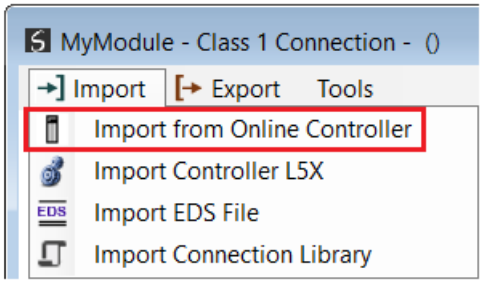


Figure 3.56 – Import from Online Controller.

The Import Connection Parameters form will open.

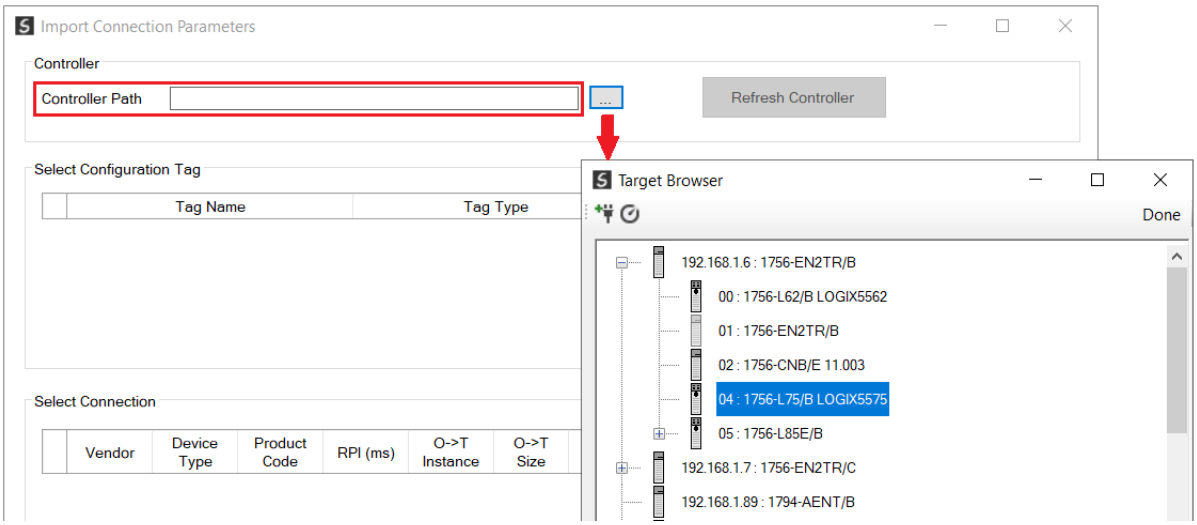


Figure 3.57 – Import Connection Parameters – Controller Path.

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.

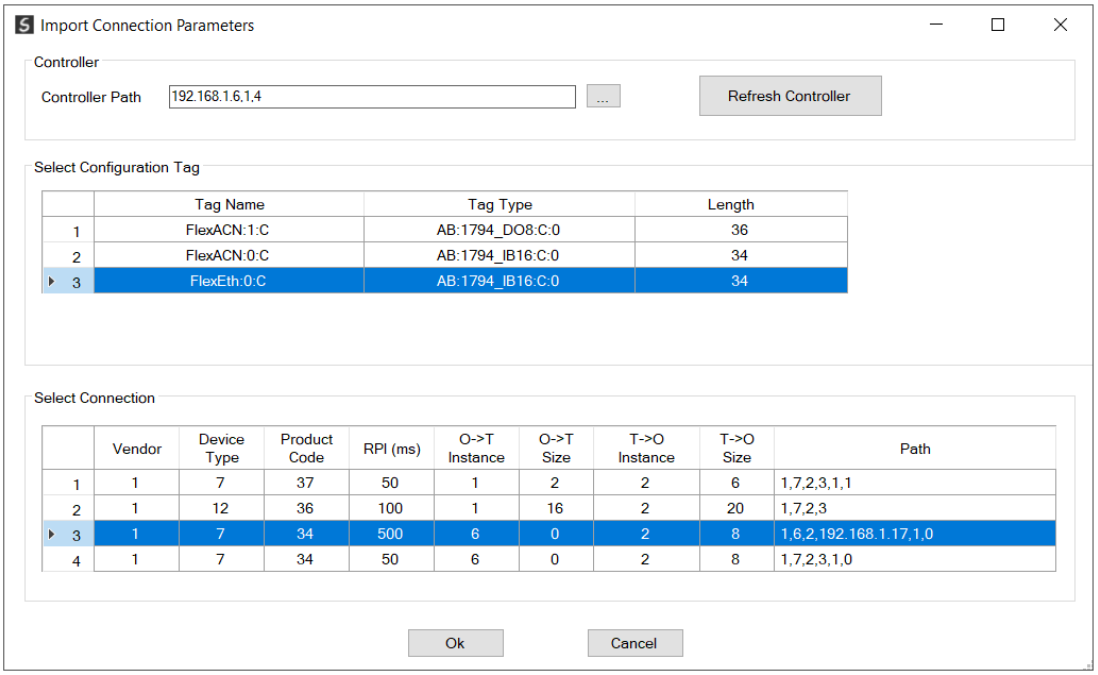


Figure 3.58 – Import Connection Parameters – Select Connection.

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.

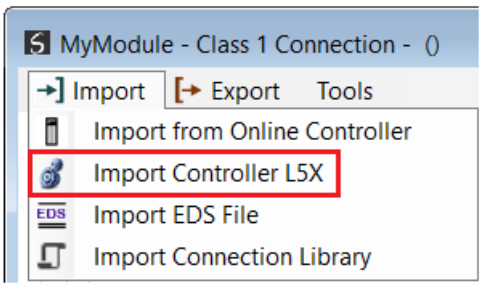


Figure 3.59 – Import from Controller L5X.

The Import Connection Parameters form will open.

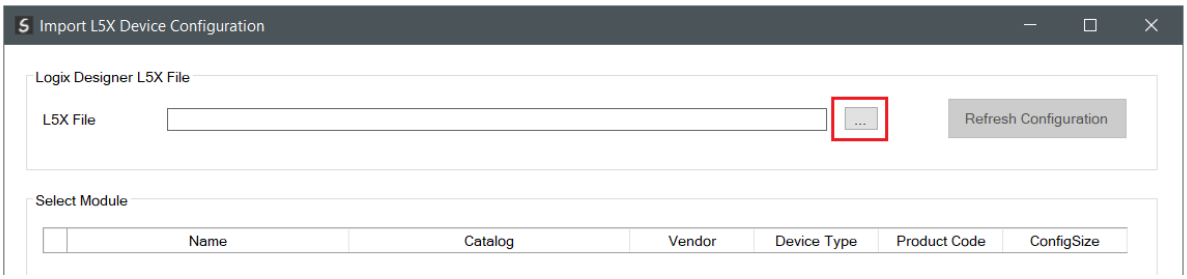
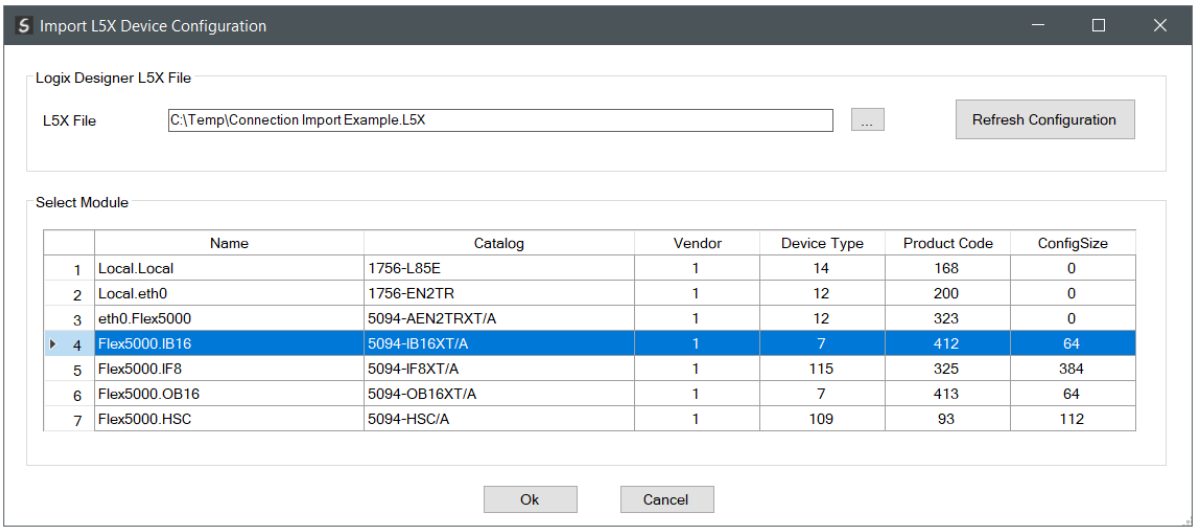


Figure 3.60 – Import L5X Device Configuration – Select L5X.

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.



	Name	Catalog	Vendor	Device Type	Product Code	ConfigSize
1	Local.Local	1756-L85E	1	14	168	0
2	Local.eth0	1756-EN2TR	1	12	200	0
3	eth0.Flex5000	5094-AEN2TRXT/A	1	12	323	0
4	Flex5000.IB16	5094-IB16XT/A	1	7	412	64
5	Flex5000.IF8	5094-IF8XT/A	1	115	325	384
6	Flex5000.OB16	5094-OB16XT/A	1	7	413	64
7	Flex5000.HSC	5094-HSC/A	1	109	93	112

Figure 3.61 – 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.

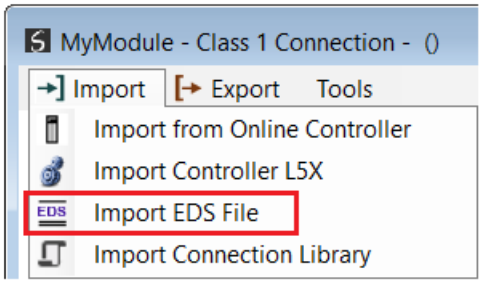


Figure 3.62 – Import EDS File.

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

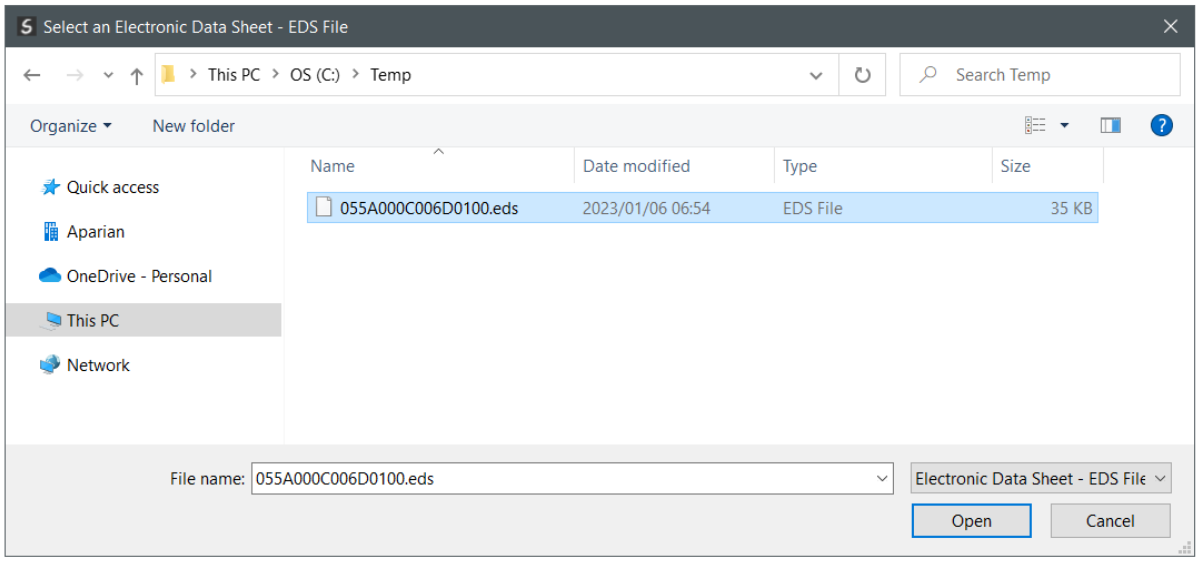


Figure 3.63 – Browse to EDS File.

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.

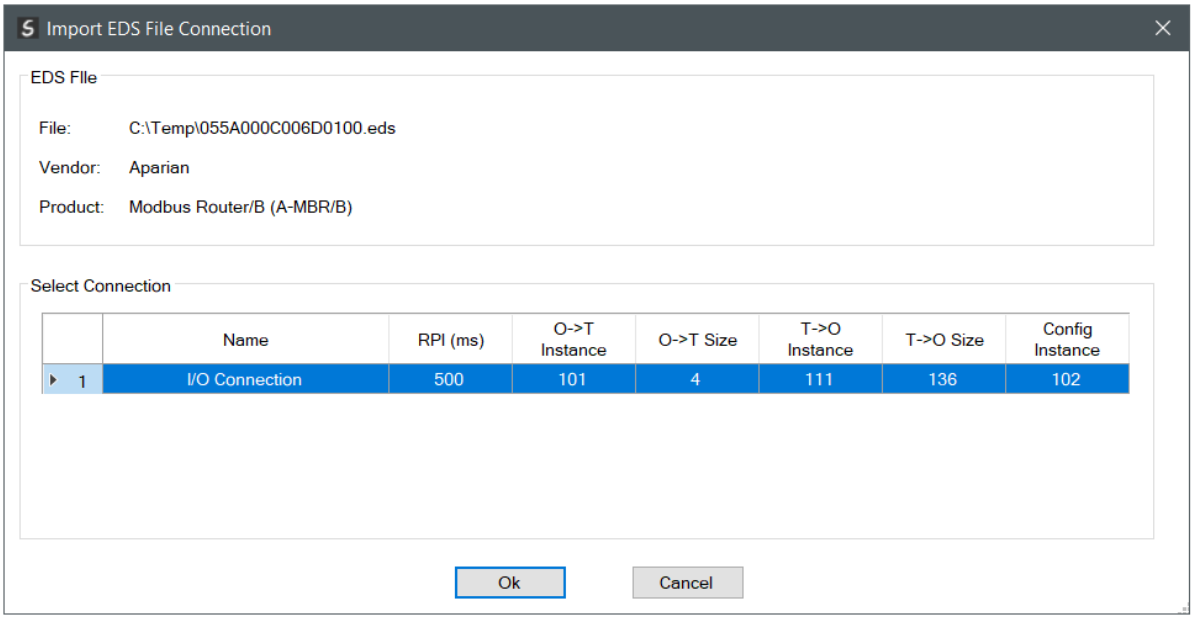


Figure 3.64 – Select Connection.

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.

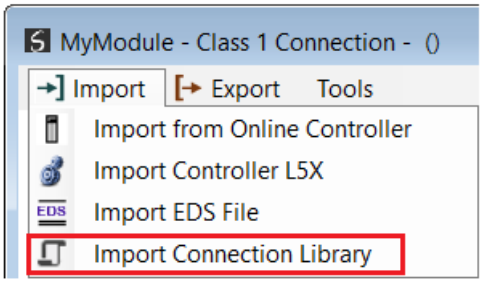


Figure 3.65 – Import Connection Library File.

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.

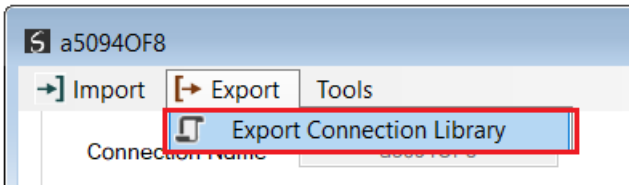


Figure 3.66 – Export Connection Library File.

3.6.2.2.2. 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*.

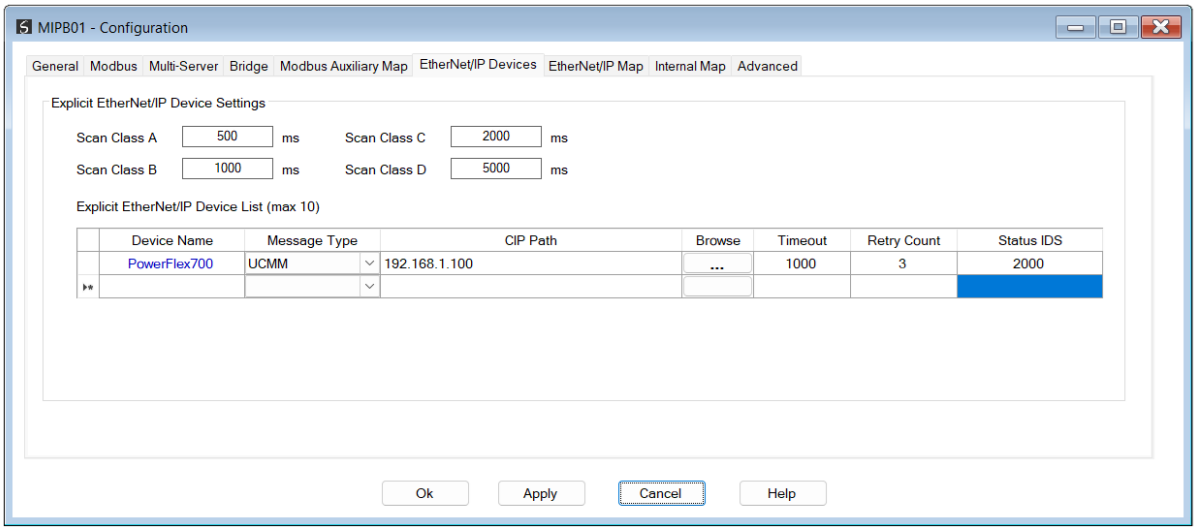


Figure 3.67 – EtherNet/IP Devices – Configuration.

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*.

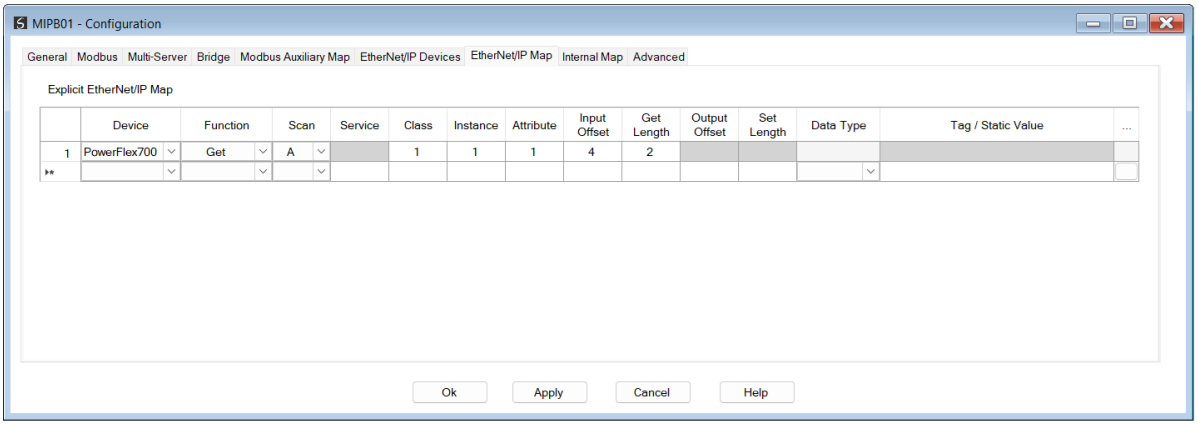


Figure 3.68 – EtherNet/IP Map configuration.

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 A-MBIPB 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 A-MBIPB 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 A-MBIPB 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.

3.6.2.2.3. 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.

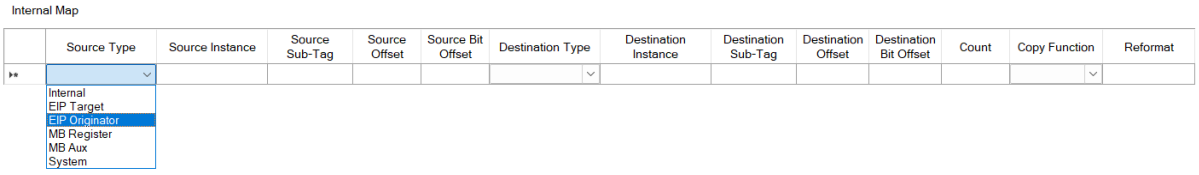


Figure 3.69 – IDS Copy – EtherNet/IP Originator Source Type.

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

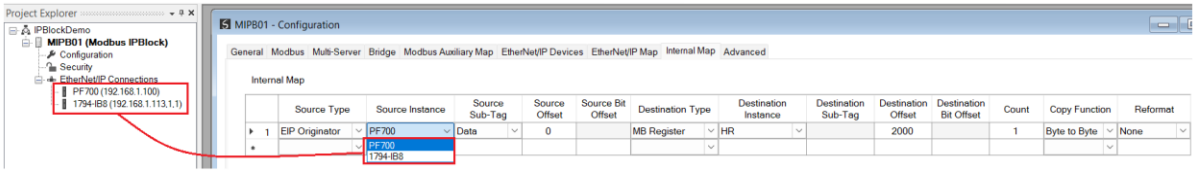


Figure 3.70 – IDS Copy – EtherNet/IP Originator Source Instance.

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.

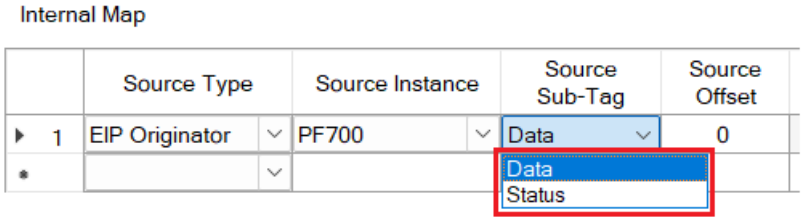


Figure 3.71 – IDS Copy – EtherNet/IP Originator Status.

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.

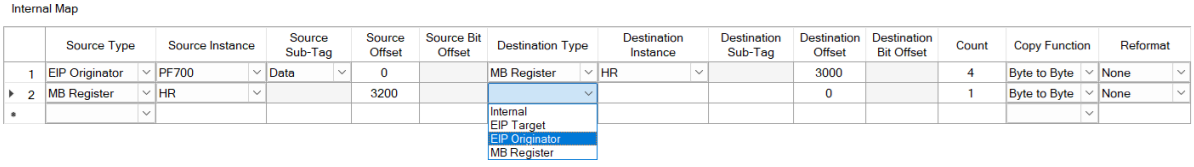


Figure 3.72 – IDS Copy – EtherNet/IP Originator Destination Type.

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

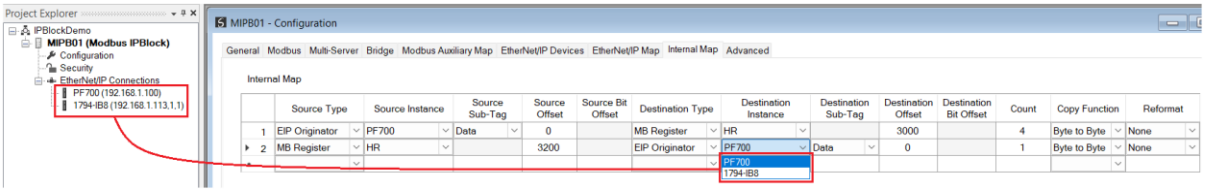


Figure 3.73 – IDS Copy – EtherNet/IP Originator Destination Instance.

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

3.7. BRIDGE MODE

In Bridge Mode, the A-MBIPB can operate as a transparent link for Modbus, between different media (TCP and RTU485). This allows, for example, a Modbus client on Modbus TCP to communicate with Modbus devices on RS485 and vice versa.

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

In the example below, any Modbus request received on the Ethernet TCP port for a Modbus Server at Node address 5 (to the IP address of the A-MBIPB 192.168.1.100), from the Modbus Client at IP 192.168.1.50, will be routed to the RS485-A 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).

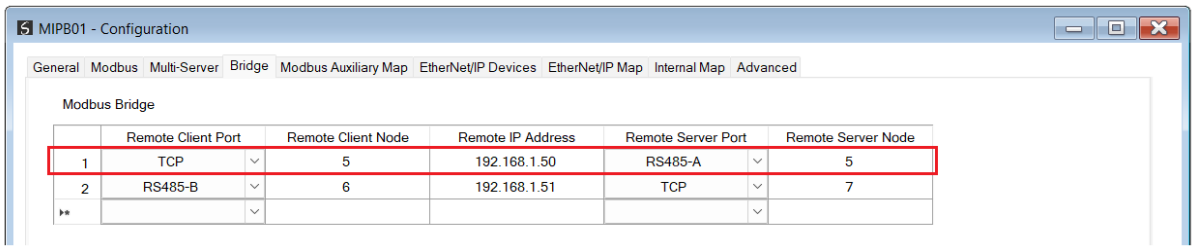


Figure 3.74 – Bridge Mode Map – TCP to RS485-A

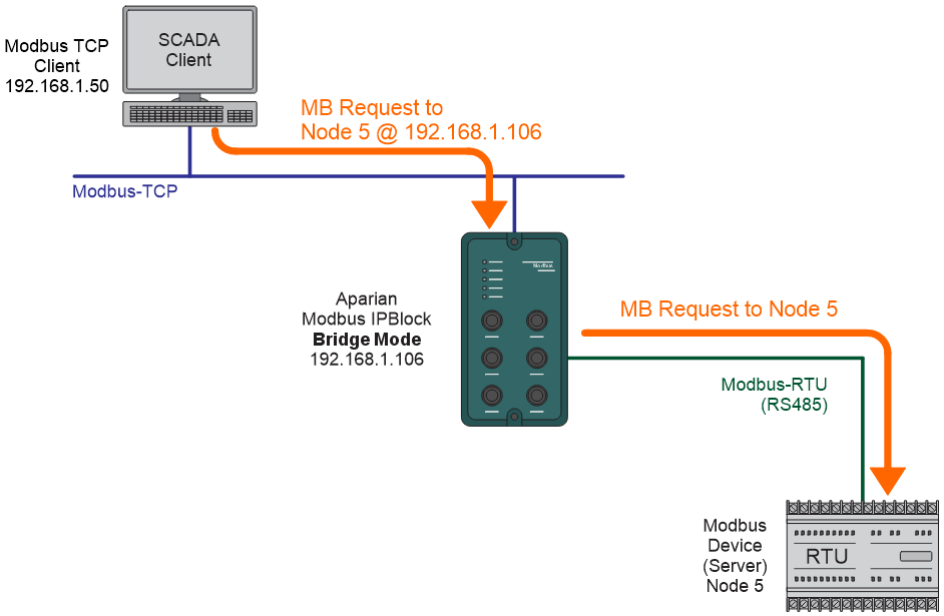


Figure 3.75 – Bridge Mode Diagram – TCP to RS485-A.

In the next example below, any Modbus request received on the RS485-B 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-B).

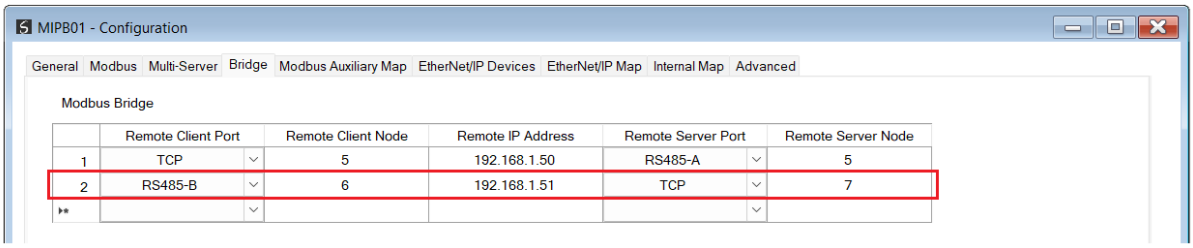


Figure 3.76 – Bridge Mode Map – RS485-B to TCP.

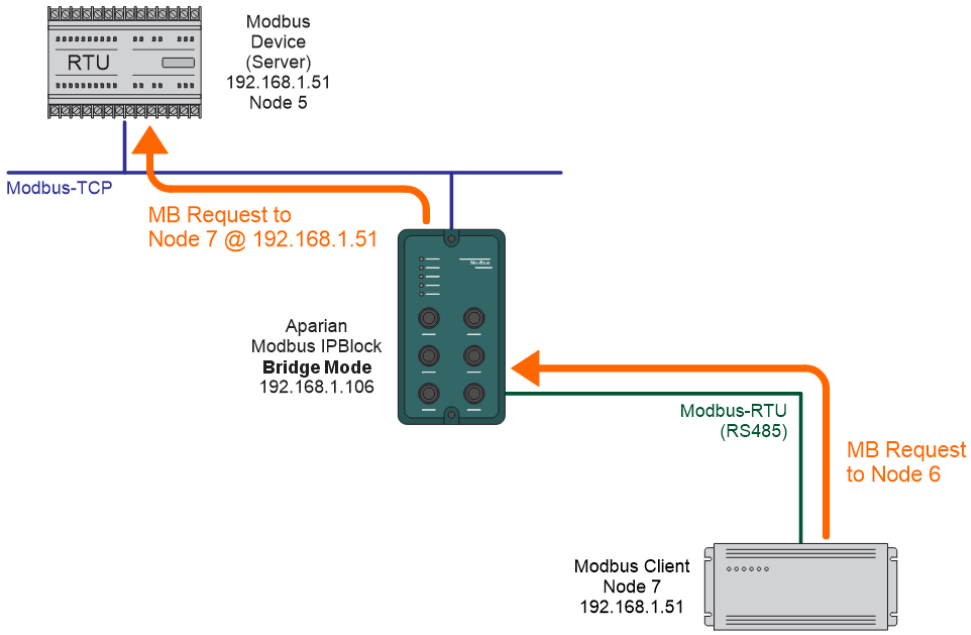


Figure 3.77 – Bridge Mode Diagram – RS485-B to TCP.

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

NOTE: When routing from TCP to Serial (RS485 A/B), 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.

3.8. CONCENTRATOR MODE

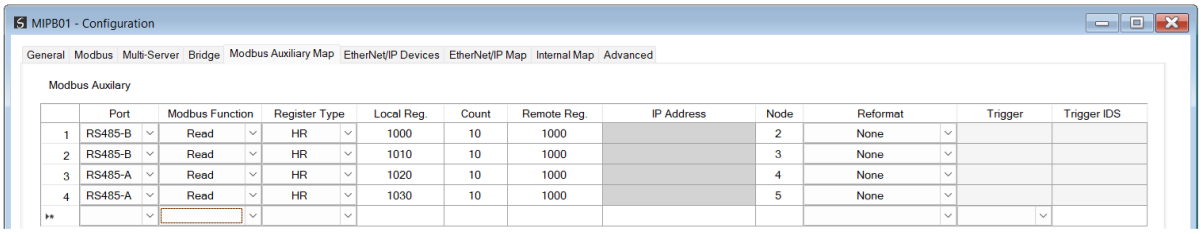
In Concentrator Mode, the A-MBIPB will collect data from various Modbus devices and provide the consolidated data to one or more Modbus Clients. The A-MBIPB will be a Modbus Server on one port, and a Modbus Client on the other ports. This will allow a Modbus SCADA to efficiently read the data from many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

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 A-MBIPB which is serving as a proxy for the Modbus RTU devices in the field. The A-MBIPB will collect data from the Modbus RTU devices via two separate RS485 (Modbus RTU) networks and save it in its internal Modbus Registers. Modbus HR 1000 will be read from each node and placed at 10 intervals in the A-MBIPB internal MB registers (e.g., data from node 2 will be at HR1000, data from node 3 will be at HR1010, data from node 4 will be at HR11020, etc.).

The Modbus SCADA will then be able to read that data from the A-MBIPB at the specific Modbus Registers in a single read. (HR1000..HR1039)

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	Trigger	Trigger IDS
1	RS485-B	Read	HR	1000	10	1000		2	None		
2	RS485-B	Read	HR	1010	10	1000		3	None		
3	RS485-A	Read	HR	1020	10	1000		4	None		
4	RS485-A	Read	HR	1030	10	1000		5	None		

Figure 3.78 – Concentrator Mode – Modbus Auxiliary Mapping.

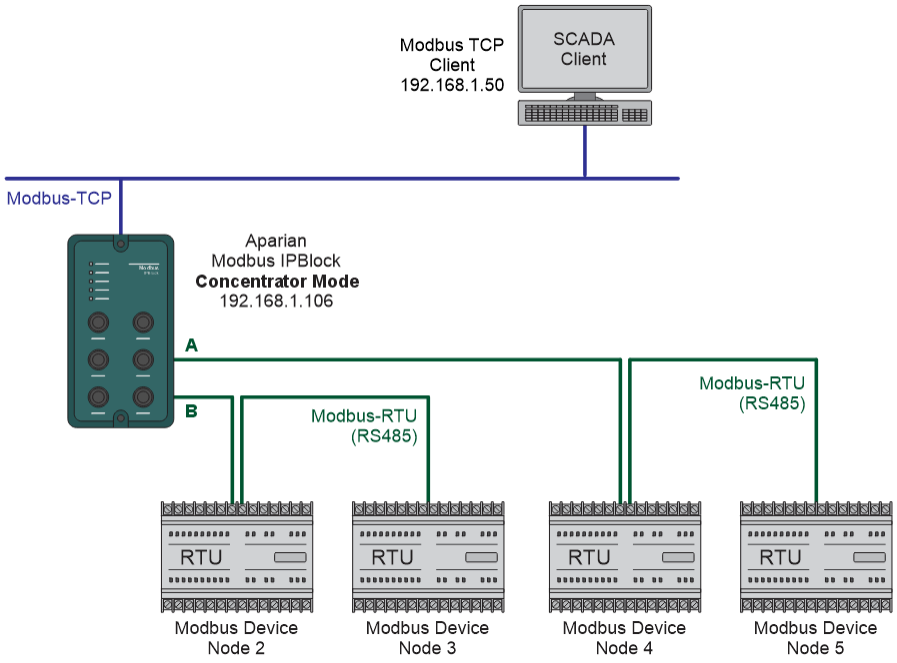


Figure 3.79 – Concentrator Mode Diagram.

3.9. 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.

3.9.1. COPY FROM

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

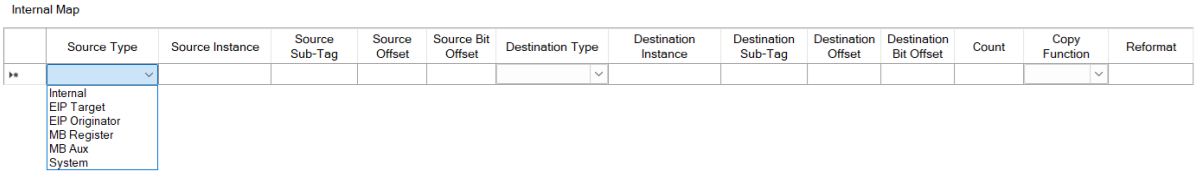


Figure 3.80 – Internal Map – Source Type.

3.9.1.1. INTERNAL

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

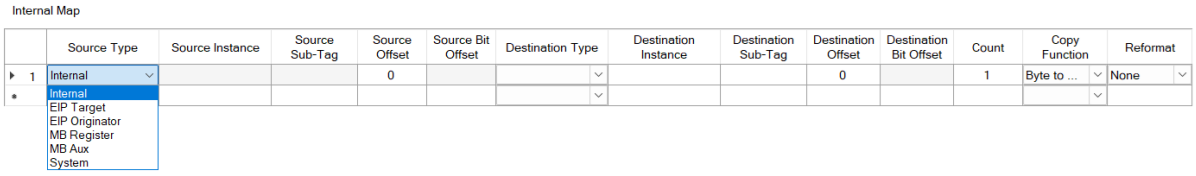


Figure 3.81 – IDS Copy – Internal Source Type.

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.

3.9.1.2. 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.

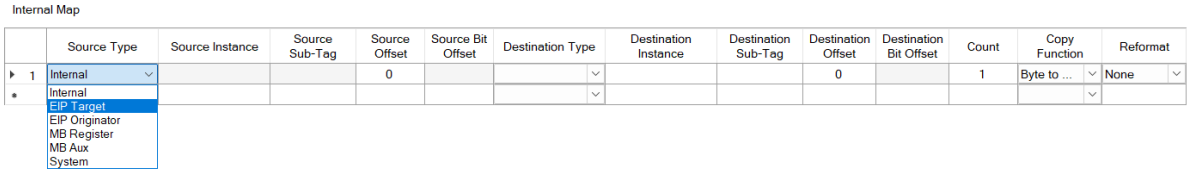


Figure 3.82 – IDS Copy – EtherNet/IP Target Source Type.

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.

3.9.1.3. EIP ORIGINATOR

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

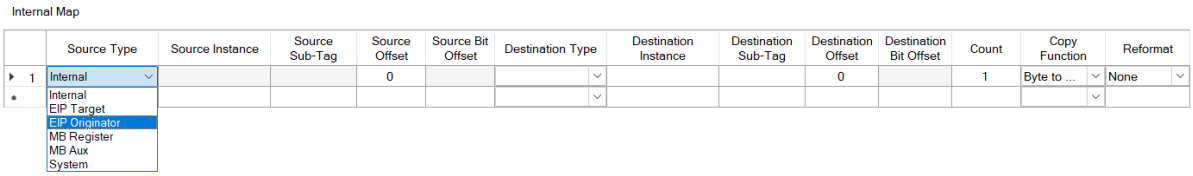


Figure 3.83 – IDS Copy – EtherNet/IP Originator Source Type.

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

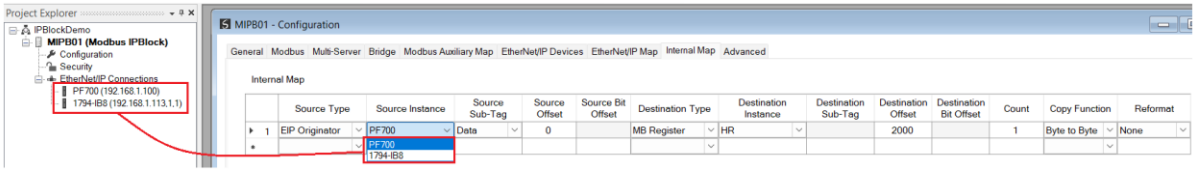


Figure 3.84 – IDS Copy – EtherNet/IP Originator Source Instance.

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.

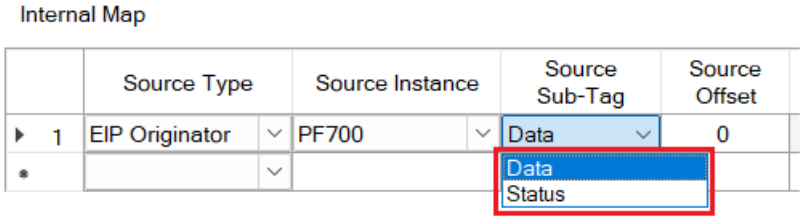


Figure 3.85 – IDS Copy – EtherNet/IP Originator Status.

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.

3.9.1.4. MB REGISTER

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

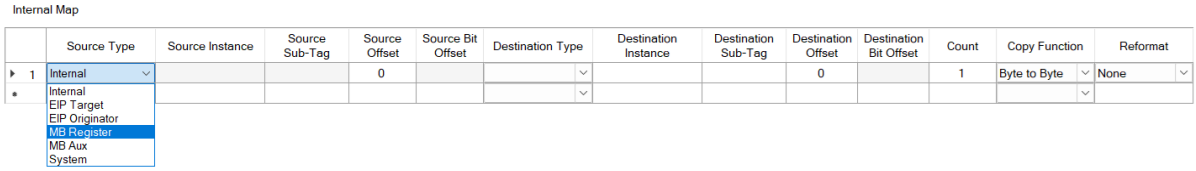


Figure 3.86 – IDS Copy - Modbus Register Source Type.

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

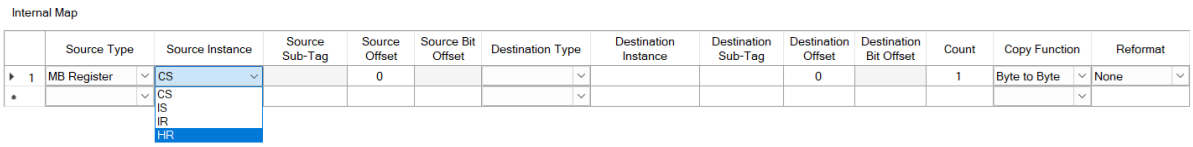


Figure 3.87 – IDS Copy - Modbus Register Source Instance.

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.

3.9.1.5. 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 A-MBIPB is operating in Enhanced mode as a Modbus Client.

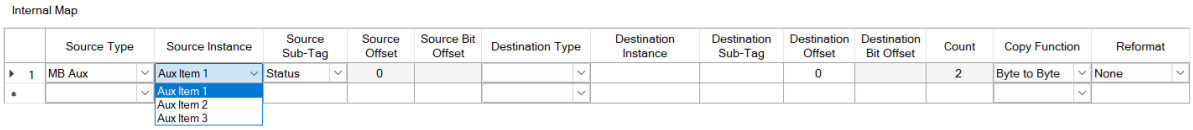


Figure 3.88 – IDS Copy - Modbus Register Source Instance.

The MB Aux format is as shown below:

Parameter	Data Type	Description
MB Aux Status	SINT	Bit 0 – Last Successful (0=Failed, 1=Success) Bit 1 – Waiting (0=Not waiting, 1= Waiting for Trigger) Bit 2 – Not Started (0=Started, 1=Not Started)
MB Aux Count	USINT	Successful transaction count (rolls-over at 255)

Table 3.18 – Modbus Aux Item Status

3.9.1.6. SYSTEM

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

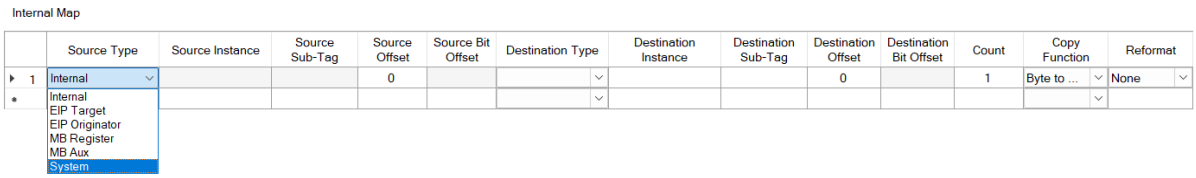


Figure 3.89 – IDS Copy – System Information.

The module's System information has the following format.

Parameter	Data Type	Description
Status	DINT	Module Status. Bit 0 - Configuration Valid (0=Invalid, 1=Valid) Bit 1 - EtherNet/IP Target (0=Not-Owned, 1=Owned) Bit 2 - Controller Mode (0=Program, 1=Run) Bit 3 - EtherNet/IP Originator (0=Failures, 1=All Connected Ok) Bit 4 - Modbus Status (0=Failures,1=All Ok) Bit 5 - Modbus Mode (0=Client, 1=Server) Bit 6 - Modbus Disabled By Controller (0=Normal, 1=Disabled)

		Bit 7 - NTP Status (0=Fail, 1=Ok) Bit 8 - Link Status (0=Down, 1=Up) Bit 9 - Link Status (0=Down, 1=Up) Bit 10 - Link Status (0=Down, 1=Up) Bit 11 - Device Level Ring Enabled (0=Disabled, 1=Enabled) Bit 12 - Device Level Ring State (0=Linear, 1=Ring) Bit 13 - Device Level Ring Status (0=Ok, 1=Fault)
ConfigCRC	INT	The module configuration signature.
Transaction Rate	INT	The transaction rate is the number of Modbus messages per second that the module is currently routing.
Mode	DINT	Operational Mode: 4 – Enhanced 5 – Bridge 6 – Concentrator
Temperature	REAL	Temperature of the CPU.
AuxMapItemStatus[100]	SINT[100]	Auxiliary Mapping Item Status for each mapped item. Bit 0 – Last Successful (0=Failed, 1=Success) Bit 1 – Waiting (0=Not waiting, 1= Waiting for Trigger) Bit 2 – Not Started (0=Started, 1=Not Started)

Table 3.19 – System Information Format.

3.9.2. COPY TO

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

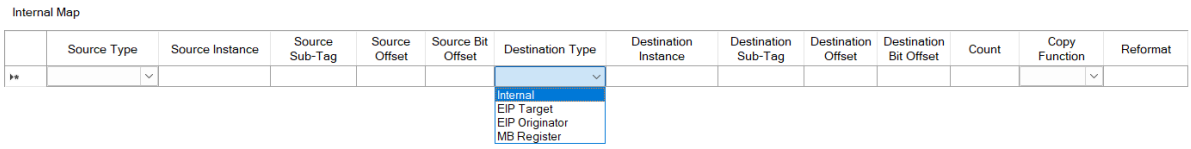


Figure 3.90 – Internal Map – Destination Type.

3.9.2.1. INTERNAL

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

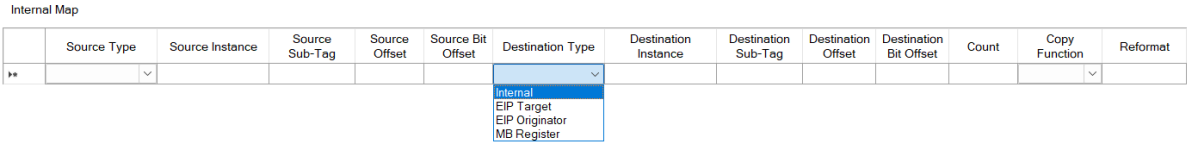


Figure 3.91 – IDS Copy – Internal Source Type.

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.

3.9.2.2. EIP TARGET

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

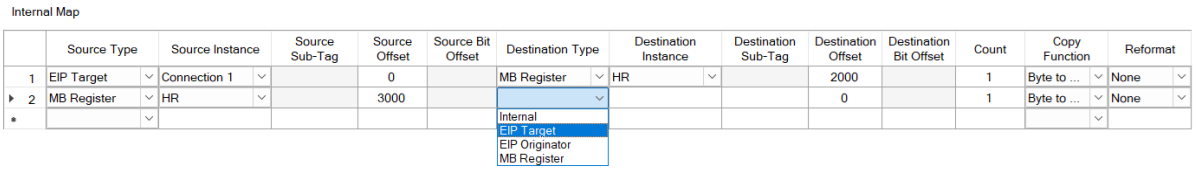


Figure 3.92 – IDS Copy – EtherNet/IP Target Destination Type.

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.

3.9.2.3. 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.

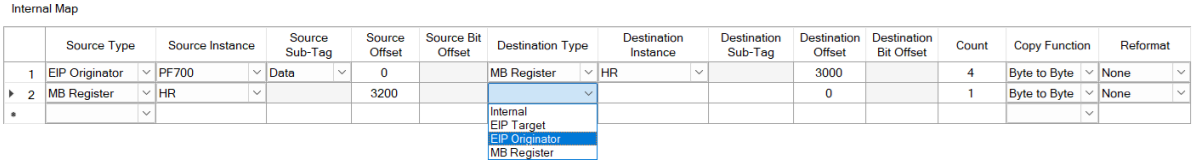


Figure 3.93 – IDS Copy – EtherNet/IP Originator Destination Type.

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

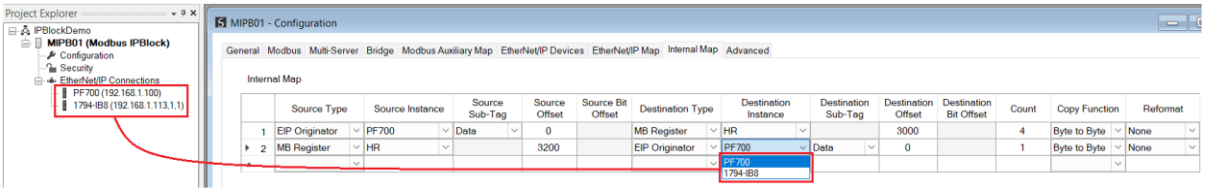


Figure 3.94 – IDS Copy – EtherNet/IP Originator Destination Instance.

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.

3.9.2.4. MB REGISTER

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

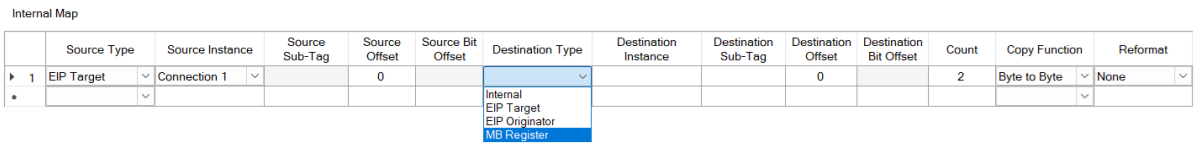


Figure 3.95 – IDS Copy - Modbus Register Destination Type.

The destination instance will be the Modbus register type required.

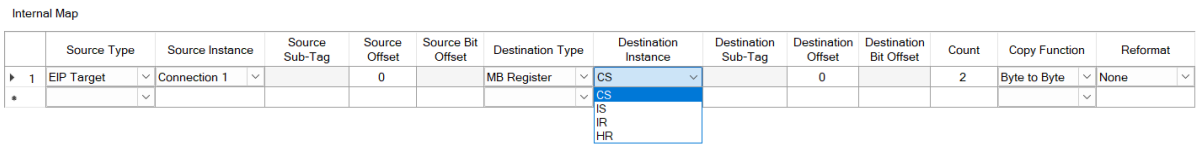


Figure 3.96 – IDS Copy - Modbus Register Destination Instance.

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. 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*.

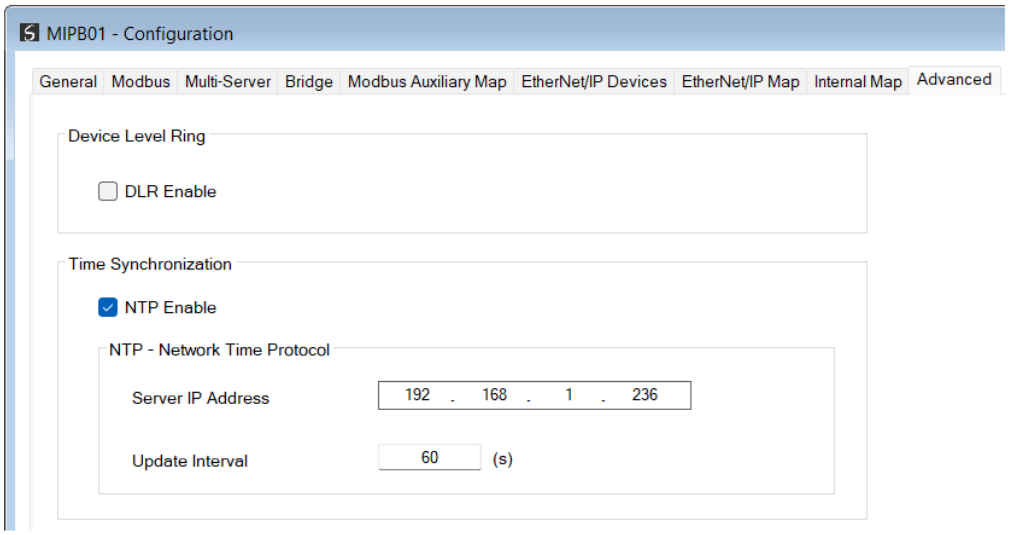


Figure 3.97 – Advanced configuration

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 A-MBIPB 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 A-MBIPB will request time from the NTP Server.

Table 3.20 – Advanced configuration parameters.

3.11. 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 IPBlock is not on a local network.

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

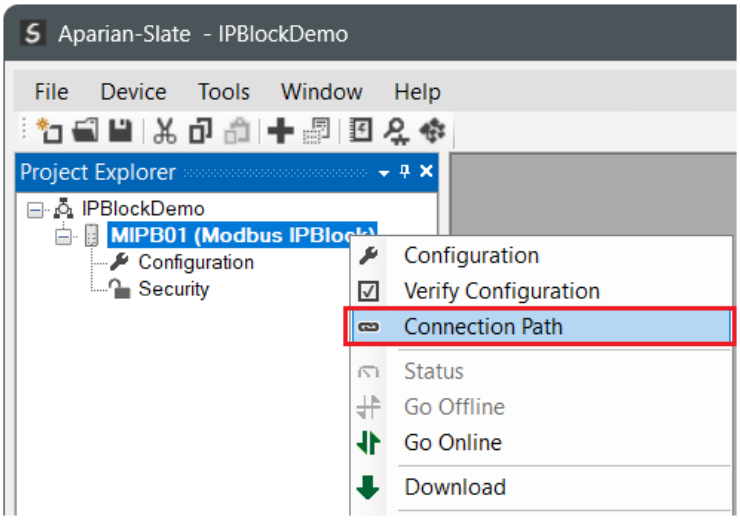


Figure 3.98 - Selecting Connection Path.

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

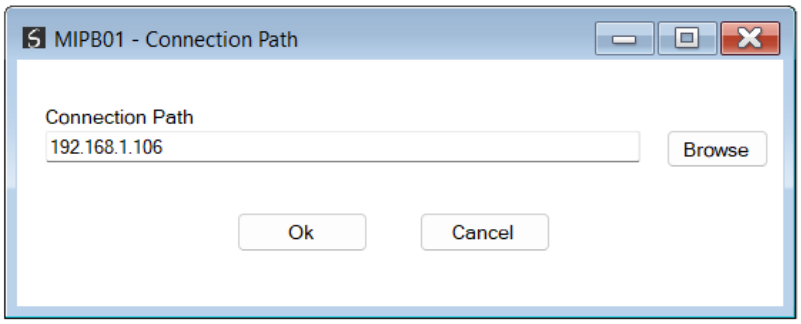


Figure 3.99 - Connection Path.

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

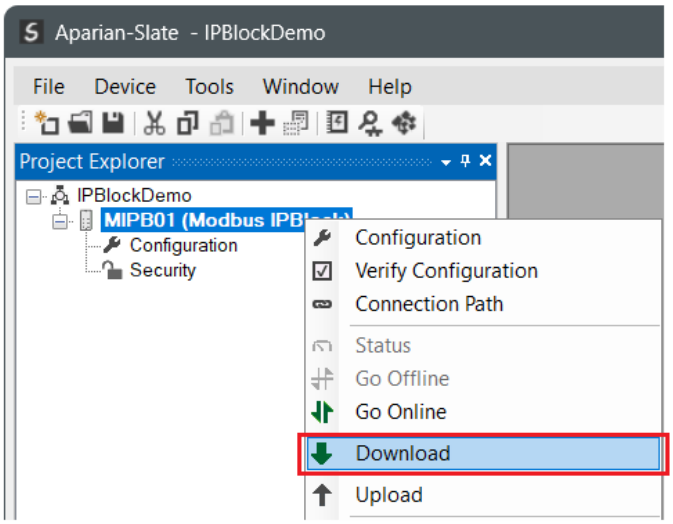


Figure 3.100 - Selecting Download.

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

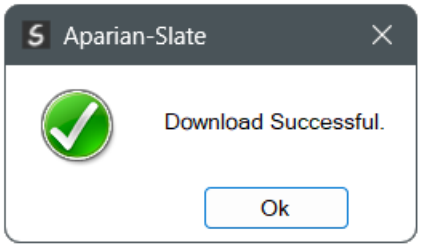


Figure 3.101 - Successful download.

The module is now configured and will start operating immediately.

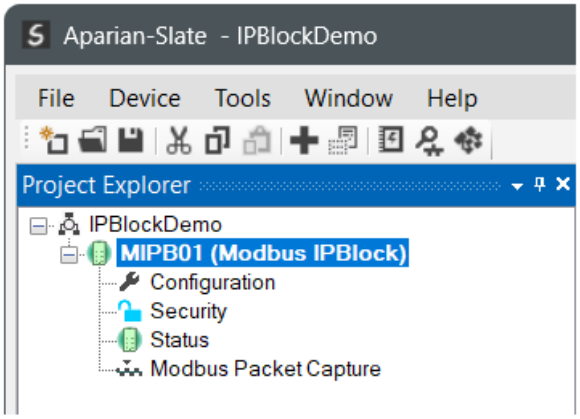


Figure 3.102 - Module online.

4. DEVICE FIRMWARE UPDATE

The A-MBIPB 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.

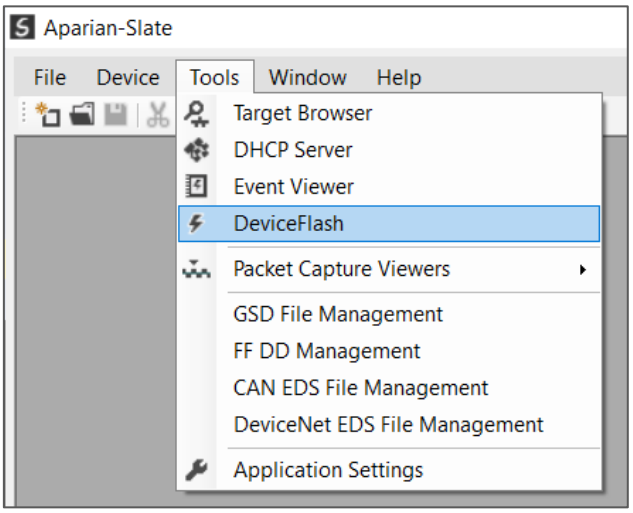


Figure 4.1 – Select DeviceFlash utility from Slate.

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

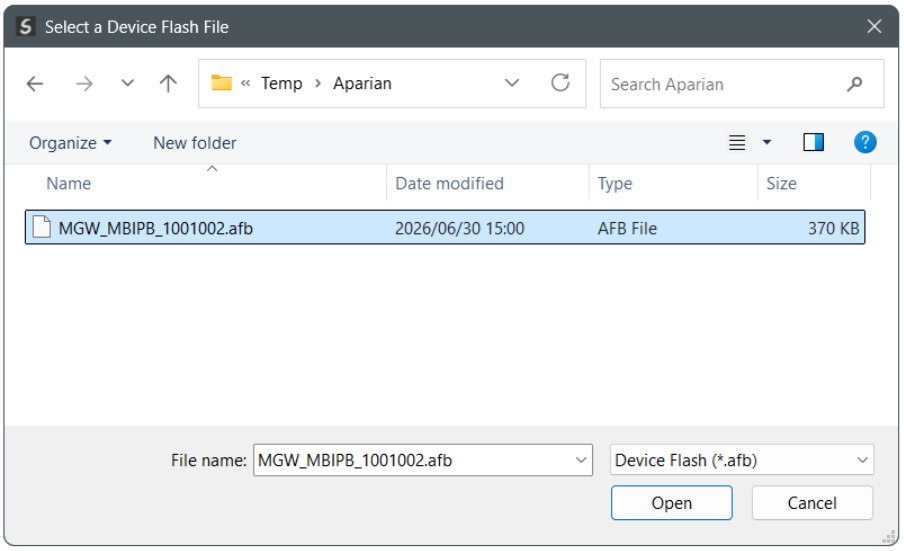


Figure 4.2 – Select the binary file.

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

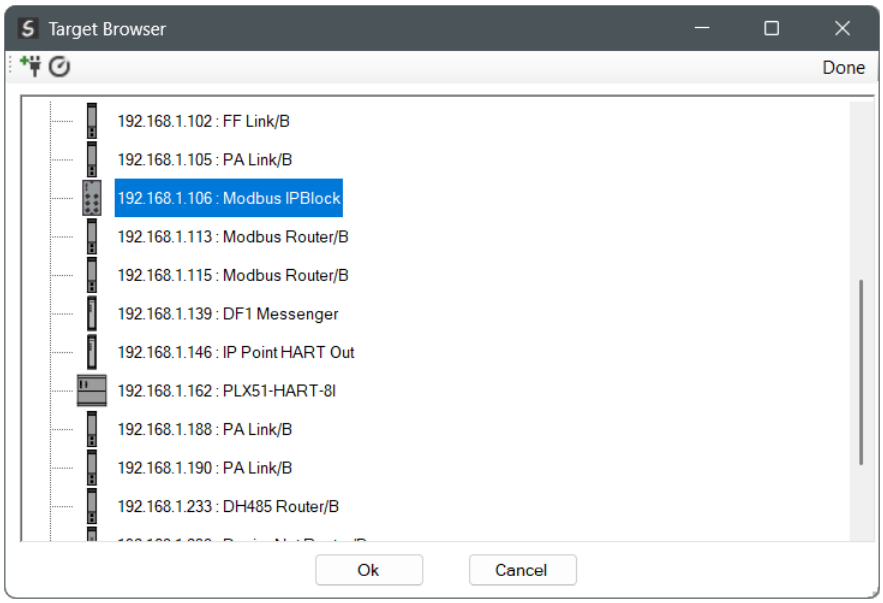


Figure 4.3 – Select the device to be updated.

- 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.

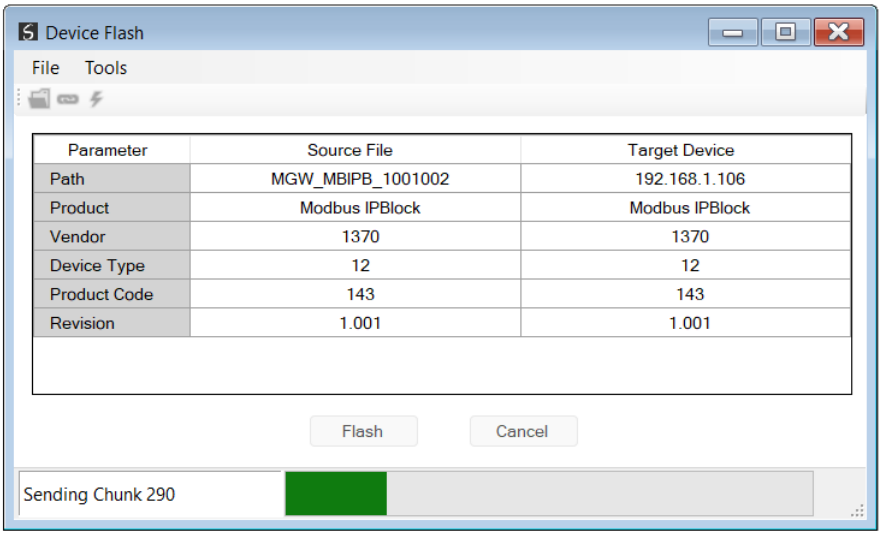


Figure 4.4 – Firmware update busy.

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

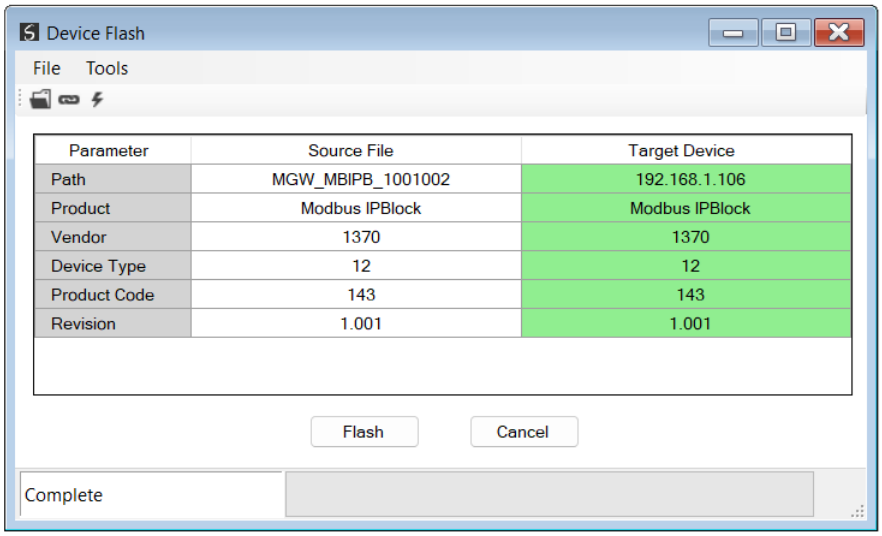


Figure 4.5 – Firmware update successfully completed.

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 A-MBIPB 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.

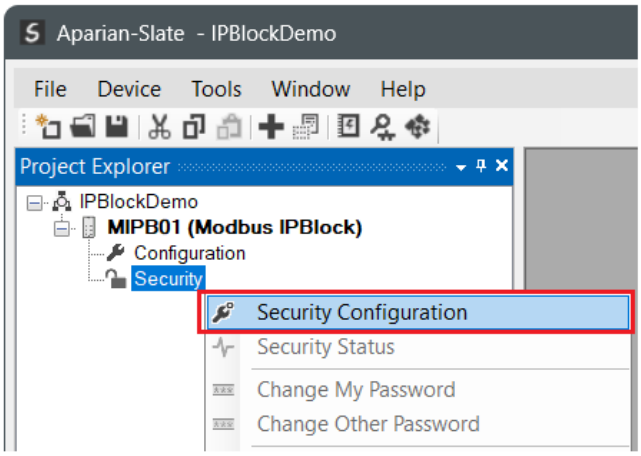


Figure 5.1 – Selecting Security Configuration.

The configuration form will open and display multiple configuration tabs.

5.1.1.1. General Configuration

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

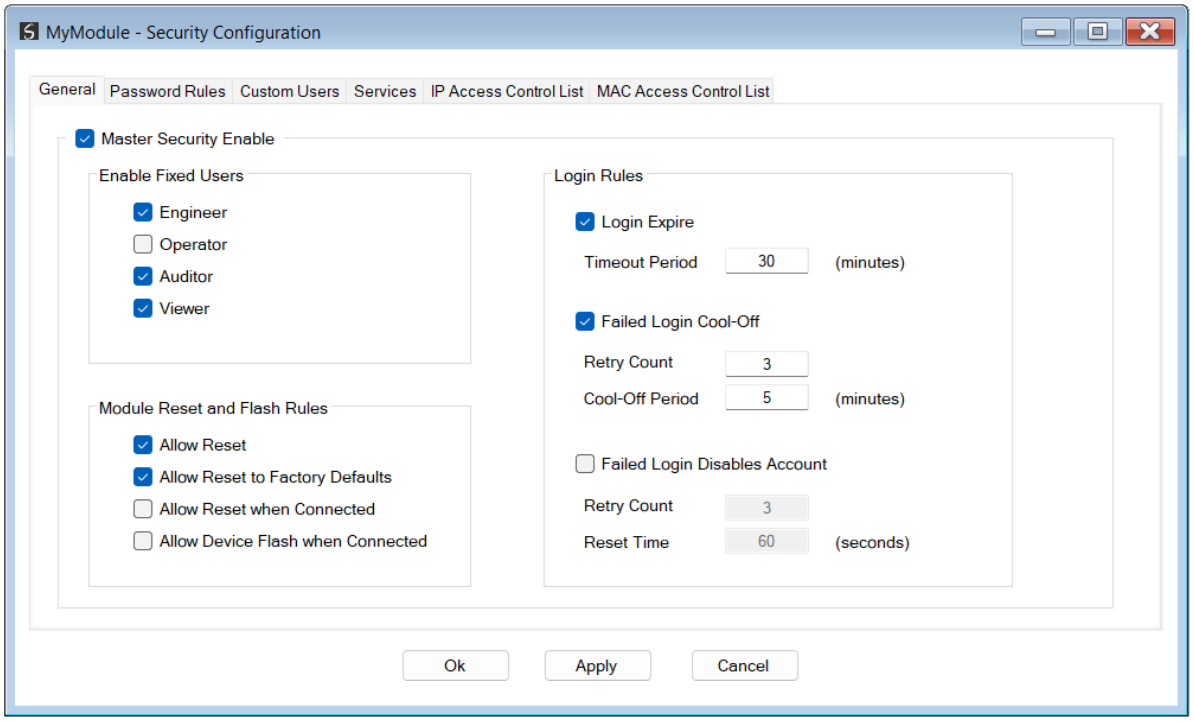


Figure 5.2 – Security Configuration – General.

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 the Retry Count 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 Retry Count	The number of retries before the Login Cool-Off period is applied.
Failed Login Cool-Off Period	The Cool-Off duration, in minutes. Once a user has entered an incorrect password the Retry Count 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 the Retry Count times in succession before the Reset Time has expired. **NOTE**: Once the user's account has been disabled, an Administrator would need to reset their password. **NOTE**: This option will not be applicable for the Administrator user account.
Failed Login Disables Account Retry Count	The number of retries while the Reset Time is active before the user's account is disabled.
Failed Login Disables Account Reset Time	The Reset Time duration, in seconds. Once a user has entered an incorrect password, the Disable Account Reset Time will be applied and starts counting down to zero. If the user enters the incorrect Password the Retry Count times before the Reset Time has expired, the user's account will be disabled. Once the Reset Time has expired (gone back to zero), then the login process can be restarted. Min: 1 second. Max: 65000 seconds (18 hours)

Table 5.2 – Security Configuration – General.

5.1.1.2. PASSWORD RULE CONFIGURATION

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

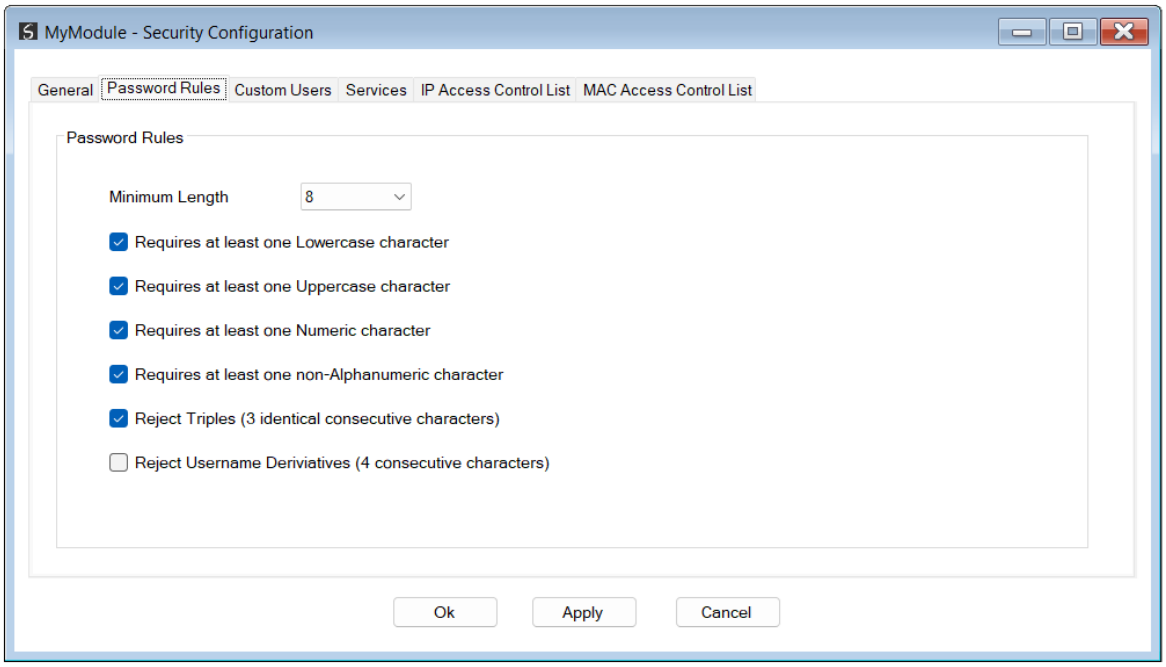


Figure 5.3 – Security Configuration – Password Rules.

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.

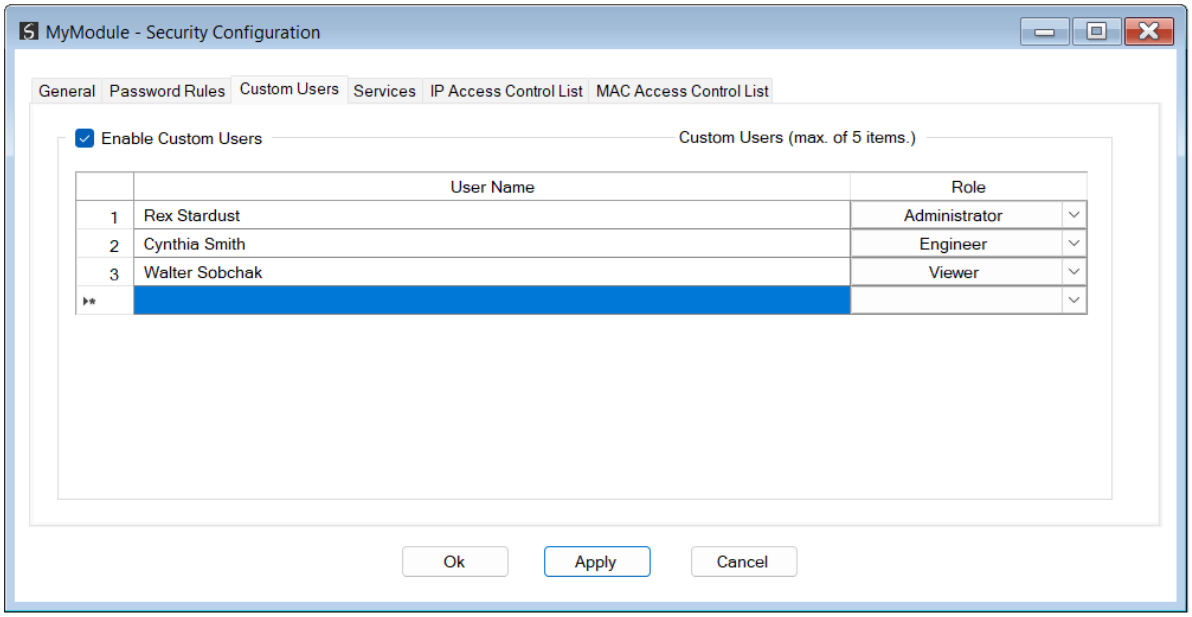


Figure 5.4 – Security Configuration – Custom Users.

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.

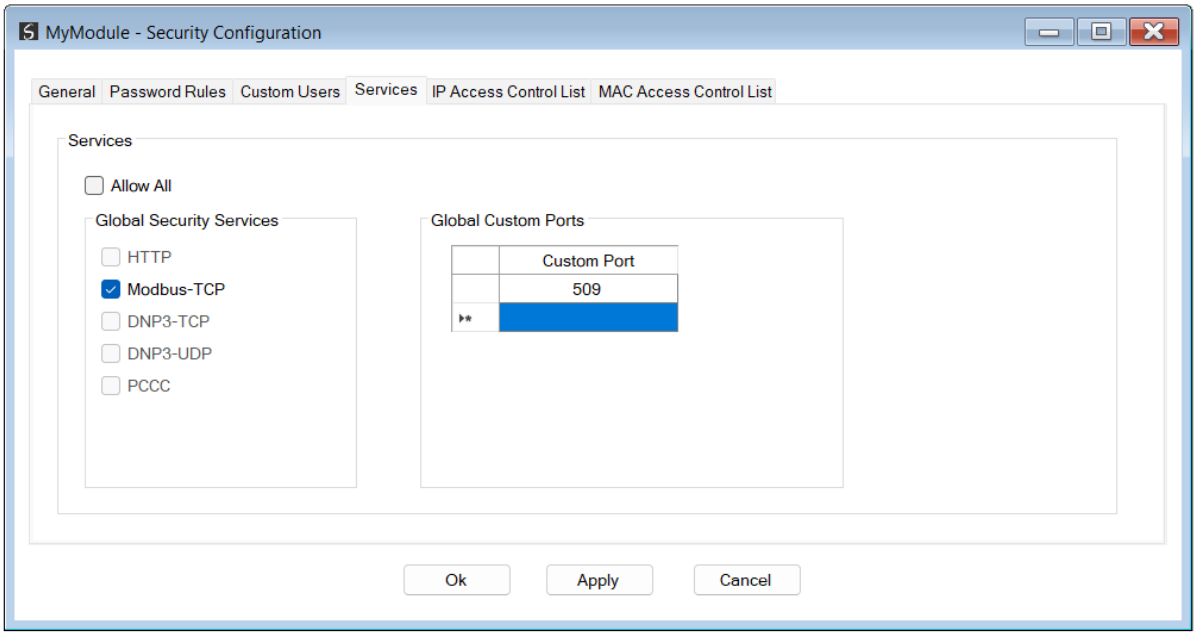


Figure 5.5 – Security Configuration – Services.

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.

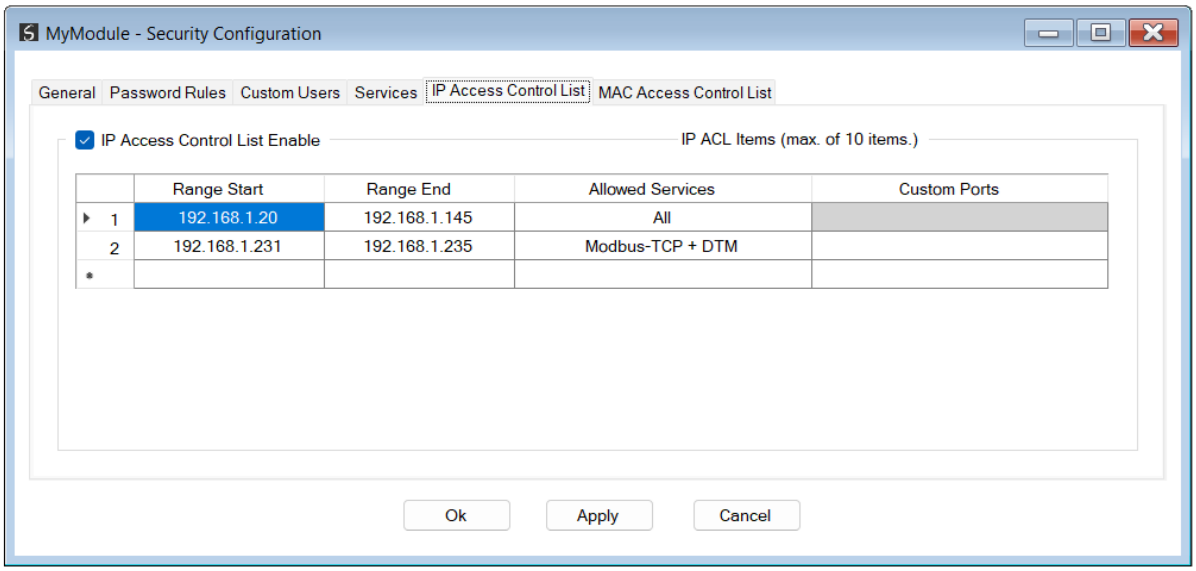


Figure 5.6 – Security Configuration – IP Access Control List.

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.
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.

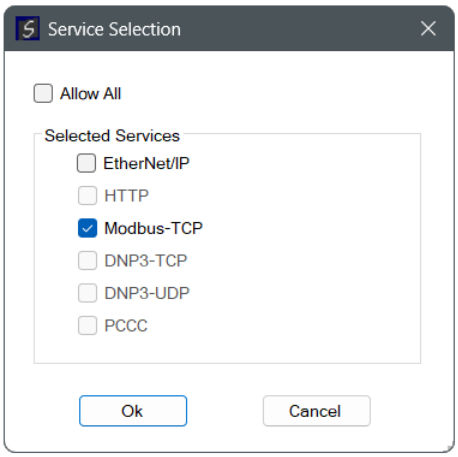


Figure 5.7 – Security Configuration – Service Selection.

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.

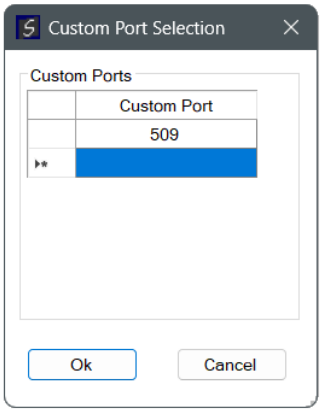


Figure 5.8 – Security Configuration – Custom Ports.

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.

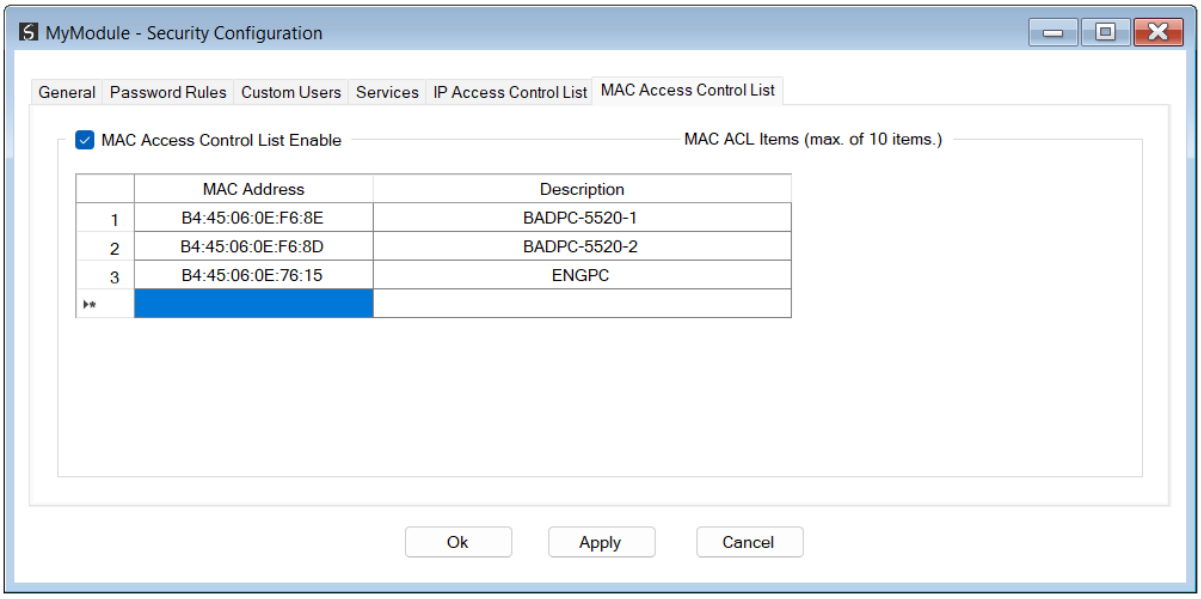


Figure 5.9 – Security Configuration – MAC Access Control List.

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.

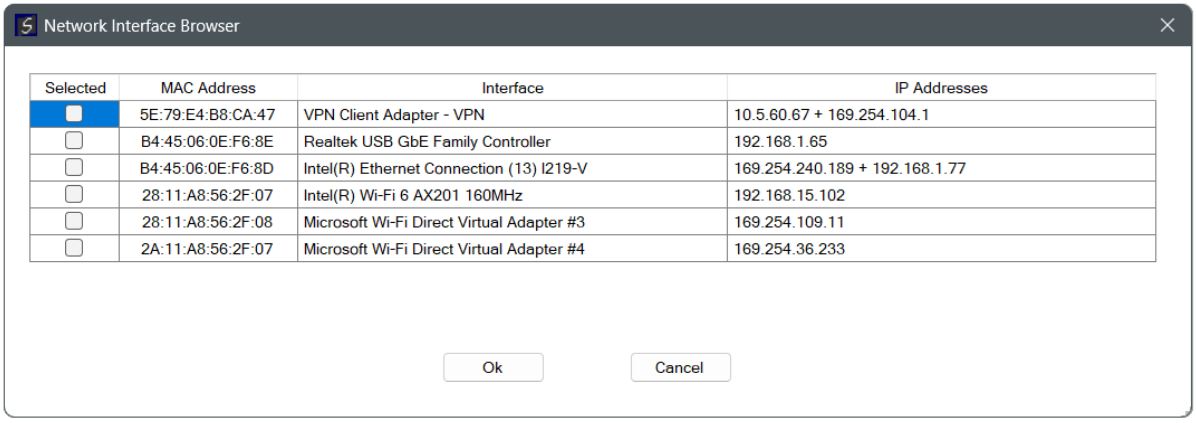


Figure 5.10 – Security Configuration – Network Interface Browser.

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.

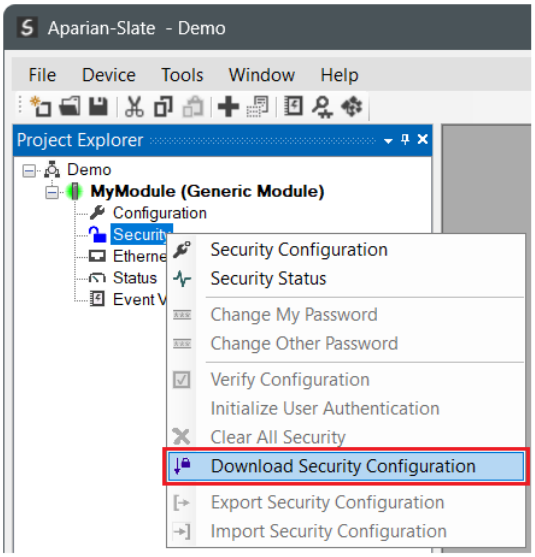


Figure 5.11 – Download Security Configuration.

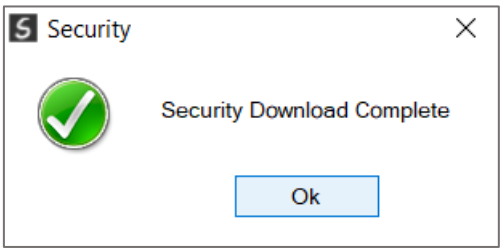


Figure 5.12 – Download Security Configuration Complete.

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.

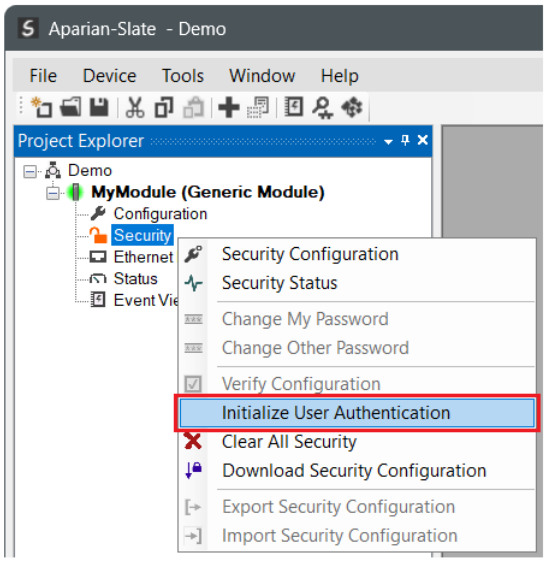


Figure 5.13 – Initialize User Authentication.

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

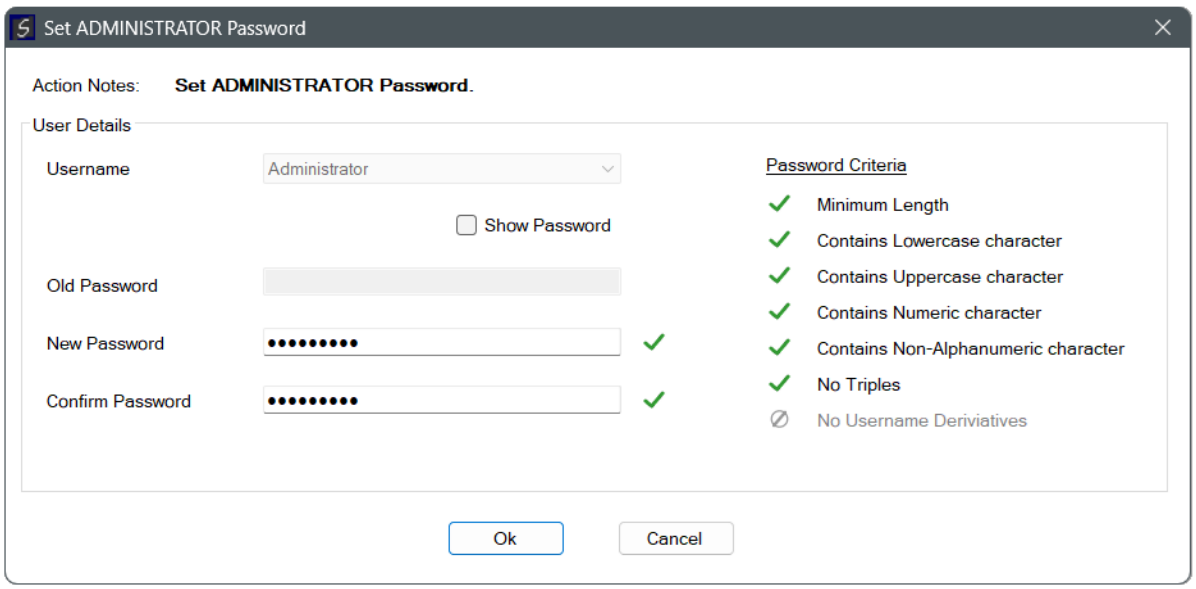


Figure 5.14 – Set Administrator Password.

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.

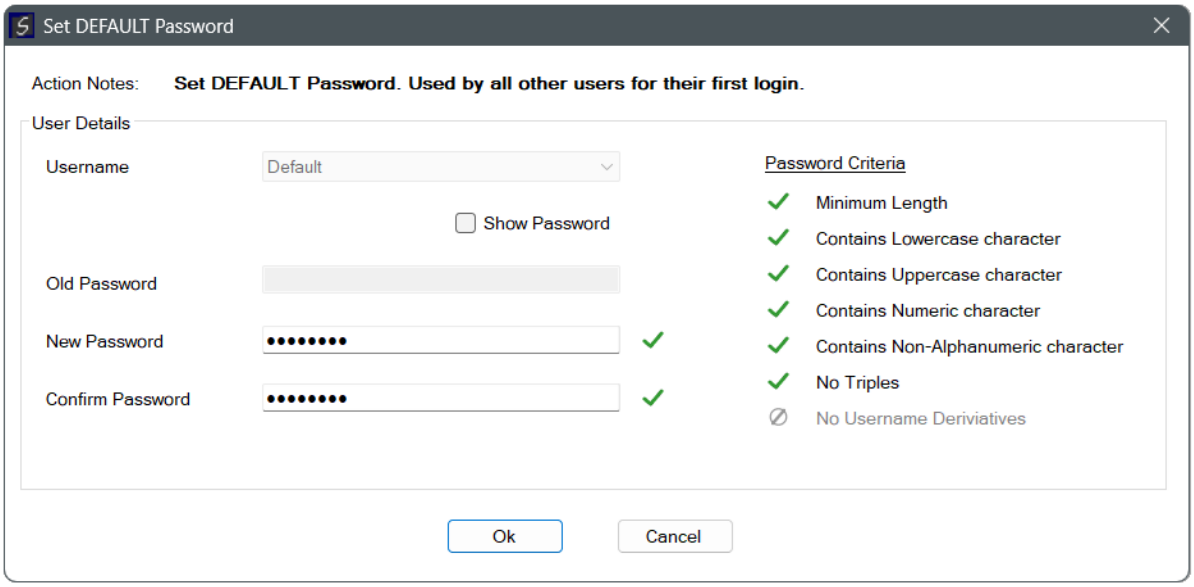


Figure 5.15 – Set Default Password.



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.

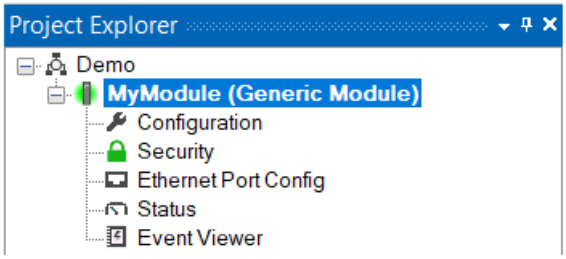


Figure 5.16 – Secured Module.

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***.

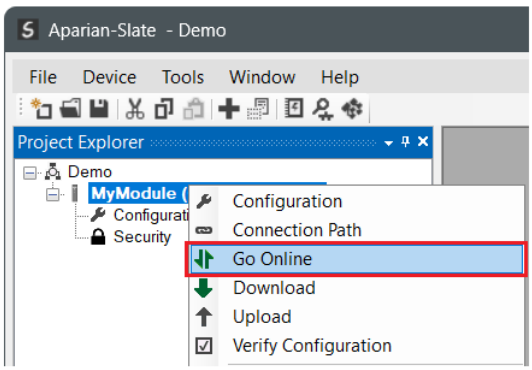


Figure 5.17 – Go Online.

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

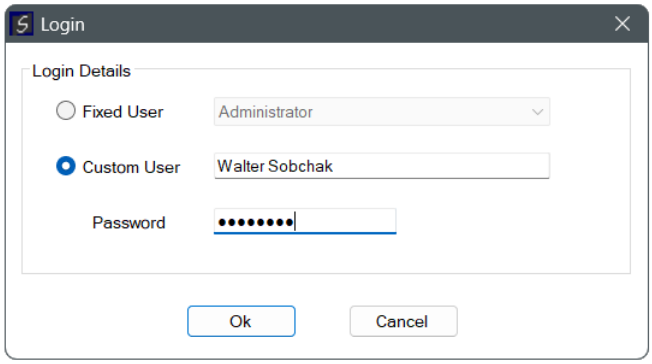


Figure 5.18 – Custom User Login.

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.

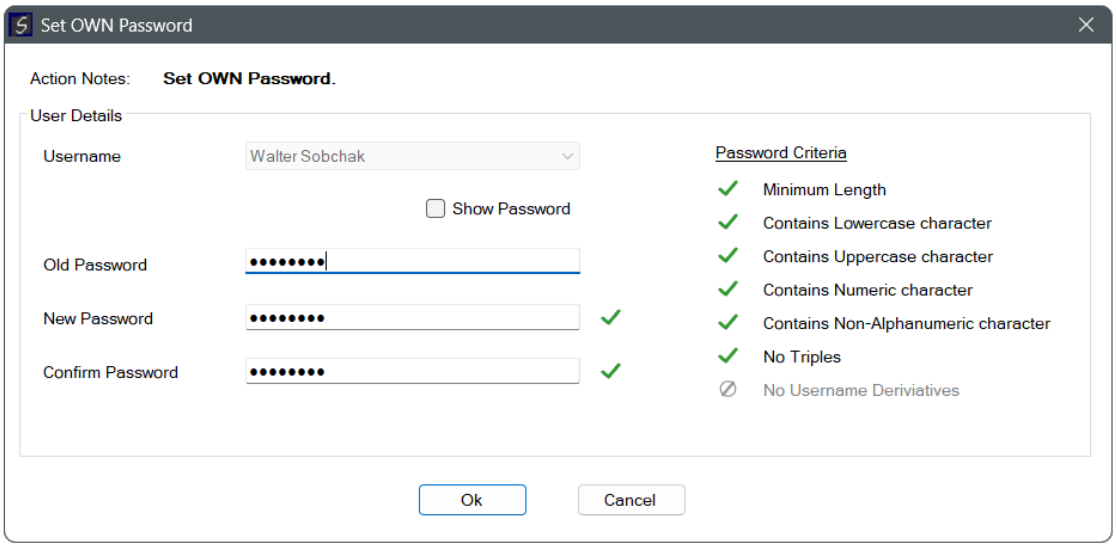


Figure 5.19 – Change Own Password.

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.

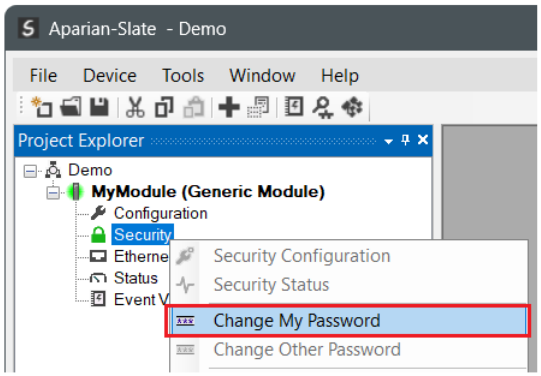


Figure 5.20 – Select Change My Password.

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.

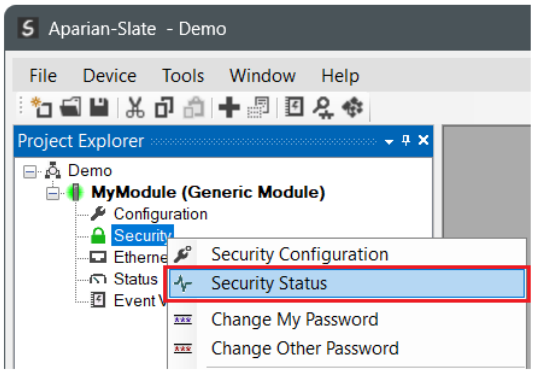


Figure 5.21 – Select Security Status.

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.

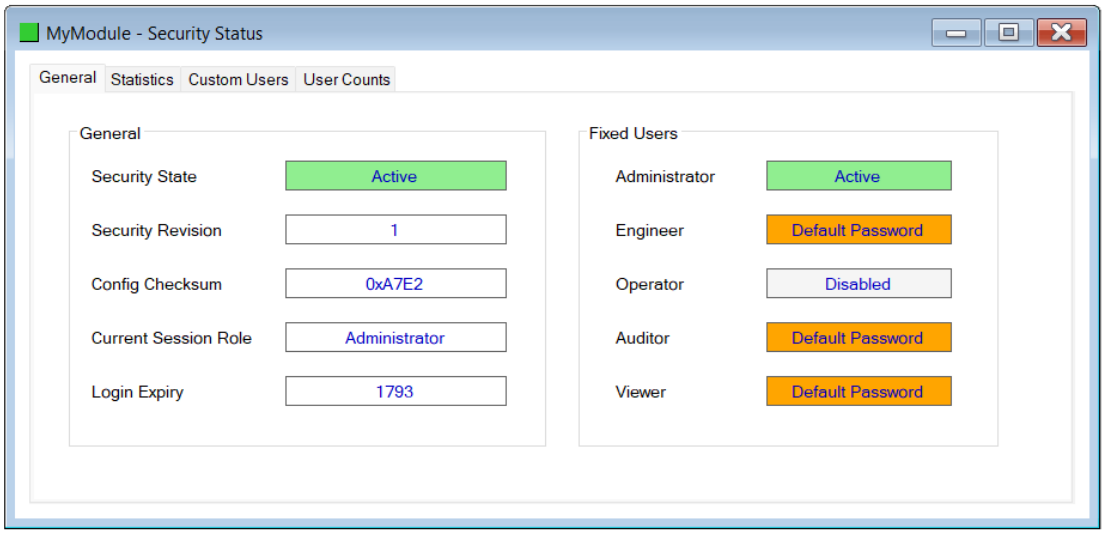


Figure 5.22 – Security Status – General.

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.

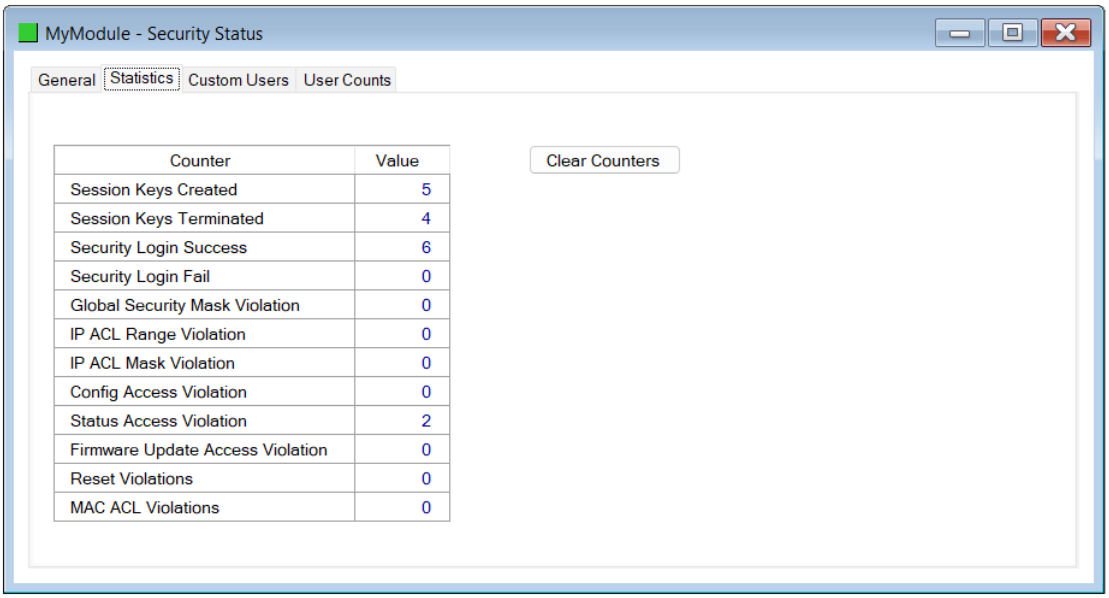


Figure 5.23 – Security Status – Statistics.

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:

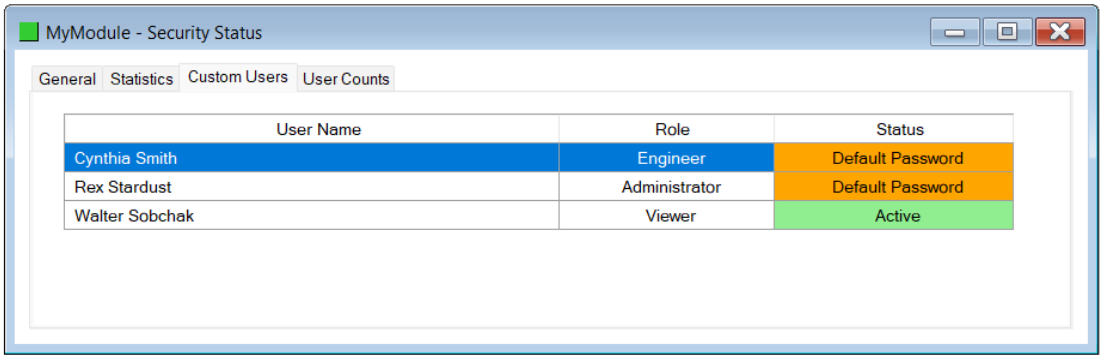


Figure 5.24 – Security Status – Custom Users.

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.

The User Counts status tab displays the following:

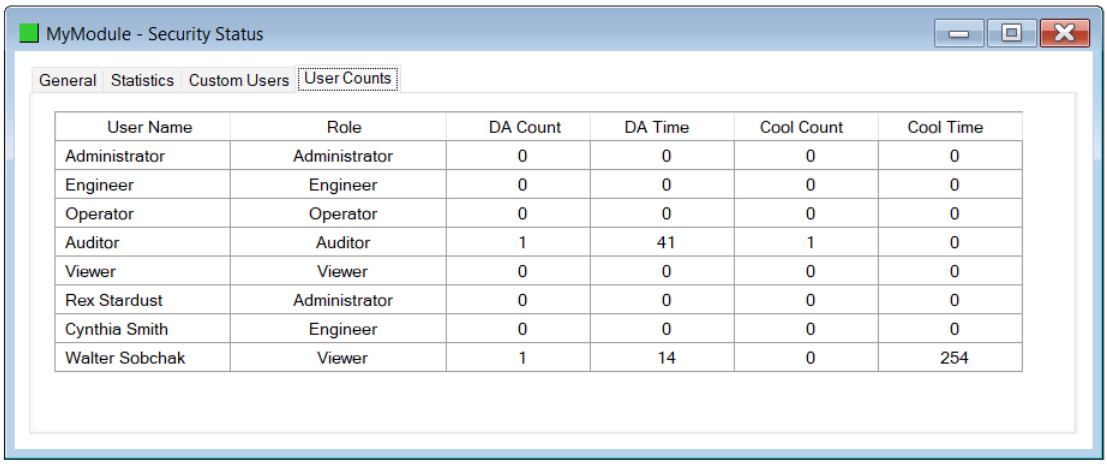


Figure 5.25 – Security Status – User Counts

The following statistics are displayed for each user:

Parameter	Description
User Name	The configured user name.
Role	The configured role for this specific user.
DA Count	The number of failed login occurrences with respect to Disabling the account.

DA Time	The time (seconds) since the last failed login occurrence with respect to Disabling the account.
Cool Count	The number of failed login occurrences with respect to account Cool Down period.
Cool Time	The time (seconds) remaining for the account cool down period.

Table 5.13 – Security Status – User Count

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.

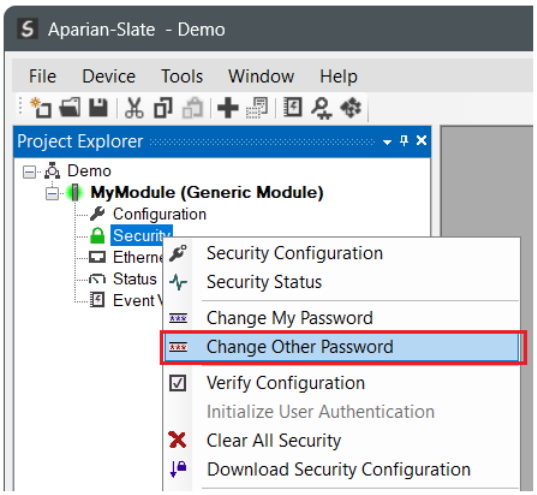


Figure 5.26 – Select Change Other Password.

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.

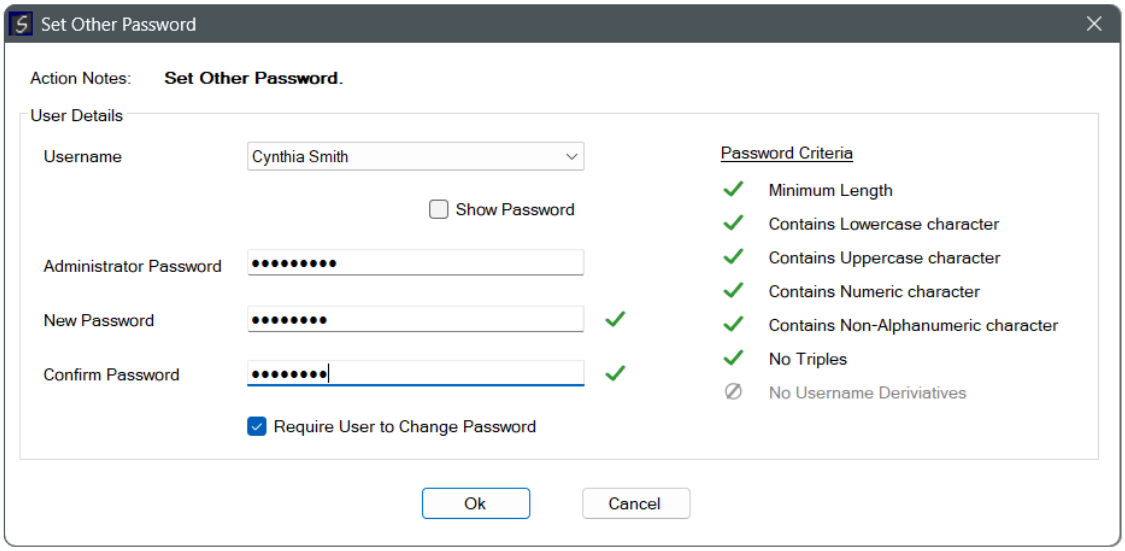


Figure 5.27 – Set Other Password.

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.

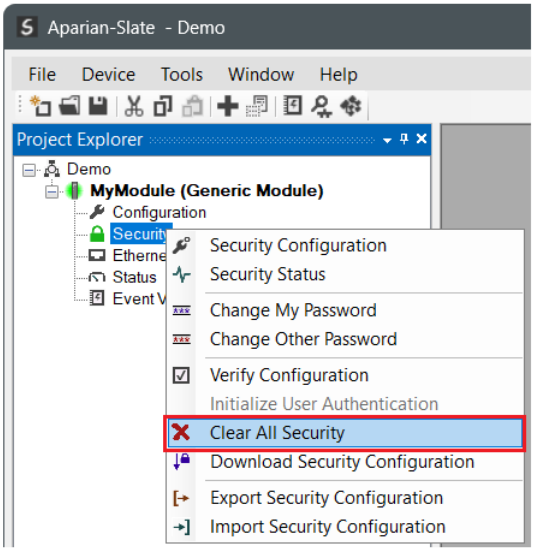


Figure 5.28 – Select Clear All Security.

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.

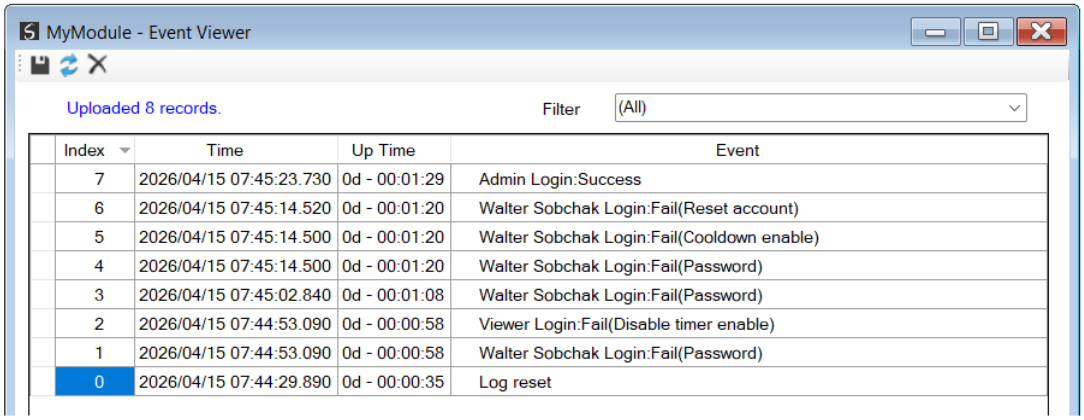


Figure 5.29 – Event Log.

6. OPERATION

6.1. ENHANCED MODE

In Enhanced Mode the A-MBIPB can be configured in any of the four combinations:

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

Table 6.1 – Enhanced Mode Operating Combinations.

6.1.1. EIP TARGET – MODBUS SERVER

When the A-MBIPB 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 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 A-MBIPB, 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 A-MBIPB with an EtherNet/IP Class 1 connection.

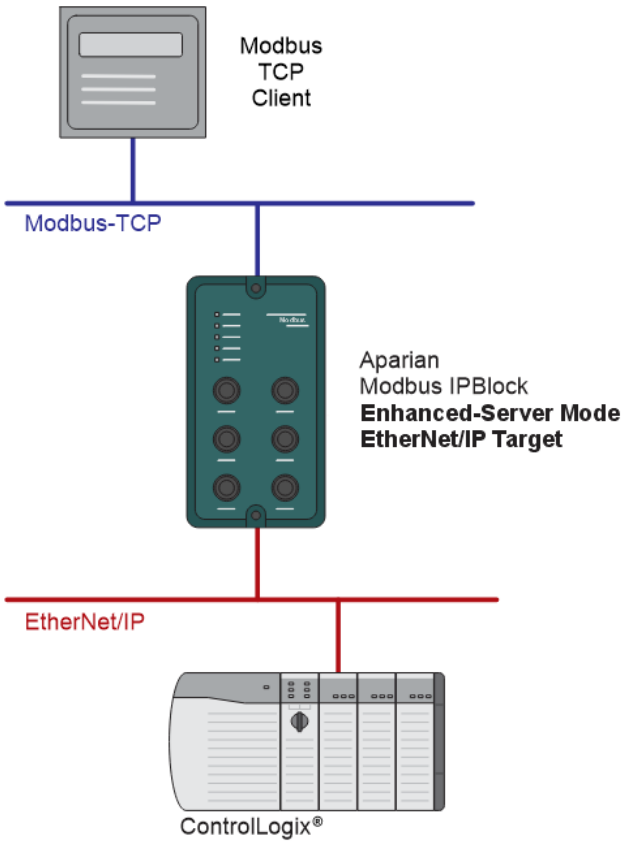


Figure 6.1 – Modbus TCP Client to Logix Assembly operation.

For this example, the remote Modbus TCP Client will write data to Modbus Holding Register 4000 in the A-MBIPB. The A-MBIPB will map the received Modbus data (at Holding Register 4000) to the input assembly of the A-MBIPB 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).

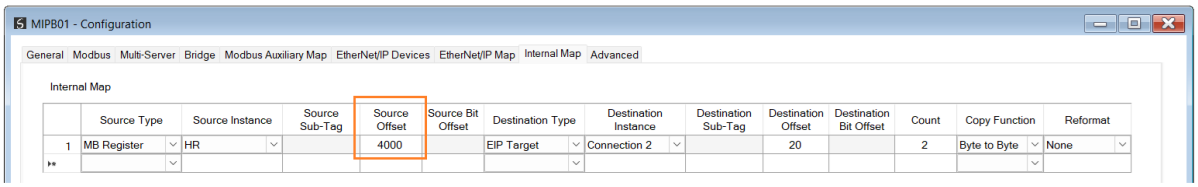


Figure 6.2 – Internal Mapping from Modbus Register to Logix input assembly,

6.1.2. EIP Originator – Modbus Server

When the A-MBIPB 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 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 A-MBIPB, 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.

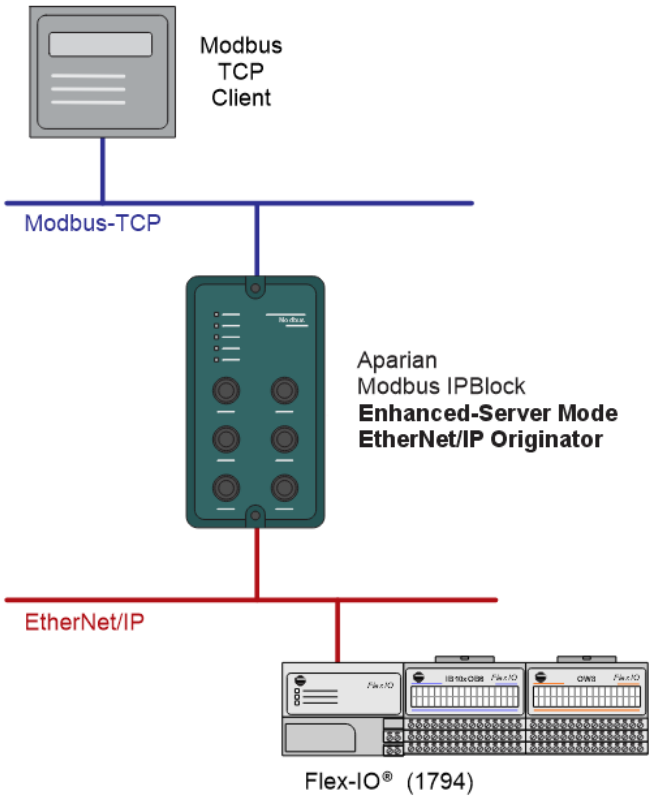


Figure 6.3 – Modbus TCP Client to EtherNet/IP IO operation,

For this example, the remote Modbus TCP Client will write data to Modbus Holding Register 2000 in the A-MBIPB. The A-MBIPB 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 A-MBIPB.

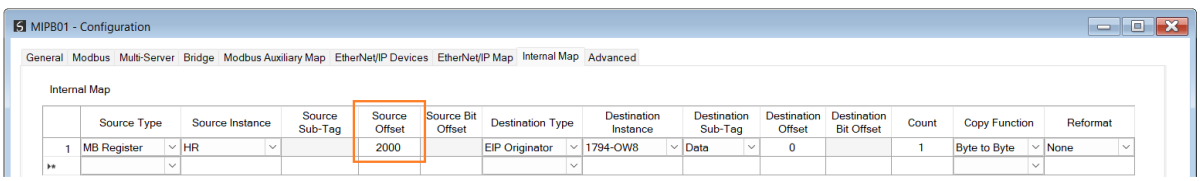


Figure 6.4 – Internal Mapping from Modbus Register to EtherNet/IP IO.

6.1.3. EIP TARGET – MODBUS CLIENT

When the A-MBIPB 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 or RTU485.

In the example below the A-MBIPB (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.

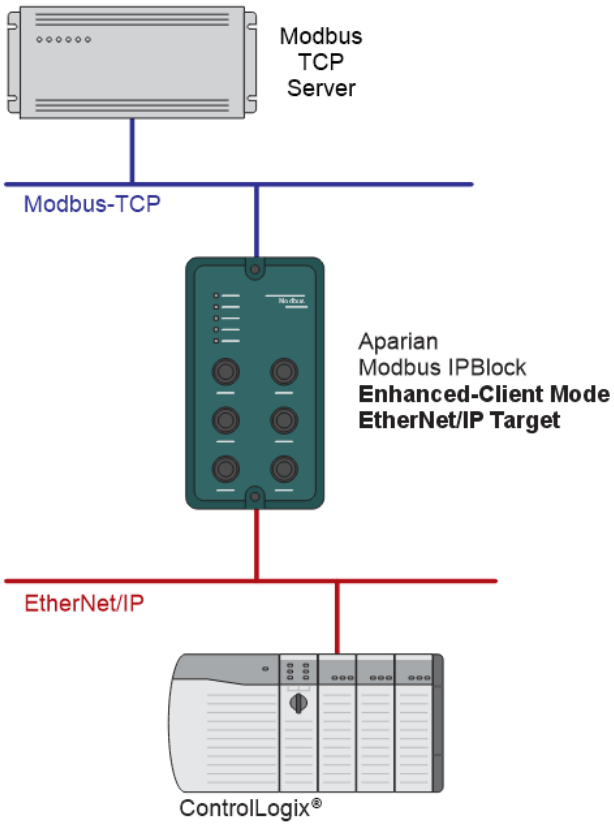


Figure 6.5 – Modbus Client to EtherNet/IP operation.

For this example, the user will configure the Modbus Auxiliary Map to read data from a Modbus Server device on Ethernet. The A-MBIPB 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.

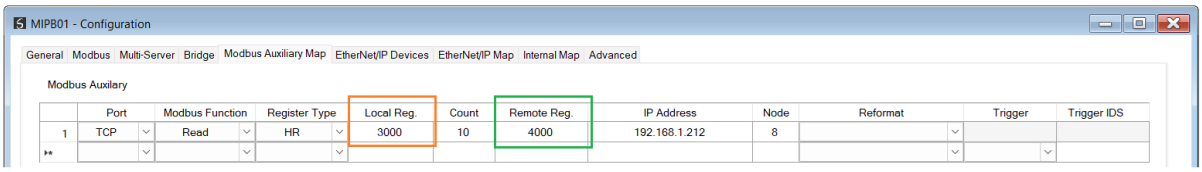


Figure 6.6 – Modbus Client Aux Mapping.

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.

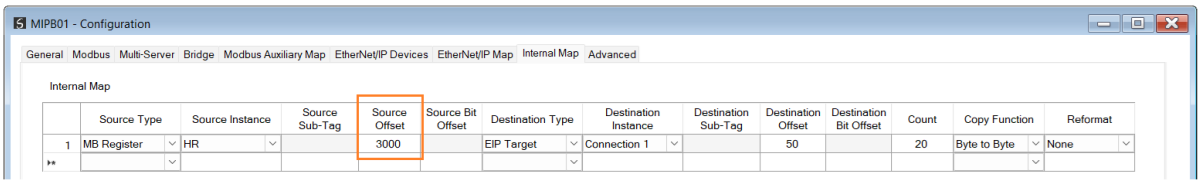


Figure 6.7 – Internal Mapping from Modbus Register to EtherNet/IP.

6.1.4. EIP ORIGINATOR – MODBUS CLIENT

When the A-MBIPB 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 or RTU485.

In the example below the A-MBIPB (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.

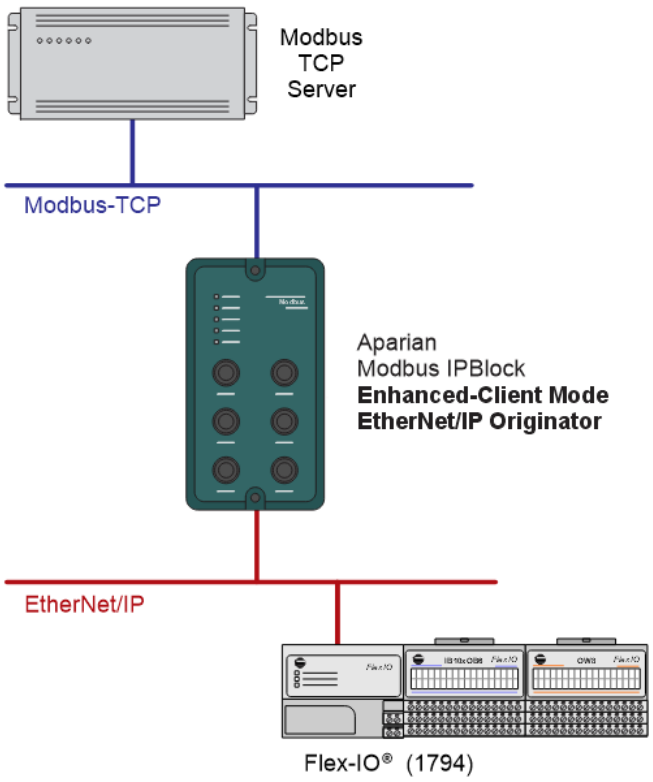


Figure 6.8 – Modbus Client to EtherNet/IP operation.

For this example, the user will configure the Modbus Auxiliary Map to read data from a Modbus Server device on Ethernet. The A-MBIPB 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.

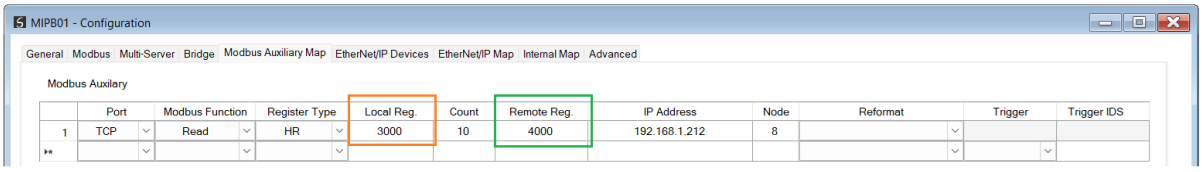


Figure 6.9 – Modbus Client Aux Mapping.

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.

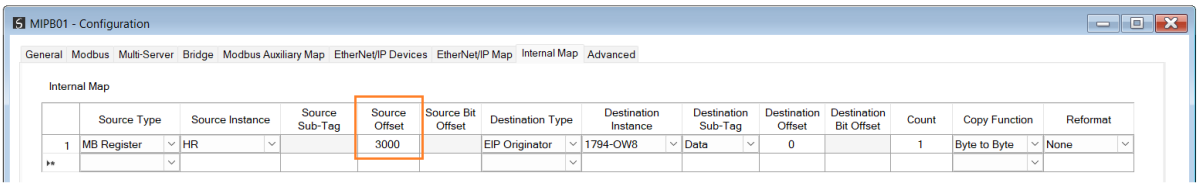


Figure 6.10 – Internal Mapping from Modbus Register to EtherNet/IP.

6.2. BRIDGE MODE

In Bridge Mode, the A-MBIPB can operate as a transparent link for Modbus between different media (TCP and RTU485). This allows, for example, a Modbus client on Modbus TCP to communicate with Modbus devices on RS485 and vice versa.

In the example below, any Modbus request received on the Ethernet TCP port for a Modbus Server at Node address 5 (to the IP address of the A-MBIPB 192.168.1.100), from the Modbus Client at IP 192.168.1.50, will be routed to the RS485-A 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).

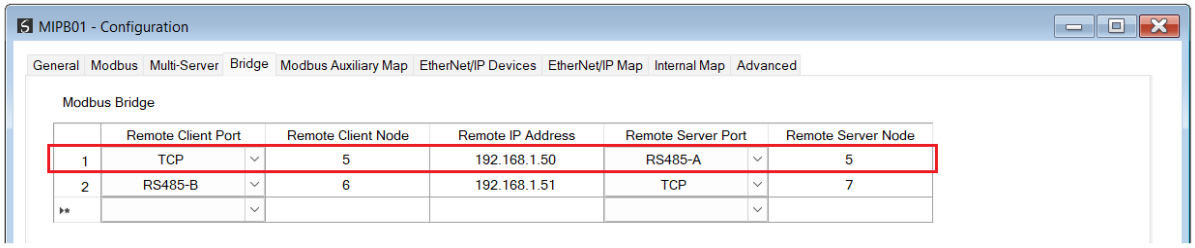


Figure 6.11 – Bridge Mode Map – TCP to RS485-A.

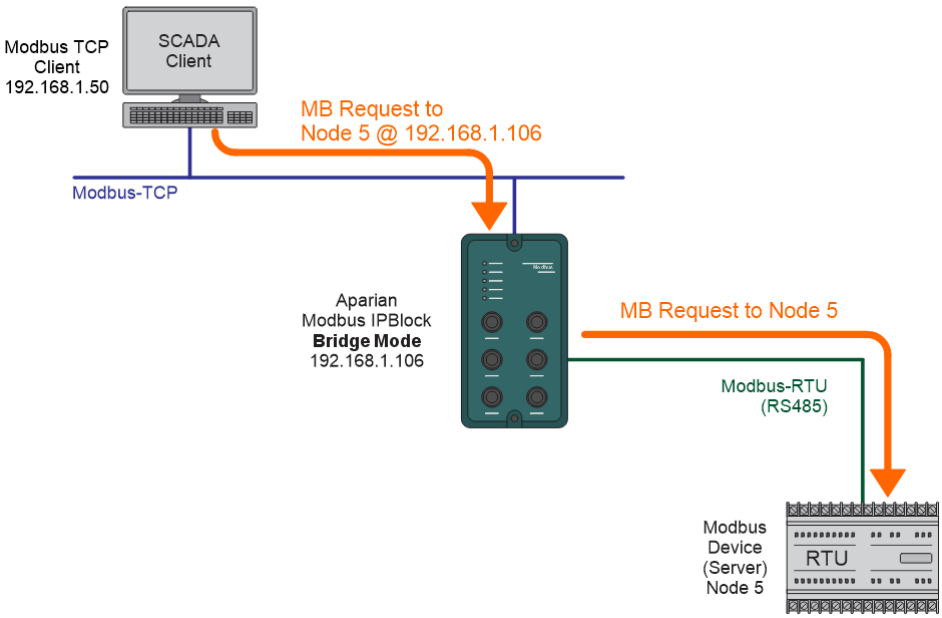


Figure 6.12 – Bridge Mode Diagram – TCP to RS485-A.

In the next example below, any Modbus request received on the RS485-B 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-B).

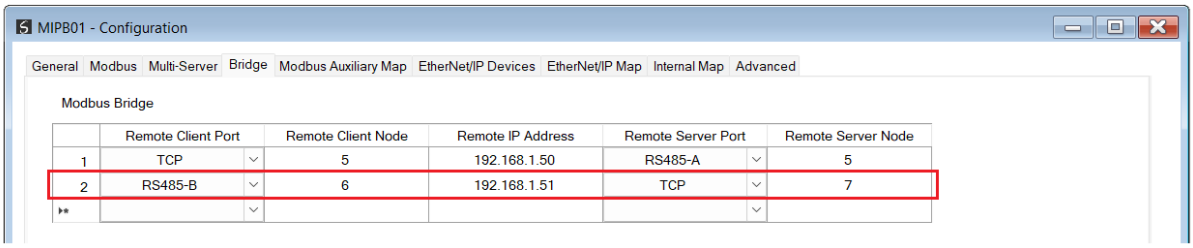


Figure 6.13 – Bridge Mode Map – RS485-B to TCP.

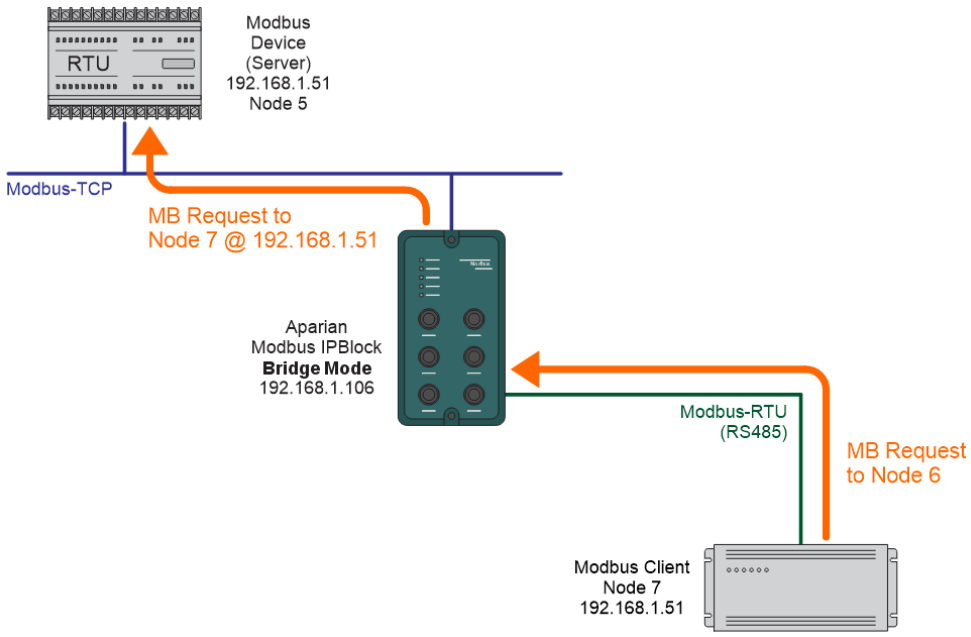


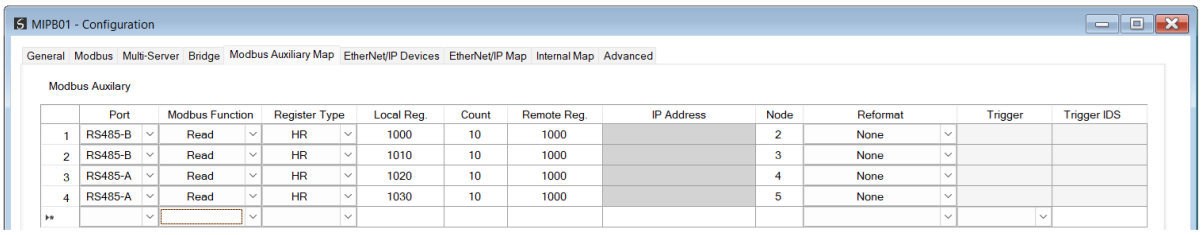
Figure 6.14 – Bridge Mode Diagram – RS485-B to TCP.

6.3. CONCENTRATOR MODE

In Concentrator Mode, the A-MBIPB will collect data from various Modbus devices and provide the consolidated data to one or more Modbus Clients. The A-MBIPB 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 A-MBIPB.

In the example below the Modbus SCADA is communicating on Modbus TCP to the A-MBIPB which is serving as a proxy for the Modbus RTU devices in the field. The A-MBIPB will collect data from the Modbus RTU devices via two separate RS485 (Modbus RTU) networks and save it in its internal Modbus Registers. Modbus HR 1000 will be read from each node and placed at 10 intervals in the A-MBIPB internal MB registers (e.g., data from node 2 will be at HR1000, data from node 3 will be at HR1010, data from node 4 will be at HR11020, etc.).

The Modbus SCADA will then be able to read that data from the A-MBIPB at the specific Modbus Registers in a single read. (HR1000..HR1039)



	Port	Modbus Function	Register Type	Local Reg.	Count	Remote Reg.	IP Address	Node	Reformat	Trigger	Trigger IDS
1	RS485-B	Read	HR	1000	10	1000		2	None		
2	RS485-B	Read	HR	1010	10	1000		3	None		
3	RS485-A	Read	HR	1020	10	1000		4	None		
4	RS485-A	Read	HR	1030	10	1000		5	None		

Figure 6.15 – Concentrator Mode – Modbus Auxiliary Mapping.

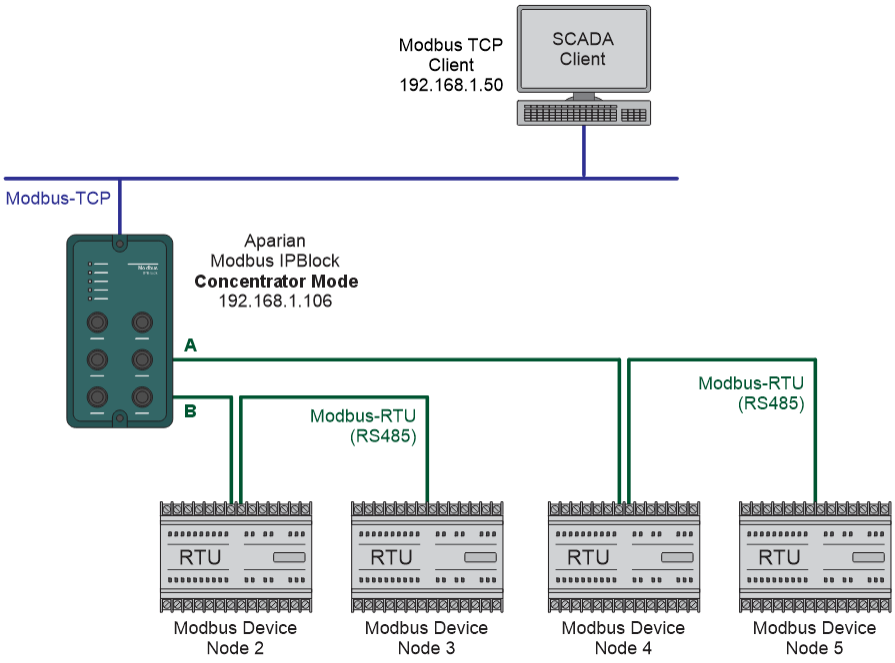


Figure 6.16 – Concentrator Mode Diagram.

6.4. 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.

6.4.1. SYSTEM STATUS

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

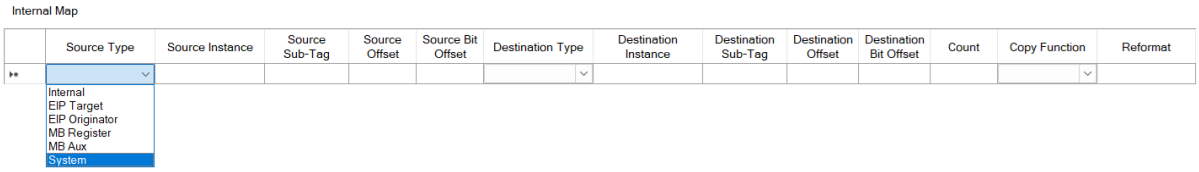


Figure 6.17 – IDS Copy – System Information.

The module's system information is described in Table 3.19 – System Information Format..

6.4.2. 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 A-MBIPB is operating in Enhanced mode as a Modbus Client.

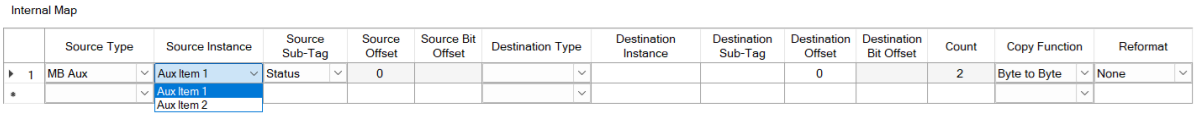


Figure 6.18 – IDS Copy - Modbus Register Source Instance.

The MB Aux format is as shown below:

Parameter	Data Type	Description
MB Aux Status	SINT	Bit 0 – Last Successful (0=Failed, 1=Success) Bit 1 – Waiting (0=Not waiting, 1= Waiting for Trigger) Bit 2 – Not Started (0=Started, 1=Not Started)
MB Aux Count	USINT	Successful transaction count (rolls-over at 255)

Table 6.2 – Modbus Aux Item Status

6.4.3. ETHERNET/IP IO DEVICE STATUS

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

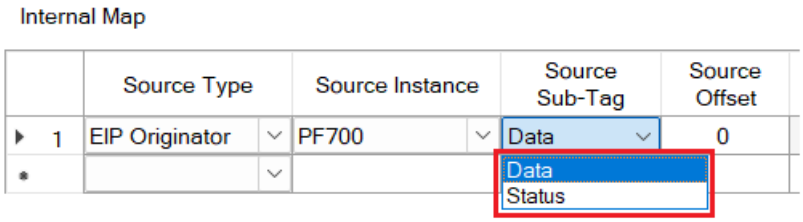


Figure 6.19 – IDS Copy – EtherNet/IP Originator Status.

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 6.3 – EtherNet/IP Originator Connection Status.

7. DIAGNOSTICS

7.1. LEDS

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

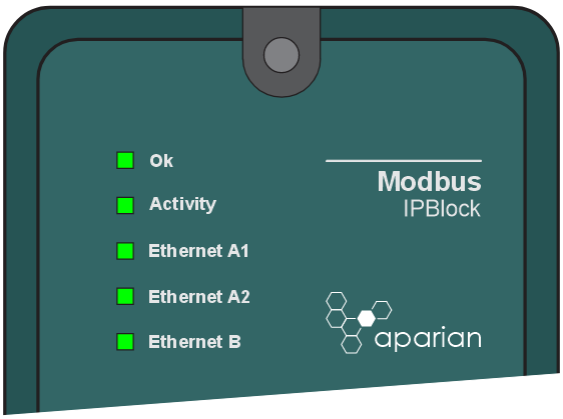


Figure 7.1 - Modbus IPBlock front view.

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.
Ethernet A1 / A2 / 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 three Ethernet ports A1, A2 and B. Each LEDs represents each specific port.

Activity	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. Checksum error).

Table 7.1 - Module LED operation.

7.2. MODULE STATUS MONITORING IN SLATE

The Modbus IPBlock 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.

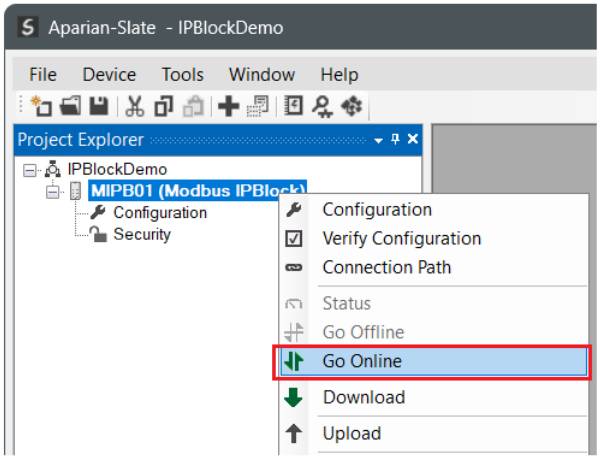


Figure 7.2 - Selecting to Go Online.

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

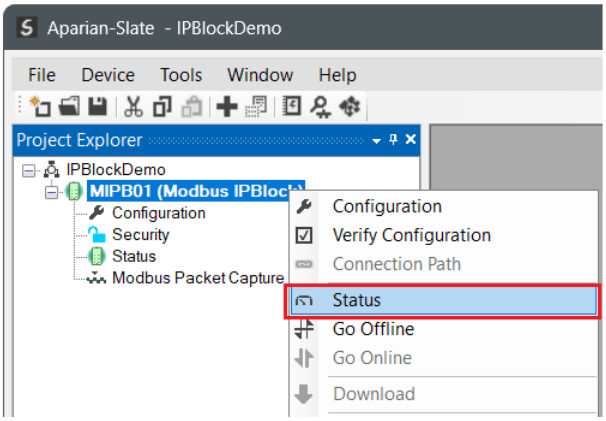


Figure 7.3 - Selecting online Status.

The Status monitoring window can be opened by either double-clicking on the *Status* item in the Project Explorer tree, or by right-clicking on the module and selecting *Status*.

The status window contains multiple tabs to display the current status of the module.

7.2.1. GENERAL

The General tab displays the general status for the local A-MBIPB module.

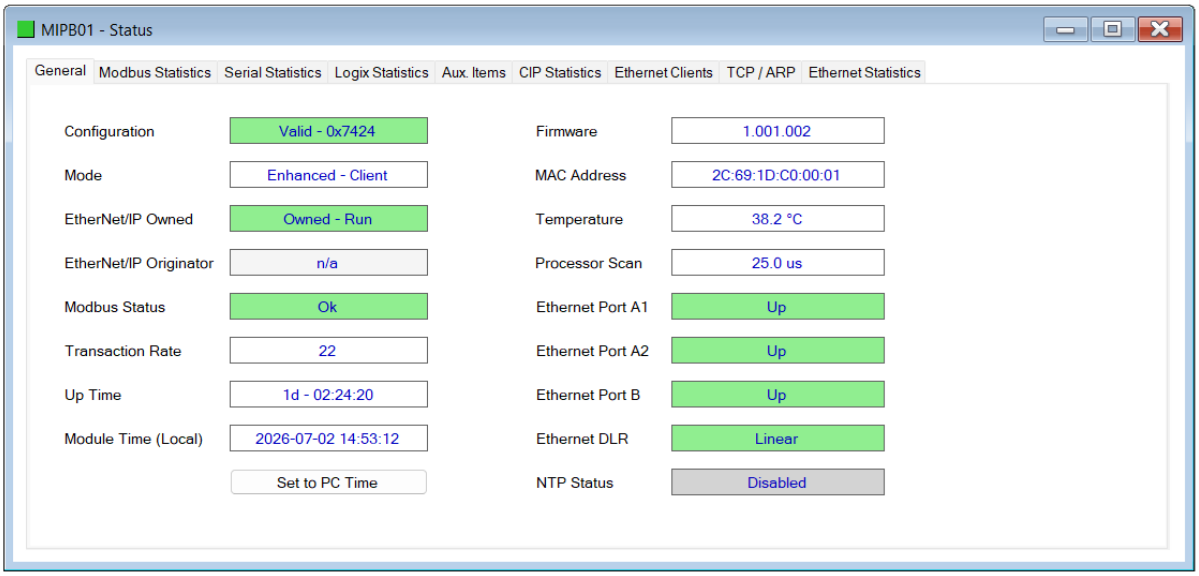


Figure 7.4 - Status monitoring – General.

The General tab displays the following general parameters:

Parameter	Description
Configuration	Indicates whether the current configuration is valid or not, as well as the configuration checksum.
Mode	Indicates the current operating mode: 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	When the module is configured as an EtherNet/IP Originator, this will show if all the Class 1 and Explicit Message connections to EtherNet/IP target devices are established and returning valid data.
Modbus Status	When the module is operating as a **Modbus 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. If Modbus communication has been disabled by the EtherNet/IP controller, then it will show Disabled.
Transaction Rate	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's CPU.
Processor Scan	The amount of time (microseconds) taken by the module's processor in the last scan.
Ethernet Port A1 / A2 / B	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.

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

7.2.2. MODBUS STATISTICS

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

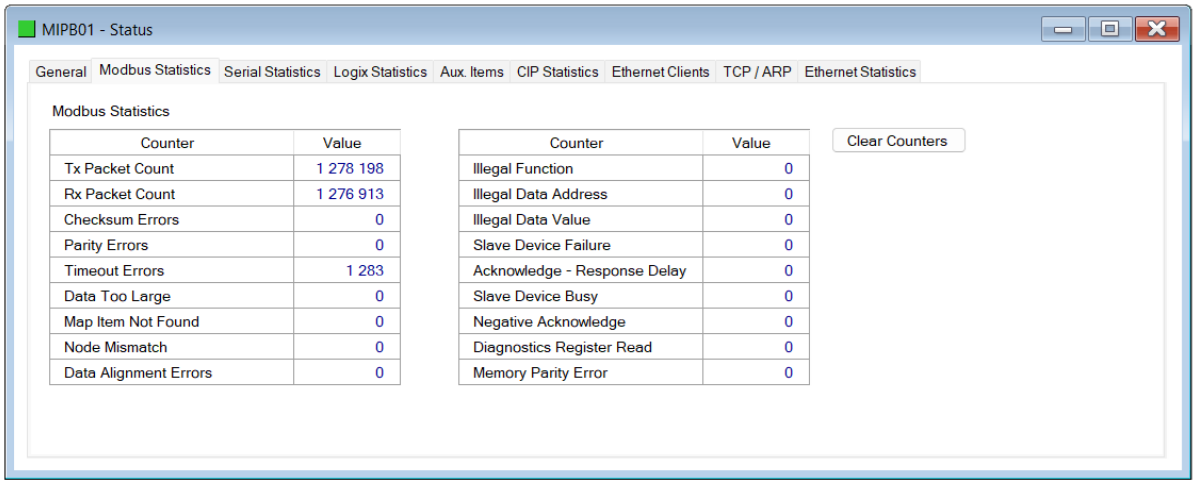


Figure 7.5 - Status monitoring – Modbus Statistics.

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 7.3 – Modbus statistics.

7.2.3. SERIAL STATISTICS

The Serial Statistics tab displays the Serial statistics for the Modbus Read and Write Message Exchanges when the module is a Modbus Server or Modbus Client.

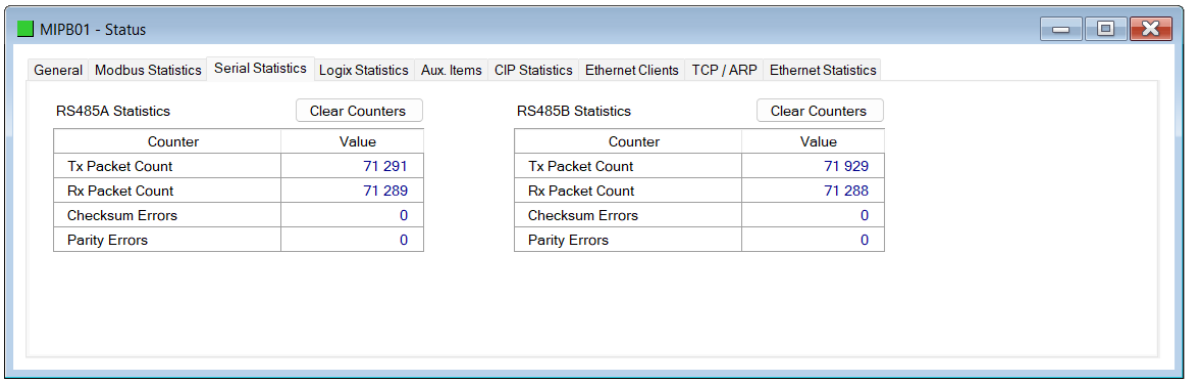


Figure 7.6. - Status monitoring – Serial Statistics

The Serial Statistics tab displays the following parameters:

Statistic	Description
RS485-A	
Tx Packet Count	The number of RS485 (port A) packets sent by the module.
Rx Packet Count	The number of RS485 (port A) packets received by the module.
Checksum errors	The number of corrupted RS485 (port A) packets received by the module.
Parity errors	The number of RS485 (port A) bytes with parity errors received by the module.
RS485-B	
Tx Packet Count	The number of RS485 (port B) packets sent by the module.
Rx Packet Count	The number of RS485 (port B) packets received by the module.
Checksum errors	The number of corrupted RS485 (port B) packets received by the module.
Parity errors	The number of RS485 (port B) bytes with parity errors received by the module.

Table 7.4 - Serial Statistics Tab

7.2.4. LOGIX STATISTICS

The following Logix statistics are only relevant when the module is running in EtherNet/IP Originator mode.

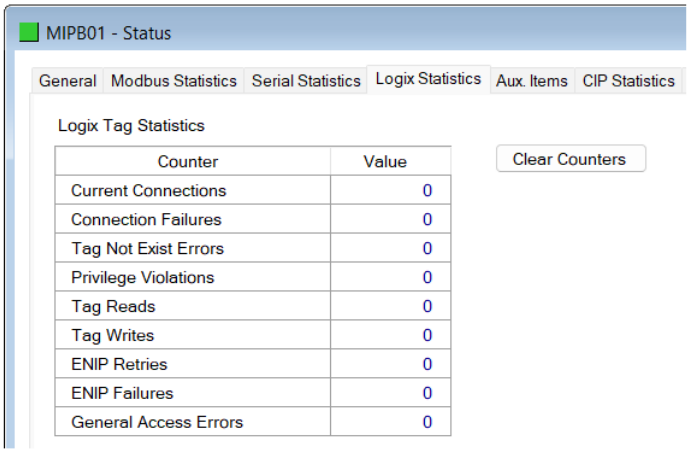


Figure 7.7 - Status monitoring – Logix Statistics.

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 IPBlock module.
Tag Writes	The number of tag write transactions executed by the Modbus IPBlock 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 7.5 - Tag Mapping statistics.

7.2.5. 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**. If the transaction shows **Waiting**, then it indicates that this mapped item has a trigger and the mapped item has not yet been executed. This provides quick visual feedback as to which items are currently active.

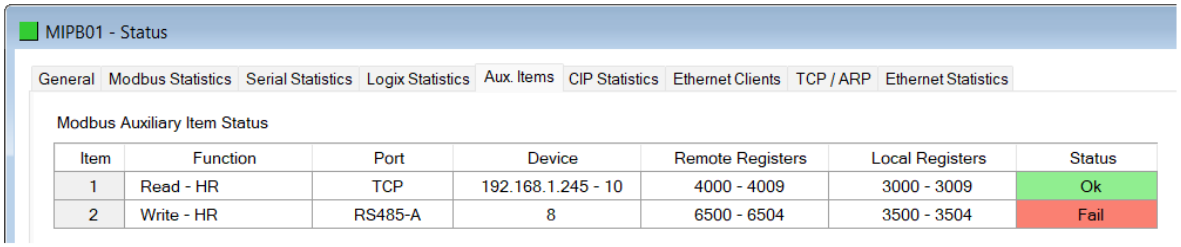


Figure 7.8 - Auxiliary Item status.

7.2.6. CIP STATISTICS

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

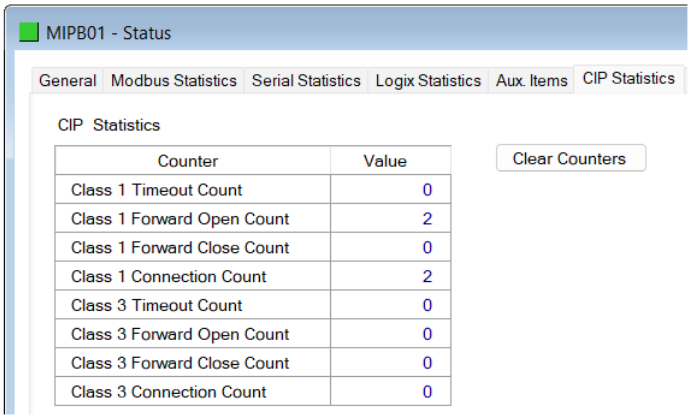


Figure 7.9 - Status monitoring – CIP Statistics.

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 7.6 – CIP statistics.

7.2.7. 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.

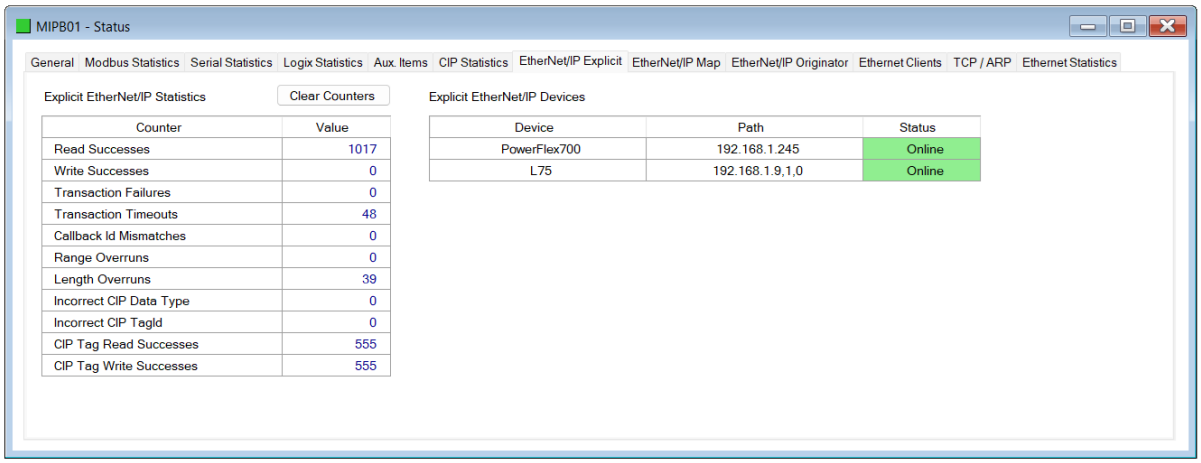


Figure 7.10 - Status monitoring – EtherNet/IP Explicit.

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 7.7 – EtherNet/IP Explicit Statistics.

7.2.8. 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.

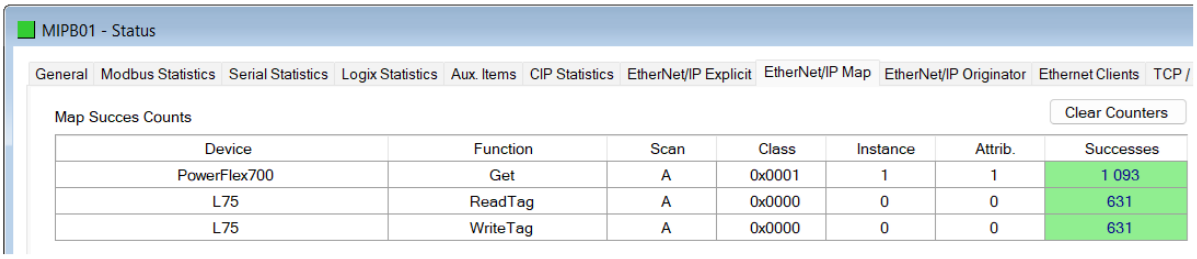


Figure 7.11 - Status monitoring – EtherNet/IP Map.

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

7.2.9. 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.

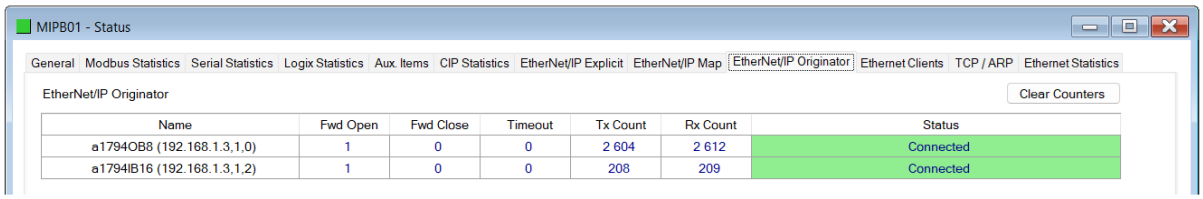


Figure 7.12 - Status monitoring – EtherNet/IP Originator.

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 7.8 – EtherNet/IP Class 1 status and statistics

7.2.10. ETHERNET CLIENTS

The Ethernet Clients tab displays details of the Ethernet and EtherNet/IP clients connected to the A-MBIPB.

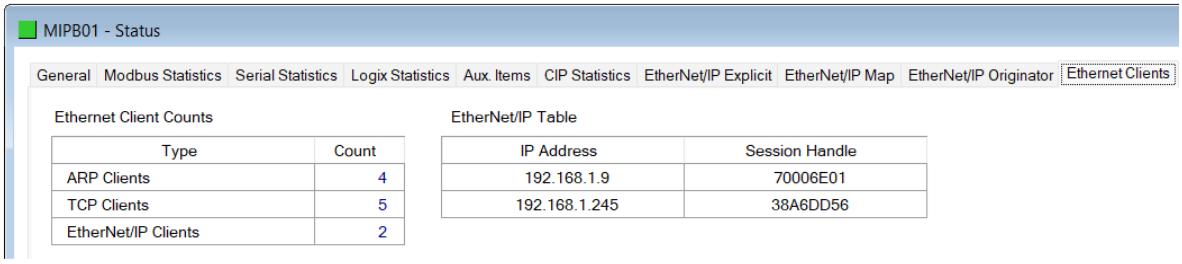


Figure 7.13 –Status monitoring – Ethernet Client Statistics.

7.2.11. TCP / ARP

The TCP/ARP tab displays details of the internal Ethernet ARP and TCP lists of the A-MBIPB.

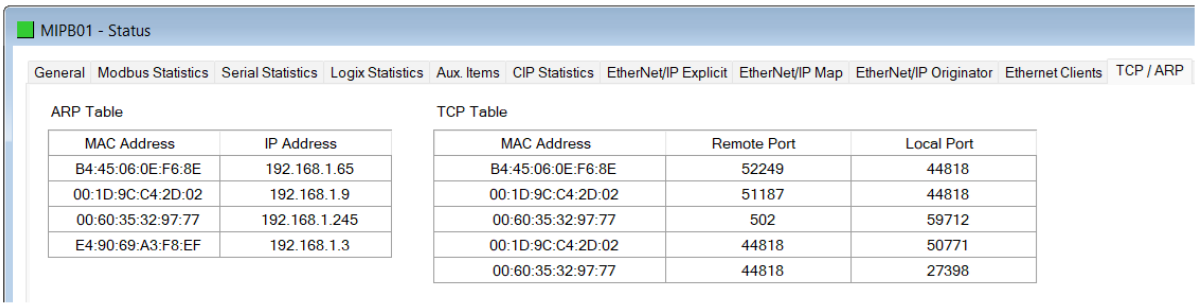


Figure 7.14 – Status monitoring – Ethernet TCP / ARP Statistics.

7.3. TARGET DEVICE STATUS MONITORING IN SLATE

The A-MBIPB 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.

7.3.1. ETHERNET/IP

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

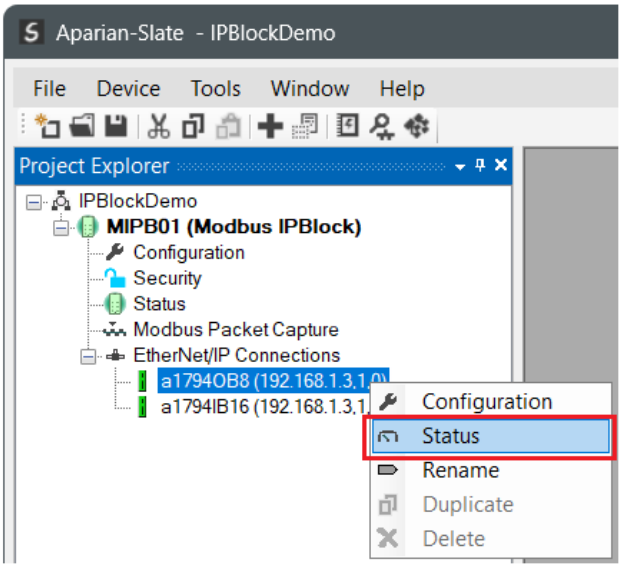


Figure 7.15 – EtherNet/IP Device Status – Status selection.

7.3.1.1. GENERAL

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

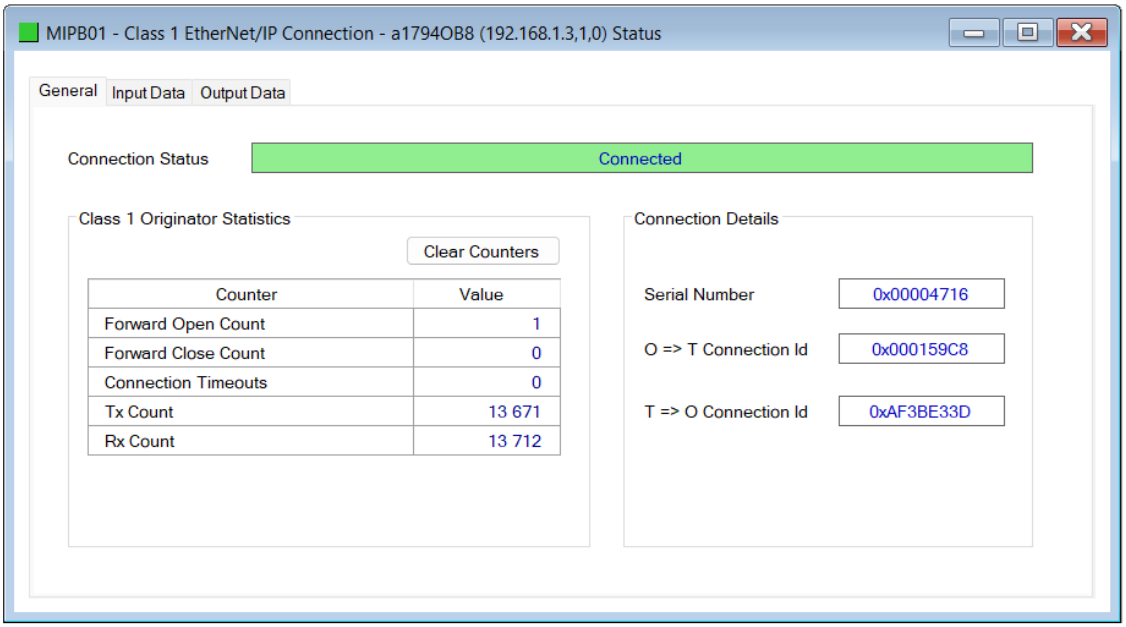


Figure 7.16 – EtherNet/IP Device Status – General Status.

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 7.9 – EtherNet/IP Class 1 Device status and statistics .

7.3.1.2. INPUT DATA

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

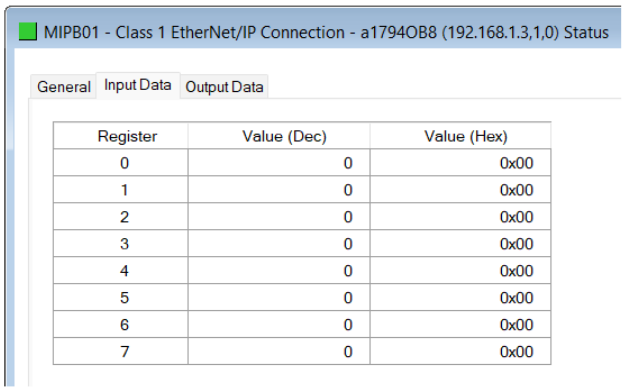


Figure 7.17 – EtherNet/IP Device Status – Input Data.

7.3.1.3. OUTPUT DATA

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

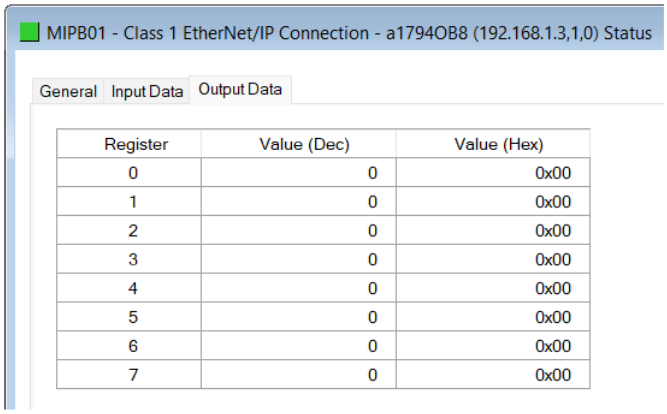


Figure 7.18 – EtherNet/IP Device Status – Output Data.

7.4. MODBUS DIAGNOSTICS

The Modbus IPBlock supports Modbus diagnostics (function code 8) which can be read from the Modbus Client. This functionality will only be available when the Modbus IPBlock is running in Modbus Server 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 7.10 - Tag Mapping statistics.

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

7.5. 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.

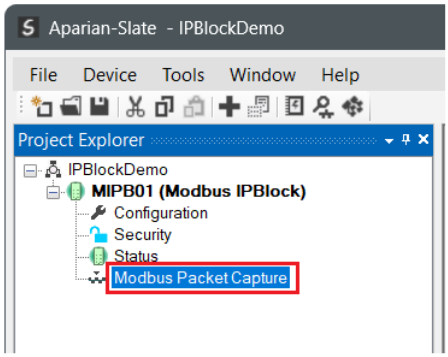


Figure 7.19 - Selecting Modbus Packet Capture.

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

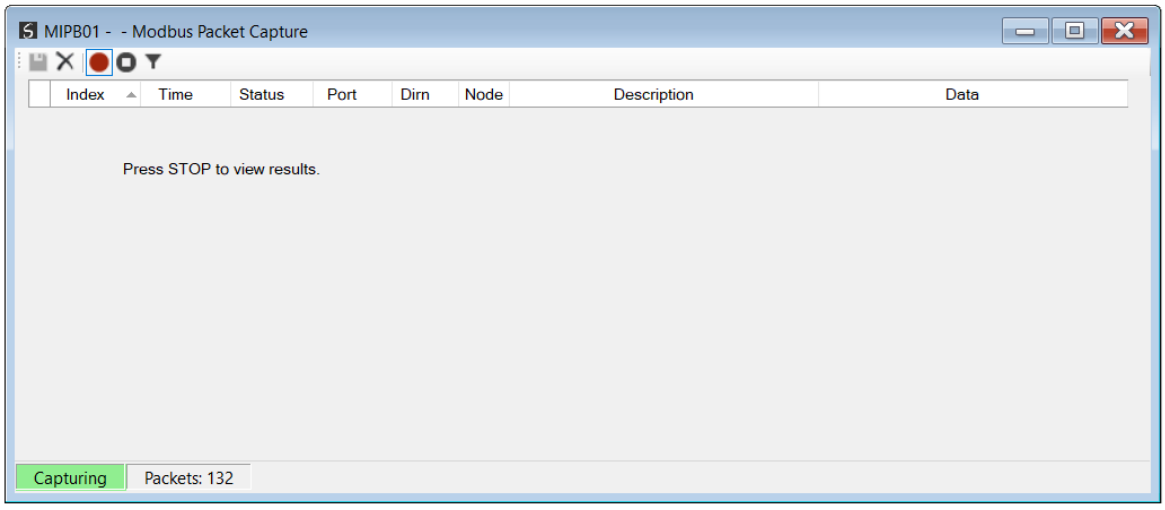


Figure 7.20 – Modbus packet capture.

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

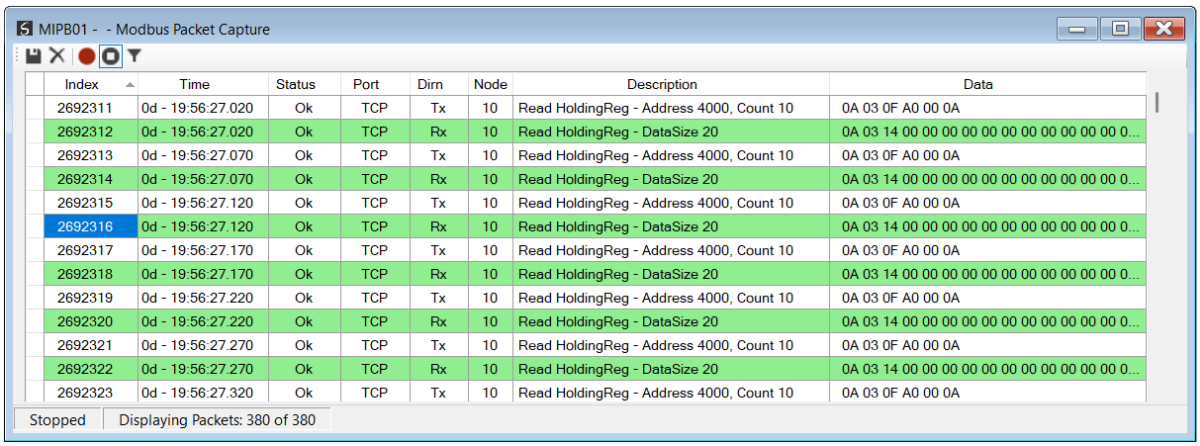


Figure 7.21 – Modbus Packet Capture complete.

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, RTU485-A, RTU485-B)
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 7.11 – Modbus Packet Capture fields.

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

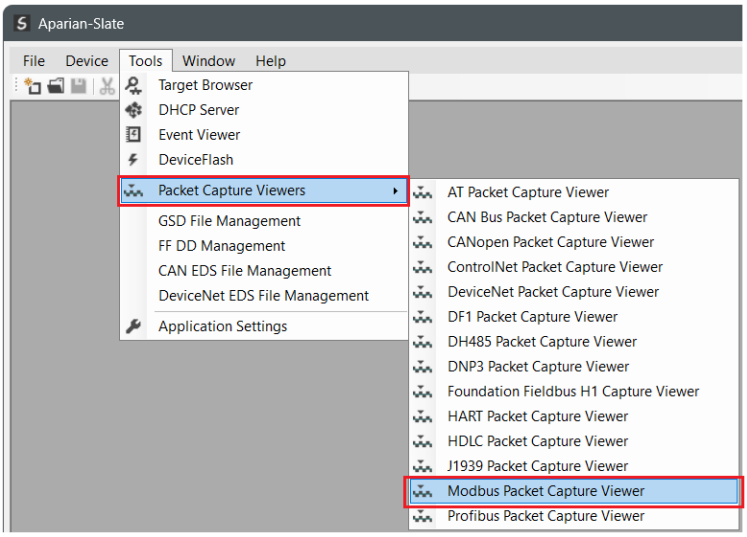


Figure 7.22 - Selecting the Modbus Packet Capture Viewer.

7.6. MODULE EVENT LOG

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

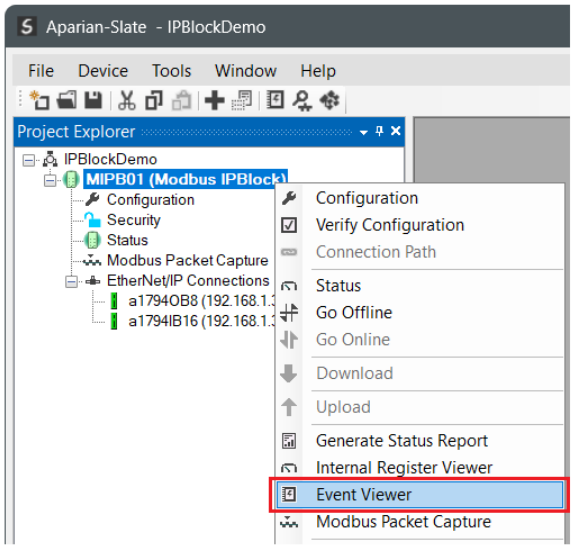
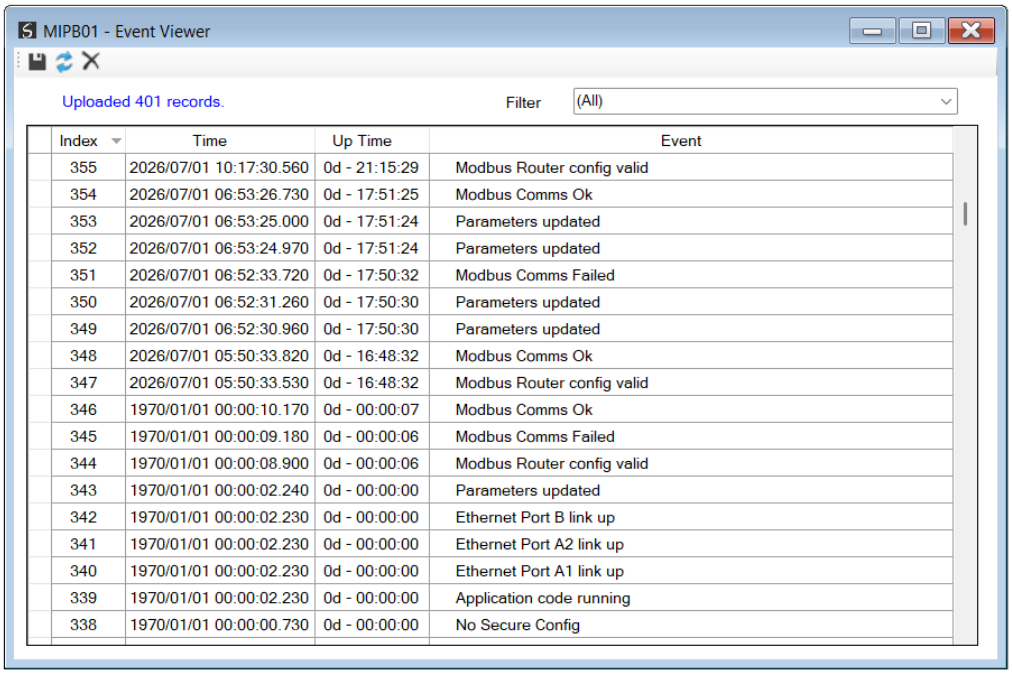


Figure 7.23 - Selecting the module Event Log.

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
355	2026/07/01 10:17:30.560	0d - 21:15:29	Modbus Router config valid
354	2026/07/01 06:53:26.730	0d - 17:51:25	Modbus Comms Ok
353	2026/07/01 06:53:25.000	0d - 17:51:24	Parameters updated
352	2026/07/01 06:53:24.970	0d - 17:51:24	Parameters updated
351	2026/07/01 06:52:33.720	0d - 17:50:32	Modbus Comms Failed
350	2026/07/01 06:52:31.260	0d - 17:50:30	Parameters updated
349	2026/07/01 06:52:30.960	0d - 17:50:30	Parameters updated
348	2026/07/01 05:50:33.820	0d - 16:48:32	Modbus Comms Ok
347	2026/07/01 05:50:33.530	0d - 16:48:32	Modbus Router config valid
346	1970/01/01 00:00:10.170	0d - 00:00:07	Modbus Comms Ok
345	1970/01/01 00:00:09.180	0d - 00:00:06	Modbus Comms Failed
344	1970/01/01 00:00:08.900	0d - 00:00:06	Modbus Router config valid
343	1970/01/01 00:00:02.240	0d - 00:00:00	Parameters updated
342	1970/01/01 00:00:02.230	0d - 00:00:00	Ethernet Port B link up
341	1970/01/01 00:00:02.230	0d - 00:00:00	Ethernet Port A2 link up
340	1970/01/01 00:00:02.230	0d - 00:00:00	Ethernet Port A1 link up
339	1970/01/01 00:00:02.230	0d - 00:00:00	Application code running
338	1970/01/01 00:00:00.730	0d - 00:00:00	No Secure Config

Figure 7.24 – Module Event Log.

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

7.7. 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.

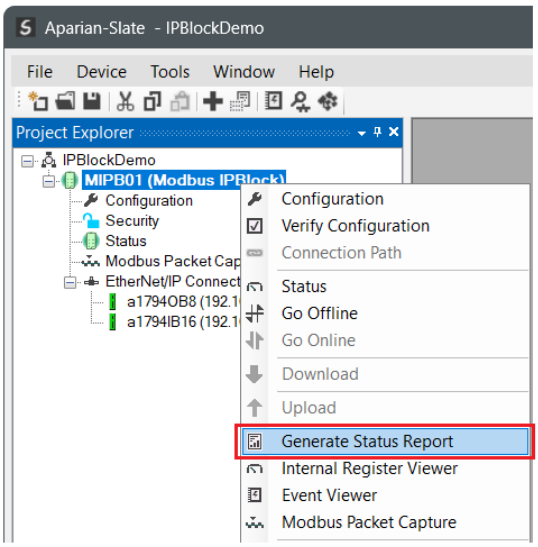


Figure 7.25 – Module Status Report.

7.8. INTERNAL REGISTER VIEWER

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

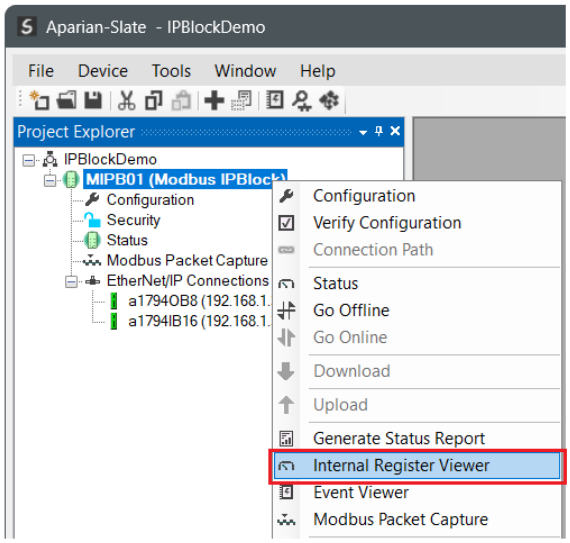


Figure 7.26 – Internal Register Viewer - Select

Once opened, the register type can be selected.

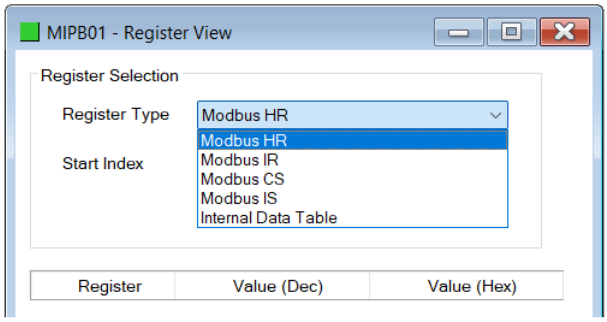


Figure 7.27 – Internal Register Viewer – Register Type

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

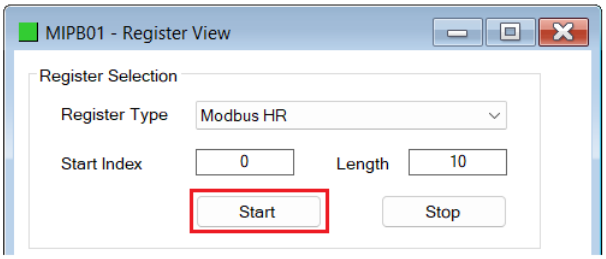


Figure 7.28 – Internal Register Viewer – Indexes

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

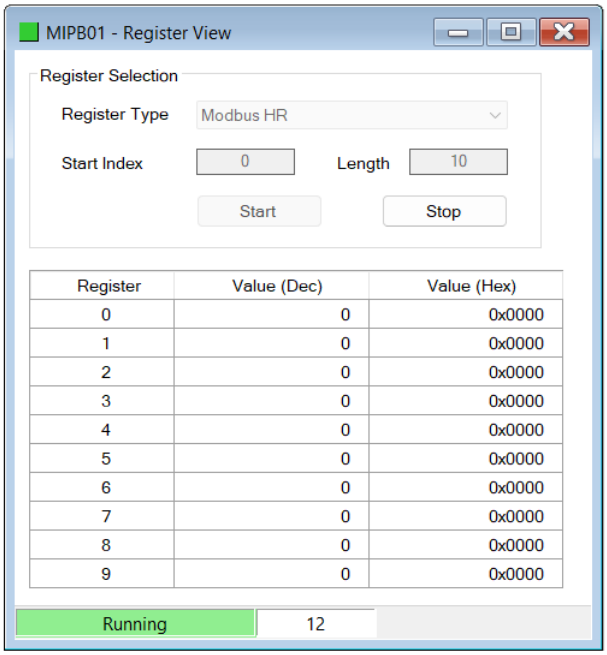


Figure 7.29 – Internal Register Viewer – Running

8. TECHNICAL SPECIFICATIONS

8.1. DIMENSIONS

The enclosure dimensions are detailed below. All dimensions are in millimeters.

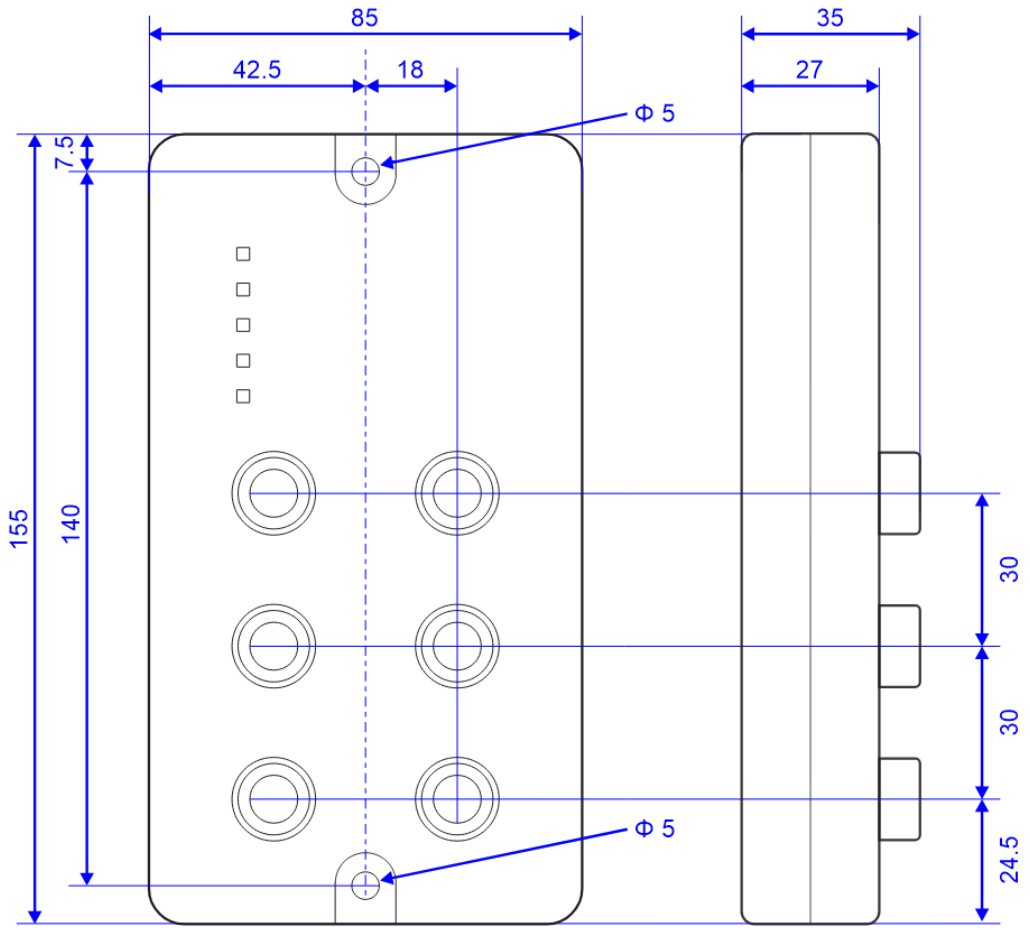


Figure 8.1 – Modbus IPBlock enclosure dimensions.

8.2. ELECTRICAL

Specification	Rating
Power requirements	Input: 10 – 32V DC, (60 mA @ 24 VDC) Class 2
Power consumption	1.5 W maximum
Module Connector	M12 4-pin A-coded male
Required Mating Connector	M12 4-pin A-coded female

Earth connection	Yes, terminal based
Emissions	IEC61000-6-4
ESD Immunity	EN 61000-4-2
Radiated RF Immunity	IEC 61000-4-3
EFT/B Immunity	EFT: IEC 61000-4-4
Surge Immunity	Surge: IEC 61000-4-5
Conducted RF Immunity	IEC 61000-4-6

Table 8.1 - Electrical specification.

8.3. ENVIRONMENTAL

Specification	Rating
Enclosure rating	IP67
Temperature	-20 – 70 °C
Relative Humidity	5% to 90% - No condensation
Pollution Degree	2
Altitude	< 2000 m

Table 8.2 - Environmental specification.

8.4. ETHERNET

Specification	Rating
Module Connector	M12 4-pin D-coded female
Required Mating Connector	M12 4-pin D-coded male
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
Embedded switch	Yes, 2 x Ethernet ports
Device Level Ring (DLR)	Supported
Network Time Protocol (NTP)	Supported

Table 8.3 - Ethernet specification.

8.5. SERIAL PORT (RS485)

Specification	Rating
Module Connector	M12 5-pin B-coded female
Required Mating Connector	M12 5-pin B-coded male
Electrical Isolation	1500 Vrms for 1 minute. (Port to remainder of board.) Note: Ports A and B are not isolated from each other.
BAUD	1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200
Parity	None, Even, Odd
Data bits	8
Stop bits	1
Termination	125 Ω - Software enabled

Table 8.4 – RS485 Serial Port specification.

8.6. MODBUS CLIENT – ENHANCED MODE

Specification	Rating
Modes Supported	Modbus TCP, Modbus RTU485
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 8.5 – Modbus Client specification.

8.7. MODBUS SERVER – ENHANCED MODE

Specification	Rating
Modes Supported	Modbus TCP 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 8.6 – Modbus Server specification.

8.1. ETHERNET/IP TARGET

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

Table 8.7 – EtherNet/IP Target specification.

8.2. 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 8.8 – EtherNet/IP Originator specification.

8.3. CERTIFICATIONS

Certification	Mark
CE Mark	
RoHS2 Compliant	
UL Mark File: E494895	 CLASS 1, DIV 2, GROUPS A, B, C, D **PENDING**
ODVA Conformance	 **PENDING**
UKCA	

Table 8.9 – Certifications

9. INDEX

A

assembly instance, 46, 47

C

class 1, 46, 47, 48
Concentrator, 126
Connection path, 82
Contact Us, 14
ControlLogix, 7

D

DC power, 15
DHCP, 20, 21, 22
dimensions, 146

E

Ethernet Bridge, 45
Ethernet connector, 18
Ethernet TCP, 7
Ethernet UDP, 9
Ethernet/IP, 64
EtherNet/IP, 7, 23, 64
EtherNet/IP Devices, 63
EtherNet/IP Devices configuration, 63, 64
EtherNet/IP MAP, 65
EtherNet/IP Map configuration, 65, 81

F

firmware upgrade, 27

G

General parameters, 27

I

input assembly, 128
input voltage, 16

L

LED, 123, 124
Logix controller, 46, 48, 126, 130
Logix platform, 7

M

Mapping Mode, 28
MODBUS, 1, 7, 9, 10, 14, 15, 25, 26, 27, 28, 30, 31, 45, 46, 47, 50, 52, 78, 81, 123, 124, 127, 128, 129, 131, 140, 141, 142, 146
MODBUS general configuration, 27
Modbus IPBlock, 7, 9, 14, 15, 25, 26, 45, 46, 47, 50, 81, 123, 124, 131, 142, 146
Modbus IPBlock general configuration, 34, 36, 38, 52, 67
Modbus IPBlock Internal Data Space, 72

P

Port parameters, 32
Protocol, 20

R

requested packet interval (RPI), 46, 48
Rockwell Automation, 24
RS485, 17
RS485, 18
RSLinx, 24
RSLogix 5000, 45, 47, 49

S

Serial, 9, 148
Slate, 9, 14, 20, 22, 24, 25, 26, 27, 124, 136, 142
SLATE Configuration Utility, 89, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111
statistics, 124, 135, 138, 139
Support email, 14

T

Target Browser, 22, 23, 64, 82

U

UDTs, 14, 47, 49, 50
User Defined Types (UDTs), 47, 49

W

web server, 124